POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Bayesian Neural Network to Factor-in Structural Attributes in Infrastructure Probabilistic Deterioration Models

SAID ALI KAMAL FAKHRI
Département de génie civil, géologique et des mines

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
génie civil

Octobre 2023

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Bayesian Neural Network to Factor-in Structural Attributes in Infrastructure Probabilistic Deterioration Models

présenté par **Said Ali Kamal FAKHRI**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Charles L. BÉRUBÉ, président
James-A. GOULET, membre et directeur de recherche
Kostas PAPAKONSTANTINOU, membre externe

DEDICATION

To my parents, who have given me everything and more...

ACKNOWLEDGEMENTS

I am enormously grateful to my academic advisor, Professor James-A. Goulet. Your work is inspiring, and your passion is contagious. Thank you for believing in me and welcoming me into your group – it has been a great pleasure.

A huge thanks to Zachary Hamida for helping me over the last two years. This work would not have been possible without your constant guidance, mentorship, and helpful insights. You are one of the most hardworking and kind individuals I have ever met.

Major thanks to the Institut de valorisation des données (IVADO) and Ministère des Transports du Québec (MTQ) for funding this research. Your generosity allowed me to carry out this work.

To all my colleagues, thank you for the warm company, fun times, and deep conversations.

RÉSUMÉ

Les infrastructures de transport jouent un rôle crucial dans la croissance économique en facilitant la circulation des personnes, des biens et des services. Au fil du temps, l'état des infrastructures se détériore en raison du vieillissement, de l'utilisation et des facteurs environnementaux. Bien que les inspections visuelles périodiques constituent l'approche la plus courante pour surveiller l'état des structures à grande échelle, elles sont coûteuses et subjectives. Il en résulte peu de points de données, souvent peu fiables, pour chaque structure. Malgré cela, la plupart des propriétaires d'infrastructures publiques, s'appuient de plus en plus sur des modèles de détérioration basés sur les inspections visuelle pour planifier les activités d'entretien et de réhabilitation. Cependant, se fier uniquement à un nombre limité d'inspections visuelles pour chaque structure n'est pas suffisant pour modéliser de manière fiable leur détérioration. La prise en compte des attributs structurels (par exemple, l'âge, l'emplacement, etc.) peut compenser le manque d'inspections en permettant le partage d'informations entre les structures. Ces attributs sont de bons prédicteurs de détérioration et sont souvent facilement disponible. Une récente méthode de régression par le noyaux a réussi à inclure des attributs structurels dans un modèle probabiliste de détérioration des infrastructures tout en quantifiant l'incertitude des inspecteurs. Bien qu'elle soit capable de modéliser de manière fiable la détérioration, cette méthode nécessite des ressources de calcul considérables et ne peut inclure que peu d'attributs structurels. Ces problèmes rendent l'approche basée sur les noyaux impropre à la modélisation de la détérioration des grands réseaux d'infrastructures ; Pourtant, trouver une méthode alternative n'est pas anodin, car elle doit être probabiliste, efficace sur le plan informatique et capable d'inclure de nombreux attributs. Bien que les réseaux de neurones bayésiens (BNN) soient bien adaptés aux grands ensembles de données et possèdent bon nombre des qualités souhaitées, leur intégration dans le modèle de détérioration existant a été limitée par leurs mécanismes d'inférence reposant sur l'échantillonnage ou l'optimisation basée sur le gradient. Cependant, l'inférence sous forme fermée dans les BNN a récemment été rendue possible grâce à une méthode probabiliste appelée *tractable approximate Gaussian inference* (TAGI). Cette recherche vise à fusionner un BNN entrainé par TAGI avec un modèle probabiliste de détérioration des infrastructures à grande échelle pour apprendre la relation entre les attributs structurels et les taux de détérioration. La méthode proposée est vérifiée sur un ensemble de données d'inspection visuelle synthétique et ses performances sont comparées à l'approche existante en utilisant les données du réseau de ponts de la province de Québec. Il est démontré que la nouvelle méthode est plusieurs fois plus rapide que la méthode existante sans compromettre les performances

prédictives. Il intègre de manière transparente tous les attributs disponibles, ce qui supprime la tâche fastidieuse et chronophage consistant à identifier les variables les plus importantes à chaque fois que de nouvelles analyses sont effectuées. Ces avantages ont incité à étendre la méthode proposée pour estimer conjointement les paramètres de l'inspecteur dans toutes les catégories structurelles (poutres, platelage, etc.), plutôt que de devoir s'appuyer sur la configuration actuelle par catégorie. Ils ont également permis d'automatiser le traitement de bout en bout de l'ensemble des données composées de milliers de structures englobant des millions d'inspections visuelles. L'effet du cadre conjoint sur les paramètres des inspecteurs est examiné et ses performances sont comparées à l'approche par catégorie utilisant plusieurs catégories structurelles. Dans l'ensemble, l'efficacité et l'évolutivité des cadres conjoints et par catégorie proposés permettent de modéliser de manière fiable la détérioration de grands réseaux d'infrastructures.

ABSTRACT

Transportation infrastructure plays a crucial role in fueling the economic growth by facilitating the movement of people, goods, and services. Over time, the condition of the infrastructure deteriorates due to aging, usage, and environmental factors. Although periodic visual inspections are the most common approach to monitoring structural health at a large scale, they are costly and subjective. This results in few and often unreliable data points for each structure. Despite this, most public infrastructure owners increasingly rely on deterioration models based on visual inspections for planning maintenance and rehabilitation activities. Yet, depending solely on a limited number of visual inspections for each structure is not sufficient to reliably model their deterioration. Factoring in structural attributes (e.g., age, location, etc.) can compensate for the lack of inspections by allowing information-sharing between structures. These attributes are good predictors of deterioration and are often readily available. A recent kernel-based regression method has successfully included structural attributes in an infrastructure probabilistic deterioration model while also quantifying the inspectors' uncertainty. Despite being capable of reliably modeling deterioration, this method requires considerable computational resources and can only include few structural attributes. These issues make the kernel-based approach unfit for modeling the deterioration of large infrastructure networks; yet, finding an alternative method is not trivial, as it must be probabilistic, computationally efficient, and capable of including many attributes. Although Bayesian neural networks (BNN) are well-suited for large datasets and have many of the desired qualities, their integration into the existing deterioration model has been restricted by their inference mechanisms relying on sampling or gradient-based optimization. However, closed-form inference in BNNs was recently made possible by a probabilistic method called tractable approximate Gaussian inference (TAGI). This research aims to fuse a TAGI-trained BNN with a large-scale infrastructure probabilistic deterioration model to learn the relation between the structural attributes and the deterioration rates. The proposed method is verified on a synthetic visual inspection dataset and its performance is compared against the existing kernel-based approach using the bridge network data from the province of Quebec. The new method is shown to be orders of magnitude faster than the existing one without compromising the predictive performance. It seamlessly incorporates all the available attributes, which removes the tedious and time-consuming task of identifying the most important ones each time new analyses are performed. These advantages prompted the extension of the proposed method to jointly estimate the parameters of inspector' across all structural categories (beams, slabs, etc.), rather than having to rely on the current category-wise setup. They

also enabled automating the end-to-end processing of all the data composed of thousands of structures encompassing millions of visual inspections. The effect of the joint framework on inspectors' parameters is examined, and its performance is compared against the category-wise approach using several structural categories. Overall, the efficiency and scalability of the proposed joint and category-wise frameworks allow for reliably modeling the deterioration of large infrastructure networks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

Mathematical Notation

Unless specified otherwise, the following mathematical notation is used throughout the thesis:

- a: scalars are denoted by lower case letters.
- $\boldsymbol{a}$: vectors are denoted by bold lower case letters.
- A: random variables are denoted by italic capital letters.
- $\boldsymbol{A}$: random vectors or matrices with random variables are denoted by bold italic capital letters.
- $\mathbf{A}$: matrices with deterministic values are denoted by bold capital letters.
- $\mathcal{A}$: calligraphic font denotes a set.
- $\mathtt{A}$: capital typewriter font denotes the number of elements in a vector or a set.

Symbols

$\mathbf{A}$	Transition matrix
A	Fraction of an inspected element area with little damage
$\boldsymbol{a}$	Vector of normalized kernel regression weights
$\mathcal{B}$	Set of bridges
B	Fraction of an inspected element area with medium damage
$\mathbf{C}$	Observation matrix
C	Fraction of an inspected element area with important damage
$\mathcal{C}$	Structural element category
D	Fraction of an inspected element area with very important damage
$f(\cdot)$	Probability density function
$\mathbf{G}$	Grid of kernel regression reference points
$\boldsymbol{h}^{(j)}$	Vector of hidden units in the j-th layer of the neural network
$\mathbf{I}$	Identity matrix
i	Index of inspectors
$\mathbf{J}$	Kalman smoother gain matrix
j	Index of bridges
$\mathbf{K}$	Kalman gain matrix
$\boldsymbol{k}(\cdot)$	Multivariate kernel function
$k(\cdot)$	Univariate kernel function

$\mathcal{L}$	Log-likelihood
$\boldsymbol{L}$	Lower triangular matrix or the Cholesky factor of a covariance matrix
$\boldsymbol{l}$	Elements of the Cholesky factor $\boldsymbol{L}$ predicted by the Bayesian neural network
$\tilde{\boldsymbol{l}}$	Transformation of $\boldsymbol{l}$ to bound the diagonal elements of $\boldsymbol{L}$ to the positive domain
l	Lower bound of health condition
ℓ	Kernel length
$\mathtt{M}$	Number of kernel regression reference points in one dimension
$\mathcal{N}(\cdot)$	Gaussian distribution
$\mathtt{M}$	Total number of kernel regression reference points
n	Transformation function parameter
$o(\cdot)$	Transformation function
$o(\cdot)^{-1}$	Inverse transformation function
p	Index of structural elements
$p_{1:2}$	Deterioration speed model parameters
$\mathtt{Q}$	Number of regression covariates
$\mathbf{Q}$	Process error covariance matrix
$\mathbb{R}$	Set of real numbers
$\mathbf{R}$	Observation error covariance matrix
$\mathtt{T}$	Timestamp of the last observation
t	Time stamp
$\mathcal{U}(\cdot)$	Uniform distribution
u	Upper bound of health condition
v	Observation error
v^2	Square of the observation error
$\overline{v^2}$	Expected value of the square of the observation error, i.e., $\overline{v^2} = \mathbb{E}[V^2]$
$\widetilde{\overline{v^2}}$	Transformation of $\overline{v^2}$ to bound it in the positive domain
$\boldsymbol{w}$	Transition model process error
$\boldsymbol{w}_0$	Kernel regression model error
$\boldsymbol{x}$	Vector of hidden states in the unconstrained space
$\tilde{\boldsymbol{x}}$	Vector of hidden states in the constrained space
$\dot{\boldsymbol{x}}_z$	Vector of hidden states associated with covariates
x	State of deterioration condition in the unconstrained space
$\tilde{x}$	State of deterioration condition in the constrained space
$\dot{x}$	State of deterioration speed in the unconstrained space
$\tilde{\dot{x}}$	State of deterioration speed in the constrained space
$\ddot{x}$	State of deterioration acceleration in the unconstrained space

$\tilde{\ddot{x}}$	State of deterioration acceleration in the constrained space
y	Condition observation in the unconstrained space
$\tilde{y}$	Condition observation in the constrained space
y_a	Condition observation of little damage category
y_b	Condition observation of medium damage category
y_c	Condition observation of important damage category
y_d	Condition observation of very damaged category
$\boldsymbol{z}$	Vector of structural attributes
Δt	Time-step duration
μ	Expected value
$\boldsymbol{\mu}$	Vector of expected values
$\varphi(\cdot)$	A non-linear activation function employed by a neural network
σ	Standard deviation
$\boldsymbol{\Sigma}$	Covariance matrix
τ	Total number of timestamps
$\boldsymbol{\theta}$	Vector of all model parameters
$\boldsymbol{\theta}_{\texttt{BNN}}$	Vector of Bayesian neural network parameters
$\boldsymbol{\theta}_{\texttt{I}}$	Vector of inspectors' parameters
$\boldsymbol{\theta}_{\texttt{S}}$	Vector of initial state and process error parameters
∞	Infinity

Acronyms

AGVI	Approximate Gaussian Variance Inference
BNN	Bayesian Neural Network
CPU	Central Processing Unit
DAG	Directed Acyclic Graph
FNN	FeedForward Neural Network
GA	Genetic Algorithm
GMA	Gaussian Multiplicative Approximation
GPU	Graphics Processing Unit
KF	Kalman Filter
KR	Kernel Regression
KS	Kalman Smoother
MLE	Maximum Likelihood Estimation
NR	Newton-Raphson

PDF	Probability Density Function
SSM	State-Space Model
SSM-KR	State-Space Model + Kernel Regression
TAGI	Tractable Approximate Gaussian Inference

LIST OF APPENDICES

CHAPTER 1 Introduction

1.1 Motivation

Deterioration of transportation infrastructure from aging, usage, and environmental exposure is an issue faced by most industrialized countries [4]. For instance, in Canada, 80% of the roads and bridges are over 20 years old and 40% of them are in fair or worse condition [5]. Monitoring and maintaining deteriorating infrastructure is essential for prolonging its life, reducing economic costs, and ensuring public safety. This task becomes challenging when dealing with large networks of structures on limited budgets, as done by most public infrastructure managers. In order to make informed decisions and efficiently use the available resources, infrastructure managers increasingly rely on deterioration models to gain insight on the current state of their assets and understand how their condition is expected to evolve in the future. Therefore, reliable deterioration models are critical for effectively allocating limited resources and maintaining the safety and serviceability of the infrastructure.

The predominant approach to modeling infrastructure deterioration is based on periodic visual inspections that assess structural health. However, relying solely on raw visual inspection data is not sufficient because inspections are subjective and infrequent. This results in few highly uncertain data points for each structure over a long period of time. The lack of inspections arises over and over again, particularly for newly built structures and for those with replaced elements, where the past history of inspections becomes irrelevant. Considering additional sources of information can improve the deterioration models based on visual inspections by enabling information-sharing between structures. In particular, infrastructure managers have access to structural attributes like location, age, and traffic load, which have been shown to be good predictors of structural deterioration [6]. For instance, the deterioration of a bridge located in a cold climate is expected to be different from one that is located in a warm one due to the exposure to different environmental loads. Nonetheless, the scale of the problem makes it challenging to incorporate all available structural attributes into a network-scale deterioration model. Existing approaches are inefficient and limited to using only a subset of all the available attributes due to the computational constraints [7]. The computational time is further amplified when modeling the deterioration of large networks of structures, where each one is made up of many elements belonging to different categories. This is because the analysis of each element category requires a hand-picked selection of attributes. These issues make the existing approaches unfit for modeling the deterioration of large networks of structures.

Yet, integrating structural attributes within a stochastic deterioration model based on visual inspections is not trivial. The approach must be probabilistic for a seamless integration, and computationally efficient to be practical for large-scale applications. The aim of this work is to develop a hybrid framework that uses visual inspections to model infrastructure deterioration and factors-in structural attributes to improve the model's predictions by sharing information between structures. Unlike the existing approaches [7], the proposed model is computationally efficient and is not restricted to using only few attributes. Moreover, the method allowed for the end-to-end data process automation, which will facilitate its adoption by the infrastructure managers. All these advantages make the proposed framework well-suited for modeling deterioration of large networks of structures.

1.2 Visual Inspections on Networks of Bridges

Infrastructure managers typically monitor the structural health using routine visual inspections. In this work, infrastructure deterioration is modeled using the visual inspections and structural attributes from a network of approximately 10000 bridges located in the province of Quebec, Canada, shown in Figure 1.1. Structural attributes are included in the deterioration model as they can encode common deterioration patterns between similar bridges. For instance, bridges located in the southern region of Quebec will experience more freeze-thaw cycles than those in the north, which is expected to accelerate their deterioration.

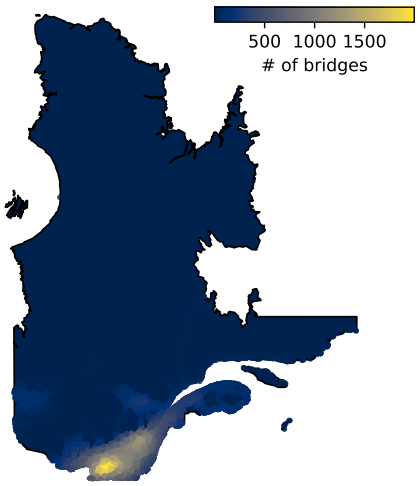


Figure 1.1 Density map of the network of bridges in Quebec, Canada.

Figure 1.2 illustrates the hierarchy of the data used in this thesis. The network of structures consists of a set of bridges $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{\mathrm{B}}\}$. Each bridge $\mathcal{B}_j$ is made up of many elements

e_p^j, where the notation indicates that e_p^j is the p-th element from the j-th bridge. The bridges' elements are further grouped into a set of structural categories such as beams, slabs, pavement, and so on, denoted by $\{\mathcal{C}_1^j, \mathcal{C}_2^j, \ldots, \mathcal{C}_{\mathtt{S}}^j\}$. That is, $\{e_1^j, e_2^j, \ldots, e_{\mathtt{E}_{(\mathcal{C}_i)}}^j\} \in \mathcal{C}_i^j$, where $\mathtt{E}_{(\mathcal{C}_i)}$ is the number of elements in the i-th category of the j-th bridge, $\mathcal{C}_i^j$. The structural attributes associated with an element include the material it is made of, in addition to the attributes it inherits from its parent bridge (e.g., location, traffic load, etc.).

The inspection of a bridge consists of examining each of its elements according to the guidelines set by the Ministry of Transportation Quebec [8]. The condition of each element is evaluated by visually examining it and assigning fractions of its area to four damage categories: A: *Nothing to little*, B: *Medium*, C: *Important*, and D: *Very Important* [8]. For example, an element with a rating of $y_{ABCD} = \{y_A = 80\%,\ y_B = 0\%,\ y_C = 0\%,\ y_D = 20\%\}$ implies that 80% of it is *undamaged* while 20% of it is *severely damaged*. Modeling the deterioration with four metrics while accounting for their interdependency increases the complexity of the analysis. Hence, the four damage categories are aggregated into a single metric $\tilde{y}$ using a weighted sum, as done in the work of Hamida and Goulet [3]:

$$\tilde{y} = \omega_A y_A + \omega_B y_B + \omega_C y_C + \omega_D y_D,$$

where the weights are defined as $\omega_A = 1$, $\omega_B = 0.75$, $\omega_C = 0.5$, and $\omega_D = 0.25$. This results in $\tilde{y} = 100$ corresponding to the perfect condition and $\tilde{y} = 25$ corresponding to the worst condition.

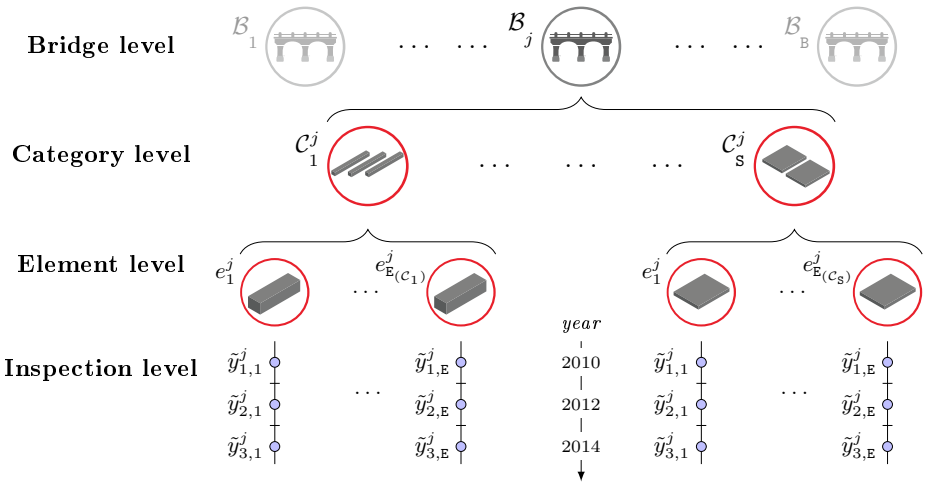


Figure 1.2 Graphical illustration of the components making up the bridge database employed in this thesis. Each bridge $\mathcal{B}_j$ has many elements, which are grouped into structural categories $\mathcal{C}_i^j$. The inspections are done on an element level, whereby the p-th element on the j-th bridge, denoted by e_p^j, is assigned a condition rating of $\tilde{y}_{t,p}^j$ at the time step t.

1.3 Research Objectives

The objective of this thesis is to develop a scalable and efficient probabilistic deterioration model that overcomes the limitations of the existing approach based on state-space models (SSM) and kernel regression (KR). The core objectives include the following:

- Develop a probabilistic deterioration model that is scalable to entire infrastructure networks.
- Enable a seamless integration of the structural attribute information in the probabilistic deterioration model.
- Use the advantages of the proposed model to develop an automated end-to-end framework for modeling infrastructure deterioration for all structural categories.

1.4 Thesis Outline

This thesis is organized as follows: Chapter 2 presents a review of the existing methods for modeling infrastructure deterioration. Since the core objective of this work is to address the limitations of the existing kernel-based approach, it is covered in greater depth. Chapter 2 also provides the background for the tractable approximate Gaussian inference method (TAGI), which is used to learn the parameters of the Bayesian neural network (BNN) employed in this thesis. Chapter 3 presents the methodology behind the proposed infrastructure deterioration model, which couples a state-space model with a TAGI-trained BNN. Chapter 4 verifies the new model on synthetic data. After verification, this model is compared against the existing kernel-based approach on synthetic and real datasets. Chapter 5 extends the proposed method to enable the joint estimation of each inspector's bias and variance using all their observations across different structural categories. The effect of the joint estimation of the inspectors' uncertainty parameters is examined using real data for several structural categories. Finally, Chapter 6 presents the conclusion of the thesis, the limitations of the proposed method, and directions for future work.

CHAPTER 2 Literature Review

This chapter presents a review of the state-of-the-art methods for modeling infrastructure deterioration based on visual inspections. The background for the tractable approximate Gaussian inference method is also provided, as it is used to estimate the parameters of the proposed deterioration model.

2.1 Infrastructure Deterioration Models Based on Visual Inspections

This section presents methods for modeling infrastructure deterioration based on visual inspections.

2.1.1 Discrete Markov Models

The predominant approach behind probabilistic infrastructure deterioration models based on visual inspections relies on the discrete Markov processes [9–11]. In general, these are defined by a set of states $\mathcal{S} = \{s_1, s_2, \ldots, s_{\mathtt{S}}\}$ and a transition matrix $\mathbf{P}$, where its elements p_{ij} represent the probability of transitioning from state s_i at time t to state s_j at time $t+1$. In the context of infrastructure deterioration modeling, the states represent the condition of the structural element. For example, $s_1 \equiv excellent$, $s_2 \equiv good$, $s_3 \equiv fair$, and $s_4 \equiv poor$. In this case, the transition matrix $\mathbf{P}$ is defined as follows,

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ 0 & p_{22} & p_{23} & p_{24} \\ 0 & 0 & p_{33} & p_{34} \\ 0 & 0 & 0 & p_{44} \end{bmatrix},$$

where the elements below the main diagonal are zero because the condition of the structural element cannot improve over time without maintenance or repair. Figure 2.1 illustrates the state transition diagram for this example. The transition probability matrix $\mathbf{P}$ can be estimated from the inspection data using the maximum likelihood estimation (MLE) method [12]. For a sequence of observed transitions, the log-likelihood function is given by

$$\mathcal{L}(p) = \sum_{i,j}^{\mathtt{S}} \mathtt{N}_{ij} \log\left(p_{ij}\right)$$

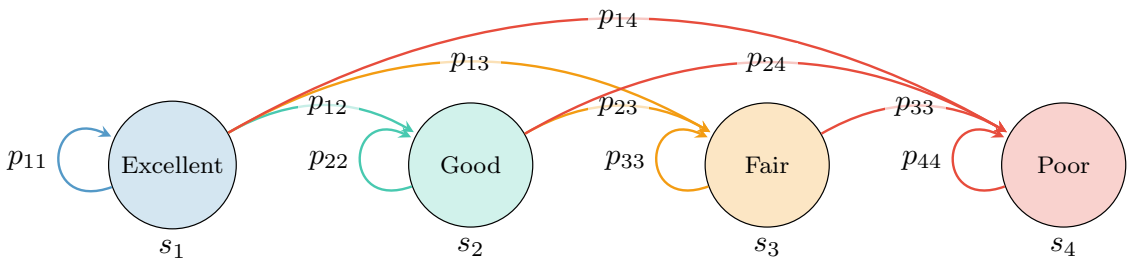


Figure 2.1 Markov chain representing the deterioration process of a structural element with the arrows indicating the possible transitions between states.

where $\mathtt{N}_{ij}$ is the number of observed transitions from state s_i to state s_j [12]. Maximizing the log-likelihood yields the following point estimates for the transition probabilities,

$$\hat{p}_{ij} \approx \frac{\mathtt{N}_{ij}}{\sum_{j=1}^{\mathtt{S}} \mathtt{N}_{ij}},$$

which corresponds to the fraction of times the given transitions are observed in the data [12].

Despite the widespread adoption of discrete Markov processes for modeling infrastructure deterioration, they have some notable drawbacks. Namely, the transition matrix $\mathbf{P}$ is typically assumed to be time-invariant (homogenous) to reduce the number of model parameters and computational demand [11, 13–15]. However, this assumption is not realistic for many infrastructure systems. To include the non-homogeneity of the deterioration process while using a homogenous Markov model requires categorizing the structures into age groups and estimating a transition matrix for each [11,13,16]. Factoring in other covariates that influence the deterioration process, such as traffic load, climate, and material properties, can be done in a similar manner [13,14]. However, such an approach considerably increases the number of transition matrices that must be estimated and requires data to be manually categorized into groups [13]. Moreover, each group must have a sufficient amount of data for its transition matrix estimates to be accurate. Alternative approaches to consider the non-homogeneity of the deterioration process include inhomogeneous Markov models with time-varying transition matrices [17] and semi-Markov models with transition rate functions [18]. The lack of adoption of these approaches in practice may be attributed to the challenges associated with accurately estimating the additional parameters resulting in a marginal improvement in the predictive performance compared to the standard homogenous Markov models [16,19]. Much of the recent developments in discrete Markov models for infrastructure deterioration focused on addressing their limitations by combining them with other methods. These developments are discussed in Section 2.2.1.

2.1.2 Regression Models

Regression consists of finding the relation between a response variable y and a set of explanatory variables $\boldsymbol{x}$ [20]. In the context of infrastructure deterioration, the response variable is typically the condition of the structural element and the explanatory variables are the covariates that influence the element's deterioration process such as its age and material properties. The regression data consists of $\mathtt{D}$ pairs of covariates and associated response variables, $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_{\mathtt{D}}, y_{\mathtt{D}})\}$, and the objective is to learn a mapping $g : \boldsymbol{x} \mapsto y$.

Early application of regression to model infrastructure deterioration primarily consisted of linear regression [21, 22]. Using linear regression in this context is not ideal because it treats the condition rating as a continuous variable $g(\boldsymbol{x}) \in \mathbb{R}$, even though it is discrete [23, 24]. To overcome this limitation, ordinal logistic regression was used to take into account the response being a discrete variable with more than two categories [23, 24]. Feedforward neural networks (FNN) are another popular approach for modeling infrastructure deterioration based on regression [19, 25–27]. FNNs can be viewed as a composition of many simple non-linear functions [20]. Their application to structural health modeling had already been explored two decades ago [26, 27]; however, there has been a resurgence of interest that may be attributed to the success of neural networks in other domains [28]. Much of the recent work explored the usage of more advanced architectures such as convolutional neural networks, which are effective at learning meaningful representation of the input data [29], and recurrent neural networks, which are suited for processing sequential data [30].

Although widely used, the regression-based approaches employed for modeling infrastructure deterioration are incapable of quantifying uncertainty in their predictions. This is a major limitation because the deterioration process is affected by many factors which cannot be fully accounted for by the covariates. Despite the importance of uncertainty quantification in this context, a review of the literature did not reveal any attempts at modeling infrastructure deterioration with methods such as Gaussian process regression [20] or Bayesian neural networks [12]. Although these methods report uncertainty in their predictions, regression-based approaches are still ill-equipped for modeling deterioration based on visual inspections. For one, the inspection data is highly imbalanced since most of the structures are kept in good health and are seldom left to deteriorate to the point of failure. This uneven distribution in the condition ratings biases the regression models towards predicting better condition states. Moreover, having few inspections per element limits the capacity of regression methods to effectively capture the temporal dependence of the deterioration process. The reviewed methods also neglect to factor-in the uncertainty associated with each inspector, which is problematic because the inspection process is inherently subjective [31]. For instance, two inspectors

may assign different condition ratings to the same structural element depending on their experience level. Contrary to the regression methods, an approach based on state-space models has been shown to be effective at probabilistically modeling infrastructure deterioration while also quantifying the uncertainty of each inspector [3]. This framework is described in the following section.

2.1.3 State-Space Models

State-space models (SSM) describe the behavior of a system over time using probabilistic transition and observation models [20]. SSM have been shown to be effective at modeling infrastructure deterioration on a network-scale [3]. In this context, the transition model describes the physical deterioration process given by

$$\overbrace{\boldsymbol{x}_t = \mathbf{A}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t}^{\text{transition model}}, \ \overbrace{\boldsymbol{w}_t : \boldsymbol{W} \sim \mathcal{N}(\boldsymbol{w}; \mathbf{0}, \mathbf{Q})}^{\text{process errors}}, \tag{2.1}$$

where the state of a structural element at time t is represented by a vector $\boldsymbol{x}_t$ containing the condition x_t, speed $\dot{x}_t$, and acceleration $\ddot{x}_t$, $\mathbf{A}$ is the state transition matrix, $\boldsymbol{w}_t$ is the process error, and $\mathbf{Q}$ is the process error covariance matrix. The state transition matrix and its associated process error covariance matrix are defined assuming a constant-acceleration kinematic model following

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \sigma_W^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix},$$

where Δt is the time step between successive states and σ_W^2 is the variance of the process error [3]. The visual inspections are characterized by the observation model, which is given by

$$\overbrace{y_t = \mathbf{C}\boldsymbol{x}_t + v_t}^{\text{observation model}}, \ \overbrace{v_t : V \sim \mathcal{N}(v; \mu_V(i), \sigma_V^2(i))}^{\text{observation errors}}, \tag{2.2}$$

where y_t represents the observation at time t, $\mathbf{C} = [1\ 0\ 0]$ is the observation matrix, and v_t is the observation error with $\mu_V(i)$ and $\sigma_V^2(i)$ representing the relative bias and variance of the i-th inspector [32].

At each time step t, the deterioration state is inferred using the Kalman filter (KF) [33], which consists of the prediction and update steps. The prediction step estimates the state

$\boldsymbol{x}_t$ from $\boldsymbol{x}_{t-1}$ using the transition model from Equation 2.1, such that,

$$\begin{aligned}\boldsymbol{\mu}_{t|t-1} &= \mathbf{A}\boldsymbol{\mu}_{t-1|t-1},\\ \boldsymbol{\Sigma}_{t|t-1} &= \mathbf{A}\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}^\intercal + \mathbf{Q}.\end{aligned}$$

Here, $\boldsymbol{\mu}_{t|t-1} \equiv \mathbb{E}[\boldsymbol{X}_t|y_{1:t-1}]$ and $\boldsymbol{\Sigma}_{t|t-1} \equiv \mathrm{cov}[\boldsymbol{X}_t|y_{1:t-1}]$ are the expected value and covariance for the state at time t, given all the observations up to time $t-1$, which are denoted by $y_{1:t-1} \equiv \{y_1, y_2, \ldots, y_{t-1}\}$. In the absence of an observation at time t, the prediction step is repeated; otherwise, if an observation is available, the update step is applied. The update step finds the posterior knowledge of $\boldsymbol{x}_t$ conditioned on all the observations $y_{1:t}$ using the Gaussian conditional equations:

$$\begin{aligned}f\left(\boldsymbol{x}_t \mid y_{1:t}\right) &= \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}\right),\\ \boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t\left(y_t - \mu_{y,t|t-1}\right),\\ \boldsymbol{\Sigma}_{t|t} &= \left(\mathbf{I} - \mathbf{K}_t\mathbf{C}\right)\boldsymbol{\Sigma}_{t|t-1},\\ \mathbf{K}_t &= \boldsymbol{\Sigma}_{t|t-1}\mathbf{C}^\intercal/\sigma^2_{y,t|t-1},\\ \mu_{y,t|t-1} &= \mathbf{C}\boldsymbol{\mu}_{t|t-1} + \mu_V(i),\\ \sigma^2_{y,t|t-1} &= \mathbf{C}\boldsymbol{\Sigma}_{t|t-1}\mathbf{C}^\intercal + \sigma_V(i)^2,\end{aligned}$$

where $\mu_{y,t|t-1}$ and $\sigma^2_{y,t|t-1}$ are the expected value and variance of the predicted observation, which are obtained using the observation model from Equation 2.2, and $\mathbf{K}_t$ is the Kalman gain matrix, which weighs the importance of the prior predicted state relative to the observation. The KF can be expressed concisely as

$$\left(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}, \mathcal{L}_t\right) = \text{Kalman filter}\left(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1}, y_t, \mathbf{A}, \mathbf{Q}, \mathbf{C}, \mu_V(i), \sigma^2_V(i)\right),$$

where $\mathcal{L}_t$ denotes the likelihood for the observation y_t, given all the previous observations $y_{1:t-1}$. Conveniently, the likelihood $\mathcal{L}_t$ is parametrized by $\mu_{y,t|t-1}$ and $\sigma^2_{y,t|t-1}$, which are readily available:

$$\mathcal{L}_t = f\left(y_t \mid y_{1:t-1}\right) = \mathcal{N}\left(y_t; \mu_{y,t|t-1}, \sigma^2_{y,t|t-1}\right).$$

Once the state estimates are obtained by the KF, they are then refined using the Kalman

smoother (KS) [34], which is given by the following equations:

$$
\begin{aligned}
f\left(\boldsymbol{x}_t \mid y_{1:\mathtt{T}}\right) &= \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\mu}_{t|\mathtt{T}}, \boldsymbol{\Sigma}_{t|\mathtt{T}}\right), \\
\boldsymbol{\mu}_{t|\mathtt{T}} &= \boldsymbol{\mu}_{t|t} + \mathbf{J}_t\left(\boldsymbol{\mu}_{t+1|\mathtt{T}} - \boldsymbol{\mu}_{t+1|t}\right), \\
\boldsymbol{\Sigma}_{t|\mathtt{T}} &= \boldsymbol{\Sigma}_{t|t} + \mathbf{J}_t\left(\boldsymbol{\Sigma}_{t+1|\mathtt{T}} - \boldsymbol{\Sigma}_{t+1|t}\right)\mathbf{J}_t^{\intercal}, \\
\mathbf{J}_t &= \boldsymbol{\Sigma}_{t|t}\mathbf{A}^{\intercal}\boldsymbol{\Sigma}_{t+1|t}^{-1},
\end{aligned}
$$

where $\boldsymbol{\mu}_{t|\mathtt{T}}$ and $\boldsymbol{\Sigma}_{t|\mathtt{T}}$ are the posterior mean and covariance of the smoothed state estimates at time t, given all the observations up to time $\mathtt{T}$, which denotes the time step of the last observation. The KS can be expressed in a compact form as

$$
\left(\boldsymbol{\mu}_{t|\mathtt{T}}, \boldsymbol{\Sigma}_{t|\mathtt{T}}\right) = \text{Kalman smoother}\left(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}, \boldsymbol{\mu}_{t+1|t}, \boldsymbol{\Sigma}_{t+1|t}, \boldsymbol{\mu}_{t+1|\mathtt{T}}, \boldsymbol{\Sigma}_{t+1|\mathtt{T}}, \mathbf{A}\right).
$$

Using the KF to model deterioration requires imposing two constraints on the state estimates to ensure the condition x_t is monotonically decreasing over time and to keep the model predictions within a predefined range of values [3]. The first constraint is enforced at each time step by restricting the deterioration rate $\dot{x}_t$ to be negative using a probability density function (PDF) truncation method [35]. The second constraint is imposed before and after analyses to maintain the hidden states and observations within a predefined range of health indexes $[l, u]$. Here, $l \equiv 25$ represents the worst damage state and $u \equiv 100$ represents the perfect condition. This transformation is required because SSM models the observations and hidden states as Gaussian random variables, which are unbounded. Hence, the original observations $\tilde{y}_t \in [l, u]$ are first transformed into an unbounded domain $(-\infty, \infty)$ to carry out analyses. Afterwards, the predicted hidden states are back-transformed into the original space $[l, u]$ for interpretation. Appendix A presents more details related to this space transformation.

Figure 2.2 illustrates an example of time series depicting structural deterioration modeled by the SSM [3]. The blue points represent the inspections performed on the element, denoted by $\tilde{y}_t$, while the asterisks correspond to the offset associated to each inspectors' relative bias μ_{V_i}. Each inspector's variability is quantified by his or her standard deviation σ_{V_i}, represented by the blue error bars. The SSM is capable of predicting the deterioration speed alongside the deterioration condition. The model predictions are shown by the black dashed line with the shaded area corresponding to their uncertainty.

Although SSM can effectively model deterioration, they solely rely on the visual inspection data without sharing information between structural elements. This is especially problematic when there are only few inspections available for each element. This limitation was addressed

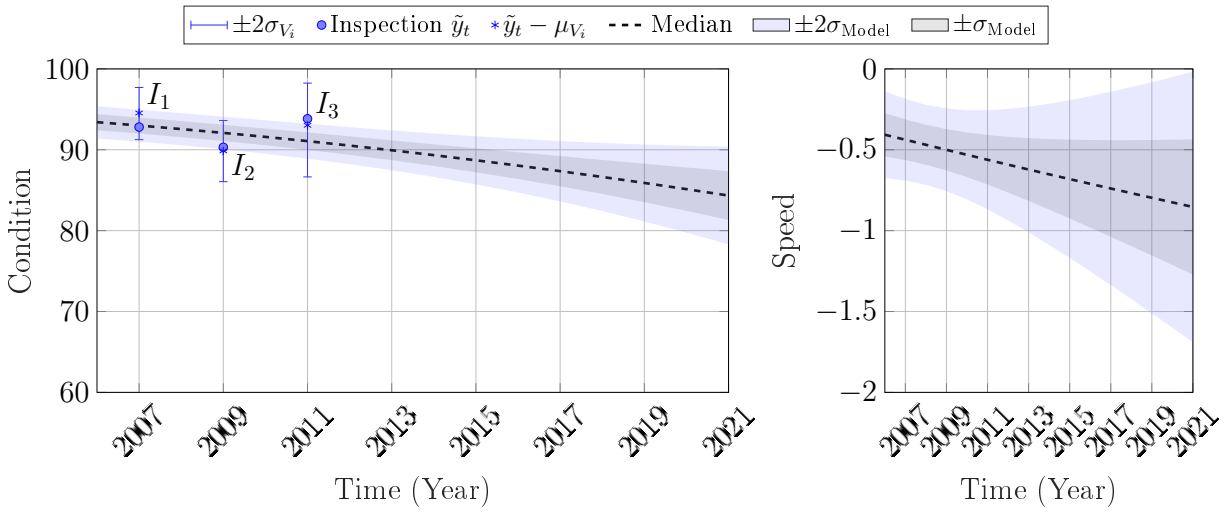


Figure 2.2 Example time series depicting structural deterioration modeled by the SSM.

by coupling the SSM framework with a kernel-based regression (KR) method [7], which is detailed in Section 2.2.2.

2.2 Hybrid Infrastructure Deterioration Models

This section starts with an overview of the recent approaches for modeling infrastructure deterioration with hybrid models. This is followed by a detailed description of the hybrid framework made up of a state-space model combined with kernel regression, since the core objective of this thesis is to develop a method that overcomes its limitations.

2.2.1 General Overview

A number of recent approaches sought to model infrastructure deterioration with hybrid frameworks consisting of two models. In general, this is done to avoid reliance on the visual inspections, which are highly subjective and limited in number [31]. The reliance on visual inspections is overcome by relating the information characterizing the physical understanding of the deterioration process to the model based on visual inspections. For instance, Zambon et al. [36] combined a semi-Markov model with an analytical carbonation-induced corrosion model, and they demonstrated the use of their framework on a single bridge. However, scaling this method to a large network of bridges is not practical, as relating the analytical model parameters to the transition probabilities of the semi-Markov model requires in-depth knowledge about each structure. Specifically, this approach requires information from the original design documents and additional tests and measurements on top of the visual inspections

(e.g., to determine carbonation depths at various points of interest). Collecting this data is a time-consuming and costly process. Dizaj et al. [37] proposed a similar method, where they estimated the transition probabilities of a non-homogeneous Markov model using an empirical model for chloride-induced corrosion. They tested their framework on a single reinforced concrete column from the UW-PEER database [38]; however, scaling their approach to a network of structures is impractical as it also requires detailed information on each structure. In contrast, Hamida and Goulet [7] demonstrated the use of their hybrid framework for modeling the deterioration of a network of bridges. Their method consists of the state-space model presented in Section 2.1.3, which serves as the base deterioration model using the visual inspections, and a kernel regression model, which enhances the base model's predictions by sharing information between structural elements using their attributes such as age and material type. Although this approach was demonstrated to work on a network level, it is still hindered by limitations that prevent it from being employed in practice. Namely, the kernel regression model can only use a limited number of explanatory variables, and even with a low number of inputs, it is computationally demanding. Moreover, bridges are composed of elements belonging to different categories (beams, slabs, etc.), and a network-level analysis involves processing each one separately. Hence, each category requires identifying the best combination of explanatory variables for the kernel regression model, as well as expertise for selecting its hyperparameters.

While the aforementioned hybrid frameworks focused on including additional information in the deterioration analyses, either through physics-based models or by including explanatory variables, some hybrid frameworks combined two models where one simply aids the other in obtaining the best possible performance. For instance, Yosri et al. [39] coupled a genetic algorithm (GA) to refine the estimates of transition probabilities of a Markov model. Similarly, Alogdianakis et al. [40] combined a neural network with GA, using the latter to both find the best network architecture and select the optimal subset of explanatory variables. In such hybrid models, the underlying base model is not modified. The hybrid framework is merely used to optimize the base model's performance. As such, these frameworks still suffer from the limitations of the base models, be it a Markov model (Section 2.1.1) or a regression model (Section 2.1.2).

Hybrid models that counteract the limitations of visual inspections by integrating other sources of information have shown promising results. In particular, the approach based on state-space models and kernel regression has been validated on a network-scale, albeit with limitations [7]. Given that one of the objectives of this thesis is to develop a method that overcomes the shortcomings of this existing approach, it is covered in detail in the following section.

2.2.2 State-Space Models with Kernel Regression

Augmenting the state-space model (SSM) framework presented in Section 2.1.3 with a kernel-based regression method (KR) enables sharing information between structural elements. This reduces the dependence of SSM on visual inspections alone, which are limited in number for each structural element. The purpose of KR is to include structural attributes $\boldsymbol{z}$ in the deterioration analyses to improve the prior for each element's initial state $\boldsymbol{x}_0 = [x_0\ \dot{x}_0\ \ddot{x}_0]^\intercal$ by sharing information between structures. More precisely, KR predicts the deterioration rate $\dot{x}_0$ based on each element's attributes that are defined for each bridge by a vector $\boldsymbol{z}_j = [z_j^1\ \dots\ z_j^{\mathtt{Q}}]^\intercal$. The KR method involves discretizing each of the $\mathtt{Q}$ covariates' domains with $\mathtt{M}$ reference points, which are then permuted to form a grid of $\mathtt{N} \equiv \mathtt{M}^{\mathtt{Q}}$ points denoted by $\mathbf{G} = [\boldsymbol{g}_1\ \dots\ \boldsymbol{g}_{\mathtt{N}}]^\intercal \in \mathbb{R}^{\mathtt{N}\times\mathtt{Q}}$. Each i-th grid point in $\mathbf{G}$ corresponds to a unique combination of the reference points and is associated with a deterioration rate $\dot{x}_i \in \dot{\boldsymbol{x}}_{\boldsymbol{z}} = [\dot{x}_1\ \dots\ \dot{x}_{\mathtt{N}}]^\intercal$. KR estimates each element's deterioration rate $\dot{x}_0$ based on the proximity of its attributes $\boldsymbol{z}_j$ to different grid points using a weighted sum given by

$$\dot{x}_0 = \boldsymbol{a}^\intercal \dot{\boldsymbol{x}}_{\boldsymbol{z}} + w_0,\ w_0 : W_0 \sim \mathcal{N}\left(0, \sigma_{W_0}^2\right), \tag{2.3}$$

where w_0 is the process error and $\boldsymbol{a} = [a_1\, a_2\ \dots\ a_{\mathtt{N}}]^\intercal$ is the vector of weights, which is obtained using a multivariate kernel function $\boldsymbol{k}(\cdot)$ following

$$a_i = \frac{\boldsymbol{k}\left(\boldsymbol{z}_j, \mathbf{G}_{(i)}, \boldsymbol{\ell}\right)}{\sum_{n=1}^{\mathtt{N}} \boldsymbol{k}\left(\boldsymbol{z}_j, \mathbf{G}_{(n)}, \boldsymbol{\ell}\right)},\ i = 1, \dots, \mathtt{N}, \tag{2.4}$$

where $\boldsymbol{\ell} = [\ell_1 \dots \ell_{\mathtt{Q}}]^\intercal$ are the kernel bandwidths and $\mathbf{G}_{(i)}$ is the i-th point on the grid. The kernel function $\boldsymbol{k} : \mathbb{R}^{\mathtt{Q}} \to \mathbb{R}$ is defined as a product of univariate kernels $k(\cdot)$ given by

$$\boldsymbol{k}\left(\boldsymbol{z}_j, \mathbf{G}_{(i)}, \boldsymbol{\ell}\right) = k\left(\frac{z_j^1 - g_i^1}{\ell_1}\right) \cdot \ldots \cdot k\left(\frac{z_j^{\mathtt{Q}} - g_i^{\mathtt{Q}}}{\ell_{\mathtt{Q}}}\right).$$

The parameter estimation of the SSM-KR model is detailed in the work of Hamida and Goulet [7]. Although coupling SSM with KR leads to improved predictions of the deterioration state, this hybrid model suffers from several limitations that are presented next.

Limitations of the SSM-KR Model

Using SSM-KR to model infrastructure deterioration on a large scale is hindered by the lack of scalability and efficiency of the KR method. Namely, KR can only use a limited number of

structural attributes as its covariates, and even with a few covariates, it requires significant computational resources. Additionally, the performance of KR is highly dependent on the selection of its hyperparameters, such as kernel types, which requires expertise [7]. Despite these limitations, the KR method cannot be easily replaced with any other regression technique. Its replacement has to be capable of reporting prediction uncertainty to be compatible with SSM, as well as scalable and computationally efficient to overcome the limitations of KR. Finding a regression method that meets all these criteria is not trivial. For example, Gaussian process regression quantifies prediction uncertainty, but it is difficult to scale it given its non-parametric nature [20]. In contrast, Bayesian neural networks (BNN) are effective on larger datasets, but the inference of BNN parameters is dominated by sampling-based or gradient-based methods, making them computationally inefficient if coupled with the SSM framework. However, the method called tractable approximate Gaussian inference (TAGI) was recently developed by Goulet et al. [1], which enables analytical inference in BNN, making it a suitable candidate for replacing KR as the regression method in the proposed hybrid model. The next section provides an overview of the TAGI method.

2.3 Tractable Approximate Gaussian Inference

TAGI is a computationally efficient analytical method for inferring the parameters of a BNN [1]. Although TAGI can be employed to infer the parameters of various network architectures such as convolutional neural networks and generative adversarial networks [41], it is described here in the context of a fully-connected feedforward neural network (FNN), which is the architecture employed in this thesis.

A FNN consists of inputs $\boldsymbol{x} \in \mathbb{R}^{\mathtt{X}}$ and outputs $\boldsymbol{y} \in \mathbb{R}^{\mathtt{Y}}$ connected by $\mathtt{L}$ hidden layers, graphically depicted in Figure 2.3. These connections are established through matrix-vector multiplication and addition with the network parameters $\boldsymbol{\theta} = \{\mathbf{W}, \boldsymbol{b}\}$, followed by a non-linear activation function $\varphi(\cdot)$. Specifically, the hidden units $\boldsymbol{h}^{(j)} \in \mathbb{R}^{|j|}$ in layer j are used to obtain the hidden units in the next layer $\boldsymbol{h}^{(j+1)} \in \mathbb{R}^{|j+1|}$ following

$$\boldsymbol{h}^{(j+1)} = \mathbf{W}^{(j)} \left(\varphi(\boldsymbol{h}^{(j)})\right) + \boldsymbol{b}^{(j)}, \ j = 0, 1, \ldots, \mathtt{L}, \tag{2.5}$$

where $\mathbf{W}^{(j)} \in \mathbb{R}^{|j+1| \times |j|}$ and $\boldsymbol{b}^{(j)} \in \mathbb{R}^{|j+1|}$ are the weights and biases in layer j with $|j|$ and $|j+1|$ indicating the number of units in layers j and $j+1$, respectively. Note that the activated units in layer $j = 0$ are defined as $\boldsymbol{x} \equiv \varphi(\boldsymbol{h}^{(0)})$.

TAGI uses Equation 2.5 to analytically propagate uncertainty forward while making several assumptions and approximations. First, it treats the network parameters as independent

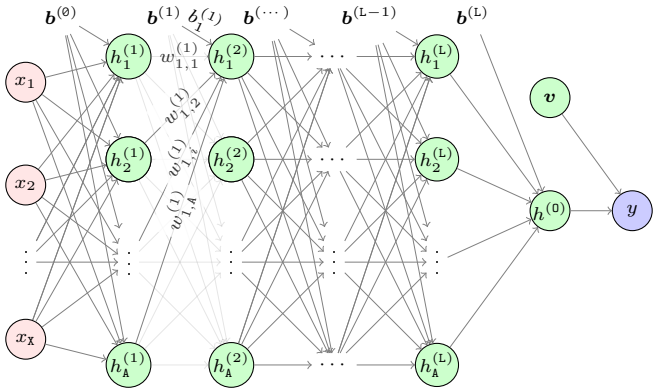


Figure 2.3 Graphical representation of a feedforward neural network consisting of $\mathtt{L}$ hidden layers with $\mathtt{A}$ hidden units in each one. The network maps the relation between the inputs $\boldsymbol{x} = [x_1 \cdots x_{\mathtt{X}}]^\intercal$ and the unit at the output layer $h^{(\mathtt{0})}$. To simplify visualization, the output is shown here as one-dimensional. The observation y is connected to the output layer unit $h^{(\mathtt{0})}$ and the error term v through the observation model defined in Equation 2.6. The parameters connecting layer j with layer $j+1$ consist of a vector of biases $\boldsymbol{b}^{(j)}$ and a matrix of weights $\mathbf{W}^{(j)}$, such that $w^{(j)}_{m,i} \in \mathbf{W}^{(j)}$ represents the weight connecting the i-th unit in layer j to the m-th unit in layer $j+1$. This figure is adapted from [1].

Gaussian random variables, so that $\boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \mathbf{I} \cdot \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}\right)$. TAGI then uses the Gaussian Multiplicative Approximation (GMA) [1,42] to approximate the product of two Gaussian random variables as a Gaussian. Finally, it locally linearizes the activation function $\varphi(\cdot)$ at the expected value of its input and assumes that the hidden units within a same layer are independent. The justifications for these assumptions and approximations, along with their detailed derivations can be found in [1]. The relation between the network's outputs $\boldsymbol{h}^{(\mathtt{0})}$ and the observed system responses $\boldsymbol{y}$ is described by the following observation model,

$$\boldsymbol{y} = \boldsymbol{h}^{(\mathtt{0})} + \boldsymbol{v}, \;\; \boldsymbol{v} : \boldsymbol{V} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I} \cdot \boldsymbol{\sigma}^{\mathbf{2}}_{\boldsymbol{V}}\right), \tag{2.6}$$

where $\boldsymbol{v}$ represents independent zero-mean Gaussian errors. TAGI performs closed-form inference in several stages. First, it leverages the properties of Gaussian conditional equations to infer the posterior expected value and covariance for the hidden units in the output layer $\boldsymbol{h}^{(\mathtt{0})}$ given the observations $\boldsymbol{y}$, such that,

$$\begin{aligned} f(\boldsymbol{h}^{(\mathtt{0})}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{h}^{(\mathtt{0})}; \boldsymbol{\mu}_{\boldsymbol{H}^{(\mathtt{0})}|\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{H}^{(\mathtt{0})}|\boldsymbol{y}}), \\ \boldsymbol{\mu}_{\boldsymbol{H}^{(\mathtt{0})}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\boldsymbol{H}^{(\mathtt{0})}} + \boldsymbol{\Sigma}^{\intercal}_{\boldsymbol{Y}\boldsymbol{H}^{(\mathtt{0})}} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{Y}} \left(\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{Y}}\right), \\ \boldsymbol{\Sigma}_{\boldsymbol{H}^{(\mathtt{0})}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\boldsymbol{H}^{(\mathtt{0})}} - \boldsymbol{\Sigma}^{\intercal}_{\boldsymbol{Y}\boldsymbol{H}^{(\mathtt{0})}} \boldsymbol{\Sigma}^{-1}_{\boldsymbol{Y}} \boldsymbol{\Sigma}_{\boldsymbol{Y}\boldsymbol{H}^{(\mathtt{0})}}. \end{aligned} \tag{2.7}$$

Next, TAGI recursively infers the posterior for the hidden units $\boldsymbol{h}^{(j)}$ and parameters $\boldsymbol{\theta}^{(j)}$ on each layer j using (1) a diagonal structure for the covariance of the parameters $\boldsymbol{\theta}$ and the hidden layers $\boldsymbol{h}^{(j)}$ and (2) the inherent conditional independence between the hidden layers embedded in the FNN structure, i.e., $\boldsymbol{H}^{(j-1)} \perp \boldsymbol{H}^{(j+1)} | \boldsymbol{h}^{(j)}$. This layer-wise inference is performed using the Rauch-Tung-Striebel (RTS) smoother [34] as follows,

$$
\begin{aligned}
f(\boldsymbol{h}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{h}; \boldsymbol{\mu}_{\boldsymbol{H}|\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{H}|\boldsymbol{y}}), \\
\boldsymbol{\mu}_{\boldsymbol{H}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\boldsymbol{H}} + \mathbf{J}_{\boldsymbol{H}} \left(\boldsymbol{\mu}_{\boldsymbol{H}^+|\boldsymbol{y}} - \boldsymbol{\mu}_{\boldsymbol{H}^+} \right), \\
\boldsymbol{\Sigma}_{\boldsymbol{H}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\boldsymbol{H}} + \mathbf{J}_{\boldsymbol{H}} \left(\boldsymbol{\Sigma}_{\boldsymbol{H}^+|\boldsymbol{y}} - \boldsymbol{\Sigma}_{\boldsymbol{H}^+} \right) \mathbf{J}_{\boldsymbol{H}}^{\intercal}, \\
\mathbf{J}_{\boldsymbol{H}} &= \boldsymbol{\Sigma}_{\boldsymbol{H}\boldsymbol{H}^+} \boldsymbol{\Sigma}_{\boldsymbol{H}^+}^{-1},
\end{aligned}
$$

$$
\begin{aligned}
f(\boldsymbol{\theta}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\boldsymbol{\theta}|\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}|\boldsymbol{y}}), \\
\boldsymbol{\mu}_{\boldsymbol{\theta}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\boldsymbol{\theta}} + \mathbf{J}_{\boldsymbol{\theta}} \left(\boldsymbol{\mu}_{\boldsymbol{H}^+|\boldsymbol{y}} - \boldsymbol{\mu}_{\boldsymbol{H}^+} \right), \\
\boldsymbol{\Sigma}_{\boldsymbol{\theta}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}} + \mathbf{J}_{\boldsymbol{\theta}} \left(\boldsymbol{\Sigma}_{\boldsymbol{H}^+|\boldsymbol{y}} - \boldsymbol{\Sigma}_{\boldsymbol{H}^+} \right) \mathbf{J}_{\boldsymbol{\theta}}^{\intercal}, \\
\mathbf{J}_{\boldsymbol{\theta}} &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{H}^+} \boldsymbol{\Sigma}_{\boldsymbol{H}^+}^{-1},
\end{aligned}
$$

where the following shorthand notations are defined for simplicity: $\{\boldsymbol{\theta}^+, \boldsymbol{H}^+\} \equiv \{\boldsymbol{\theta}^{(j+1)}, \boldsymbol{H}^{(j+1)}\}$ and $\{\boldsymbol{\theta}, \boldsymbol{H}\} \equiv \{\boldsymbol{\theta}^{(j)}, \boldsymbol{H}^{(j)}\}$. Given that TAGI parameters $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{B}\}$ are initialized with a weakly informed prior, the inference is performed over multiple successive passes over the training set. To prevent overfitting, this iterative learning process is stopped once the performance on the validation set degrades.

Despite TAGI providing closed-form inference for the parameters of BNN while matching state-of-the-art performance on various regression tasks [1], it treats the error variance $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$ in Equation 2.6 as a hyperparameter. This requires identifying an optimal value for $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$ separately from the analytical framework, which is computationally expensive. It also restricts $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$ to be homoscedastic, i.e., constant across the input covariates. These limitations were addressed in the work of Deka [2], which is presented next.

2.4 Observation Error Variance Inference with TAGI

TAGI can be extended to infer the error variance $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$ from the observation model in Equation 2.6 by coupling it with the Approximate Gaussian Variance Inference (AGVI) method [2, 43]. AGVI establishes the relation between the error variance $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$, the error $\boldsymbol{v}$, and the square of the error $\boldsymbol{v}^2$. The key relation is formed by noting that the expected value for the square of the error $\boldsymbol{v}^2$ is equal to the variance of the error $\boldsymbol{v}$, and treating this variance as a

Gaussian random variable. Using the definition of variance to express this statement as an equation yields

$$\text{var}\left[\boldsymbol{V}\right] = \mathbb{E}\left[\boldsymbol{V}^2\right] - \cancelto{\mathbf{0}}{\mathbb{E}\left[\boldsymbol{V}\right]^2} = \boldsymbol{\sigma}_{\boldsymbol{V}}^2 \equiv \overline{\boldsymbol{v}^2} : \overline{\boldsymbol{V}^2} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}}, \mathbf{I} \cdot \boldsymbol{\sigma}_{\overline{\boldsymbol{V}^2}}^2\right),$$

where we leverage the definition of $\boldsymbol{v}$ as a zero-mean Gaussian error with a variance of $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$, and denote the expected value of $\boldsymbol{v}^2$ as $\overline{\boldsymbol{v}^2}$. Establishing this relation enables AGVI to connect the priors of $\overline{\boldsymbol{v}^2}$, $\boldsymbol{v}^2$, and $\boldsymbol{v}$ as follows,

$$f\left(\overline{\boldsymbol{v}^2}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}}, \mathbf{I} \cdot \boldsymbol{\sigma}_{\overline{\boldsymbol{V}^2}}^2\right), \qquad \text{Expected value of square of errors} \qquad (2.8)$$

$$f\left(\boldsymbol{v}^2 \mid \overline{\boldsymbol{v}^2}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}}, \mathbf{I} \cdot \left(3\boldsymbol{\sigma}_{\overline{\boldsymbol{V}^2}}^2 + 2\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}}^2\right)\right), \qquad \text{Square of errors} \qquad (2.9)$$

$$f(\boldsymbol{v} \mid \overline{\boldsymbol{v}^2}) = \mathcal{N}(\mathbf{0}, \mathbf{I} \cdot \boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}}). \qquad \text{Errors} \qquad (2.10)$$

Figure 2.4 illustrates these connections as a directed acyclic graph (DAG). The detailed derivation of these relations is presented in the work of Deka and Goulet [2, 43].

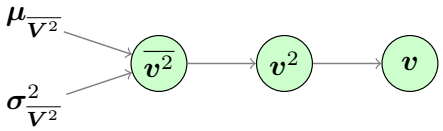


Figure 2.4 Graphical representation of the relation between the error $\boldsymbol{v}$, the square of the error $\boldsymbol{v}^2$, and the expected value of the square of the error $\overline{\boldsymbol{v}^2}$, which models the error variance $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$. The arrows indicate the dependence between the units. This figure is adapted from [2].

AGVI is coupled with TAGI by introducing additional hidden units in the neural network's output layer. Each additional unit corresponds to the expected value of the square of the error $\overline{v}_i^2 \in \overline{\boldsymbol{v}^2} = [\overline{v}_1^2 \cdots \overline{v}_{\text{Y}}^2]^\intercal$ which models the error variance $\sigma_{V_i}^2 \in \boldsymbol{\sigma}_{\boldsymbol{V}}^2 = [\sigma_{V_1}^2 \cdots \sigma_{V_{\text{Y}}}^2]^\intercal$ associated with the observation $y_i \in \boldsymbol{y} = [y_1 \cdots y_{\text{Y}}]^\intercal$. This formulation allows the error variance $\boldsymbol{\sigma}_{\boldsymbol{V}}^2$ to be treated as heteroscedastic, i.e., to vary across the input covariates. From Equations 2.8–2.10, the forward propagation of uncertainty through the DAG in Figure 2.4 relies only on the prior mean and variance of $\overline{\boldsymbol{v}^2}$, which are predicted by the neural network. More precisely, $\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}}$ and $\boldsymbol{\sigma}_{\overline{\boldsymbol{V}^2}}^2$ correspond to the mean and variance of the additional hidden units from the output layer of the network. For instance, if the target variable is two-dimensional, then the output layer of the network will have four units, where the first two correspond to the prior prediction for the expected value of the observation $\boldsymbol{y}$ and the last two correspond to the prior prediction for the expected value for the square of the error $\overline{\boldsymbol{v}^2}$. Figure 2.5 illustrates how the DAG from Figure 2.4 is integrated into the neural network architecture with a slight

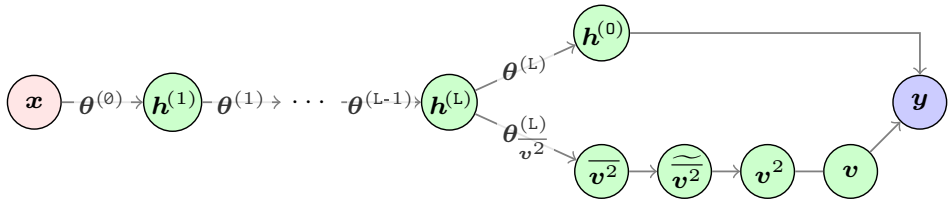


Figure 2.5 Compact representation of the network architecture from Figure 2.3 coupled with the DAG from Figure 2.4. The network parameters connecting layer j to layer $j+1$ are denoted as $\boldsymbol{\theta}^{(j)}$. This figure is adapted from [2].

modification. Namely, the output units of the network corresponding to $\overline{\boldsymbol{v}^2}$ are restricted to the positive domain using an exponential activation function, which yields $\widetilde{\overline{\boldsymbol{v}^2}} = \exp(\overline{\boldsymbol{v}^2})$. This is done because $\overline{\boldsymbol{v}^2}$ models the observation error's variance $\boldsymbol{\sigma}^2_{\boldsymbol{V}}$, which must be positive. When $\overline{\boldsymbol{v}^2}$ is treated as a Gaussian, its transformation $\widetilde{\overline{\boldsymbol{v}^2}}$ is described by a log-normal PDF [2], whose moments can be computed analytically as follows,

$$\boldsymbol{\mu}_{\widetilde{\overline{\boldsymbol{V}^2}}} = \exp\left(\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}} + \tfrac{1}{2}\boldsymbol{\sigma}^2_{\overline{\boldsymbol{V}^2}}\right), \tag{2.11}$$

$$\boldsymbol{\sigma}^2_{\widetilde{\overline{\boldsymbol{V}^2}}} = \left(\exp\left(\boldsymbol{\sigma}^2_{\overline{\boldsymbol{V}^2}}\right) - 1\right) \odot \exp\left(2\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}} + \boldsymbol{\sigma}^2_{\overline{\boldsymbol{V}^2}}\right), \tag{2.12}$$

where $\odot$ denotes element-wise multiplication. To preserve the relations in Equations 2.8–2.10, $\widetilde{\overline{\boldsymbol{v}^2}}$ is approximated as a Gaussian random variable.

Inferring the error variance $\boldsymbol{\sigma}^2_{\boldsymbol{V}}$ using the network architecture shown in Figure 2.5 requires modifying the hidden vector $\boldsymbol{h}^{(0)}$ in the Gaussian conditional update defined in Equation 2.7. Specifically, the hidden vector $\boldsymbol{h}^{(0)}$ is concatenated with the error vector $\boldsymbol{v}$, forming the augmented hidden vector $\boldsymbol{h} = [\boldsymbol{h}^{(0)}; \boldsymbol{v}]$. The posterior mean and covariance of the augmented hidden vector $\boldsymbol{h}$ are then computed using the Gaussian conditional equations in the same manner as in Equation 2.7, which yields

$$\boldsymbol{\mu}_{\boldsymbol{H}|\boldsymbol{y}} = \boldsymbol{\mu}_{\boldsymbol{H}} + \boldsymbol{\Sigma}^{\intercal}_{\boldsymbol{YH}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{Y}}\left(\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{Y}}\right), \tag{2.13}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{H}|\boldsymbol{y}} = \boldsymbol{\Sigma}_{\boldsymbol{H}} - \boldsymbol{\Sigma}^{\intercal}_{\boldsymbol{YH}}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{Y}}\boldsymbol{\Sigma}_{\boldsymbol{YH}}. \tag{2.14}$$

Following this, the posterior PDF of the error $f(\boldsymbol{v}|\boldsymbol{y})$ is marginalized from the joint PDF $f(\boldsymbol{h}^{(0)}, \boldsymbol{v}|\boldsymbol{y})$ defined by Equations 2.13–2.14. The posterior moments of the error vector $\boldsymbol{v}$ are then used to obtain the posterior moments of the square of the error $\boldsymbol{v}^2$ using the Gaussian

multiplicative approximation (GMA) [42], resulting in

$$\boldsymbol{\mu}_{\boldsymbol{V}^2|\boldsymbol{y}} = \boldsymbol{\mu}^2_{\boldsymbol{V}|\boldsymbol{y}} + \boldsymbol{\sigma}^2_{\boldsymbol{V}|\boldsymbol{y}}, \tag{2.15}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{V}^2|\boldsymbol{y}} = 2\left(\boldsymbol{\sigma}^2_{\boldsymbol{V}|\boldsymbol{y}}\right)^2 + 4\boldsymbol{\sigma}^2_{\boldsymbol{V}|\boldsymbol{y}} \odot \boldsymbol{\mu}^2_{\boldsymbol{V}|\boldsymbol{y}}. \tag{2.16}$$

Continuing our way through the DAG from Figure 2.5, the posterior of $\widetilde{\overline{\boldsymbol{v}^2}}$ is found from the prior and posterior information about the square of the error $\boldsymbol{v}^2$ using the RTS smoother [34],

$$\begin{aligned}
\boldsymbol{\mu}_{\widetilde{\overline{\boldsymbol{V}^2}}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\widetilde{\overline{\boldsymbol{V}^2}}} + \mathbf{J}_{\widetilde{\overline{\boldsymbol{V}^2}}}(\boldsymbol{\mu}_{\boldsymbol{V}^2|\boldsymbol{y}} - \boldsymbol{\mu}_{\boldsymbol{V}^2}), \\
\boldsymbol{\Sigma}_{\widetilde{\overline{\boldsymbol{V}^2}}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\widetilde{\overline{\boldsymbol{V}^2}}} + \mathbf{J}_{\widetilde{\overline{\boldsymbol{V}^2}}}(\boldsymbol{\Sigma}_{\boldsymbol{V}^2|\boldsymbol{y}} - \boldsymbol{\Sigma}_{\boldsymbol{V}^2})\mathbf{J}^{\intercal}_{\widetilde{\overline{\boldsymbol{V}^2}}}, \\
\mathbf{J}_{\widetilde{\overline{\boldsymbol{V}^2}}} &= \boldsymbol{\Sigma}_{\widetilde{\overline{\boldsymbol{V}^2}}\boldsymbol{V}^2}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{V}^2},
\end{aligned}$$

where the covariance between the square of the error $\boldsymbol{v}^2$ and $\widetilde{\overline{\boldsymbol{v}^2}}$ is simply $\boldsymbol{\Sigma}_{\widetilde{\overline{\boldsymbol{V}^2}}\boldsymbol{V}^2} = \boldsymbol{\Sigma}_{\widetilde{\overline{\boldsymbol{V}^2}}}$ [2]. Finally, the posterior of $\overline{\boldsymbol{v}^2}$, which models the error variance $\boldsymbol{\sigma}^2_{\boldsymbol{V}}$, is found from the prior and posterior information about the transformed random variable $\widetilde{\overline{\boldsymbol{v}^2}}$. The posterior moments of $\overline{\boldsymbol{v}^2}$ are given by

$$\begin{aligned}
\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}|\boldsymbol{y}} &= \boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}} + \mathbf{J}_{\overline{\boldsymbol{V}^2}}(\boldsymbol{\mu}_{\widetilde{\overline{\boldsymbol{V}^2}}|\boldsymbol{y}} - \boldsymbol{\mu}_{\widetilde{\overline{\boldsymbol{V}^2}}}), \\
\boldsymbol{\Sigma}_{\overline{\boldsymbol{V}^2}|\boldsymbol{y}} &= \boldsymbol{\Sigma}_{\overline{\boldsymbol{V}^2}} + \mathbf{J}_{\overline{\boldsymbol{V}^2}}(\boldsymbol{\Sigma}_{\widetilde{\overline{\boldsymbol{V}^2}}|\boldsymbol{y}} - \boldsymbol{\Sigma}_{\widetilde{\overline{\boldsymbol{V}^2}}})\mathbf{J}^{\intercal}_{\overline{\boldsymbol{V}^2}}, \\
\mathbf{J}_{\overline{\boldsymbol{V}^2}} &= \boldsymbol{\Sigma}_{\overline{\boldsymbol{V}^2}\widetilde{\overline{\boldsymbol{V}^2}}}\boldsymbol{\Sigma}^{-1}_{\widetilde{\overline{\boldsymbol{V}^2}}},
\end{aligned}$$

where the covariance between the transformed random variable $\widetilde{\overline{\boldsymbol{v}^2}}$ and the original random variable $\overline{\boldsymbol{v}^2}$ is

$$\boldsymbol{\Sigma}_{\overline{\boldsymbol{V}^2}\widetilde{\overline{\boldsymbol{V}^2}}} = \mathbf{I} \cdot \left(\boldsymbol{\sigma}_{\overline{\boldsymbol{V}^2}} \odot \exp(\boldsymbol{\mu}_{\overline{\boldsymbol{V}^2}} + \tfrac{1}{2}\boldsymbol{\sigma}^2_{\overline{\boldsymbol{V}^2}})\right).$$

The remainder of TAGI's inference procedure remains unchanged. Namely, the posteriors for the hidden units $\boldsymbol{h}^{(j)}$ and the parameters $\boldsymbol{\theta}^{(j)}$ for each layer j are inferred in a recursive manner as detailed in Section 2.3. Coupling TAGI with AGVI, henceforth simply referred to as TAGI, enables analytical inference of both the heteroscedastic observation error variance and the neural network's parameters and hidden units.

Figure 2.6 illustrates an application of TAGI to a one-dimensional heteroscedastic regression problem. Such variation in system response over the input space x is typically observed in real-world applications due to lack of access to all the explanatory variables [2]. Note

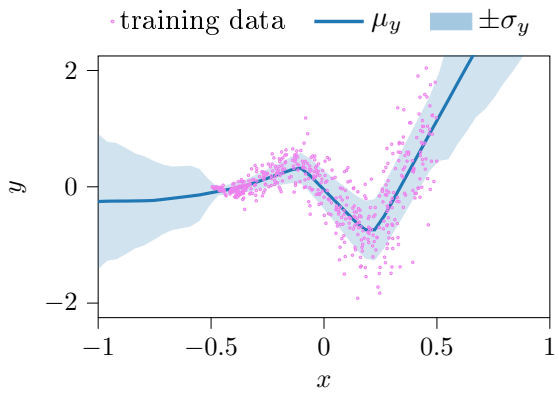


Figure 2.6 Illustration of TAGI's capacity to learn a mapping to a target variable with heteroscedastic variance. This figure is adapted from [2].

that the regression problem tackled in this thesis is significantly more complex than the one shown in Figure 2.6; contrary to this simple example, the objective is predicting a target variable that is not directly observable. Learning such a latent relation is challenging as the target variable (i.e., deterioration speed) used for training the regression method has to be inferred from a limited amount of data, which consists of few highly uncertain visual inspections corresponding to the deterioration condition of an element. Nonetheless, the next chapter presents how the proposed infrastructure deterioration model is able to learn the aforementioned latent relation by using TAGI.

CHAPTER 3 Hybrid Framework for Modeling Infrastructure Deterioration

The hybrid framework described in Section 2.2.2 combines a state-space model (SSM) with the kernel regression (KR) to model infrastructure deterioration. However, the SSM-KR approach has several limitations that make it impractical for large-scale applications. Specifically, it is limited to using few structural attributes, lacks computational efficiency, and requires fine-tuning to perform effectively. This chapter presents a new hybrid framework that addresses these limitations. The proposed model couples the SSM method from Section 2.1.3 with a Bayesian neural network (BNN) that employs TAGI (Section 2.4) as an inference engine. This new SSM-BNN approach offers significant improvements over SSM-KR in terms of scalability, computational efficiency, and ease of use.

This chapter starts with presenting the methodology of the new SSM-BNN model in Section 3.1. Following this, Section 3.2 outlines all the parameters of the new model. Finally, Section 3.3 presents the estimation procedure for the model parameters.

3.1 Modeling Infrastructure Deterioration with SSM-BNN

In the proposed SSM-BNN model, the SSM backbone (Section 2.1.3) models the infrastructure deterioration, while the BNN component is used to enhance the SSM predictions by learning the relation between the attributes of structural elements (age, location, etc.) and their deterioration rates. Thereafter, this learned relation is used to define better priors for the initial deterioration rates of elements. To illustrate this, let us examine how a fully-trained SSM-BNN estimates the evolution of the deterioration state of a p-th structural element from the j-th bridge, denoted by e_p^j.

The estimation of the deterioration state of a given element starts with transforming its visual inspection data $\tilde{\boldsymbol{y}}_{t,p}^j = [\tilde{y}_{1,p}^j \ \dots \ \tilde{y}_{\mathsf{T},p}^j]^\intercal \in [l, u]$ into an unbounded domain $\boldsymbol{y}_{t,p}^j = [y_{1,p}^j \ \dots \ y_{\mathsf{T},p}^j]^\intercal \in (-\infty, \infty)$ using the function $o(\cdot)$ detailed in Appendix A. Here, l represents the worst damage state and u represents the perfect condition. The transformed observations $\boldsymbol{y}_{t,p}^j$ are then passed to the SSM, which relies on the Kalman filter (KF) to estimate the element's deterioration state over time. The KF starts with a guess about the element's initial state and propagates this prior knowledge forward in time using the transition model, generating predictions that are updated using the observations (see Section 2.1.3). Defining the initial state of an element $\boldsymbol{x}_{0,p}^j = [x_{0,p}^j \ \dot{x}_{0,p}^j \ \ddot{x}_{0,p}^j]^\intercal$ is an important aspect that affects the quality of the KF predictions. Much like the SSM-KR model presented in Section 2.2.2,

the SSM-BNN model improves initialization of the KF for each element e_p^j by predicting its initial deterioration rate $\dot{x}_{0,p}^j$ based on its structural attributes $\boldsymbol{z}_p^j$. The SSM-BNN method defines the full initial state of each element as follows,

$$\begin{bmatrix} x_{0,p}^j \\ \dot{x}_{0,p}^j \\ \ddot{x}_{0,p}^j \end{bmatrix} \sim \mathcal{N}\left(\overbrace{\begin{bmatrix} \mu_{0,p}^j \\ \dot{\mu}_{\boldsymbol{z}} \\ 0 \end{bmatrix}}^{\boldsymbol{\mu}_{0,p}^j}, \overbrace{\begin{bmatrix} \sigma_{0,p}^j & 0 & 0 \\ 0 & \dot{\sigma}_{\boldsymbol{z}} & 0 \\ 0 & 0 & \ddot{\sigma}_0 \end{bmatrix}^2}^{\boldsymbol{\Sigma}_{0,p}^j} \right). \tag{3.1}$$

Here, $\mu_{0,p}^j$ and $\sigma_{0,p}^j$ correspond to the expected value and standard deviation of the element's initial condition, which are defined as

$$\mu_{0,p}^j \doteq y_{1,p}^j, \tag{3.2}$$

$$\sigma_{0,p}^j \doteq \max\left\{\sigma_0, \sigma_{V,t=1}\right\}, \tag{3.3}$$

where $y_{1,p}^j$ is the first observation on the element, $\sigma_{V,t=1}$ is the standard deviation corresponding to the uncertainty of the inspector that performed the first observation, and σ_0 is the global standard deviation of the initial condition. The prior uncertainty of each element's initial condition $\sigma_{0,p}^j$ is defined with a maximum operation to prevent having a prior with a variance lower than the first observation. The expected value and standard deviation of the element's initial deterioration speed, denoted by $\dot{\mu}_{\boldsymbol{z}}$ and $\dot{\sigma}_{\boldsymbol{z}}$, are predicted by the BNN based on the element's attributes $\boldsymbol{z}_p^j$:

$$(\dot{\mu}_{\boldsymbol{z}}, \dot{\sigma}_{\boldsymbol{z}}) \doteq \text{BNN}\left(\boldsymbol{z}_p^j\right).$$

These values define the prior knowledge for the deterioration speed and are clipped as $\dot{\mu}_{\boldsymbol{z}} \in [-3, 0]$ and $\dot{\sigma}_{\boldsymbol{z}} \in [0.05, 0.5]$ to keep the predictions of the BNN within a reasonable range and ensure numerical stability of the deterioration model. Finally, the element's acceleration state is initialized with an expected value of zero and the global standard deviation parameter $\ddot{\sigma}_0$. In contrast to the parameters defined on an element-by-element basis, the global parameters are shared among all the elements.

After defining the element's initial state $\boldsymbol{x}_{0,p}^j$, it is passed into the KF, together with the element's transformed observations $\boldsymbol{y}_{t,p}^j$. The KF then predicts the deterioration state of the element over time $\{\boldsymbol{x}_{0,p}^j, \ldots, \boldsymbol{x}_{\tau,p}^j\}$, where τ denotes the time-horizon of the prediction. At each time step of the KF, the deterioration rate $\dot{x}_{t,p}^j$ is restricted to be negative by requiring that $\dot{\mu}_{t,p}^j + 2\dot{\sigma}_{t,p}^j \leq 0$ [3]. If this condition is not satisfied, the deterioration rate is constrained to the negative domain by truncating its PDF [3, 35]. Following the completion of the KF,

the Kalman smoother (KS) is used to refine the KF state estimates. The smoothed state estimates are then back-transformed to the original space $\{\tilde{\boldsymbol{x}}^j_{0,p}, \ldots, \tilde{\boldsymbol{x}}^j_{\tau,p}\}$ using the inverse of the transformation function $o(\cdot)^{-1}$ (see Appendix A). Figure 3.1 graphically illustrates the overview of this process while omitting the aforementioned finer details.

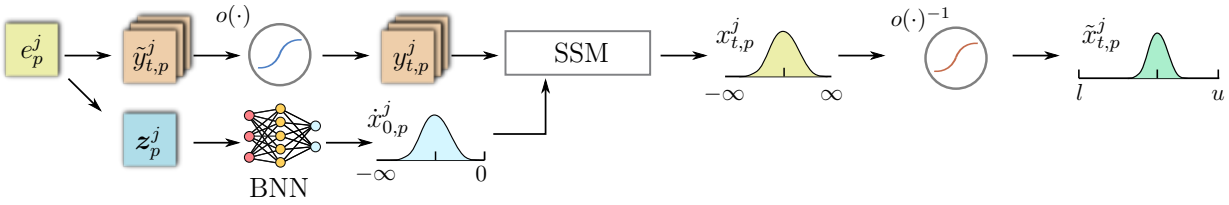


Figure 3.1 Overview of the procedure for estimating the deterioration state of the p-th element from the j-th bridge, denoted by e^j_p, using the proposed SSM-BNN framework. The original visual inspections of the element $\tilde{y}^j_{t,p}$ are first transformed into an unbounded domain using the sigmoid function $o(\cdot)$, yielding the transformed observations $y^j_{t,p}$. The BNN then uses the structural attributes of the element $\boldsymbol{z}^j_p$ to define its initial deterioration rate $\dot{x}^j_{0,p}$. The initial state $\boldsymbol{x}^j_{0,p}$ and the transformed observations $y^j_{t,p}$ are then passed into SSM to predict the element's deterioration state over time $\boldsymbol{x}^j_{t,p} = [x^j_{t,p}\ \dot{x}^j_{t,p}\ \ddot{x}^j_{t,p}]^\intercal$ for $t \in \{1, 2, \ldots, \tau\}$, where τ denotes the time-horizon of the prediction. These predicted states are finally back-transformed to the original space using $o^{-1}(\cdot)$, resulting in $\tilde{\boldsymbol{x}}^j_{t,p} = [\tilde{x}^j_{t,p}\ \tilde{\dot{x}}^j_{t,p}\ \tilde{\ddot{x}}^j_{t,p}]^\intercal$.

3.2 SSM-BNN Model Parameters

The parameters of the SSM-BNN model include the inspectors' relative biases $\mu_{V_{1:\mathrm{I}}}$ and standard deviations $\sigma_{V_{1:\mathrm{I}}}$ that characterize the observation errors (Equation 2.2), the standard deviation of the process noise error σ_W that characterizes the imperfection of the transition model (Equation 2.1), and the standard deviations defining the prior of the initial state $\{\sigma_0, \dot{\sigma}_0, \ddot{\sigma}_0\}$ for the Kalman filter. In this framework, the standard deviation of the initial deterioration rate $\dot{\sigma}_0$ is described by the parameters p_1 and p_2 as follows,

$$\dot{\sigma}_0^2 = p_1^2\,(u - \tilde{\mu}_1) + p_2^2, \tag{3.4}$$

where $\tilde{\mu}_1$ represents the expected value of the condition at $t = 1$. This relation is intended to improve the estimation of the deterioration rate given that each element has a low number of inspections over time [3]. The intention is to counteract the low number of observations in the time series by incorporating an informative prior, so that the elements having a near perfect condition ($u - \tilde{\mu}_1 \approx 0$) will start with a lower uncertainty about their deterioration rate compared to worn out elements, whose uncertainty is defined in proportion to their condition.

The value of $\widetilde{\mu}_1$ is initially set to the first observation y_1, but once the smoothed estimates are obtained, it is redefined as the smoothed estimate for the condition at the first time step $\widetilde{\mu}_1 = \mu_{1|\mathtt{T}}$. As a result, the Kalman filter and smoother need to be run twice: the first time to get the smoothed estimates and define $\widetilde{\mu}_1$, and the second time to get the final smoothed estimates. The relation in Equation 3.4 was proposed by Hamida and Goulet [3] based on experiments with the real and synthetic data. The remaining parameters of the SSM-BNN include the weights and biases of the BNN represented by $\boldsymbol{\theta}_{\mathtt{BNN}}$ and the space transformation parameter n, which governs the curvature of the transformation function used to bound the hidden states and observations (see Appendix A). All the model parameters are compactly organized in the following set,

$$\boldsymbol{\theta} = \{\mu_{V_{1:\mathtt{I}}}, \sigma_{V_{1:\mathtt{I}}}, \sigma_W, \sigma_0, \ddot{\sigma}_0, p_1, p_2, \boldsymbol{\theta}_{\mathtt{BNN}}, n\}.$$

3.3 Parameter Estimation Procedure

This section presents an overview of the estimation procedure for all the model parameters, followed by a more detailed exposition of the recursive process for estimating the BNN parameters.

3.3.1 Estimation of the Global Model Parameters

The parameters of the proposed SSM-BNN model are estimated using a combination of gradient-based optimization and approximate Bayesian inference [12]. The estimation of $\boldsymbol{\theta}$ is divided into three stages: 1) estimation of the inspectors' parameters $\boldsymbol{\theta}_{\mathtt{I}} = \{\mu_{V_{1:\mathtt{I}}}, \sigma_{V_{1:\mathtt{I}}}\}$, 2) estimation of the process noise and initial state parameters $\boldsymbol{\theta}_{\mathtt{S}} = \{\sigma_W, \sigma_0, \ddot{\sigma}_0, p_1, p_2\}$, and 3) estimation of the BNN parameters $\boldsymbol{\theta}_{\mathtt{BNN}}$. These parameters are estimated from multiple passes over the data, which is split into training, validation, and test sets. The data is split while ensuring that the elements of a same structure cannot be simultaneously in the training set and the validation or test sets. The number of passes over the data is controlled by measuring performance using the likelihood on the validation set. The likelihood function corresponds to the joint prior predictive probability of all the observations on the network of bridges, given the model parameters $\boldsymbol{\theta}$, and assuming that the observations are conditionally independent given the hidden states $\boldsymbol{x}_{t,p}^{j}$. The natural logarithm of the likelihood is taken to ensure numerical stability, resulting in the following log-likelihood function

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{j=1}^{\mathtt{B}} \sum_{p=1}^{\mathtt{E}_j} \sum_{t=1}^{\mathtt{T}_p} \ln f(y_{t,p}^{j} | y_{1:t-1,p}^{j}, \boldsymbol{\theta}),$$

where $\mathtt{B}$ denotes the total number of bridges in the network, $\mathtt{E}_j$ denotes the number of structural elements in the j-th bridge, and $\mathtt{T}_p$ denotes the number of inspections for the p-th element in the j-th bridge. The log-likelihood of the validation set is evaluated after each estimation stage ($\boldsymbol{\theta}_{\mathtt{I}}$, $\boldsymbol{\theta}_{\mathtt{S}}$, $\boldsymbol{\theta}_{\mathtt{BNN}}$). If there is no improvement, the parameter update at that stage is discarded.

To illustrate the full estimation procedure, let us examine it with a single pass over the data. The estimation procedure starts with learning $\boldsymbol{\theta}_{\mathtt{S}}$ and σ_V with the Newton-Raphson (NR) gradient optimization algorithm [3]. Here, the σ_V parameter is the standard deviation assigned to each inspector. In this initial step, all inspectors are assumed to be unbiased ($\mu_{V_{1:\mathtt{I}}} = 0$), each having the same observation error $\mathcal{N}(0, \sigma_V^2)$. The optimization is carried out using NR to maximize the log-likelihood $\mathcal{L}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}_{\mathtt{S}}$ and σ_V. After completing this stage, the inspectors' parameters $\boldsymbol{\theta}_{\mathtt{I}}$ are learned while keeping the rest of the model parameters fixed. The inspectors' parameters are estimated using the Approximate Gaussian Variance Inference (AGVI) method presented in Section 2.4 in the context of Bayesian neural networks. For details related to using AGVI to estimate inspectors' uncertainty, refer to the work of Laurent et al. [32]. Following the estimation of the inspectors' parameters $\boldsymbol{\theta}_{\mathtt{I}}$, the estimates of the SSM parameters $\boldsymbol{\theta}_{\mathtt{S}}$ are refined using NR, as in the initial step above. The final stage involves estimating the parameters of the BNN $\boldsymbol{\theta}_{\mathtt{BNN}}$ while keeping $\boldsymbol{\theta}_{\mathtt{I}}$ and $\boldsymbol{\theta}_{\mathtt{S}}$ fixed. The estimation of $\boldsymbol{\theta}_{\mathtt{BNN}}$ relies on the Tractable Approximate Gaussian Inference (TAGI) method (Sections 2.3–2.4) employed in a recursive manner, which is detailed in the next section. The sequential estimation of $\boldsymbol{\theta}_{\mathtt{I}}$, $\boldsymbol{\theta}_{\mathtt{S}}$, and $\boldsymbol{\theta}_{\mathtt{BNN}}$ is repeated until the improvement in the validation-set's log-likelihood is negligible. Since the number of options for the transformation parameter n is limited, its optimal value is determined by repeating the full estimation procedure for all of its possible values $n \in \{1, 2, 3, 4, 5\}$. The full estimation procedure is summarized by the Algorithm 1.

Algorithm 1: The pseudocode for estimation of the model parameters $\boldsymbol{\theta}$.

for $n \in \{1, 2, 3, 4, 5\}$ **do**
  optimize $\{\sigma_W, \sigma_V, \sigma_0, \ddot{\sigma}_0, p_1, p_2\}$ using **NR**
  while validation set log-likelihood improvement $\geq 0.1\%$ **do**
    optimize $\boldsymbol{\theta}_{\mathtt{I}} = \{\mu_{V_{1:I}}, \sigma_{V_{1:I}}\}$ using **AGVI**
    optimize $\boldsymbol{\theta}_{\mathtt{S}} = \{\sigma_W, \sigma_0, \ddot{\sigma}_0, p_1, p_2\}$ using **NR**
    optimize $\boldsymbol{\theta}_{\mathtt{BNN}}$ using **recursive estimation** (Section 3.3.2)
  end
end

3.3.2 Recursive Estimation of the BNN Parameters

The parameters of the BNN are estimated through a recursive process, where the BNN plays a role in generating the response variables that it is trained with. More precisely, the BNN uses the structural attributes of each element $\boldsymbol{z}_p^j$ to define their initial deterioration speeds $\dot{x}_{0,p}^j \in \boldsymbol{x}_{0,p}^j$. The initial state vector of each element $\boldsymbol{x}_{0,p}^j$ is then passed into the SSM to obtain the smoothed estimates of their initial speeds $f(\dot{x}_{0,p}^j | y_{1:\mathtt{T},p}^j) = \mathcal{N}\left(\dot{\mu}_{0|\mathtt{T},p}^j, (\dot{\sigma}_{0|\mathtt{T},p}^j)^2\right)$, which are used to train the BNN. Figure 3.2 illustrates the full recursive estimation procedure, which is explained in detail in the following paragraph.

Initially, at iteration $i = 0$, the parameters of the BNN $\boldsymbol{\theta}_{\mathtt{BNN}}$ are calibrated to predict the smoothed estimates of the deterioration speed produced by the SSM approach (Section 2.1.3), where the priors for the initial states $\boldsymbol{x}_{0,p}^j$ are defined as follows,

$$\begin{bmatrix} x_{0,p}^j \\ \dot{x}_{0,p}^j \\ \ddot{x}_{0,p}^j \end{bmatrix} \sim \mathcal{N}\left(\overbrace{\begin{bmatrix} \mu_{0,p}^j \\ 0 \\ 0 \end{bmatrix}}^{\boldsymbol{\mu}_{0,p}^j}, \overbrace{\begin{bmatrix} \sigma_{0,p}^j & 0 & 0 \\ 0 & g(\gamma p_1, p_2) & 0 \\ 0 & 0 & \ddot{\sigma}_0 \end{bmatrix}^2}^{\boldsymbol{\Sigma}_{0,p}^j} \right). \tag{3.5}$$

Here, the prior expected value and standard deviation for each element's condition, denoted by $\mu_{0,p}^j$ and $\sigma_{0,p}^j$, are defined as in Equations 3.2–3.3. The initial acceleration is also defined as was presented in Section 3.1, with an expected value of zero and a standard deviation

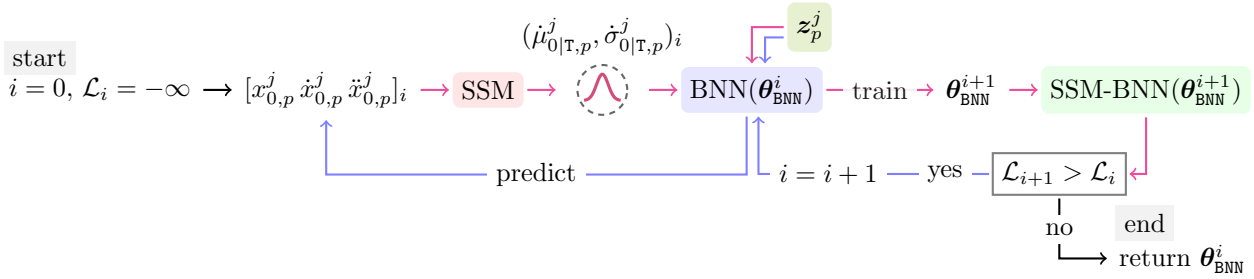


Figure 3.2 The recursive process of estimating the BNN parameters $\boldsymbol{\theta}_{\mathtt{BNN}}$. At $i = 0$, the initial states $[x_{0,p}^j\ \dot{x}_{0,p}^j\ \ddot{x}_{0,p}^j]_i$ are defined with Equation 3.5 and passed through the SSM to obtain the smoothed initial speeds $(\dot{\mu}_{0|\mathtt{T},p}^j, \dot{\sigma}_{0|\mathtt{T},p}^j)_i$. These are then used to train the BNN, so that it learns to predict them based on the structural attributes $\boldsymbol{z}_p^j$. The SSM-BNN model then evaluates the log-likelihood of the validation set $\mathcal{L}_{i+1}$ and compares it to the previous iteration $\mathcal{L}_i$. If there is an improvement, the BNN uses the structural attributes $\boldsymbol{z}_p^j$ to define the priors for the initial deterioration speeds $\dot{x}_{0,p}^j$, and the recursive loop is continued. Otherwise, the loop is terminated and the BNN parameters from the previous iteration $\boldsymbol{\theta}_{\mathtt{BNN}}^i$ are returned. The *forward* and *backward* steps of the loop are represented by the magenta and blue arrows.

parameter shared by all the elements $\ddot{\sigma}_0$. Lastly, the prior of the initial speed is defined with an expected value of zero and a standard deviation parameter $g(\gamma p_1, p_2)$, which corresponds to the linear relation from Equation 3.4 with a slope parameter p_1 scaled by a positive factor $\gamma \in \mathbb{R}^+$. Scaling the slope of the linear model provides a broad prior for the initial speeds, which is intended to provide the SSM model with more flexibility to fit the low number of highly variable observations in the time series and improve the first set of smoothed estimates used for training the BNN model. A scaling factor of $\gamma = 5$ is employed in this thesis based on the experiments with synthetic data.

After defining the priors for the initial states $\boldsymbol{x}_{0,p}^j$, they are propagated forward in time with the Kalman filter (KF). The KF estimates are then improved using the Kalman smoother (KS), yielding the smoothed initial speeds $(\dot{\mu}_{0|\mathtt{T},p}^j, \dot{\sigma}_{0|\mathtt{T},p}^j)$. These are then used to train the BNN, whose objective is to learn to predict the initial deterioration speed based on the structural attributes $\boldsymbol{z}_p^j$. The BNN is trained using the TAGI method presented in Sections 2.3–2.4. Following the first iteration, the SSM-BNN model is used in subsequent iterations ($i > 0$) to obtain the smoothed initial speeds $(\dot{\mu}_{0|\mathtt{T},p}^j, \dot{\sigma}_{0|\mathtt{T},p}^j)$ as outlined in Section 3.1, with one exception. The prior standard deviation for each element's deterioration speed $\dot{\sigma}_{0,p}^j$ is defined with Equation 3.4, rather than with the standard deviation predicted by the actively trained BNN model. This is done to avoid a premature convergence caused by the predicted deterioration speeds having narrow priors, which restricts the capacity of the deterioration model to fit the observations. Given the recursive nature of estimating the BNN parameters, this can result in the BNN learning a poor representation of the relation between the structural attributes and the deterioration speeds. The recursive procedure of estimating the smoothed initial speeds $(\dot{\mu}_{0|\mathtt{T},p}^j, \dot{\sigma}_{0|\mathtt{T},p}^j)$ with SSM-BNN and then using them to estimate the BNN parameters $\boldsymbol{\theta}_{\mathtt{BNN}}$ is repeated until there is no improvement in the validation set log-likelihood $\mathcal{L}$.

CHAPTER 4 Case Studies

4.1 Introduction

This chapter presents two case studies comparing the proposed SSM-BNN model with the existing SSM-KR model presented in Section 2.2.2. Prior to comparing these two methods, the proposed SSM-BNN model is first verified on synthetic data, where the true parameters and states are known. The synthetic visual inspection data is designed to have characteristics similar to the real visual inspection data from the network of bridges in the province of Quebec. Section 4.2 describes the real and synthetic data. Section 4.3 compares the two deterioration models on the synthetic data, while Section 4.4 compares them on the real data.

4.2 Real and Synthetic Data Description

The real data consists of visual inspections performed on roughly 10000 bridges located in the Quebec province of Canada. Each bridge is characterized by a set of attributes such as age, location, and material type. The inspection frequency for each bridge ranges from once a year to once every five years, with the majority of the bridges undergoing four or five inspections in total from 2007 up to 2023. To ensure that the synthetic data is representative of the real data, it is generated with characteristics similar to those derived from the real visual inspections on the beam elements. The synthetic data consists of $\mathtt{E} = 20000$ elements, each designed to have a service life of $\tau = 60$ years. The deterioration curves for each synthetic element are generated using the transition model defined in Equation 2.1 with the standard deviation of the process noise set to $\sigma_W = 5 \times 10^{-3}$. To ensure that these curves exhibit the characteristics of the real deterioration, several criteria are enforced on the synthetic elements' states $\boldsymbol{x}_{t,p}^{j}$ [3]. For instance, a slow deterioration is imposed by requiring that the condition of each element at half of it service life has degraded by no more than 15%, i.e., $x_{\frac{\tau}{2},p}^{j} > 0.85 \times x_{1,p}^{j}$. The full set of criteria is provided in the work of Hamida and Goulet [3]. Following the creation of the synthetic deterioration curves, a set of 300 synthetic inspectors is created, with each inspector's error defined by $v_t : V \sim \mathcal{N}(\mu_{V_i}, \sigma_{V_i}^2)$. The bias and standard deviation of each inspector's error are sampled from uniform distributions, following $\mu_{V_i} \sim \mathcal{U}(-4, 4)$ and $\sigma_{V_i} \sim \mathcal{U}(1, 6)$. The synthetic visual inspection data is generated using the observation model defined in Equation 2.2 by sampling synthetic observations from the true synthetic deterioration states. The synthetic inspections are created while ensuring

that the inspection frequency of the synthetic elements matches the inspection frequency of the real beam elements. The true deterioration states and observations are generated in the transformed space using $n = 4$ as the transformation function parameter. The synthetic structural attributes z^j are generated using the relation proposed by Hamida and Goulet [7]:

$$z^j = \ln(|\dot{x}_0^j|) + w_0, \;\; w_0 : W_0 \sim \mathcal{N}(0, 0.1^2). \tag{4.1}$$

This relation results in unevenly distributed synthetic attributes as illustrated by the plot of their histogram in Figure 4.1. Using synthetic attributes that exhibit a long-tail distribution allows verifying the robustness of the proposed framework.

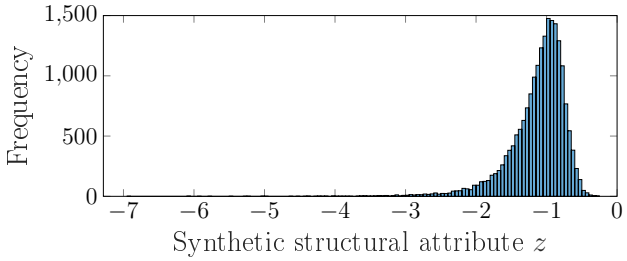


Figure 4.1 Histogram of the synthetic structural attribute.

4.3 Analyses on Synthetic Data

The SSM-BNN framework is verified on synthetic data using a simple neural network architecture. Namely, one hidden layer is employed with 128 hidden units and ReLU activation functions. The parameters of the BNN are initialized using He's method [44]. The training data is standardized to have a zero mean and unit variance, and the network is trained with a batch size of 16. To prevent overfitting, the training data is further split into training and validation sets using an 85/15 split. In order to use the SSM-KR framework as a benchmark, its KR component is configured following the work of Hamida and Goulet [7]. For details related to the training procedure and hyperparameters of KR, refer to Appendix B.

Figure 4.2 illustrates the latent relation between the synthetic structural attribute z and the initial deterioration speed $\dot{x}_0$ learned by the BNN, along with the training data and the ground truth. The true speed lies within two standard deviations of the mean predicted by the BNN for nearly the full range of inputs; the exception is only in the range of inputs outside the training data $z \in [-0.4, 0)$. The BNN also correctly captures the heteroscedasticity of the data and assigns large uncertainty to the ranges with a lower density of training data $z \in [-7, -2]$.

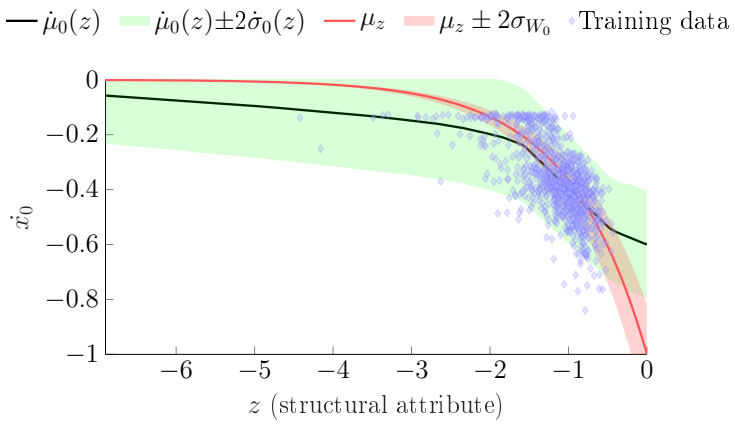
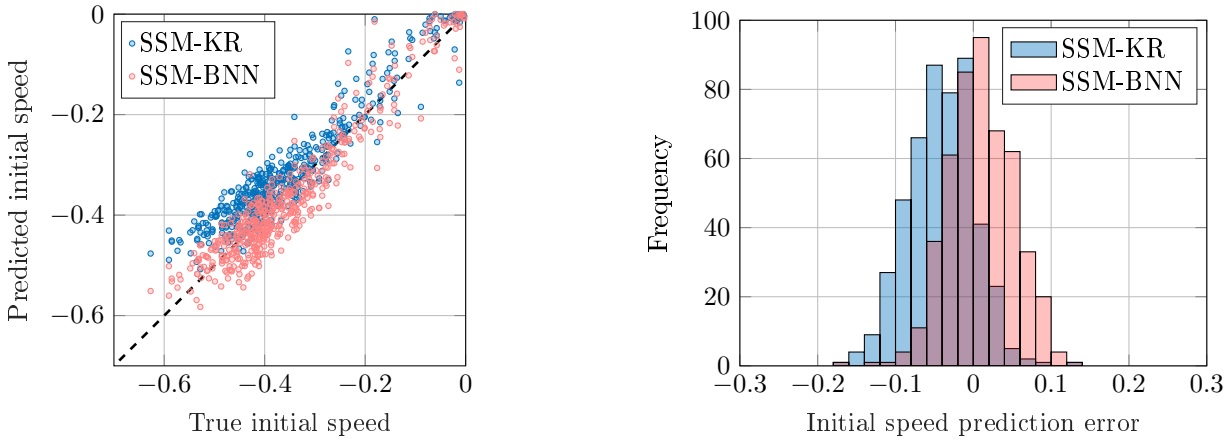


Figure 4.2 The latent relation between the synthetic structural attribute z and the deterioration speed $\dot{x}_0$ learned by the BNN. The black line represents the expected values predicted by the BNN with the shaded green area representing uncertainty. The true relation is depicted by the red line with the shaded red area representing its variability. Having this variability makes the synthetic attributes resemble reality, where structures with different attributes can deteriorate at the same speed.

Having verified that SSM-BNN is capable of learning the latent relation between the synthetic structural attribute and the initial deterioration speed, we now proceed to compare it with SSM-KR on the synthetic data. To evaluate the performance on a network scale, we randomly sample 500 elements from the synthetic data and examine the capacity of the two models for predicting the deterioration conditions and speeds. First, we examine the predictive capacity of the two models to estimate the initial deterioration speeds. Figure 4.3a presents the scatter plot of the true initial deterioration speed $\dot{x}_0$ versus the predicted initial deterioration speed $\dot{\mu}_0$ for the synthetic data. The initial speed values estimated by both models exhibit a reasonable spread around the diagonal line of the scatter plot. In Figure 4.3b, the histogram showing the difference between the true and predicted values demonstrates that SSM-KR model is slightly biased compared to the SSM-BNN model.

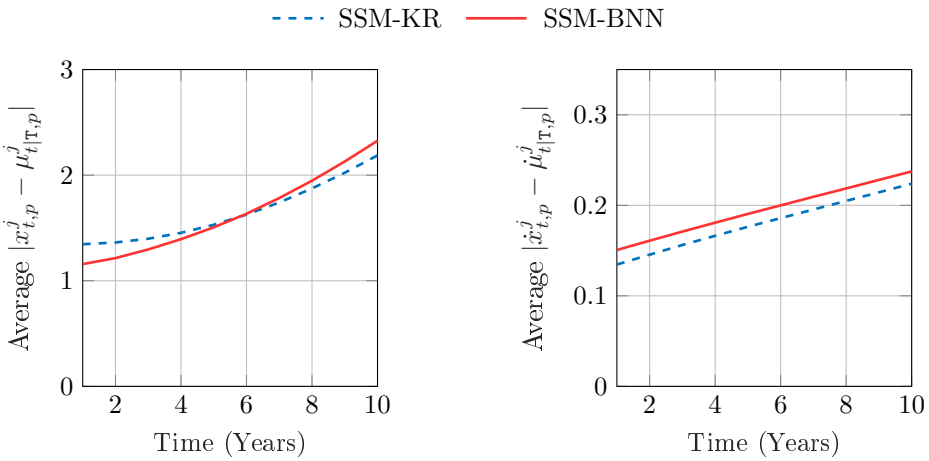
Figure 4.4 illustrates the capacity of the two models to predict the condition and speed across different time intervals following the last inspection points. Both models produce reliable estimates, as evident from their absolute average prediction errors being within reasonable ranges. Figure 4.4a demonstrates that the SSM-BNN model achieves better predictions for the deterioration condition closer to the last inspection point, while the SSM-KR model achieves better predictions further away. The SSM-KR model also achieves slightly lower prediction errors for the deterioration speed throughout the full prediction time interval, as shown in Figure 4.4b. Despite the marginal disparity in terms of the predictive capacity, the SSM-BNN model completed training in approximately half the time required by the SSM-



(a) Scatter plot of the true speed $\dot{x}_0$ versus the predicted speed $\dot{\mu}_0$.

(b) Histogram of the difference between the true speed $\dot{x}_0$ and the predicted speed $\dot{\mu}_0$.

Figure 4.3 Scatter plot and histogram comparing the true initial deterioration speeds $\dot{x}_0$ of the synthetic elements with those predicted by SSM-BNN and SSM-KR.



(a) Absolute average error for the predicted condition.

(b) Absolute average error for the predicted speed.

Figure 4.4 Absolute average errors for the difference between the predicted values for condition $\mu^j_{t,p}$ and speed $\dot{\mu}^j_{t,p}$ and the true values for condition $x^j_{t,p}$ and speed $\dot{x}^j_{t,p}$ across different time intervals following the last inspection points.

KR model (Table 4.1). The computational difference between the two models becomes more pronounced when increasing the number of training data and structural attributes, which significantly slows down the SSM-KR model [7]. Furthermore, while the SSM-KR model requires hyperparameter tuning to achieve the performance displayed here, the SSM-BNN model is able to match that with a default network configuration.

Table 4.1 Training time for SSM-KR and SSM-BNN on the synthetic data.

Model	Training time (min)
SSM-KR	57.4
SSM-BNN	**28.5**

Some challenges with the inspection data are illustrated by the deterioration analysis of a synthetic element e_1^{912} in Figure 4.5. The inspector I_1 is noticeably biased towards overestimating the element's condition, which shifts upwards the deterioration condition estimates of both the SSM-BNN and SSM-KR. Although the estimated initial speed for both models is close to the true value, the overestimation of the deterioration condition forces both models to compensate by overestimating the deterioration speed. Despite the challenges posed by the variable and biased inspection data in this example, both models perform equally well, as illustrated by their predictions almost overlapping and the true state being within two standard deviations of their predictions. Overall, the synthetic analyses presented here verified the functionality of the SSM-BNN framework. The SSM-BNN method demonstrated a comparable performance to the SSM-KR method on a synthetic dataset with a single structural

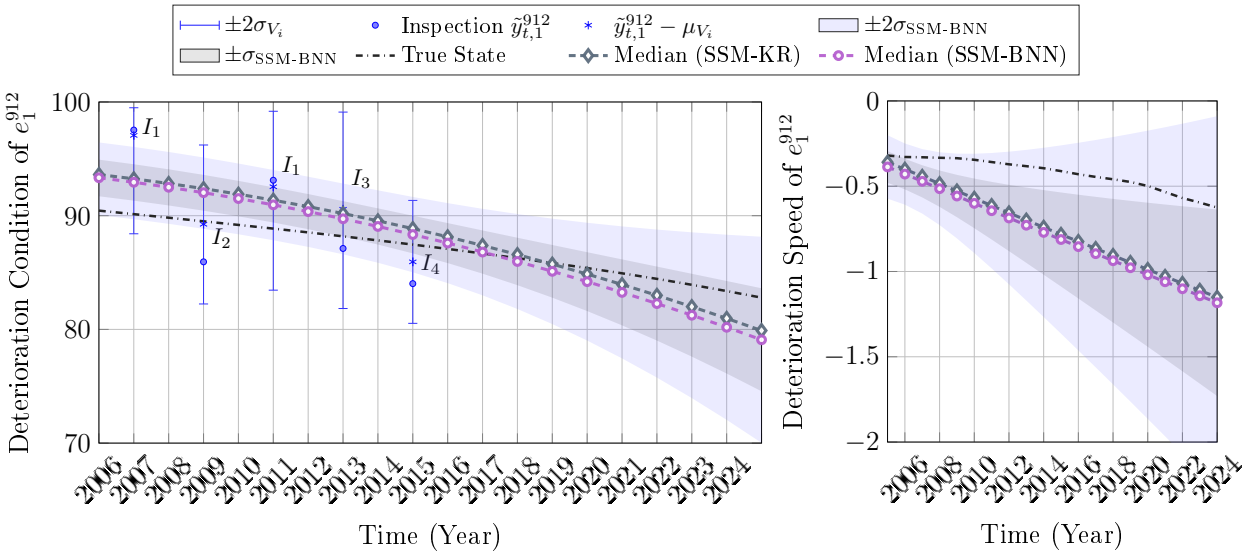
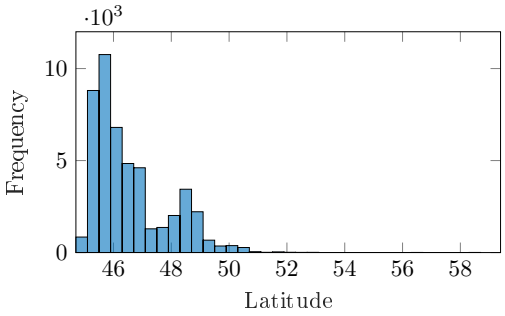


Figure 4.5 Predicted deterioration of a synthetic element e_1^{912}. The predictions of SSM-BNN are shown in purple while the predictions of SSM-KR are shown in green. The true deterioration is represented by the dashed black line. The shaded area represents the uncertainty associated with the predictions of SSM-BNN. The synthetic inspections are represented by the blue circles. The blue asterisks represent the inspections corrected by the estimated bias of the associated inspector and the blue error bars indicate the inspector's estimated standard deviation.

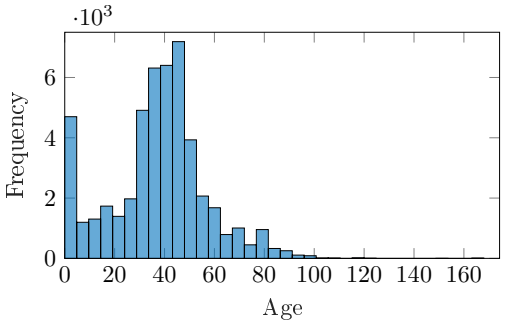
attribute. Although SSM-KR resulted in lower errors for the forecasted speed, training the SSM-BNN model was two times faster and it achieved better estimates of the initial speeds. Moreover, unlike the SSM-KR model, the SSM-BNN model did not require any hyperparameter tuning. The next section compares the two models on the real data to demonstrate the performance and scalability advantages offered by the SSM-BNN framework.

4.4 Analyses on Real Data

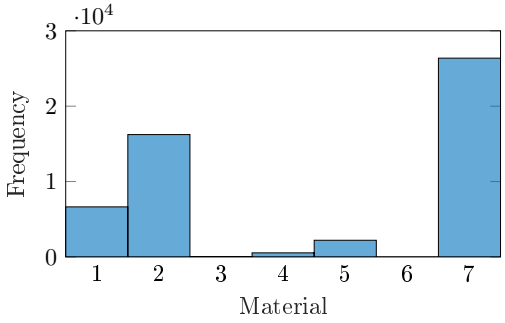
The SSM-BNN and SSM-KR models are compared using all the inspection data for the beam elements' category. The two models are initially compared using only four covariates in the regression analyses due to the computational limitations of the KR. The distributions of these covariates are shown in Figure 4.6. The four covariates consist of three structural attributes and the condition at time $t = 1$, which is estimated as the average of the first three inspections [7]. The structural attributes include the elements' material type, latitude,



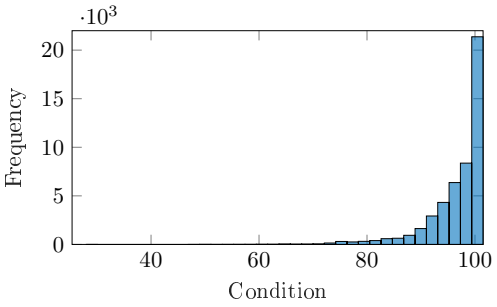
(a) Histogram of the latitudes.



(b) Histogram of the structures' ages, calculated at the date of their first inspection.



(c) Histogram of materials: 1 ≡ weathering steel, 2 ≡ regular steel, 3 ≡ aluminum, 4 ≡ wood, 5 ≡ high-performance concrete, 6 ≡ prestressed concrete, 7 ≡ regular concrete.



(d) Histogram of the conditions at time $t = 1$ approximated as the average of the first three inspections.

Figure 4.6 Histograms of the four covariates used in the regression analyses.

and age at the time of the first inspection. Contrary to KR, BNN has no issues with using a higher number of covariates. Hence, the SSM-BNN model is also trained with all the available structural attributes, which are summarized in Table 4.2. This is done to determine the impact of the additional covariates on the model's predictive capacity.

Table 4.2 Summary of the structural attributes used in the regression analyses.

Attributes	**Value Ranges**	**Unit**
material	-	-
age	$[0, 165]$	years
latitude	$[44.99, 58.67]$	-
longitude	$[-79.51, -57.25]$	-
total length	$[3.674, 1801.4]$	meters
slab length	$[3.674, 1801.4]$	meters
total width	$[3.35, 120]$	meters
surface area	$[17, 52420]$	square meters
number of lanes	$[0, 13]$	lanes
percentage of trucks	$[0, 100]$	-
annual average daily traffic	$[0, 178000]$	cars per day

The architecture employed for the BNN consists of one hidden layer with 128 units. The activation function is chosen to be ReLU and the weights and biases of the BNN are initialized using He's method [44]. A batch size of 16 is used for training the BNN. The material covariate is one-hot encoded [45] while the rest of the covariates and the response variables are standardized to have zero mean and unit variance. The data used to train the BNN is randomly split into 85% training set and 15% validation set.

Figure 4.7 illustrates the performance of SSM-KR versus SSM-BNN on a single beam element from a bridge in Quebec. The plot shows the predictions by SSM-KR and SSM-BNN using four covariates and SSM-BNN using all twelve covariates. As shown in Figure 4.7, the predictions of all three models are nearly identical. The predicted deterioration conditions are also well aligned with the inspection data shown in blue, despite it indicating that the element's condition is improving over time. The test observation shown in red is hidden during the training to assess the models' predictive capacity. In all three cases, the test observation nearly overlaps with the model predictions. Note that the inspection data does not represent the true deterioration condition, but rather its estimate affected by each inspector's observation error. Thus, it is not expected, nor is it necessary, for the model predictions to match the inspection data exactly. However, examining the difference between the expected values of the model predictions $\tilde{\mu}_{t|\mathtt{T}-1}$ and the test observations $\tilde{y}_{t=\mathtt{T}}$ on a network scale can reveal if the model is biased to overestimating or underestimating the deterioration condi-

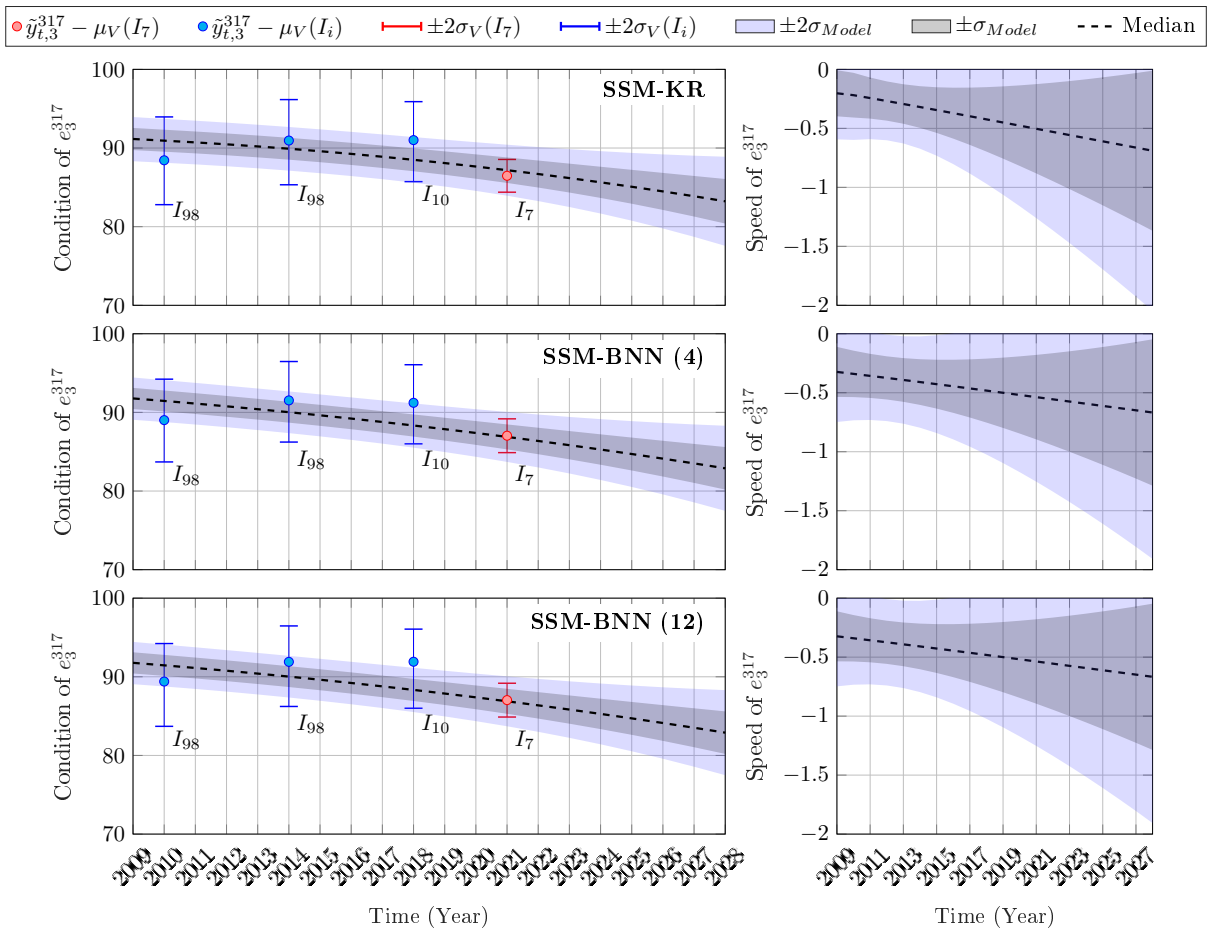


Figure 4.7 The deterioration of element e_3^{317}, as predicted by SSM-KR and SSM-BNN (4) using 4 covariates and SSM-BNN (12) using all 12 covariates. The predicted deterioration is represented by the dashed black lines, with the shaded areas representing the confidence intervals. The observations $\tilde{y}_{t,3}^{317}$ corrected by each inspector's estimated bias $\mu_V(I_i)$ are represented by the circles, with the error bars representing each inspector's estimated variance. The inspection shown in red is hidden from the models during training, and is used to gauge their predictive capacity.

tion. To examine this, ≈11000 test observations are used to generate the scatter plots shown in Figure 4.8. Different markers are used to represent the test observations corresponding to different forecast durations since the inspection frequency vary for different bridges. Figure 4.8 shows that the majority of the scatter points are roughly equally distributed on both sides of the diagonals indicating that none of the examined models are biased.

The global performance of the models is measured using 1004 beam elements from an independent test set of bridges whose data was not used to train the models. The performance metrics include the log-likelihood and training time of the models, which are presented in

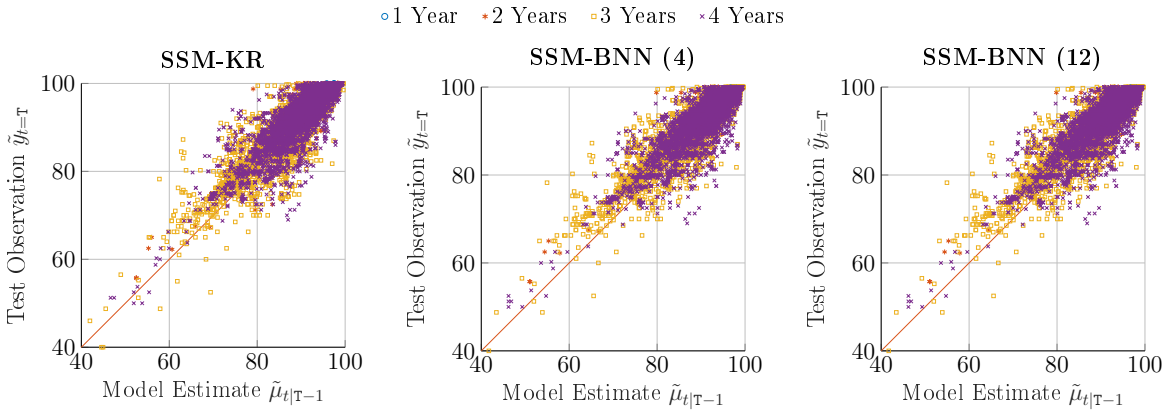


Figure 4.8 Scatter plots of the test observations $\tilde{y}_{t=\mathrm{T}}$ versus the model predictions $\tilde{\mu}_{t|\mathrm{T}-1}$ for ≈ 11000 beam elements. The predictions are obtained by SSM-KR and SSM-BNN using 4 covariates and SSM-BNN using all 12 covariates. Different markers correspond to different forecast durations since the inspection frequency is not uniform among different bridges.

Table 4.3. The SSM-BNN models significantly outperform SSM-KR in terms of computational time, which is measured using a system equipped with an Intel Xeon 6248R CPU, 256GB memory, and an NVIDIA Quadro RTX 5000 GPU. The training of SSM-BNN models is nearly two orders of magnitude faster than the training of SSM-KR. The SSM-BNN models also achieve $\approx 6\%$ improvement in log-likelihoods compared to the SSM-KR model. The inclusion of the additional covariates in the SSM-BNN model did not result in a better log-likelihood, as much of the relevant information is already captured by the four covariates, which were found to be the most influential in the work of Hamida and Goulet [7]. Moreover, using the log-likelihood to measure if an increased number of covariates improves the deterioration speed estimates is limited because the deterioration speed is a latent variable, which limits its impact on the log-likelihood, in contrast to other parameters such as the inspectors' variances σ_{V_i}.

Table 4.3 The performance and training time of SSM-KR on beam elements against SSM-BNN with 4 covariates and SSM-BNN with 12 covariates.

model	# of covariates	log-likelihood	total training time (hrs)
SSM-KR	4	-14180	189.3
SSM-BNN	4	**-13343**	2.9
SSM-BNN	12	-13364	**2.8**

CHAPTER 5 Network-scale Joint Estimation of Inspectors' Parameters

5.1 Introduction

The previous chapter demonstrated that the SSM-BNN outperforms the SSM-KR in terms of computational efficiency while matching its predictive capacity. The lack of computational efficiency constraints the SSM-KR to using a limited number of input covariates, which requires identifying the most influential ones (i.e., feature engineering) for each element category. In contrast, the SSM-BNN can handle a large number of input covariates, which eliminates the need for manual feature engineering and introduces the potential to automate the data processing and parameter estimation. Not being restricted to processing each element category one by one also enables pooling the observations of each inspector in the dataset for estimating their parameters. This is in contrast to the category-wise framework presented in Section 3, which processes each element category separately and estimates the inspectors' parameters using only their observations on a single element category. The category-wise framework also results in different estimates of the inspectors' parameters for different categories of elements; whereas, intuitively, one would expect the inspectors' performance to remain consistent across different element categories. Moreover, pooling the observations allows the inspectors' parameters to be estimated using more data, which should result in more accurate estimates for inspectors with few observations for a single element category. Consequently, this chapter presents a network-scale joint estimation framework based on the SSM-BNN model that learns the inspectors' parameters using their inspections over all the different elements. The results compare the inspectors' parameters estimated using the joint framework with those estimated using the category-wise framework.

5.2 Parameter Estimation

The proposed joint estimation framework pools the inspections for each inspector across all element categories. This is in contrast to the SSM-BNN framework presented in Section 3, which estimates the inspectors' parameters separately for each element category. The parameters of the joint training framework are the same as those of the category-wise framework presented in Section 3.2. The only difference is that in the joint framework, the inspectors' parameters $\{\mu_{V_{1:\mathrm{I}}}, \sigma_{V_{1:\mathrm{I}}}\}$ and the transformation function parameter n are shared across different element categories. So the set of parameters for the joint framework are effectively split into a global set $\boldsymbol{\theta}^G$, which includes parameters that are shared across different element

categories, and a local set $\boldsymbol{\theta}^L$, which includes parameters that are estimated separately for each element category:

$$\boldsymbol{\theta}^G = \{\mu_{V_{1:\mathtt{I}}}, \sigma_{V_{1:\mathtt{I}}}, n\} \quad \& \quad \boldsymbol{\theta}^L = \{\sigma_W, \sigma_0, \ddot{\sigma}_0, p_1, p_2, \boldsymbol{\theta}_{\mathtt{BNN}}\}.$$

The procedure for estimating the parameters of the joint network-scale framework is shown in Algorithm 2. As with the category-wise framework, the parameters are estimated from multiple passes over the data, which is split into training, validation, and test sets. The split is done while ensuring that the elements of a same structure are not simultaneously in the training set and the validation or test sets. The full estimation procedure is repeated for all the possible values of n since it can only take on a few values. For each n, the estimation process starts with learning the parameters of each inspector using the AGVI method, as outlined in the work of Laurent et al. [32]. The estimation of the inspectors' parameters is continued until the improvement in the sum of the validation set log-likelihoods of all element categories is negligible, i.e., $\Delta \sum_{j=1}^{\mathtt{C}} \mathcal{L}_j < 0.1\%$, where $\mathtt{C}$ denotes the total number of different element categories. The next step involves the estimation of local parameters $\boldsymbol{\theta}^L$ for each element category separately. The procedure for estimating these parameters is the same as for the category-wise framework, which is presented in detail in Section 3.2. In short, it consists of estimating the parameters of the SSM using Newton-Raphson (NR), followed by estimating the parameters of the BNN using the method presented in Section 3.3.2. The global estimation of inspectors' parameters $\boldsymbol{\theta}_{\mathtt{I}}$, followed by the local estimation of the SSM parameters $\boldsymbol{\theta}_{\mathtt{S}}$ and the BNN parameters $\boldsymbol{\theta}_{\mathtt{BNN}}$ is repeated until the improvement in the sum of the validation set log-likelihoods of all element categories is less than a predefined threshold.

Algorithm 2: The pseudocode for the joint estimation framework.

```
for n ∈ {1, 2, 3, 4, 5} do
    while Δ Σ_{j=1}^{C} L_j ≥ 0.1% do
        while Δ Σ_{j=1}^{C} L_j ≥ 0.1% do
            optimize θ_I = {μ_{V1:I}, σ_{V1:I}} using AGVI
        end
        for j ∈ {beams, slabs, columns, ...} do
            while ΔL_j ≥ 0.1% do
                optimize θ_S^j = {σ_W, σ_0, σ̈_0, p_1, p_2}^j using NR
                optimize θ_BNN^j using recursive estimation (Section 3.3.2)
            end
        end
    end
end
```

5.3 Results

This section starts with comparing the SSM-BNN model trained using the joint network-scale training framework with the SSM-BNN model trained using only the beam elements. The BNN configuration used for training on the beam elements is the same as the one presented in Section 4.4. In the network-scale training, all structural categories share the same BNN configuration. Namely, a BNN with a single hidden layer with 128 hidden units and ReLU activation function. The training is done with a batch size of 16 and the parameters of the BNN are initialized using He's method [44]. The categorical covariates are one-hot encoded while the rest of the covariates and the response variables are standardized to have zero mean and unit variance. To prevent overfitting, the training data for the BNN is randomly split into an 85% training set and 15% validation set.

Figure 5.1 presents the histograms comparing the inspectors' parameters estimated with the category-wise approach using the beam elements to those estimated with the joint framework using all the element categories. The inspector biases μ_{V_i} obtained with the joint framework are centered closer to zero compared to the ones obtained using only beams. Also, the inspector standard deviations σ_{V_i} estimated with the joint framework are generally higher compared to those estimated with the category-wise approach. This is likely due to inspectors having larger variations among their observations across categories compared to their observations on the beam elements only.

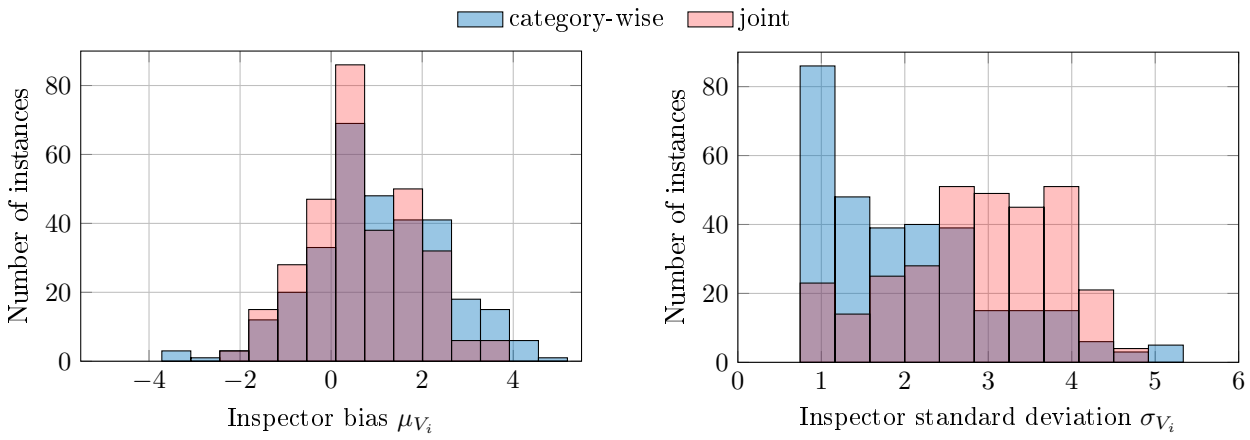


Figure 5.1 Histograms of the inspector parameters obtained using just the beam elements (category-wise) versus using all the element categories (joint).

While the histograms in Figure 5.1 show the overall distributions of the inspectors' parameters estimated by the two frameworks, Figure 5.2 illustrates the changes in the parameters of each

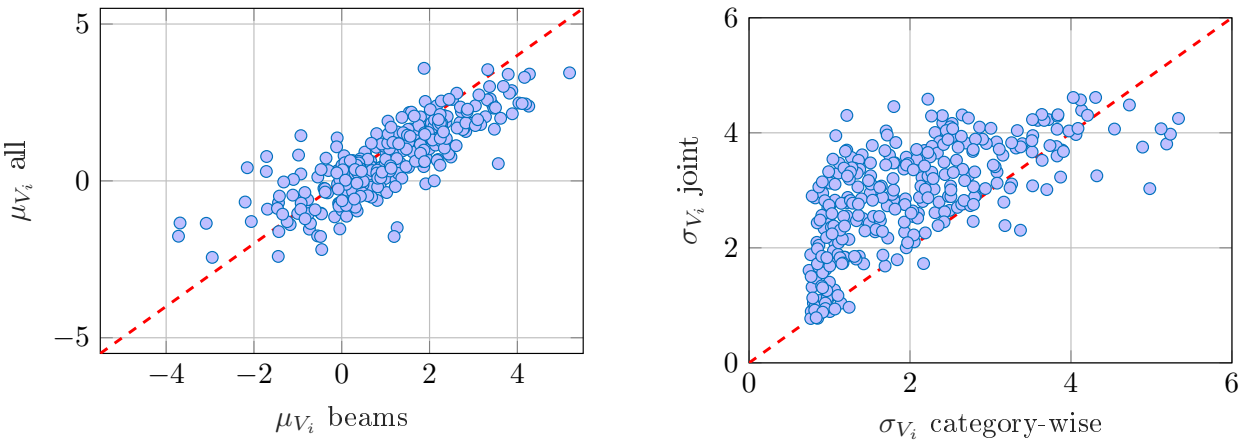


Figure 5.2 Scatter plots of the inspector biases μ_{V_i} and standard deviations σ_{V_i} estimated with the category-wise training using just the beam elements versus those estimated with the joint framework using all the element categories.

individual inspector. From these scatter plots, we can see that a large portion of inspectors' parameters change when using the joint estimation framework, with the biases exhibiting a smaller change compared to the standard deviations. On average, the number of observations per inspector increased by a factor of seven (or equivalently by 4607 inspections) when using the joint estimation framework. One might expect that a larger change in the number of observations would be accompanied by a larger difference between the parameters estimated by the two frameworks (e.g., $\Delta\mu_{V_i} = |\mu_{V_i}^{cat} - \mu_{V_i}^{joint}|$). That is, if an inspector's observations shift from being sparse in the category-wise training to becoming abundant in the joint training, then one would expect two significantly different estimates for that inspector's parameters. However, this is not observed from the results. For instance, inspector I_{64} had a considerable increase in their number of observations ($\Delta_{\text{obs}} = 2428$, a 694% increase), yet their parameters only marginally changed: $\Delta\mu_{V_{64}} = 0.44$ and $\Delta\sigma_{V_{64}} = 0.08$. In contrast, the inspector with the smallest increase in the number of observations (5 obs. $\rightarrow$ 22 obs.) had a comparatively larger change in parameters: $\Delta\mu_{V_{157}} = 2.02$ and $\Delta\sigma_{V_{157}} = 0.77$. Figure 5.3 illustrates the changes in the error distributions for these two inspectors. For reference, the smallest percentage increase in the number of observations is 236%.

The two frameworks are also compared using the following five structural categories: exterior walls, front walls, slabs, beams, and pavement. The category-wise training is done on each of these element categories separately, while the joint estimation is performed using all of them together. Both frameworks employ the same configuration for the BNN as described in the beginning of this section. Table 5.1 compares the log-likelihoods of the two frameworks on the validation and test sets. The joint framework performs better on three out of the

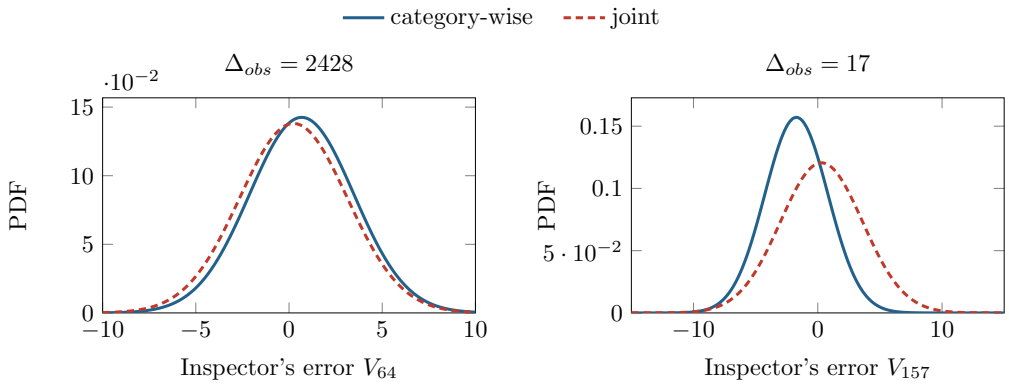


Figure 5.3 Illustration of the observation errors distributions estimated by the category-wise and joint frameworks for inspectors I_{64} and I_{157}. The number of observations for I_{64} increased by 2428, whereas inspector I_{157} only had an increase of 17.

five element categories on the validation set and four out of the five element categories on the test set. These results indicate that the inspectors' parameters estimated with the joint framework are better. The data in Table 5.1 also suggests that the category-wise framework may overfit the validation data for some categories (e.g, exterior walls and pavement). This is possible given that the category-wise framework employs more parameters than the joint estimation framework.

Table 5.1 Comparison of the joint estimation framework versus the category-wise estimation framework using five structural categories. **Bold** numbers indicate better performance. The percent changes (evaluated as $\% \ \Delta = 1 - \mathcal{L}_{\text{joint}}/\mathcal{L}_{\text{cat}}$) shown in green/red indicate improvement/degradation in performance of the joint framework compared to the category-wise framework.

Category	Validation set			Test set		
	$\mathcal{L}_{\text{cat}}$	$\mathcal{L}_{\text{joint}}$	$\% \ \Delta$	$\mathcal{L}_{\text{cat}}$	$\mathcal{L}_{\text{joint}}$	$\% \ \Delta$
Exterior walls	**-1.16E+04**	-1.20E+04	-3.6%	-5.47E+03	**-5.44E+03**	0.7%
Front walls	-5.79E+03	**-5.75E+03**	0.7%	-2.94E+03	**-2.92E+03**	0.8%
Slabs	-8.18E+03	**-8.13E+03**	0.6%	-3.02E+03	**-2.93E+03**	3.0%
Beams	-2.12E+04	**-2.10E+04**	1.1%	**-1.23E+04**	-1.25E+04	-2.4%
Pavement	**-6.64E+03**	-7.02E+03	-5.7%	-4.69E+03	**-3.32E+03**	24%
Total	**-5.34E+04**	-5.39E+04	-0.9%	-2.84E+04	**-2.74E+04**	3.5%

The analyses also reveal that the inspectors with smaller uncertainty within the joint framework are more likely to have their category-wise parameters clustered closer together. Figure 5.4 illustrates this with a scatter of the inspectors' parameters estimated with the

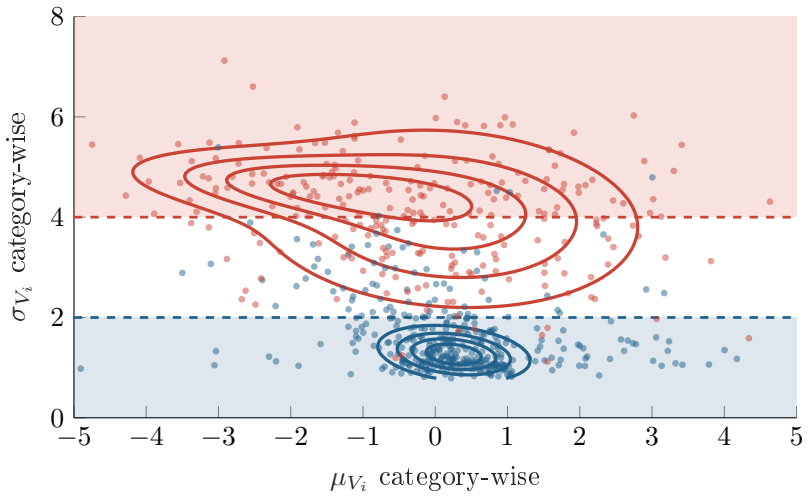


Figure 5.4 Illustration of the variability between inspectors' parameters estimated by the category-wise framework. The scatter points correspond to the inspectors' parameters estimated with the category-wise framework using five structural categories. The **blue** points are associated with the accurate inspectors ($\sigma_{V_i}^{joint} < 2$), whereas the **red** ones correspond to the inspectors with larger uncertainty ($\sigma_{V_i}^{joint} > 4$). These thresholds are established based on the inspectors' parameters estimated with the joint framework. The contours represent the estimated density of the scatter points.

category-wise approach on five different element categories. Although the scatter points correspond to the estimates from the category-wise approach, they are color coded based on the inspector uncertainty estimated from the joint framework. The blue points correspond to the inspectors with low uncertainty in the joint framework ($\sigma_{V_i}^{joint} < 2$) and the red ones correspond to the inspectors with large uncertainty in the joint framework ($\sigma_{V_i}^{joint} > 4$). The contours represent the density of the scatter points estimated with a mixture of two Gaussians [20]. The parameters corresponding to the inspectors with less uncertainty (blue points) exhibit a tighter spread compared to the ones with higher uncertainty (red points). This suggests that inspectors that consistently perform well across different elements tend to perform similarly well across them all. Although inspectors' biases μ_{V_i} exhibit a larger spread compared to their standard deviations σ_{V_i}, their impact on the inspectors' errors are less important. For instance, consider an inspector whose observation error is characterized by a standard Gaussian distribution with a mean $\mu_{V_i} = 0$ and a variance $\sigma_{V_i}^2 = 1$. In this case, increasing the bias estimate by one unit ($\Delta\mu_{V_i} = 1$) would result in a minor change in the range of observation error's highly likely values, provided the standard deviation remains constant. In contrast, if the bias remains unchanged, a shift of $\Delta\sigma_{V_i} = 1$ would double the range of the highly likely observations. Figure 5.5 presents a graphical illustration of these scenarios.

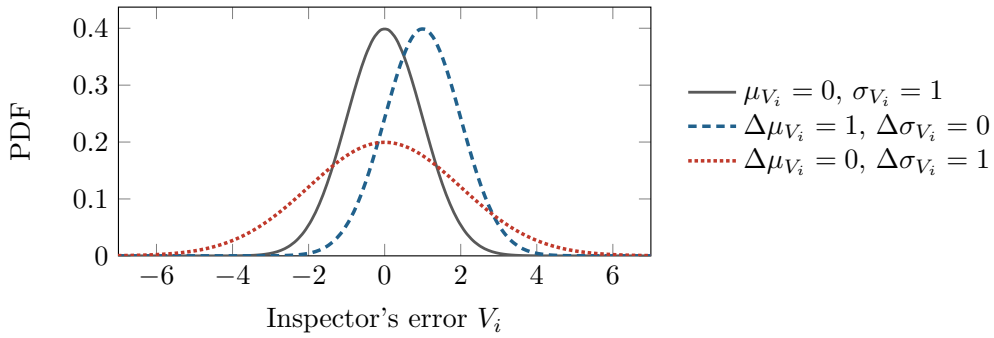


Figure 5.5 Illustration of the effects of a unit increase in bias and standard deviation estimates on the range of highly probable observations. The figure highlights how bias and standard deviation adjustments impact the distribution of likely values within the context of observation error. The black curve corresponds to the original distribution of observation error, whereas the red and blue curves correspond to the distributions resulting from the bias and standard deviation adjustments, respectively.

Overall, the joint framework significantly increases the number of observations per inspector, which is expected to bring their estimated parameters closer to the true values. This is supported by the joint framework achieving better log-likelihoods on most of the validation and test sets (Table 5.1). Additionally, estimating inspectors' parameters using all of their observations is more congruent with reality, where the performance of inspectors is not expected to change drastically when inspecting elements from different categories. Finally, the joint estimation framework takes roughly 4 days to train on the entire dataset of ≈ 10000 bridges with hundreds of elements each, comprising ≈ 1.6 million inspections. In contrast, the SSM-KR framework, which can only be trained individually on each element category, takes over 30 days. Note that this estimate does not include the time spent on feature engineering for SSM-KR, whereas the joint estimation framework is fully automated.

CHAPTER 6 Conclusion

6.1 Thesis Conclusions

This thesis proposed a new probabilistic method for modeling the deterioration of large infrastructure networks using visual inspections and structural attributes (e.g., age, location, etc.). The proposed method relies on an established state-space model (SSM) as its foundation, enhancing its capabilities by coupling it with a tailored Bayesian neural network (BNN). The SSM models the deterioration using the visual inspections while the BNN improves the SSM predictions using the structural attributes, which encode common deterioration patterns among similar structures. The parameters of the BNN are learned analytically using the tractable approximate Gaussian inference (TAGI) method, allowing the BNN to seamlessly integrate with the SSM.

The proposed method sought to address the limitations of the existing approach that coupled SSM with kernel regression (KR), which lacks scalability and computational efficiency. The proposed model was compared against the existing SSM-KR approach on a synthetic dataset and a real dataset consisting of beam elements from all the bridges in the province of Quebec. The results showed that the proposed model significantly outperforms the existing method in terms of computational time while matching its predictive capacity. Moreover, the proposed model can include many structural attributes, unlike the existing approach, which is limited to considering only a few. This removes the need to identify the most optimal subset of attributes for each structural category, which is a time-consuming and menial task. Overall, these benefits make the proposed method more suitable for modeling the deterioration of large networks of structures.

The aforementioned advantages of the proposed SSM-BNN model motivated its extension to a joint estimation framework, where the inspectors' errors are estimated jointly across all categories. While the proposed category-wise model yields distinct errors for the same inspector across individual categories, the joint estimation framework maintains consistent error estimates for the inspectors across all categories. The joint framework also estimates the inspectors' errors using more data, as the observations made by each inspector across various categories are consolidated. The two frameworks were compared on multiple structural categories from the real dataset. The joint estimation framework achieved a better performance on either the test or validation set in seven out of ten instances, indicating that it is a better approach for estimating the inspectors' parameters.

6.2 Limitations

This section outlines the limitations of the proposed deterioration model.

6.2.1 State Constraint on the Deterioration Speed

To model the decline in the condition of a structure over time, the predicted deterioration speed at each time step must be negative. To accomplish this, the proposed framework restricts the deterioration speed to the negative domain by using a probability density function (PDF) truncation method [3, 35]. The PDF truncation is applied if $\dot{\mu}^j_{t,p} + 2\dot{\sigma}^j_{t,p} > 0$ [3]. However, this typically causes an overestimation of the deterioration speed when it is close to zero or when it has large uncertainty. One possible way to overcome this is by performing a space transformation in the transition model to ensure the speed is always negative.

6.2.2 Lack of Physics in the Model

The proposed model incorporates the structural attributes which help identify structures with similar deterioration patterns. This indirectly factors-in the physics of the deterioration process, albeit in a limited way, by relating the structural attributes to the visual inspections. For instance, older structures should deteriorate faster as they are exposed to the environment for a longer period of time, increasing the interaction between the structural components and damaging agents (e.g., chlorides, carbonation, etc.). However, early stages of such deterioration are seldom captured by routine surface-level visual inspections. Although the transition equation of the state-space model is rooted in physics, its derivation is based on simple kinematics. The proposed model could benefit from integrating an analytical physics model for corrosion or developing a transition model with more nuanced physics.

6.3 Future Work

This section discusses future research which can potentially improve the proposed deterioration model and extend the other methods used in this work.

6.3.1 Time-varying Inspectors' Errors

The proposed model assumes that the parameters characterizing the inspectors' errors are constant. However, this is not consistent with the reality where the inspectors' errors are likely to change over time. For instance, an inspector's performance is expected to improve with gaining more experience. To accommodate this non-stationarity, the transition model

for the mean and variance characterizing each inspector's error could be modified to be time-varying within the AGVI method [2, 46].

6.3.2 Estimating the Initial Conditions with the BNN

The definition of the initial states impacts the trajectory of the time series modeled by SSM. The effect of the initial states is particularly pronounced when the number of observations is small. In the context of structural deterioration, the state vector is composed of the condition, speed, and acceleration. The proposed model employs a BNN for defining the initial deterioration speed of each structural element based on its attributes. However, the other two states are initialized without learning any dependence on the available data. The initial condition is defined as a function of the first few inspections performed on the structural element, and the initial acceleration is assumed to be zero. While the impact of the initial acceleration is minimal, the initial condition can significantly alter the trajectory of the deterioration since condition states are directly observed via inspections.

The overall performance of the deterioration model is expected to improve by refining the initialization of the state vector. One may accomplish this by using regression to infer the deterioration speed and condition while accounting for the correlation between them. A survey of the literature did not reveal a regression method with such capabilities. However, extending the tractable approximate Gaussian inference (TAGI) method can enable BNN to predict a multidimensional output with a full covariance matrix. One possible approach is to relate the hidden units from the last layer of a BNN to the predicted response variables and the elements of their full covariance matrix. Then, it is possible to leverage the ideas from the multivariate AGVI method [2, 43] to ensure that the predicted covariance matrix is positive semi-definite, and to form a relation between the predicted random variables representing the response variables and the elements of their full covariance matrix.

Methodology

Recall that the observation model employed by TAGI (Equation 2.6) assumes that the observation errors are independent. Removing this assumption enables factoring in the correlation between the response variables $\boldsymbol{y} = [y_1 \ \dots \ y_{\mathsf{Y}}]^\intercal$. Specifically, the new observation model is defined with a full error covariance matrix as follows,

$$\boldsymbol{y} = \boldsymbol{h}^{(0)} + \boldsymbol{v}, \ \boldsymbol{v} : \boldsymbol{V} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{V}}\right).$$

Here, $\boldsymbol{h}^{(0)}$ represents the vector of hidden units corresponding to the predicted response variables, $\boldsymbol{v}$ denotes the vector of observation errors, and $\boldsymbol{\Sigma_V}$ is the full error covariance matrix. To enable analytical inference of $\boldsymbol{\Sigma_V}$, we must modify the TAGI method presented in Section 2.4. First, the output layer of the neural network requires additional hidden units, which will correspond to the covariance between the observation errors. Next, we must ensure the predicted covariance matrix is symmetric and positive semi-definite. One way to accomplish this is to form $\boldsymbol{\Sigma_V}$ using the Cholesky decomposition [47] as follows,

$$\boldsymbol{\Sigma_V} = \boldsymbol{L}\boldsymbol{L}^{\intercal},$$

where $\boldsymbol{L}$ is a lower triangular matrix with positive diagonal entries. So instead of predicting the elements forming the full covariance matrix, the neural network will predict the elements forming its Cholesky factor $\boldsymbol{L}$. That is, the additional hidden units from the output layer will correspond to the elements of the Cholesky factorization of $\boldsymbol{\Sigma_V}$. Given that the elements of the Cholesky factor are treated as Gaussian random variables, the full error covariance matrix is obtained using the Gaussian multiplicative approximation (GMA) as outlined in the work of Deka and Goulet [43]. Figure 6.1 illustrates the compact representation of the neural network architecture with these modifications. The predicted diagonal elements of $\boldsymbol{L}$ are restricted to the positive domain using the exponential activation function, denoted by $\boldsymbol{l} \rightarrow \widetilde{\boldsymbol{l}}$ in Figure 6.1. The inference procedure is analogous to the one presented in Sections 2.3–2.4, which consists of applying the Gaussian conditional and RTS smoother equations.

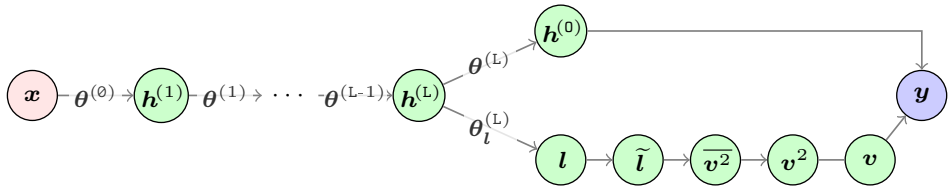


Figure 6.1 Compact representation of the network architecture from Figure 2.3 with modifications to enable inference of the full error covariance matrix. The neural network's parameters connecting layer j to layer $j+1$ are denoted by $\boldsymbol{\theta}^{(j)}$. This figure is adapted from [2].

Results on a Toy Problem

To verify the proposed methodology, synthetic data is generated with one-dimensional inputs $x \in \mathbb{R}$ and three-dimensional outputs $\boldsymbol{y} \in \mathbb{R}^3$ following

$$\boldsymbol{y} = (x+1) \cdot \sin(\pi x) + \boldsymbol{v}, \;\; \boldsymbol{v} : \boldsymbol{V} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma_V}\right). \tag{6.1}$$

The correlation between the response variables $\boldsymbol{y}$ is factored-in through the full heteroscedastic observation error covariance matrix $\boldsymbol{\Sigma_V}$ given by

$$\boldsymbol{\Sigma_V} = \begin{bmatrix} \sigma_{V_1}^2 & \sigma_{V_1}\sigma_{V_2}\rho_{12} & \sigma_{V_1}\sigma_{V_3}\rho_{13} \\ \vdots & \sigma_{V_2}^2 & \sigma_{V_2}\sigma_{V_3}\rho_{23} \\ \text{sym.} & \dots & \sigma_{V_3}^2 \end{bmatrix},$$

where its elements are defined using the following relations,

$$\boldsymbol{\sigma_V^2}(x) = \begin{bmatrix} \sigma_{V_1}^2 \\ \sigma_{V_2}^2 \\ \sigma_{V_3}^2 \end{bmatrix} = \begin{bmatrix} 0.1 \cdot x + 0.25 \\ 0.5 \cdot x^2 + 0.1 \\ 0.5 \cdot (x + 1.5)^2 \end{bmatrix}, \quad \boldsymbol{\rho} = \begin{bmatrix} \rho_{12} \\ \rho_{13} \\ \rho_{23} \end{bmatrix} = \begin{bmatrix} 0.1 \\ -0.3 \\ -0.25 \end{bmatrix}.$$

These relations are chosen to avoid the need to standardize the training data. The observations are generated using Equation 6.1 with 2000 randomly sampled inputs following $\sim \mathcal{U}(-0.75, 0.75)$. Figure 6.2 demonstrates how the variances change as a function of the input x.

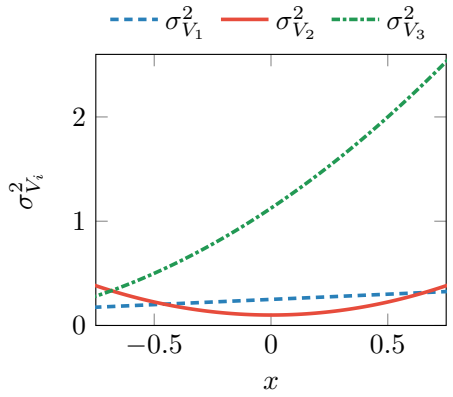


Figure 6.2 Illustration of the heteroscedastic nature of the variance functions used.

The neural network architecture employed consists of a single hidden layer with 400 units and ReLU activations. Training is done using a batch size of one to streamline the coding of the proposed method. To prevent overfitting, data is randomly split into training and validation sets using a 70/30 ratio. Figure 6.3 illustrates the training data alongside the outputs predicted by the neural network. These plots demonstrate the model's capacity to effectively predict correlated outputs by leveraging their interdependence. Figure 6.4 shows that the proposed method is capable of learning this interdependence by accurately inferring the full observation error covariance matrix. The next step in future work is to verify

the proposed method on synthetic network-scale visual inspection data, where the observed deterioration condition is correlated with the deterioration speed.

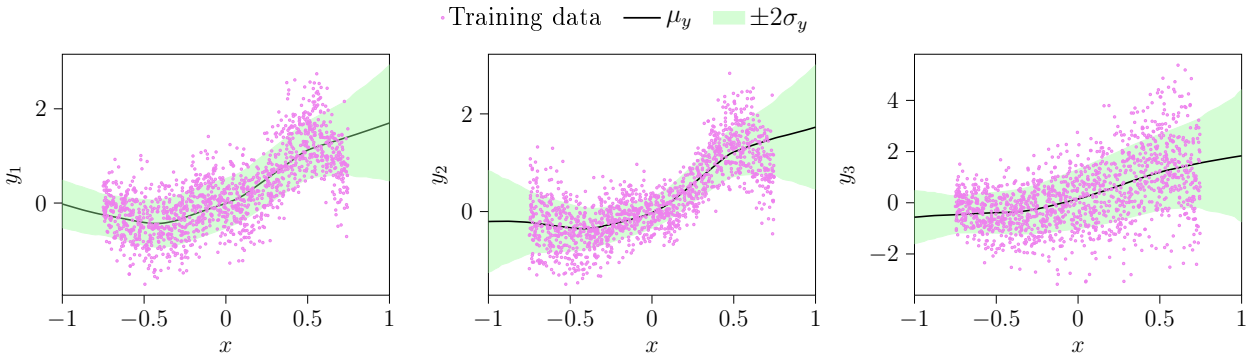


Figure 6.3 Illustration of correlated and highly variable outputs predicted by a Bayesian neural network relying on the proposed inference method.

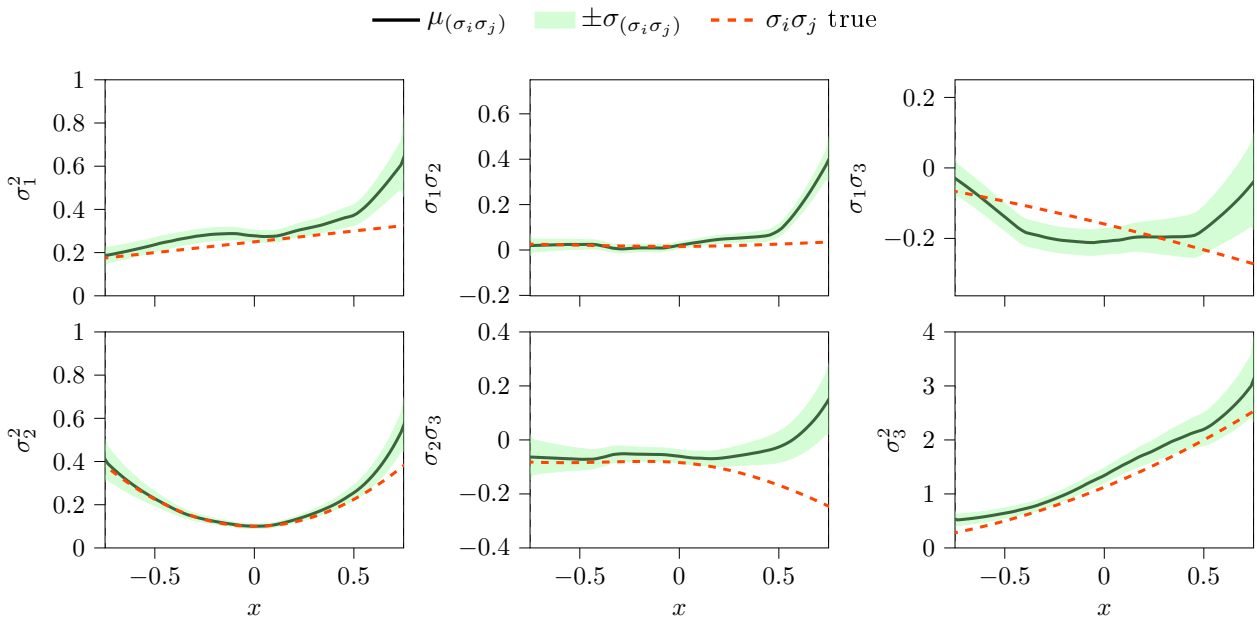


Figure 6.4 Graphical visualization of the full observation error covariance matrix. The true relations are shown by the dashed red lines, while those inferred by the Bayesian neural network are shown by the solid black lines with the shaded regions representing uncertainty.

REFERENCES

[1] J.-A. Goulet, L. Nguyen, and S. Amiri, "Tractable approximate Gaussian inference for Bayesian neural networks," *Journal of Machine Learning Research*, vol. 22, no. 251, pp. 1–23, 2021.

[2] B. Deka, "Analytical bayesian parameter inference for probabilistic with engineering applications," Ph.D. dissertation, Polytechnique Montréal, Montreal, Canada, 2022.

[3] Z. Hamida and J.-A. Goulet, "Modeling infrastructure degradation from visual inspections using network-scale state-space models," *Structural Control and Health Monitoring*, vol. 27, no. 9, p. e2582, 2020, e2582 stc.2582.

[4] C. Boller, P. Starke, G. Dobmann, C.-M. Kuo, and C.-H. Kuo, "Approaching the assessment of ageing bridge infrastructure," *Smart Structures and Systems*, vol. 15, no. 3, pp. 593 – 608, 2015.

[5] ACEC, CCA, CPRA, CPWA, CSCE, CUTA, CNAM, and FCM, "Canadian infrastructure report card," 2019. [Online]. Available: http://canadianinfrastructure.ca/en/index.html

[6] S. M. Piryonesi and T. E. El-Diraby, "Data analytics in asset management: Cost-effective prediction of the pavement condition index," *Journal of Infrastructure Systems*, vol. 26, no. 1, p. 04019036, 2020.

[7] Z. Hamida and J.-A. Goulet, "Network-scale deterioration modelling of bridges based on visual inspections and structural attributes," *Structural Safety*, vol. 88, p. 102024, 2021.

[8] MTQ, *Manuel d'Inspection des Structures*, Ministère des Transports, de la Mobilité Durable et de l'Électrification des Transports, 2014.

[9] M. Santamaria, J. Fernandes, and J. C. Matos, "Overview on performance predictive models – Application to bridge management systems." International Association for Bridge and Structural Engineering (IABSE), 2019.

[10] Z. Mirzaei, B. T. Adey, L. Klatter, and J. S. Kong, "Overview of existing bridge management systems-report by the iabmas bridge management committee," in *6th International Conference on Bridge Maintenance, Safety and Management (IABMAS)*, 2012.

[11] A. K. Agrawal, A. Kawaguchi, and Z. Chen, "Deterioration rates of typical bridge elements in new york," *Journal of Bridge Engineering*, vol. 15, no. 4, pp. 419–429, 2010.

[12] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[13] F. Wang, C.-C. B. Lee, and N. G. Gharaibeh, "Network-level bridge deterioration prediction models that consider the effect of maintenance and rehabilitation," *Journal of Infrastructure Systems*, vol. 28, no. 1, p. 05021009, 2022.

[14] A. Hatami and G. Morcous, "Deterioration models for life-cycle cost analysis of bridge decks in nebraska," *Transportation Research Record*, vol. 2313, no. 1, pp. 3–11, 2012.

[15] S. Ranjith, S. Setunge, R. Gravina, and S. Venkatesan, "Deterioration prediction of timber bridge elements using the markov chain," *Journal of Performance of Constructed Facilities*, vol. 27, no. 3, pp. 319–325, 2013.

[16] I. Zambon, A. Vidovic, A. Strauss, J. Matos, and J. Amado, "Comparison of stochastic prediction models based on visual inspections of bridge decks," *Journal of Civil Engineering and Management*, vol. 23, no. 5, pp. 553–561, 2017.

[17] N. Li, R. Haas, and W. Xie, "Development of a new asphalt pavement performance prediction model," *Canadian Journal of Civil Engineering*, vol. 24, no. 4, pp. 547–559, AUG 1997.

[18] D. Mizutani and X.-X. Yuan, "Infrastructure deterioration modeling with an inhomogeneous continuous time Markov chain: A latent state approach with analytic transition probabilities," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 13, pp. 1730–1748, 2023.

[19] M. Santamaria Ariza, I. Zambon, H. S. Sousa, J. A. Campos e Matos, and A. Strauss, "Comparison of forecasting models to predict concrete bridge decks performance," *Structural Concrete*, vol. 21, no. 4, pp. 1240–1253, 2020.

[20] J.-A. Goulet, *Probabilistic Machine Learning for Civil Engineers*. MIT Press, 2020.

[21] D. Veshosky, C. R. Beidleman, G. W. Buetow, and M. Demir, "Comparative analysis of bridge superstructure deterioration," *Journal of Structural Engineering*, vol. 120, no. 7, pp. 2123–2136, 1994.

[22] H. H. West, R. M. McClure, E. J. Gannon, H. L. Riad, and B. E. Siverling, "A nonlinear deterioration model for the estimation of bridge design life," *The Pennsylvania Transportation Institute*, 1989.

[23] M. Ilbeigi and M. E. Meimand, "Statistical forecasting of bridge deterioration conditions," *Journal of Performance of Constructed Facilities*, vol. 34, no. 1, p. 04019104, 2020.

[24] P. Lu, H. Wang, and D. Tolliver, "Prediction of Bridge Component Ratings Using Ordinal Logistic Regression Model," *Mathematical Problems in Engineering*, vol. 2019, p. 9797584, Apr. 2019.

[25] Y.-H. Huang, "Artificial neural network model of bridge deterioration," *Journal of Performance of Constructed Facilities*, vol. 24, no. 6, pp. 597–602, 2010.

[26] J. Sobanjo, "A neural network approach to modeling bridge deterioration," *Computing in Civil Engineering*, pp. 623–626, 1997.

[27] P. Rattanasuwan, "Estimating NBI ratings using a neural network," Master's thesis, University of Wisconsin-Madison, Wisconsin, USA, 1998.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[29] H. Liu and Y. Zhang, "Bridge condition rating data modeling using deep learning algorithm," *Structure and Infrastructure Engineering*, vol. 16, no. 10, pp. 1447–1460, 2020.

[30] P. Miao, H. Yokota, and Y. Zhang, "Deterioration prediction of existing concrete bridges using a lstm recurrent neural network," *Structure and infrastructure engineering*, vol. 19, no. 4, pp. 475–489, 2023.

[31] B. A. Graybeal, B. M. Phares, D. D. Rolander, M. Moore, and G. Washer, "Visual inspection of highway bridges," *Journal of nondestructive evaluation*, vol. 21, no. 3, pp. 67–83, 2002.

[32] B. Laurent, B. Deka, Z. Hamida, and J.-A. Goulet, "Analytical inference for inspectors' uncertainty using network-scale visual inspections," *Journal of Computing in Civil Engineering*, vol. 37, no. 5, p. 04023022, 2023.

[33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45 0021–9223, 1960.

[34] H. E. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450 0001–1452, 1965.

[35] D. Simon and D. L. Simon, "Constrained kalman filtering via density function truncation for turbofan engine health estimation," *International Journal of Systems Science*, vol. 41, no. 2, pp. 159–171 2010.

[36] I. Zambon, A. Vidović, A. Strauss, and J. Matos, "Condition prediction of existing concrete bridges as a combination of visual inspection and analytical models of deterioration," *Applied Sciences*, vol. 9, no. 1, 2019.

[37] E. A. Dizaj, J. E. Padgett, and M. M. Kashani, "A Markov chain-based model for structural vulnerability assessmentof corrosion-damaged reinforced concrete bridges," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2203, p. 20200290, 2021.

[38] M. Berry, M. Parrish, and M. Eberhard, "Peer structural performance database user's manual," *University of California, Berkeley*, 2004.

[39] A. Yosri, Y. Elleathy, S. Hassini, and W. El-Dakhakhni, "Genetic algorithm-markovian model for predictive bridge asset management," *Journal of Bridge Engineering*, vol. 26, no. 8, p. 04021052, 2021.

[40] F. Alogdianakis, L. Dimitriou, and D. C. Charmpis, "Data-driven recognition and modelling of deterioration patterns in the us national bridge inventory: A genetic algorithm-artificial neural network framework," *Advances in Engineering Software*, vol. 171, p. 103148, 2022.

[41] L.-H. Nguyen and J.-A. Goulet, "Analytically tractable hidden-states inference in Bayesian neural networks," *Journal of Machine Learning Research*, vol. 23, pp. 1–33, 2022.

[42] B. Deka, L. H. Nguyen, S. Amiri, and J.-A. Goulet, "The Gaussian multiplicative approximation for state-space models," *Structural Control & Health Monitoring*, vol. 29, no. 3, p. e2904, 2022.

[43] B. Deka and J.-A. Goulet, "Approximate gaussian variance inference for state-space models," *International Journal of Adaptive Control and Signal Processing, In press*, 2023.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[45] A. C. Muller and S. Guido, *Introduction to machine learning with Python*. O'Reilly, 2017.

[46] B. Laurent, "Analytical inference for visual inspection uncertainty in the context transportation infrastructures," Master's thesis, Polytechnique Montréal, Montreal, Canada, 2022.

[47] C. Benoit, "Note sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindres carrés à un système d'équations linéaires en nombre inférieur à celui des inconnues (procédé du commandant cholesky)," *Bulletin géodésique*, vol. 2, no. 1, pp. 67–77, 1924.

APPENDIX A SPACE TRANSFORMATION FOR THE OBSERVATIONS AND HIDDEN STATES PREDICTED BY THE SSM

The original observations $\tilde{x} \in [l, u]$ are transformed into an unbounded domain $x \in (-\infty, \infty)$ using the following function:

$$x = o(\tilde{x}) = \begin{cases} \left[\frac{1}{\Gamma(\alpha)} \int_0^{\tilde{x}} t^{\alpha-1} e^{-t} dt\right]^{\alpha}, & \frac{u+l}{2} < \tilde{x} \leq u, \\ \tilde{x}, & \tilde{x} = \frac{u+l}{2}, \\ -\left[\frac{1}{\Gamma(\alpha)} \int_0^{\tilde{x}} t^{\alpha-1} e^{-t} dt\right]^{\alpha}, & l \leq \tilde{x} < \frac{u+l}{2}, \end{cases} \tag{A.1}$$

Here, $\Gamma(\cdot)$ is the gamma function and α is the shape parameter of the gamma distribution. The shape parameter is defined as $\alpha \equiv 2^{-n}$, where n is a positive integer that controls the curvature of the transformation function near the bounds l and u. After completing the analyses, the predictions of the deterioration model $x \in (-\infty, \infty)$ are back-transformed into the original space $\tilde{x} \in [l, u]$ using

$$\tilde{x} = o^{-1}(x) = \begin{cases} \frac{1}{\Gamma(\alpha)} \int_0^{x^{\frac{1}{\alpha}}} t^{\alpha-1} e^{-t} dt, & x > \frac{u+l}{2}, \\ x, & x = \frac{u+l}{2}, \\ -\frac{1}{\Gamma(\alpha)} \int_0^{x^{\frac{1}{\alpha}}} t^{\alpha-1} e^{-t} dt, & x < \frac{u+l}{2}. \end{cases} \tag{A.2}$$

Figure A.1 illustrates the transformation function $o(\cdot)$ with different n values. An example of the space transformation with a specific n value is shown in Figure A.2.

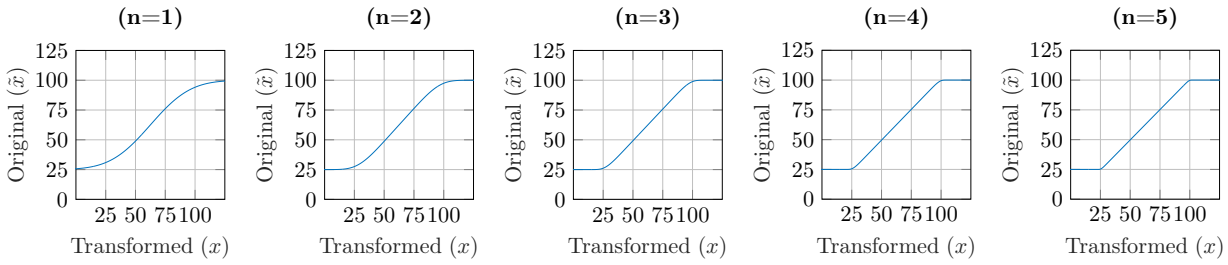


Figure A.1 Transformation function $o(\cdot)$ with different n values. Reproduced from [3].

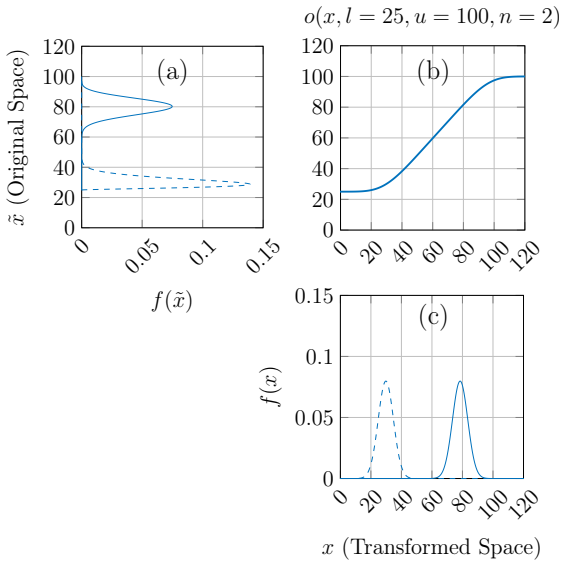


Figure A.2 Examples of state transformation using $o(\cdot)$ and $o^{-1}(\cdot)$. Reproduced from [3].

APPENDIX B KR TRAINING AND HYPERPARAMETERS FOR THE SYNTHETIC DATA

The kernel regression (KR) model predicts the deterioration speed $\dot{x}_0$ of a given element based on the proximity of its attributes z to the defined reference points and the hyperplane defined over them. The KR model in the SSM-KR framework iteratively learns the hyperplane defined by $\dot{\boldsymbol{\mu}}_{z|\mathtt{T}}$ and $\dot{\boldsymbol{\Sigma}}_{z|\mathtt{T}}$ using the following equations,

$$\begin{aligned}
\dot{\boldsymbol{\mu}}_{z|\mathtt{T}} &= \dot{\boldsymbol{\mu}}_z + \mathbf{J}_z \left(\dot{\boldsymbol{\mu}}_{0|\mathtt{T}} - \dot{\boldsymbol{\mu}}_{0|z} \right), \\
\dot{\boldsymbol{\Sigma}}_{z|\mathtt{T}} &= \dot{\boldsymbol{\Sigma}}_z + \mathbf{J}_z \left(\dot{\boldsymbol{\Sigma}}_{0|\mathtt{T}} - \dot{\boldsymbol{\Sigma}}_{0|z} \right) \mathbf{J}_z^\intercal, \\
\mathbf{J}_z &= \dot{\boldsymbol{\Sigma}}_z \mathbf{A}_\kappa^\intercal \dot{\boldsymbol{\Sigma}}_{0|z}^{-1},
\end{aligned}$$

where $\dot{\boldsymbol{\mu}}_z$ and $\dot{\boldsymbol{\Sigma}}_z$ define the prior for the hyperplane, $\dot{\boldsymbol{\mu}}_{0|z}$ and $\dot{\boldsymbol{\Sigma}}_{0|z}$ are the mean and covariance of the deterioration speed at time $t = 0$ as predicted by Equation 2.3, $\dot{\boldsymbol{\mu}}_{0|\mathtt{T}}$ and $\dot{\boldsymbol{\Sigma}}_{0|\mathtt{T}}$ are the mean and covariance of the deterioration speed at time $t = 0$ obtained from the Kalman smoother, and $\mathbf{A}_\kappa$ is the matrix containing the weight vectors associated with each structural element e_p^j, i.e. $\mathbf{A}_\kappa = \left[\boldsymbol{a}_1^1 \cdots \boldsymbol{a}_p^j \right]^\intercal$, where the weights defining $\boldsymbol{a}_p^j$ are obtained from Equation 2.4.

The KR model is trained using the radial basis function as the kernel function and $\mathtt{M} = 20$ reference points [7]. The rest of the KR model's hyperparameters are obtained following gradient optimization and are shown in Table B.1. For more details regarding the training of the KR model and its hyperparameters, refer to the work of Hamida and Goulet [7].

Table B.1 Optimal hyperparameter values of the KR model found using gradient optimization.

Hyperparameter	Value
l	0.2007
σ_{W_0}	0.1469