Variational deep learning for time series modelling and analysis : applications to dynamical system identification and maritime traffic anomaly detection

van Duong Nguyen

DOCTORAT BRETAGNE LOIRE MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Signal, Image, Vision*

Par
Duong NGUYEN

Variational Deep Learning for Time Series Modelling and Analysis

Applications to Dynamical System Identification and Maritime Traffic Anomaly Detection

Thèse présentée et soutenue à IMT Atlantique, Brest, France, le 17/12/2020
Unité de recherche : Lab-STICC CNRS UMR 6285
Thèse N° : 2020IMTA0227

Rapporteurs avant soutenance :

Patrick GALLINARI	Professeur, Sorbonne Université
Jean-Francois GIOVANNELLI	Professeur, Université de Bordeaux

Composition du Jury :

Président :	Patrick GALLINARI	Professeur, Sorbonne Université
Examinateurs :	Jean-Francois GIOVANNELLI	Professeur, Université de Bordeaux
	Stan MATWIN	Professeur, Université Dalhousie
	Cyril RAY	Maitre de conférences, Institut de Recherche de l'Ecole Navale
	Guillaume HAJDUCH	Chef de service, Collecte Localisation Satellites (CLS)
	Angélique DRÉMEAU	Maitre de conférences, ENSTA Bretagne
Dir. de thèse :	René GARELLO	Professeur, IMT Atlantique
Co-dir. de thèse :	Ronan FABLET	Professeur, IMT Atlantique

Summary (English)

Over the last decades, the ever increase in the amount of collected data has motivated data driven approaches. However, the majority of available data are unlabelled, noisy and may be partial. Furthermore, a lot of them are time series, *i.e.* data that have a sequential nature. This thesis work focuses on a class of unsupervised, probabilistic deep learning methods that use variational inference to create high capacity, scalable models for this type of data. We present two classes of variational deep learning, then apply them to two specific problems: learning dynamical systems and maritime traffic surveillance.

The first application is the identification of dynamical systems from noisy and partially observed data. We introduce a framework that merges classical data assimilation and modern deep learning to retrieve the differential equations that control the dynamics of the system. The role of the assimilation part in the proposed framework is to reconstruct the true states of the system from series of imperfect observations. Given those states, we then apply neural networks to identify the underlying dynamics. Using a state space formulation, the proposed framework embeds stochastic components to account for stochastic variabilities, model errors and reconstruction uncertainties. Experiments on chaotic and stochastic dynamical systems show that the proposed framework can remarkably improve the performance of state-of-the-art learning models on noisy and partial observations.

The second application is maritime traffic surveillance using AIS data. AIS is an automatic tracking system designed for vessels. The information richness of AIS has made it quickly become one of the most important sources of data in the maritime domain. However, AIS data are noisy, and usually irregularly sampled. We propose a multitask probabilistic deep learning architecture that can overcome these difficulties. Our model can achieve state-of-the-art performance in different maritime traffic surveillance related tasks, such as trajectory reconstruction, vessel type identification and anomaly detection, while reducing significantly the amount data to be stored and the calculation time. For the most important task—anomaly detection, we introduce a geospatial detector that uses variational deep learning to builds a probabilistic representation of AIS trajectories, then detect anomalies by judging how likely this trajectory is. This detector takes into account the fact that AIS data are geographically heterogeneous, hence the performance of the learnt probabilistic distribution is also geospatially dependent. Experiments on real data assert the relevance of the proposed method.

Key words: deep learning, variational inference, time series, state space model, recurrent neural network, dynamical system identification, anomaly detection, AIS, maritime traffic surveillance.

Résumé (Français)

Au cours des dernières décennies, l'augmentation de la quantité de données collectées a motivé l'utilisation d'approches basées sur les données. Cependant, la majorité des données disponibles sont non étiquetées, bruitées et peuvent être partiellement observées. De plus, beaucoup d'entre elles sont des séries temporelles, c'est-à-dire des données de nature séquentielle. Ce travail de thèse se focalise sur une classe de méthodes d'apprentissage profond, probabilistes et non-supervisées qui utilisent l'inférence variationnelle pour créer des modèles évolutifs de grande capacité pour ce type de données. Nous présentons deux classes d'apprentissage variationnel profond, puis nous les appliquons à deux problèmes spécifiques: l'apprentissage de systèmes dynamiques et la surveillance du trafic maritime.

La première application est l'identification de systèmes dynamiques à partir de données bruitées et partiellement observées. Nous introduisons un cadre qui fusionne l'assimilation de données classique et l'apprentissage profond moderne pour retrouver les équations différentielles qui contrôlent la dynamique du système. Le rôle de la partie d'assimilation, dans le cadre proposé, est de reconstruire les vrais états du système à partir des séries d'observations imparfaites. Étant donné ces états, nous appliquons ensuite des réseaux de neurones pour identifier la dynamique sous-jacente. En utilisant une formulation d'espace d'états, le cadre proposé intègre des composantes stochastiques pour tenir compte des variabilités stochastiques, des erreurs de modèle et des incertitudes de reconstruction. Des expériences sur des systèmes dynamiques chaotiques et stochastiques montrent que le cadre proposé peut améliorer remarquablement les performances des modèles d'apprentissage de pointe sur des observations bruitées et partielles.

La deuxième application est la surveillance du trafic maritime à l'aide des données AIS. L'AIS est un système de suivi automatique conçu pour les navires. La richesse d'informations de l'AIS en a rapidement fait l'une des sources de données les plus importantes dans le domaine maritime. Cependant, les données AIS sont bruitées et généralement échantillonnées de manière irrégulière. Nous proposons une architecture d'apprentissage profond probabiliste multitâche capable de surmonter ces difficultés. Notre modèle peut atteindre des performances très prometteuses dans différentes tâches liées à la surveillance du trafic maritime, telles que la reconstruction de trajectoire, l'identification du type de navire et la détection d'anomalie, tout en réduisant considérablement la quantité de données à stocker et le temps de calcul. Pour la tâche la plus importante - la détection d'anomalie, nous introduisons un détecteur géospatialisé qui utilise l'apprentissage profond variationnel pour construire une représentation probabiliste des trajectoires AIS, puis détecter les anomalies en jugeant la probabilité de cette trajectoire. Ce détecteur prend

en compte le fait que les données AIS sont géographiquement hétérogènes, ce qui, par conséquet fait varier la qualité de la distribution probabiliste apprise. Des expériences sur des données réelles affirment la pertinence de la méthode proposée.

Mots clés: apprentissage profond, inférence variationnelle, séries temporelles, modèle espace d'états, réseau de neurones récurrents, identification de systèmes dynamiques, détection d'anomalies, AIS, surveillance de trafic maritime.

Acknowledgement

The pursuit of a PhD is an enduring, yet exciting adventure in one's life. And I consider myself lucky for having such great guidance and companionship from the following people.

Prof. Dung-Nghi Truong-Cong and Prof. Nhat-And Che-Viet, I would like to express my sincere gratitude for your help in the early days. I could not have embarked on this amazing journey without your unstinting support.

To my thesis supervisors: Prof. René Garello, Prof. Ronan Fablet, and Assoc. Prof. Lucas Drumetz, thank you for your leadership. René, your kindness, scientific feedback, and administrative support has helped me greatly throughout the way. Ronan, your knowledge was essential, your trust was crucial, and your patience was vital. You carved out the rough shape of this work and helped me turn it into something I can be proud of. Thank you also for always having an open door for me. You are a role model for me to look up to, and I am very fortunate to have worked with you. To Lucas, I credit those insightful comments and suggestions, especially during the phases of drafting and revising manuscripts.

Dr. Quang-Thang Nguyen and Dr. Alex TP Nguyen, special thanks for your mentorship and guidance. Whenever I need advice, I know that I can always count on you two. To Oscar Chapron, I have grown to love the Breton culture more thanks to you (I am reserving the best spot in my office for your Breton flag). Said Ouala, our late-night deadline running, the abroad missions, and the ups and downs we shared will be remembered. You are the best workmate, comrade, and friend (beside Oscar) that I could ask for.

During my PhD, I spent most of my time at the Department of Signal and Communications (SC), IMT Atlantique. My gratitude goes to all my colleagues and the faculties members there. Prof. Aimee Johansen and Prof. Rebecca Clayton, thank you for your English courses and advice. Greatest thanks also to Monique Larsonneur and Martine Besnard for your keen support with various administrative tasks.

I would like to express my profound appreciation to Dr. Oliver S. Kirsebom, Fábio Frazão, and the PhD students at the Institute for Big Data Analytics (Dalhousie) for their unparalleled hospitality during the 2 internships I had there. And to Prof. Stan Matwin particularly for his kindness, his generosity, and the invaluable time he has spent on me.

Prior to this PhD, I did an internship at Collecte Localisation Satellites (CLS). During this PhD, I had a special opportunity to come back and carry out a mission there. To people of CLS, especially to Dr. Guillaume Hadjuch, Rodolphe Vadaine and Dr. Vincent Kerbaol, I sincerely thank you for the opportunities and the collaboration.

I also would like to thank Prof. Steve Brunton for having sponsored my internship at the University of Washington.

I'm grateful to my thesis committee for their time examining my thesis manuscript, their invaluable remarks, and comments.

Finally, I would like to acknowledge the Mines-Télécom Foundation for their financial support during my undergraduate and my master. My PhD is support by public funds (Ministère de l'Education Nationale, de l'Enseignement Supérieur et de la Recherche, FEDER, Région Bretagne, Conseil Général du Finistère, Brest Métropole), by Institut Mines Télécom, received in the framework of the VIGISAT program managed by "Groupement Bretagne Télédétection" (BreTel).

List of publications

Journal papers

— **D Nguyen**, R Vadaine, G Hajduch, R Garello and R Fablet, *GeoTrackNet-A Maritime Anomaly Detector using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection*, IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2019.

— S Ouala, **D Nguyen**, L Drumetz, B Chapron, A Pascual, F Collard, L Gaultier and R Fablet, *Learning Latent Dynamics for Partially-Observed Chaotic Systems*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 2020.

Preprints

— **D Nguyen**, S Ouala, L Drumetz and R Fablet, *Variational Deep Learning for the Identification and Reconstruction of Chaotic and Stochastic Dynamical Systems from Noisy and Partial Observations*, 2020.

— **D Nguyen**, S Ouala, L Drumetz and R Fablet, *EM-like learning chaotic dynamics from noisy and partial observations*, 2019.

Publications in conference proceedings

— **D Nguyen**, M Simonin, G Hajduch, R Vadaine, C Tedeschi and R Fablet, *Detection of Abnormal Vessel Behaviors from AIS data using GeoTrackNet: from the Laboratory to the Ocean*, 21st IEEE International Conference on Mobile Data Management (MDM), Maritime Big Data Workshop (MBDW), 2020.

— **D Nguyen**, S Ouala, L Drumetz and R Fablet, *Assimilation-based Learning of Chaotic Dynamical Systems from Noisy and Partial Data*, 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

— **D Nguyen**, OS Kirsebom, F Frazão, R Fablet and S Matwin, *Recurrent Neural Networks with Stochastic Layers for Acoustic Novelty Detection*, 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.

— **D Nguyen**, R Vadaine, G Hajduch, R Garello and R Fablet, *A Multi-task Deep Learning Architecture for Maritime Surveillance using AIS Data Streams*, The 5th

IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA), 2018.

— **D Nguyen**, R Vadaine, G Hajduch, R Garello and R Fablet, *An AIS-based Deep Learning Model for Vessel Monitoring*, NATO CMRE Maritime Big Data Workshop, 2018.

— S Ouala, **D Nguyen**, L Drumetz, B Chapron, A Pascual, F Collard, L Gaultier and R Fablet, *Learning ocean dynamical priors from noisy data using assimilation-derived neural nets*, 2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2019.

— S Ouala, **D Nguyen**, SL Brunton, L Drumetz and R Fablet, *Learning Constrained Dynamical Embeddings for Geophysical Dynamics*, Climate Informatics (CI), 2019.

Communications in international conferences

— **D Nguyen**, S Ouala, L Drumetz and R Fablet, *Learning Chaotic and Stochastic Dynamics from Noisy and Partial Observation using Variational Deep Learning*, 10th International Conference on Climate Informatics (CI), 2020.

— S Ouala, R Fablet, **D Nguyen**, L Drumetz, B Chapron, A Pascual, F Collard and L Gaultier, *Data assimilation schemes as a framework for learning dynamical model from partial and noisy observations*, General Assembly of the European Geosciences Union (EGU), 2019.

Acronyms

AIS Automatic Identification System

AnDA Analog Data Assimilation

ANN Artificial Neural Network

CNN Convolutional Neural Network

COG Course Over Ground

DAODEN Data-Assimilation-based Ordinary Differential Equation Network

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DL Deep Learning

DSSM Deep State Space Model

EKF Extended Kalman Filter

ELBO Evidence Lower BOund

EM Expectation-Maximisation

EnKF Ensemble Kalman Filter

EnKS Ensemble Kalman Smoother

FIVO Filtering Variational Objective

GMM Gaussian Mixture Model

GRU Gated Recurrent Unit

HMM Hidden Markov Model

IWAE Importance Weighted AutoEncoder

KDE Kernel Density Estimation

KL divergence Kullback–Leibler divergence

LGSSM Linear Gaussian State Space Model

LSTM Long Short-term Memory

LVM Latent Variable Model

MAP Maximum A Posteriori

MDA Maritime Domain Awareness

ML Machine Learning

MLP MultiLayer Perceptron

MMSI Maritime Mobile Service Identity

NLP Natural Language Processing

NN Neural Network

ODE Ordinary Differential Equation

PF Particle Filter

RK4 Runge-Kutta 4

RNN Recurrent Neural Network

ROI Region Of Interest

SDE Stochastic Differential Equation

SGD Stochastic Gradient Descent

SINDy Sparse Identification of Nonlinear Dynamics

SOG Speed Over Ground

SRNN Stochastic Recurrent Neural Network

SSM State Space Model

SVAE Sequential Variational AutoEncoder

TREAD Traffic Route Extraction and Anomaly Detection

VAE Variational AutoEncoder

VDL Variational Deep Learning

VI Variational Inference

VRNN Variational Recurrent Neural Network

Table of Contents

Part I

Introduction

A PhD is a great time in one's life to go for a big goal, an even small steps towards that will be valued.

Yoshua Bengio

Chapter 1

General Introduction

1.1 Motivation

The term "**deep learning**" (DL) was first introduced by Rina Dechter in 1986 (Dechter 1986). Nowaday, DL is usually understood as a family of **machine learing** (ML) methods that use **artificial neural networks** (ANNs) to learn features of data with multiple levels of abstraction (LeCun et al. 2015). Over the last decade, the world has witnessed an incredible development of DL. Machine learning in general, and deep learning in particular, have recently revolutionised many fields of research and application. In computer vision, DL has surpassed human-level performance for image classification, object detection, *etc.* (He et al. 2015; Russakovsky et al. 2015). Many tasks that are hard to mathematically defined such as mimicking an artistic style, generating artificial human-lookalike images, *etc.* have been achieved by neural-network-based (NN-based) models (I. Goodfellow, Pouget-Abadie, et al. 2014; Zhu et al. 2017; Karras et al. 2019). In natural language processing (NLP), the introduction of embedding models such as Word2Vec (Mikolov et al. 2013) and Glove (Pennington et al. 2014), ELMo (Peters et al. 2018) and BERT (Devlin et al. 2018) has significantly boosted the fields. NNs have helped create better machine translation (Sutskever et al. 2014; Y. Wu et al. 2016), human-like chatbots (of which Apple's Siri, Google Assistant, and Amazon Alexa are great examples), fake news detection models (Shu et al. 2017). From medicine, healthcare (Ravì et al. 2017; Esteva et al. 2019) to bioinformatics (Min et al. 2017), from chemistry (Goh et al. 2017) to agriculture (Kamilaris

et al. 2018), DL has yielded numerous state-of-the-art results and has been leveraged to obtain better solutions for complex tasks.

Among many others, two main components that build the success of deep learning are rapid advances in computational power and the ever-increasing availability of data. New hardware technologies dedicated for parallel computing such as GPUs or TPUs allow training big deep neural networks within a reasonable time. Some models might take months to train in the past now can be trained in a few hours. Alongside with those hardware accelerators, open-source libraries such as Tensorflow (Abadi et al. 2016), Pytorch (Paszke et al. 2017), MXnet (T. Chen et al. 2015), *etc.* have made DL more accessible. Thanks to those libraries, students, researchers and deep learning practitioners can spend more time on ideas and algorithms instead of on implementation aspects. With the growing popularity of the Internet, the development of sensor technologies as well as the Internet of things (IoT), more and more data are collected. As a data-driven approach, DL models require representative data, both in quality and in volume to be effective. For example, the launch of the ImageNet challenge (Russakovsky et al. 2015) is one of the main factors that evoke the rebirth of deep learning, marked by the victory of AlexNet (Krizhevsky et al. 2012). On the other hand, the success of DL encourages the collection of large data sets, because we now can extract and exploit valuable information from data.

Although the above-mentioned results are fascinating and their potential are appealing, deep learning still has many drawbacks (Marcus 2018):

— Most of the successful NN-based practical applications use **supervised learning** methods. Supervised learning is a branch of machine learning where the data are labelled, the models aim to find a mapping that predicts the labels given the data as the input. However, unlike unlabelled data, which are highly available, labelled data are expensive to obtain. For this reason, DL community has recently focused more on **unsupervised** and **semi-supervised learning** (Diederik P. Kingma and Welling 2013; Durk P. Kingma et al. 2014; Rezende, Mohamed, and Wierstra 2014; Locatello et al. 2019; Yin et al. 2018; Zhou et al. 2017; Metz et al. 2018; Vacar et al. 2019). In unsupervised learning, the data are not labelled, the models aim to uncover the structures, the patterns, the correlations existing in data. Those discoveries can be used for semi-supervised learning, where the models aim to do supervised tasks with only a small part of the data is labelled.

— Neural networks naturally do not deal well with noisy and irregularly-sampled data. The lack of explicit mathematical models makes standard neural networks

such as **multilayer perceptrons** (MLPs), **recurrent neural networks** (RNNs), **convolutional neural networks** (CNNs), *etc.* unable to distinguish noise from data. They blindly apply a series of calculations on a set of numbers (the inputs) to provide another set of numbers (the outputs). Those may cause the models to overfit the training data, or to create unexpected effects such as the adversarial examples (I. Goodfellow, Pouget-Abadie, et al. 2014; I. J. Goodfellow et al. 2014; Szegedy et al. 2014; Pajot et al. 2018). Because the calculations of DL models are performed on computational graphs, they do not accept NaN (not a number) as an input. Hence, we usually have to perform a preprocessing interpolation to fill the missing data. This step prevents DL from achieving its own **end-to-end learning** goal. Almost all DL models thus far suppose that the data are sampled regularly. In the real-world, it is rarely the case for numerous applications.

— It is difficult to embed prior knowledge into neural networks. One of the ultimate goals of DL is to perform an end-to-end learning, which means to minimise the number of hand-engineering steps and to relax as much as possible weak hypotheses. However, for tasks that DL is still not doing well, one may not want to throw away domain expertise that has been studied and verified for decades. For example, it has been well known that many atmospheric processes are chaotic (Lorenz 1963; Lorenz 1996), yet embedding this knowledge into a neural network is not trivial.

In the last few years, **variational deep learning** (VDL)—a branch of deep learning, has arisen as a very promising candidate to overcome those difficulties (Diederik P. Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014). Broadly speaking, VDL combines probabilistic modelling and deep learning to create flexible, high-capacity, expressive generative models that can scale easily. In this thesis, we focus on the sequential setting of VDL, applied to a type temporally-correlated data, called time series. We introduce two different classes of sequential VDL, then apply them to two different types of highly nonlinear, noisy and irregularly sampled time series data: observations of dynamical systems and maritime traffic data.

Three-quarter's of the Earth's surface is covered by water. Since the dawn of life, the ocean has evolved and interacted tightly with the planet and its climate. For humankind, the ocean has provided rich physical and biological resources, as well as a major transport medium. Nowadays, with the rising concern of climate change as well as the rapid growth of globalisation and global trade, the studies of oceanography and maritime traffic are attracting a lot of attention. Understanding the dynamics of the ocean helps forecast

weather condition, simulate climate change, evaluate the impact of waves and tides to the coastal areas, *etc*. Monitoring, analysing and modelling maritime traffic play an important role in maritime safety and security. Maritime traffic surveillance also contributes to **maritime domain awareness** (MDA), fishing control and smuggling detection. To meet the needs for oceanography and maritime traffic surveillance, maritime data are more and more collected. More sensors are placed in the ocean, on the surface (Sendra et al. 2015), along the coastline (Bresnahan et al. 2020) and especially in the sky (Biancamaria et al. 2016) to measure the ocean. For maritime traffic, the **automatic identification system** (AIS) provide a fine-grained, rich information source of data. Everyday, on a global scale there are hundreds of millions of AIS messages transmitted (Perobelli 2016). The huge amount of available data makes deep learning a plausible approach.

However, there are still many problems to tackle. Usually we do not have direct access to the true states of the ocean dynamics. Instead, we observe a series of damaged and potentially incomplete measurements. As mentioned above, DL does not deal well with this type of data. Nevertheless, the hidden processes of oceanographic data obey fundamental physical laws. This is again a drawback of deep learning. Similarly, AIS trajectories are just noisy, potentially irregularly-sampled observations of the underlying movement patterns of vessels. Without prior knowledge integrated, DL can hardly capture those patterns.

Conducted within the framework of ANR (French Agence Nationale de la Recherche) AI Chair OceaniX, this thesis aims to exploit deep learning for ocean sciences. Because maritime data are usually sequential, noisy and irregularly sampled, we focus on a family of sequential models which use variational inference to uncover the hidden dynamics of the learning data. We combine deep learning architectures with probabilistic models for time series, and integrate prior knowledge of the domain to create a novel framework for learning dynamical systems [1] (Part II) and a novel deep learning model for maritime surveillance using AIS data (Part III). The details will be presented in the next sections.

This thesis work is supported by public funds (Ministère de l'Education Nationale, de l'Enseignement Supérieur et de la Recherche, FEDER, Région Bretagne, Conseil Général du Finistère, Brest Métropole); by ANR (French Agence Nationale de la Recherche), under grants Melody and OceaniX; and by Institut Mines Télécom, received in the framework of the VIGISAT program managed by "Groupement Bretagne Télédétection" (BreTel). It benefits from HPC and GPU resources from Azure (Microsoft EU Ocean awards) and

1. In this thesis we introduce this framework for the identification of general dynamical systems, some specific applications of this framework in geophysical oceanography are presented in our related work in (Ouala, Duong Nguyen, Herzet, et al. 2019) and (Ouala, Duong Nguyen, Drumetz, et al. 2020).

from GENCI-IDRIS (Grant 2020-101030). The work in Part III is supported by DGA (Direction Générale de l'Armement) and by ANR under reference ANR-16-ASTR-0026 (SESAME initiative).

The primary target audience of this thesis is DL practitioners whose research interests focus on geoscience, marine science and maritime traffic. We suppose that readers have enough background on dynamical system, AIS, probabilistic and deep learning.

1.2 Outline and contributions

This thesis contains three main parts. In the first part, we provide the motivation, the formulation and the construction of general **variational deep learning** (VDL) frameworks for time series analysis. This part aims to provide the "big picture" of different deep learning models for sequential data, how they are constructed and the relations between them. In the next parts, we present our models specifically designed for domain applications: dynamical system identification (Part II) and maritime traffic surveillance using AIS data (Part III). The details are as follows:

— In Part I (Chapter 2) we introduce two classes of deep latent variable models for sequential data: **deep state space models** (DSSMs) and**sequential variational autoencoders** (SVAEs). We present the derivations of these models, starting from **latent variable models** (LVMs)—which are the bricks to build all the models in this thesis—to their two sequential extensions: **state space models** (SSMs) and **recurrent neural networks** (RNNs). DSSMs and SVAEs are then obtained by combining them with variational deep learning, to help these models become more expressive and scalable.

— Part II (Chapter 3 and Chapter 4) contains paper (Duong Nguyen, Ouala, et al. 2020b) in which we present a DSSM, called **data-assimilation-based ordinary differential equation network** (DAODEN), specifically designed for learning dynamical system. DAODEN uses state-of-the-art neural network architectures to model the dynamics of **ordinary differential equations** (ODEs) systems and possibly of **stochastic differential equations** (SDEs) systems. DAODEN contains two key components: an inference model that mimics classical data assimilation methods to reconstruct the true hidden states of the systems from noisy and potentially partial observations, and a generative model that use state-of-the-art neural networks representation of dynamical systems to retrieve the underlying dynamics of these

states. Therefore, by construction, DAODEN can obtain comparable performance with the one of models trained on ideal observations, even when DAODEN is trained on highly damaged data.

— Part III (Chapters 5, 6 and 7) contains papers (Duong Nguyen, Vadaine, et al. 2018; Duong Nguyen, Vadaine, et al. 2019; Duong Nguyen, Simonin, et al. 2020), in which we present **MultitaskAIS** and **GeoTrackNet**. MultitaskAIS is a multitask deep learning architecture for maritime traffic surveillance using AIS data (Chapter 6). The core of this architecture is an SVAE, which converts noisy and irregularly sampled AIS messages into series of clean and regularly sampled hidden states of the vessel's trajectory. These states then can be used for task-specific sub-models (such as trajectory reconstruction, vessel type identification, anomaly detection). Experiments show that MultitaskAIS can achieve state-of-the-art performance on those takes, while requiring a significantly smaller storage and computational need. GeoTrackNet is the anomaly detection submodel of MultitaskAIS. This model takes into account the fact that the performance of a model that represents AIS trajectories are location-dependant to create a geospatially-sensitive detector that can effectively detect anomalies in vessels' behaviour.

— Part IV (Chapter 7) finally summarises the contributions of this thesis, discusses the remain challenges and some directions for future work.

— The Appendix A is an application our idea for anomaly detection using VDL to acoustic anomaly detection in the appendix.

You never really understand a person until you consider things from his point of view... Until you climb inside of his skin and walk around in it.

Harper Lee

Chapter 2

Variational Deep Learning for Time Series Modelling and Analysis

When we monitor or track a process, the sequences of the obtained observations are usually temporally correlated. This type of data is called time series. Modelling time series is a challenging task, because most of the time we do not know the governing laws that define the dynamics of the considered process. These laws can be highly nonlinear, chaotic and/or stochastic. Moreover, the data that we obtain may not be the true states of the process, but rather the noisy and partial observations/measurements. Over the last few years, sequential variational deep learning has emerged as a very promising approach for time series modelling and time series analysis (R. G. Krishnan et al. 2017; J. Chung et al. 2015; Fraccaro et al. 2016). This approach combines probabilistic modelling and deep learning (usually RNN-based networks) to create high capacity, expressive models that can capture the stochasticities, variations, uncertainties and long-term correlations in the data. In this chapter, we will present the motivation, the formulation and the applications of this approach. The content presented here is the theory part of the applications in Part II and Part III.

2.1 Latent variable models (LVMs)

2.1.1 Motivation

Given a set of possibly high-dimensional observations $\mathbf{X} = \{\mathbf{x}^{(1)}, ..\mathbf{x}^{(N)}\}$, the goal of probabilistic unsupervised learning models is to learn a probability distribution $p(\mathbf{x})$ that well describes $\mathbf{X}$. **Latent variable models** (LVMs) introduce an unobserved **latent variable** $\mathbf{z}$ that helps model $p(\mathbf{x})$. The joint distribution $p(\mathbf{x}, \mathbf{z})$ is then computed as:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}), \tag{2.1}$$

where $p(\mathbf{z})$ is the **prior distribution** over $\mathbf{z}$ and conditional distribution $p(\mathbf{x}|\mathbf{z})$ is the **emission distribution** over $\mathbf{x}$, given $\mathbf{z}$. The **posterior distribution** $p(\mathbf{z}|\mathbf{x})$ can be computed using the Bayesian formula:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}. \tag{2.2}$$

If $\mathbf{z}$ is continuous, $p(\mathbf{x})$ can be obtained by marginalising over $\mathbf{z}$:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z})\mathrm{d}\mathbf{z} = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}. \tag{2.3}$$

For discrete variables, we replace the integration above by the sum over all possible value of $\mathbf{z}$. In this paper, we present only the formula for continuous variables, however, most of the ideas also apply to the discrete case. We may also use discrete $\mathbf{z}$ for some demonstrating examples.

The underlying hypothesis of VLMs is in order to generate an observation $\mathbf{x}^{(s)}$, we first draw a sample $\mathbf{z}^{(s)}$ from $p(\mathbf{z})$, then use it to draw a new sample from the emission distribution $p(\mathbf{x}|\mathbf{z}^{(s)})$. There are different ways to interpret the latent variable $\mathbf{z}$. For some applications, $\mathbf{z}$ is considered as the true physical event, of which $\mathbf{x}$ is just a corrupted observation. An example of this interpretation is the famous Kalman filter that was used in the Apollo project (Kalman 1960). Another way of interpreting $\mathbf{z}$ is that the latent variable allows us to factor the complex and possibly intractable distribution $p(\mathbf{x})$ into more tractable distributions $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$. For example, to generate a human portrait image $\mathbf{x}$, the latent variable $\mathbf{z}$ may contain the gender, age, race of that human.

Latent variables have been widely used in statistics and in machine learning. Among

many famous others, we may name **principal component analysis** (PCA), **mixture models**, **hidden Markov models** (HHMs), **state space models** (SSMs) (Bishop 2006), **variational auto-encoders** (VAEs) (Diederik P. Kingma and Welling 2013; Rezende and Mohamed 2015). We will go through some of those models later in this thesis.

2.1.2 Variational inference (VI)

One of the main task in LVMS is to calculate the posterior distribution $p(\mathbf{z}|\mathbf{x})$. However, apart from a small set of simple cases, this distribution is intractable because the integral in Eq. (2.3) does not have an analytic solution. In such situations, we have to approximate $p(\mathbf{z}|\mathbf{x})$. There are two classes of techniques for this approximation:

— *Stochastic techniques*: this class uses sampling techniques to generate an ensemble of points that represent the distribution to estimate. If the number of points is large enough, the approximations converge to the exact results. An example of the methods in this class is Gibbs sampling (Geman et al. 1984; Barbos et al. 2017; Féron et al. 2016). However, those methods are computationally expensive and do not scale well to large data sets.

— *Deterministic approximation techniques*: this class uses analytical approximations to $p(\mathbf{z}|\mathbf{x})$. They impose the assumption that the posterior comes from a particular parametric family of distributions or that it factorises in a certain way. These methods scale very well, however, they never generate the exact results. An example of the methods in this class is variational inference (Blei et al. 2017), which is the topic of this section.

Recall that the objective is to find the distributions of two variables $\mathbf{x}$ and $\mathbf{z}$ that maximise the likelihood of the set of observations $\mathbf{X}$. In practice, we usually use $\log p_{\boldsymbol{\theta}}(\mathbf{X})$ instead of $p_{\boldsymbol{\theta}}(\mathbf{X})$ to leverage some nice properties of the log function and to avoid problems with very small numbers in numerical implementation. We focus on a family of distribution $p_{\boldsymbol{\theta}}$ parameterised by a set of parameters $\boldsymbol{\theta}$. For any arbitrary distribution q, we can

decompose $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ as follows:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \int q(\mathbf{z}) \log p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{z} \tag{2.4}$$

$$= \int q(\mathbf{z}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z} \tag{2.5}$$

$$= \int q(\mathbf{z}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) q(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) q(\mathbf{z})} \mathrm{d}\mathbf{z} \tag{2.6}$$

$$= \int q(\mathbf{z}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \mathrm{d}\mathbf{z} + \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} \mathrm{d}\mathbf{z} \tag{2.7}$$

$$= \mathbb{E}_{q(\mathbf{z})} \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right] + \mathrm{KL}\left[q(\mathbf{z}) || p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right], \tag{2.8}$$

with $\mathrm{KL}\left[q||p\right]$ denotes the Kullback–Leibler divergence between two distributions q and p.

Because the KL divergence is a non-negative quantity, the first term in the right hand side of Eq. (2.8), denoted as $\mathcal{L}(\mathbf{x}, p_{\boldsymbol{\theta}}, q)$, is a lower bound of $\log p_{\boldsymbol{\theta}}(\mathbf{x})$. $\mathcal{L}(\mathbf{x}, p_{\boldsymbol{\theta}}, q)$ is called the **evidence lower bound** (ELBO). **Variational inference** (VI) suggests approximating the intractable quantity $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ by $\mathcal{L}(\mathbf{x}, p_{\boldsymbol{\theta}}, q)$. The error of this approximation, *i.e.* the difference between $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ and $\mathcal{L}(\mathbf{x}, p_{\boldsymbol{\theta}}, q)$ is $\mathrm{KL}\left[q(\mathbf{z})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right]$. Hence, to find $\log p_{\boldsymbol{\theta}}(\mathbf{x})$, we minimise $\mathrm{KL}\left[q(\mathbf{z})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right]$ w.r.t. $q(\mathbf{z})$. Because $\mathcal{L}(\mathbf{x}, p_{\boldsymbol{\theta}}, q) = \log p_{\boldsymbol{\theta}}(\mathbf{x})$ if and only if $q(\mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, a natural choice for q is $q(\mathbf{z}) = q(\mathbf{z}|\mathbf{x})$. In other words, VI converts an intractable inference problem to an optimisation problem by approximating the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ by the variational distribution $q(\mathbf{z}|\mathbf{x})$.

Note that those above are correct for any arbitrary q. Hence to make a good approximation, we should choose $q(\mathbf{z}|\mathbf{x})$ high capacity enough, as long as the ELBO is tractable. In this thesis, we focus on a family of distributions $q_{\boldsymbol{\phi}}$ that can be parameterised by a set of parameters $\boldsymbol{\phi}$.

2.1.3 Objective functions

The goal of probabilistic unsupervised learning is to maximise $\log p_{\boldsymbol{\theta}}(\mathbf{X})$. In Section 2.1.2 we introduced the ELBO, which is an lower bound of $\log p_{\boldsymbol{\theta}}(\mathbf{X})$, as an objective function for the learning. Many efforts have been conducted to tighten this bound. Among them, we might cite the **importance weighted auto-encoder** (IWAE) bound (Burda et al. 2016) and the **fitering variational objective** (FIVO) (Maddison et al. 2017), which is used for sequential data. The idea is instead of drawing only one sample from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, we draw N samples then average the importance-weighted results. These methods

guarantee that the bounds are tighter than the ELBO. However, on one hand, they use more computational resources. On the other hand, tighter variational bounds are not necessarily better (Rainforth et al. 2018). For the applications in this thesis, we empirically observed that the trade-off is not worth, hence we do not present IWAE and FIVO here.

Another example of using loose lower bounds is **maximum a posteriori** (MAP) inference. Instead of estimating a distribution for the latent variable $\mathbf{z}$, MAP inference computes only the single most likely value:

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmax}} q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}). \tag{2.9}$$

In the context of VI, the MAP solution can be explained as the case where q is parameterised by Dirac delta functions. Although the MAP bound is infinitely loose, MAP inference is still very common (Bishop 2006).

2.1.4 Optimisation methods

Given an objective function $\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi})$ (ELBO, IWAE, FIVO, MAP solution, *etc.*) as presented in Section 2.1.3, the next step is to optimise this quantity over the observations w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. In this section, we present two strategies to perform this optimisation: i) alternatively optimise $\mathcal{L}$ over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using the **expectation-maximisation** (EM) algorithm and ii) simultaneously update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using the gradient of $\mathcal{L}$.

Expectation–maximisation (EM) algorithm

The expectation-maximisation algorithm is an iterative optimisation technique for LVMs (Dempster et al. 1977; C. J. Wu 1983; Neal et al. 1998). Starting from the initial condition $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\phi}^{(0)}$, each iteration i in EM contains two steps:

— In the E step, $\boldsymbol{\phi}$ is updated to maximise $\mathcal{L}$: $\boldsymbol{\phi}^{(i)} = \underset{\boldsymbol{\phi}}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\phi})$. This step corresponds to finding the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ (because $\mathcal{L}$ is maximised when $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$).

— In the M step, $\boldsymbol{\theta}$ is updated, while $\boldsymbol{\phi}$ is held fixed: $\boldsymbol{\theta}^{(i)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}^{(i)})$. This step corresponds to increasing the objective function $\mathcal{L}$.

EM has some nice properties, such as the convergence is fast and guaranteed [1] (Ghahramani and Roweis 1999). However, EM may not apply for very general configurations,

1. EM is guaranteed to converge to a point with zero gradient.

because the inference problem may not be tractable when considering complex dependence structure.

Gradient-based techniques

Another strategy to optimise $\mathcal{L}$ is to update $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ simultaneously using the gradient of $\mathcal{L}$. This approach is widely used for neural networks. For example, in the current context, we can update $\{\boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}^{(i)}\}$ using a gradient ascent technique:

$$\{\boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}^{(i)}\} = \{\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\phi}^{(i-1)}\} + \eta \nabla_{\{\boldsymbol{\theta},\boldsymbol{\phi}\}} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi}). \tag{2.10}$$

with η is the learning rate.

Because maximising $\mathcal{L}$ is equal to minimising $-\mathcal{L}$, we can re-write Eq. (2.11) as follows:

$$\{\boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}^{(i)}\} = \{\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\phi}^{(i-1)}\} - \eta \nabla_{\{\boldsymbol{\theta},\boldsymbol{\phi}\}} (-\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi})). \tag{2.11}$$

The second notation, called gradient descent, is more common.

In deep learning, we usually use a "stochastic version" of Eq. (2.11). In each iteration, instead of evaluating $\nabla_{\{\boldsymbol{\theta},\boldsymbol{\phi}\}} - \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi})$ on the whole observation set $\mathbf{X}$, we calculate this quantity on just a subset of $\mathbf{X}$, called **mini-batch**. This technique is known as **stochastic gradient descent** (SGD) (Bottou et al. 2018). SGD is the basic form of gradient-based optimisation techniques used in DL, many variants and extensions of SGD, such as AdaGrad (Duchi et al. 2011), RMSprop (G. Hinton 2012), Adam (Diederik P. Kingma and Ba 2015), have been proposed to improve the performance of the learning. These methods have been widely available in open-source DL frameworks such as Tensorflow (Abadi et al. 2016), Pytorch (Paszke et al. 2017).

2.1.5 Variational Auto-Encoders (VAEs)

So far we have reviewed LVMs and how to overcome the intractable inference problem in LVMs using VI. In this section, we present a class of LVMs that is extremely popular in probabilistic DL: the **variational auto-encoders** (VAEs) (Diederik P. Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014). VAEs are used as bricks to build many generative models.

The architecture of VAEs is the basic form of VLMs, as show in Fig. 2.1. We have the observed variable $\mathbf{x}$, the latent variable $\mathbf{z}$, the emission distribution $p_\theta(\mathbf{x}|\mathbf{z})$ and the

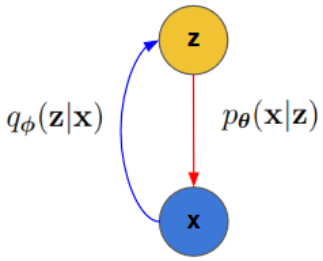


Figure 2.1 – Graphical model of a VAE. $\mathbf{x}$ is the observed variable, $\mathbf{z}$ is the latent variable. In this thesis, we use circle-shaped units for random variables, blue colour for observed variables, yellow colour for latent variables, red arrows for emission models and blue arrows for inference models.

variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ that approximates the true posterior distribution. Again, in this thesis we focus on parametric models.

To build a VAE, there are three problems to deal with: i) how to define $\mathbf{x}$ and $\mathbf{z}$, ii) how to define the emission model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ and iii) how to define the inference model $q_\phi(\mathbf{x}|\mathbf{z})$.

Most of the time we $\mathbf{x}$ is what we observe. *e.g.* the value of the pixels in an digital image, the temperature indicated by a thermometer in the room, *etc.* However, in some cases, we can use prior knowledge to convert observed data into another domain that is believed to be more suitable for the considered problem. For example, we can convert an audio record to a spectrogram to highlight some important features in the frequency domain. In this case, $\mathbf{x}$ is the spectrogram of the audio signal. The "four-hot vector" presented in Chapter 6 and Chapter 7 of this thesis is a specific representation we designed for AIS. As presented in Section 2.1.1, the interpretation of $\mathbf{z}$ is heavily context-dependent. For example, to generate an human portrait image, $\mathbf{z}$ can be the distance between two eyes, the colour of the skin, the ratio of the width to the length of the face, *etc.* If we do not have any prior knowledge of $\mathbf{z}$, we usually model $\mathbf{z}$ as a continuous variable, whose prior is an isotropic multivariate Gaussian with mean $\mathbf{0}$ and covariance matrix $\mathbf{I}$.

$$p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2.12}$$

Other distributions can also be used. However, unless we have prior knowledge of $\mathbf{z}$, we should avoid multimodal distributions whose modes are sufficiently widely separated. To understand the reason behind it, let's rewrite the ELBO as follows:

$$\mathcal{L}(\mathbf{x}, p_{\boldsymbol{\theta}}, q_\phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})\right]. \tag{2.13}$$

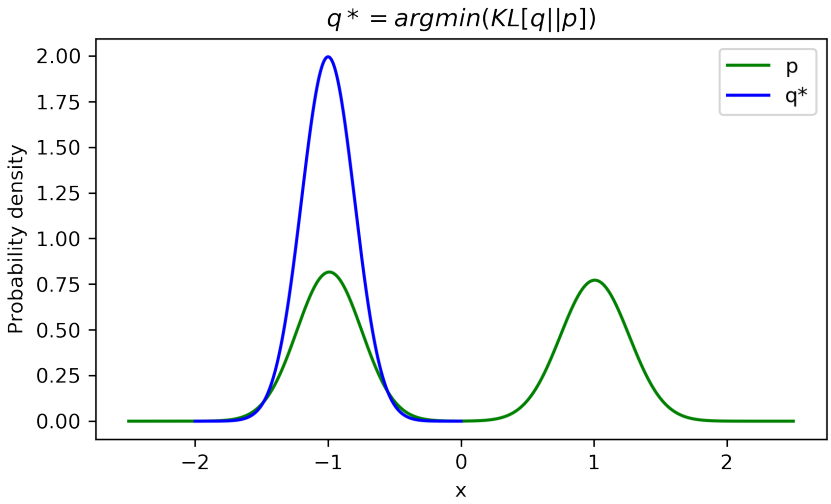


Figure 2.2 – The over concentration problem of minimising KL$(q||p)$. When p is a multimodal distribution, the optimisation might result in a distribution q that corresponds to only one mode of p.

Hence, to maximise $\mathcal{L}(\mathbf{x}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}})$, we have to simultaneously maximise $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right]$ and minimise KL $\left[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})\right]$ w.r.t. $q_{\boldsymbol{\phi}}$. Minimising KL $\left[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z})\right]$ means choosing $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ that has low probability wherever $p_{\boldsymbol{\theta}}(\mathbf{z})$ is small. If $p_{\boldsymbol{\theta}}(\mathbf{z})$ is a multimodal distribution that has sufficiently widely separated modes, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ might just choose one of those modes, as shown in Fig. 2.2

Given this distribution, we define a sufficiently complicated function that maps $\mathbf{z}$ to $\mathbf{x}$. This mapping is called the **decoder**. In DL, the emission distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ is usually modeled by a Gaussian distribution (for real-valued $\mathbf{x}$) or a Bernoulli distribution (for binary $\mathbf{z}$), whose parameters are computed by a neural network. For example:

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{z}}). \tag{2.14}$$

with

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{z}} = NN_{decoder}(\mathbf{z}). \tag{2.15}$$

Because we can not find the inverse function of the neural network $NN_{decoder}$, the inference $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ is intractable. As presented in Section 2.1.2, we will approximate this posterior distribution by a variational distribution q, using some hypotheses, such as $q_{\boldsymbol{\phi}}$ is a factorial distribution:

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \prod_i q_{\boldsymbol{\phi}}(\mathbf{z}_i|\mathbf{x}). \tag{2.16}$$

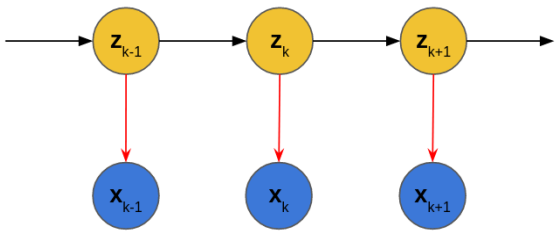


Figure 2.3 – Graphical model of an SSM. We use black arrows for transition models.

This technique—called the mean field technique, is widely used to simplify the behavior of high-dimensional stochastic models (Landau 1937; Flory 1942; Huggins 1941).

In DL, we usually use another network to model the mapping from $\mathbf{x}$ to $\mathbf{z}$, called the **encoder**:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{z}|\mathbf{x}}). \tag{2.17}$$

with

$$\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{z}|\mathbf{x}} = NN_{encoder}(\mathbf{x}). \tag{2.18}$$

The parameters of the encoder and the decoder are then optimised using gradient-based techniques with an objective function presented in Section 2.1.2.

2.2 State space models (SSMs)

In the previous section, we assume that the observations are **independent and identically distributed** (i.i.d). However, in the real world, there are many situations where there is a temporal correlation in the data, for example, the measurements of the temperature at a particular place in two days, or two words in a sentence, are related. Such data are called **time series**. In this section, we introduce a way to model this type of data, especially how to model the temporal correlation in time series.

2.2.1 Formulation

A regularly-sampled time series $\mathbf{x}_{0:T}$ is a sequence of $T+1$ observations: $\mathbf{x}_{0:T} \triangleq \{\mathbf{x}_{t_0}, .., \mathbf{x}_{t_T}\}$, where t_k refers to the time sampling. We consider cases where the sampling is regular: $t_k = t_0 + k.\delta$ with δ is the sampling resolution. For the sake of simplicity, from now on in this thesis, unless specified otherwise we use the notation $\mathbf{x}_k$ for $\mathbf{x}_{t_k}$ and $\mathbf{x}_{t+n}$ for $\mathbf{x}_{t_k+n.\delta}$. An irregularly sampled time series is a time series where some components

of $\mathbf{x}_k$, or entire $\mathbf{x}_k$ may be missing.

Given an observed time series of observations $\mathbf{x}_{0:T}$, we aim to find a model that maximise the likelihood:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_0) \prod_{k=1}^{N} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}). \tag{2.19}$$

If we assume that:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}) = p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{2.20}$$

then we have a **first-order Markov model**. In other words, in a first-order Markov model, the future states depend only on the present state. Generally, an **$\mathbf{n^{th}}$-order Markov model** assumes $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}) = p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{k-n:k-1})$. At first glance, Eq. (2.20) looks like a strong assumption, however, any n^{th}-order Markov model can be converted to a first-order Markov model by using an augmented state: $\mathbf{x}_k^{aug} = \{\mathbf{x}_k, \mathbf{x}_{k-1}, .., \mathbf{x}_{k-n+1}\}$.

Although Markov models may look appealing, they are not really useful to directly model $\mathbf{x}_{0:T}$ because the process of $\{\mathbf{x}_k\}$ may not follow the Markov assumption to any order. With the spirit of LVMs, we suppose that the data generating process of $\mathbf{x}_{0:T}$ depends on a series of latent variables $\mathbf{z}_{0:T}$. For example, let $\mathbf{x}_k$ be the values of a thermometer measuring the temperature in a room, these values maybe effected by the errors in the sensor of the thermometer, hence we can not find the direct relation between two consecutive values $\mathbf{x}_k, \mathbf{x}_{k+1}$. In this case, $\mathbf{z}$ can be the true temperature of the room, and there is a direct relation, which is the heat equation, between two consecutive values of $\mathbf{z}_k, \mathbf{z}_{k+1}$.

The joint distribution can be factorised as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}). \tag{2.21}$$

The likelihood of the observation can be obtained by marginalising the latent variables:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) \mathrm{d}\mathbf{z}_{0:T}. \tag{2.22}$$

Again, in general this integral is intractable. To compute $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$, we have to impose some hypotheses and some assumptions on the relation of $\mathbf{x}_{0:T}$ and $\mathbf{z}_{0:T}$. One way of doing so is using **state space models** (SSMs). The general form of an SSM is expressed as

follow [2]:

$$\mathbf{z}_k \sim p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}) \tag{2.23}$$

$$\mathbf{x}_k \sim p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) \tag{2.24}$$

with $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})$ is the **transition distribution** (or **prior distribution**), models the temporal evolution of $\mathbf{z}_k$, and $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ is the **emission distribution** (or the **observation distribution**), models the observation operator. The graphical model of an SSM is shown in Fig. 2.3. Note that the distributions in Eqs. (B.6) and (B.7) are general, they include Dirac delta functions that model cases where $\mathbf{z}_k$ is deterministic.

In order to determine $p_{\boldsymbol{\theta}}$, SSMs use some characteristics presented in the following sections.

2.2.2 Properties

Using the d-separation criterion, we derive some important properties of SSMs are as follows:

— The process of $\{\mathbf{z}_k\}$ is Markovian, *i.e.* the future states depend only on the current state $\mathbf{z}_k$.

— Given $\mathbf{z}_k$, $\mathbf{x}_k$ does not depend on other states or observations: $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{x}_{k+1:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$.

— The process of $\{\mathbf{x}_k\}$ is **not** Markovian, *i.e.* the future observation $\mathbf{x}_{k+1}$ depends on the present and all the past observations $\mathbf{x}_{0:k}$.

Hence, the joint distribution in SSMs can then be factorised as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_0)\prod_{k=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})\prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k). \tag{2.25}$$

Depending on the form of the transition and the emission distributions, a particular SMM may have other properties, which could be used to design particular models. For examples, if the latent variables $\mathbf{z}_k$ are discrete, we can use **hidden Markov models** (HMMs) (Rabiner 1989), **Kalman filters** (Kalman 1960) are designed for SSMs whose the transition and the emission distributions are Gaussian.

2. In this thesis, we focus on SSMs that do not have control input.

2.2.3 Posterior inference for SSMs

According to the Bayesian rule, given the whole sequence, the form of the posterior inference for SSMs is expressed as follows:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})}{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}. \tag{2.26}$$

In practice, we are usually interested in the following three inference problems:

— Filtering: $\mathbf{z}_k$ is inferred using the all the present and the past observations; *i.e.*, to we compute $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k})$.

— Smoothing: using the d-separation criterion, $\mathbf{z}_k$ depends not only on the current and the past observations but also on the future observations, hence we compute $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:T})$. Because smoothing uses more information from data than filtering, it should provide a better inference. However, smoothing requires information from the future, hence there is always a lag. On the other hand, filtering can be computed online.

— Prediction: we use the current and the past information to predict the future states, *i.e.* to compute $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+1}|\mathbf{x}_{0:k})$.

When the transition and the emission are linear and Gaussian, the Kalman filter (Kalman 1960) provides a mathematically elegant solution for the inference problem. However, when the transition and/or the emission are not linear and Gaussian anymore, the posterior becomes intractable. We have to perform some approximations. For cases where the transition and the emission can be described by differentiable functions, the **extended Kalman filter** (EKF) (Smith et al. 1962) approximates the posterior by a linearisation of $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$. The **particle filter** (Doucet et al. 2009) has a different approach. This method uses sequential important sampling to recursively approximate $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:k}|\mathbf{x}_{0:k})$ given $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:k-1}|\mathbf{x}_{0:k-1})$. In particle filters, a distribution is represented by a set of particles. The **ensemble Kalman filter** (EnKF) (Evensen 2003) bridges the idea of the Kalman filter and the particle filter by supposing that the distributions represented by the particles are Gaussian. Each of those methods also has a corresponding smoothing version.

2.2.4 Example: linear Gaussian SSMs (LGSSMs)

To better understand SSMs, let's revisit one of the most classic SSMs: the Kalman filter (Kalman 1960).

Consider a system governed by the following equations:

$$\mathbf{z}_k = \mathbf{A}_k \mathbf{z}_{k-1} + \boldsymbol{\omega}_k, \tag{2.27}$$

$$\mathbf{x}_k = \mathbf{H}_k \mathbf{z}_k + \boldsymbol{\upsilon}_k. \tag{2.28}$$

where $\mathbf{A}_k$ is a matrix, defines the *transition* of the hidden state $\mathbf{z}_k$; $\mathbf{H}_k$ is an invertible matrix, defines the *observation* operator which maps the state $\mathbf{z}_k$ to the observation $\mathbf{x}_k$ at each timestep k. $\boldsymbol{\omega}_k$ is the *process noise*, follows a *zero-mean multivariate Gaussian* $\mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ and $\boldsymbol{\upsilon}_k$ is the *observation noise*, follows another *zero-mean multivariate Gaussian* $\mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. All $\boldsymbol{\omega}_k$ and $\boldsymbol{\upsilon}_k$ are mutually independent. The covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ are called the *model error* and the *observation error*, respectively. Using the linear transformation properties of Gaussian random variables, we can re-write Eqs. (2.27) and (2.28) as:

$$\mathbf{z}_k \sim p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}) = \mathcal{N}(\mathbf{A}_k \mathbf{z}_{k-1}, \mathbf{Q}_k), \tag{2.29}$$

$$\mathbf{x}_k \sim p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) = \mathcal{N}(\mathbf{H}_k \mathbf{z}_k, \mathbf{R}_k). \tag{2.30}$$

here $\boldsymbol{\theta}$ is the set $\{\mathbf{A}_k, \mathbf{H}_k, \mathbf{Q}_k, \mathbf{R}_k\}$. Because the relationships between $\mathbf{z}_k$ and $\mathbf{z}_{k-1}$, between $\mathbf{x}_k$ and $\mathbf{z}_k$ are linear, and the two distributions $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})$, $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ are Gaussian, this model is called a **linear Gaussian SSM** (LGSSM).

The joint distribution of a LGSSM is factorised as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_0) \prod_{t=1}^{T} \mathcal{N}(\mathbf{z}_k; \mathbf{A}_k \mathbf{z}_{k-1}, \mathbf{Q}_k) \prod_{t=0}^{T} \mathcal{N}(\mathbf{x}_k; \mathbf{H}_k \mathbf{z}_k, \mathbf{R}_k). \tag{2.31}$$

To find the inference distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$, Kalman proposed a recursive algorithm to compute the marginal posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k})$ at each timestep k, given the $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1})$.

Using Bayes' rule and the independence properties of SSMs, we have:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k}) = p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_k, \mathbf{x}_{0:k-1}) \tag{2.32}$$

$$= \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{x}_{0:k-1})p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1})}{p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1})} \tag{2.33}$$

$$= p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1})\frac{p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)}{p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1})}. \tag{2.34}$$

Eq. (2.34) show that to calculate $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k})$, we first calculate $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1})$, *i.e.* to predict $\mathbf{z}_k$ using historical information in $\mathbf{x}_{0:k-1}$ (*prediction step*), then "correct" this prediction using the information provided by $\mathbf{x}_k$ (*measurement step*).

Suppose $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k}) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1}) = \mathcal{N}(\boldsymbol{\mu}_{k|k-1}, \boldsymbol{\Sigma}_{k|k-1})$. Lets compute each component in Eq. (2.34) step by step.

We have:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1}) \triangleq \mathcal{N}(\boldsymbol{\mu}_{k|k-1}, \boldsymbol{\Sigma}_{k|k-1}) \tag{2.35}$$

$$= \int p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{x}_{0:k-1})p_{\boldsymbol{\theta}}(\mathbf{z}_{k-1}|\mathbf{x}_{0:k-1})\mathrm{d}\mathbf{z}_{k-1} \tag{2.36}$$

$$= \int p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})p_{\boldsymbol{\theta}}(\mathbf{z}_{k-1}|\mathbf{x}_{0:k-1})\mathrm{d}\mathbf{z}_{k-1} \tag{2.37}$$

$$= \int \mathcal{N}(\mathbf{A}_k\mathbf{z}_{k-1}, \mathbf{Q}_k)\mathcal{N}(\boldsymbol{\mu}_{k-1}, \boldsymbol{\Sigma}_{k-1})\mathrm{d}\mathbf{z}_{k-1} \tag{2.38}$$

$$= \mathcal{N}(\mathbf{A}_k\boldsymbol{\mu}_{k-1}, \mathbf{A}_k\boldsymbol{\Sigma}_{k-1}\mathbf{A}_k^T + \mathbf{Q}_k). \tag{2.39}$$

Note that (2.38) to (2.39) is possible because the model is linear and Gaussian.

Appling Bayes' rule for the Gaussian $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) = p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{x}_{0:k-1})$ (Murphy 2012), we have:

$$\boldsymbol{\Sigma}_k^{-1} = \boldsymbol{\Sigma}_{k|k-1}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}, \tag{2.40}$$

$$\boldsymbol{\mu}_k = \boldsymbol{\Sigma}_k\mathbf{H}\mathbf{R}_k^{-1}\mathbf{x}_k + \boldsymbol{\Sigma}_k\boldsymbol{\Sigma}_{k|k-1}^{-1}\boldsymbol{\mu}_{k|k-1}. \tag{2.41}$$

Applying the matrix inversion lemma (Murphy 2012) for $\boldsymbol{\Sigma}_k^{-1}$, we have:

$$\boldsymbol{\Sigma}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\boldsymbol{\Sigma}_{k|k-1}, \tag{2.42}$$

with $\mathbf{K}_k \triangleq \boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ is the **Kalman gain matrix**.

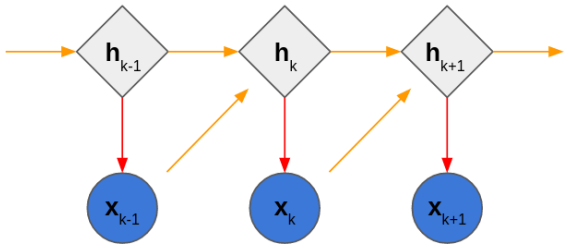


Figure 2.4 – Graphical model of a RNN. We use diamond-shaped units for deterministic variables, light yellow arrows for the recurrences.

Substituting Eq. (2.42) into Eq. (2.41), we have:

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k|k-1} + \mathbf{K}_k(\mathbf{x}_k - \mathbf{H}_k\boldsymbol{\mu}_{k|k-1}). \tag{2.43}$$

In conclusion, at each timestep k, the Kalman filter comprises two steps: i) the prediction step uses Eq. (2.39) to predict $\mathbf{z}_k$ given $\mathbf{x}_{1:k-1}$; ii) the measurement step uses Eqs. (2.42) and (2.43) to update the prediction given the additional information in the observation $\mathbf{x}_k$.

The Kalman filter can compute the exact inference distribution because the system is linear and Gaussian (Eqs. (2.39), (2.40), (2.41)). When these two conditions are not satisfied, the exact inference distribution becomes intractable and we have to use approximate inference.

2.3 Recurrent neural networks (RNNs)

In the previous section we introduced SSMs as a mean to model time series. The underlying idea of SMMs is to convert a non-Markovian series $\mathbf{x}_{0:T}$ to a Markovian series $\mathbf{z}_{0:T}$ and use some nice properties of Markov chains to factorise the transition, the emission and the inference distributions to calculate the likelihood in Eq. (2.22). In this chapter we present another way to model time series, using a type of deep neural networks called **recurrent neural networks** (RNNs).

Given a series of observations $\mathbf{x}_{0:T}$, RNN supposes that at time k, all the historical information can be encoded in a deterministic variable $\mathbf{h}_k$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_0)\prod_{k=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}) = \prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k). \tag{2.44}$$

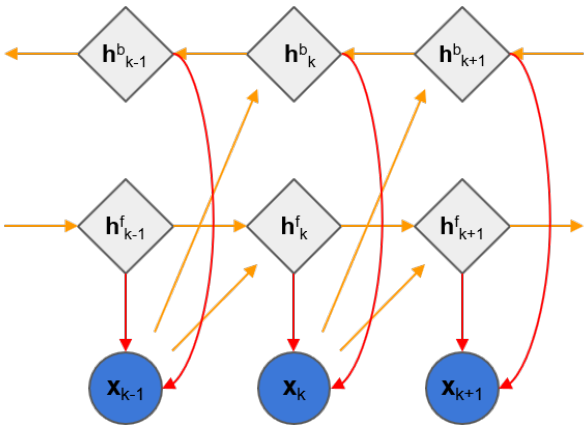


Figure 2.5 – Graphical model of a bidirectional RNN. $\mathbf{h}_k^f$ and $\mathbf{h}_k^b$ are the hidden states of the RNNs that move forwards and backwards, respectively.

The graphical model of a RNN is depicted in Fig. 2.4. At each time step k, the state $\mathbf{h}_k$ of the network is updated using information from the previous state $\mathbf{h}_{k-1}$ and the previous observation $\mathbf{x}_{k-1}$:

$$\mathbf{h}_k = h_{\boldsymbol{\theta}}(\mathbf{h}_{k-1}, \mathbf{x}_{k-1}), \tag{2.45}$$

with $h_{\boldsymbol{\theta}}$ is a differentiable non-linear updating formula of RNN, such as those in (Elman 1990). To be able to capture time correlations in data, function $h_{\boldsymbol{\theta}}$ must be high capacity. Nowadays, we usually use gated extension versions of RNNs, such as LSTMs (Hochreiter et al. 1997) or GRUs (Junyoung Chung et al. 2015).

Note that by definition, an RNN is not an SSM, because the transition of the hidden state $\mathbf{h}_k$ depends on the observation $\mathbf{x}_k$. However, we can transform an RNN to an SSM by simply redefining the hidden state $\tilde{\mathbf{h}}_k = \{\mathbf{h}_k, \mathbf{x}_k\}$. In this thesis, we keep the original definition of the hidden states of RNN. Another way to frame an RNN as an SSM is to consider the previous observation as a control input for the next timestep. However this modelling may cause some confusions with the definition of autonomous systems used in Chapter 4. Further more, we call the process of updating the hidden states $\mathbf{h}_{k-1} \longrightarrow \mathbf{h}_k$ the recurrence, to distinguish with the transition $\mathbf{z}_{k-1} \longrightarrow \mathbf{z}_k$ in SSMs.

In comparison with SSMs, we can loosely say that in order to model $\mathbf{x}_{0:T}$, RNNs use a highly non-linear recurrent function to model the time correlation and a deterministic hidden state to make the likelihood in Eq. (2.22) tractable. The trade-off is because $\mathbf{h}_k$ is deterministic, RNNs can not capture all the variation and uncertainty in data.

The basic form of RNNs is a filtering model. We can also condition the hidden states of RNNs on future observations by using **bidirectional RNNs** to perform a smoothing

inference. There are several ways to construct a bidirectional RNN, in Fig. 2.5 we depict the graphical model of the most common one. A bidirectional RNN of this type can be considered as a network of two RNNs, one moves forwards in time and the other one moves backwards.

To find the parameters $\boldsymbol{\theta}$, RNNs use gradient-based optimisation techniques with backpropagation.

2.4 Variational deep learning for noisy and irregularly sampled time series modelling and analysis

In this section we present the main focus of the theory part of this thesis: the study of variational deep learning for noisy and irregularly sampled time series modelling and analysis. We have introduced latent variable models as a tool to model complex observations, and how to use variational inference to approximate the likelihood of the observations when the true posterior distribution is intractable. We have also presented state space models and recurrent neural networks as two ways to extend LVMs for time series. However, each of those models still contains some drawbacks. Although SSMs have some nice factorisation properties, in general the posterior distribution is still intractable. Modelling and propagating uncertainty in highly nonlinear SSMs are also still an open problem. On the other hand, RNNs provide tractable solution for highly nonlinear processes, however, because the hidden states $\mathbf{h}_k$ are deterministic, they can hardly capture all the variations and uncertainties in data.

To apply these models to stochastic non-linear time series, we have two options: either to make SSMs tractable for non-linear systems, or to add stochastic factors to RNNs. The former approach is called **deep state space models** (DSSMs), while the later one is called **sequential variational auto-encoders** (SVAEs). We will go through each of them in this section.

2.4.1 Deep state space models (DSSMs)

To overcome the difficulties of non-linearity, one can use neural networks to parameterise all the unknown distributions in SSMs. This approach is hence called deep state space models.

If $\mathbf{z}_k$ is real-valued, the transition distribution can be modelled by a Gaussian:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}) = \mathcal{N}(\boldsymbol{\mu}_k^{trans}, \boldsymbol{\Sigma}_k^{trans}), \tag{2.46}$$

with $\boldsymbol{\mu}_k^{trans}$ and $\boldsymbol{\Sigma}_k^{trans}$ are functions parameterised by a neural network:

$$\boldsymbol{\mu}_k^{trans}, \boldsymbol{\Sigma}_k^{trans} = NN_{\boldsymbol{\theta}}^{trans}(\mathbf{z}_{k-1}). \tag{2.47}$$

$\boldsymbol{\Sigma}_k^{trans}$ is usually chosen to be a diagonal matrix. However, full matrices can can also be used, as discussed in (Rezende, Mohamed, and Wierstra 2014), with the trade-off is the computational cost. Although one can use a sophisticated architecture for $NN_{\boldsymbol{\theta}}^{trans}$, such as gated transition functions in (Rahul G. Krishnan et al. 2016), most of the time an MLP would give a satisfied result with a reasonable calculation time.

As in VAEs, the choice of the form of the emission distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ is heavily context dependent. In general, if $\mathbf{x}_k$ is binary, $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ is chosen to be a Bernoulli distribution, if $\mathbf{x}_k$ is real-valued, $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ is chosen to be a Gaussian distribution. Again, we parameterise $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ by a neural network $NN_{\boldsymbol{\theta}}^{gen}(\mathbf{z}_k)$. The architecture of this network depends on $\mathbf{x}_k$. For example, if $\mathbf{x}_k$ is an image, $NN_{\boldsymbol{\theta}}^{gen}$ should contain convolutional units.

$NN_{\boldsymbol{\theta}}^{trans}$ and $NN_{\boldsymbol{\theta}}^{gen}$ together create an expressive and very high capacity model for the data generating process in Eq. (2.1). However, because of the non-linearity, the inference model becomes intractable. One may use one of the methods presented in Section 2.2.3 to approximate the true posterior distribution. In DSSMs, we usually approximate $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ by another network to increase the modelling capacity as well as the scalability of the model. Applying the d-separation criterion on the graphical model of SSMs in Fig. 2.3, we can factorise the true posterior distribution as follows:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_0|\mathbf{x}_{0:T}) \prod_{k=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{x}_{0:T}) \tag{2.48}$$

$$= p_{\boldsymbol{\theta}}(\mathbf{z}_0|\mathbf{x}_{0:T}) \prod_{k=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{x}_{k:T}). \tag{2.49}$$

Eq. (2.49) means that given $\mathbf{z}_{k-1}$, the posterior of $\mathbf{z}_k$ depends only on the present and the future observations. All the relevant information from the past has been encoded in $\mathbf{z}_{k-1}$. On the basis of this property, one may naturally choose the following variational

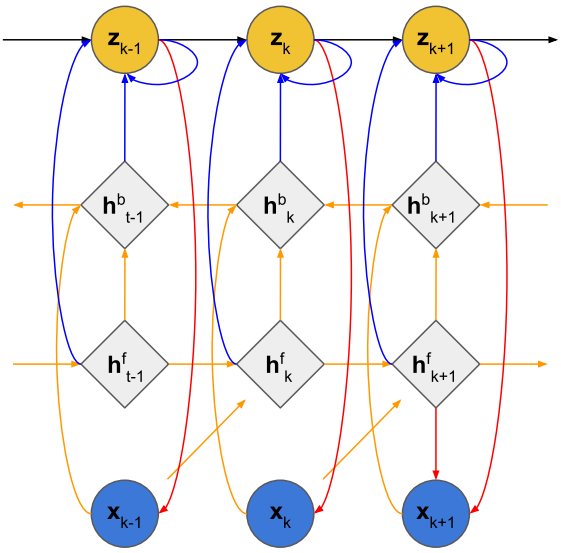


Figure 2.6 – Graphical model of a DSSM. The generative model comprises the transition (black arrows) and the emission (red arrows). The inference model contains the recurrence (yellow arrows) and the inference (blue arrows). (Note that different settings may be used).

distribution:

$$q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x}_{0:T}) \prod_{k=1}^{T} q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{x}_{k:T}). \tag{2.50}$$

However, in practice we empirically observed that conditioning the posterior on the whole observed sequence (Eq. (2.51)) gives a slightly better result. This is also in accordance with the results in (R. G. Krishnan et al. 2017). This may come from the fact that $q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ is just an approximation, and not the true solution of $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$.

$$q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x}_{0:T}) \prod_{k=1}^{T} q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{x}_{0:T}). \tag{2.51}$$

To encode the information contained in the observations $\mathbf{x}_{0:T}$, we can use a bidirectional RNN. Furthermore, to encode the relation $\mathbf{z}_k|\mathbf{z}_{k-1}$, we can leverage the transition model $NN_{\boldsymbol{\theta}}^{trans}$. $NN_{\boldsymbol{\theta}}^{trans}$ will predict the current state $\mathbf{z}_k$ given $\mathbf{z}_{k-1}$, and the inference model will "correct" this prediction. For example, in Fig. 2.6 we depict the graphical model of the DSSM used in Chapter 4. The data generating process comprises the transition (black arrows), modelled by $NN_{\boldsymbol{\theta}}^{trans}$, and the emission (red arrows), modelled by $NN_{\boldsymbol{\theta}}^{gen}$.

Formally, $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{x}_{0:T})$ is a Gaussian $\mathcal{N}(\boldsymbol{\mu}_k^{inf}, \boldsymbol{\Sigma}_k^{inf})$ parameterised by:

$$\mathbf{h}_k^f = f_{\boldsymbol{\phi}}^f(\mathbf{h}_{k-1}^f, \mathbf{x}_{k-1}), \tag{2.52}$$

$$\mathbf{h}_k^b = f_{\boldsymbol{\phi}}^b(\mathbf{h}_{k+1}^b, \mathbf{x}_k), \tag{2.53}$$

$$\boldsymbol{\mu}_k^{inf}, \boldsymbol{\Sigma}_k^{inf} = NN_{\boldsymbol{\phi}}^{inf}(\boldsymbol{\mu}_k^{trans}, \mathbf{h}_k^f, \mathbf{h}_k^b). \tag{2.54}$$

with $f_{\boldsymbol{\phi}}^f$, $f_{\boldsymbol{\phi}}^b$ are the formulas of the RNN forward and the RNN backward, respectively. $f_{\boldsymbol{\phi}}^f$, $f_{\boldsymbol{\phi}}^b$ are usually modelled by LSTMs or GRUs. Similarly to $NN_{\boldsymbol{\theta}}^{trans}$, $NN_{\boldsymbol{\phi}}^{inf}$ can be modelled by an MLP.

Now we have the full architecture of the model, the rest of the work is to define a loss function then optimise that loss w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using a gradient-based technique. Similarly to LVMs, we can use ELBO as the loss function. The ELBO for sequential data is defined as follows:

$$\mathcal{L}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})}\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T})}{q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T})}\right]. \tag{2.55}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) - \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})||p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})\right]\right] \tag{2.56}$$

Applying the factorisation in Eqs. (2.25), (2.51), we have:

$$\mathcal{L}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})}\left[\sum_{k=0}^{T}\left(\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) - \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{\mu}_k^{trans}, \mathbf{h}_k^f, \mathbf{h}_k^b)||p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})\right]\right)\right]. \tag{2.57}$$

In Eq. (2.64), we denote $q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{z}_{-1}, \mathbf{x}_{0:T}) = q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x}_{0:T})$ and $p_{\boldsymbol{\theta}}(\mathbf{z}_0|\mathbf{z}_{-1}) = p_{\boldsymbol{\theta}}(\mathbf{z}_0)$ for notational simplicity. This simplification is applied from now on for all equations in this thesis.

To optimise $\mathcal{L}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}})$ w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, we use a Monte Carlo estimator of its gradient, as presented in (Hoffman et al. 2012).

2.4.2 Sequential variational auto-encoders (SVAEs)

RNNs are highly non-linear models that can capture long-term dependencies in sequential data. However, because the hidden states $\mathbf{h}_k$ of RNNs are deterministic, they can not model stochasticity. To increase the modelling capacity of RNNs, one may augment RNNs by adding stochastic components. In this section, we will go through some of those models.

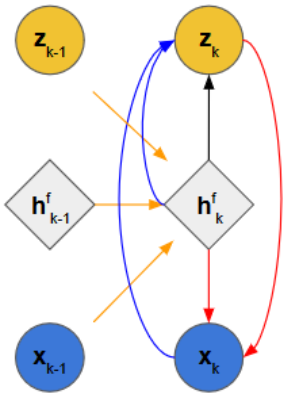


Figure 2.7 – Graphical model of a VRNN.

Variational recurrent neural networks (VRNNs)

The first architecture that we introduce first in this section is the **variational recurrent neural networks** (VRNNs) (J. Chung et al. 2015), whose the graphical model is depicted in Fig. 2.7. In VRNNs, the data generating process of observation $\mathbf{x}_k$ depends not only on the deterministic hidden state $\mathbf{h}_k$ (as in RNNs), but also on a stochastic latent variable $\mathbf{z}_k$. $\mathbf{z}_k$ on its own depends on $\mathbf{h}_k$.

The joint distribution is factorised as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}) \tag{2.58}$$

$$= \prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{h}_k)p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k) \tag{2.59}$$

The recurrence in VRNN is defined as follows:

$$\mathbf{h}_k = f_{\boldsymbol{\theta}}(\mathbf{h}_{k-1}, \mathbf{x}_{k-1}, \mathbf{z}_{k-1}). \tag{2.60}$$

with $f_{\boldsymbol{\theta}}$ is parameterised by an LSTM or a GRU. In comparison to the original recurrence of RNNs, $\mathbf{h}_k$ depends not only on the previous state $\mathbf{h}_{k-1}$ and the previous observation $\mathbf{x}_{k-1}$, but also on a stochastic variable $\mathbf{z}_{k-1}$.

An elegant way to explain VRNN is to think of $\mathbf{h}_k$ as a component that capture the patterns, the structures in data, while the the stochastic component $\mathbf{z}_k$ model the variations and uncertainties around it. For example, to model the trajectory of a vessel, $\mathbf{h}_k$ encodes the common maritime route that vessels follow, while $\mathbf{z}_k$ encode some deviations

the vessel has to make to offset the effect of the wind, the sea current, *etc.*

We can use similar architecture of the transition model and the emission model in DSSMs for VRNN. The only modification needed is the input of the transition model is now the deterministic hidden state $\mathbf{h}_k$ of the RNN and the input of the emission model is now a concatenation of $\mathbf{z}_k$ and $\mathbf{h}_k$.

In the original paper of VRNN (J. Chung et al. 2015), the inference model is a filter, factorised as:

$$q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = \prod_{k=0}^{T} q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{h}_k, \mathbf{x}_k). \tag{2.61}$$

At each timestep, $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{h}_k, \mathbf{x}_k)$ can be modelled as a Gaussian with the mean and the covariance matrix parameterised by a deep neural network:

$$q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{h}_k, \mathbf{x}_k) = \mathcal{N}(\boldsymbol{\mu}_k^{inf}, \boldsymbol{\Sigma}_k^{inf}), \tag{2.62}$$

$$\boldsymbol{\mu}_k^{inf}, \boldsymbol{\Sigma}_k^{inf} = NN_{\boldsymbol{\phi}}^{inf}(\mathbf{h_k}, \mathbf{x_k}). \tag{2.63}$$

Follow the same procedure in Eqs. (2.64), the ELBO in VRNN is computed as:

$$\mathcal{L}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})}\left[\sum_{k=0}^{T} \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k, \mathbf{z}_k) - \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{x}_k, \mathbf{h}_k)||p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k)\right]\right)\right]. \tag{2.64}$$

Stochastic recurrent neural networks (SRNNs)

In VRNNs, there is no direct transition between two consecutive stochastic latent states $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$. All the stochasticity has to pass through the deterministic state $\mathbf{h}_k$, which may create a bottleneck in the stochastic flow. The **stochastic recurrent neural networks** (SRNNs) (Fraccaro et al. 2016) suggest adding a direct link between $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$ to allow the networks to temporal correlation in the space of $\mathbf{z}_k$. Furthermore, SRNNs separate the deterministic process in the RNN from the stochasticity by removing the link between $\mathbf{z}_{k-1}$ and $\mathbf{h}_k^f$. The graphical model of a SRNN is depicted in Fig. 2.8. We can see that SRNNs are a combination of RNNs and SSMs: a RNN run along the sequence to capture the long-term dependencies, and a stochastic process to model the transition of the uncertainty through time.

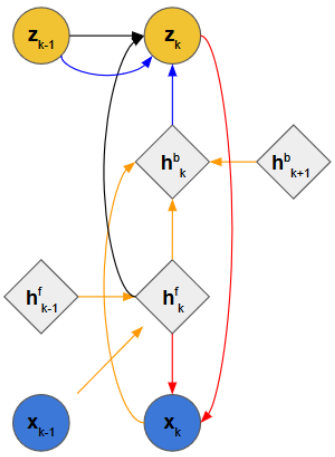


Figure 2.8 – Graphical model of a SRNN.

The joint distribution of SRNNs is computed as:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}, \mathbf{x}_{0:T}) = \prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{h}_k^f) p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{h}_k^f). \tag{2.65}$$

The architectures of the emission model $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{h}_k^f)$ and the transition model $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{h}_k^f)$ are similar to those in DSSMs and VRNNs, with the modification is the inputs.

SRNNs define the inference model as a smoother, and the variational distribution can be factorised as:

$$q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{h}_k^b). \tag{2.66}$$

The architecture of $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{h}_k^b)$ is similar to the inference network in VRNNs.

Although SRNNs may have better modelling capacity than VRNNs, VRNNs have been used more commonly because they required less computational resources and can perform the inference online (there is always a lag needed in the smoothing step in SRNNs).

2.4.3 Handling irregularly sampled data

DSSMs and SVAEs can naturally deal with noise in data. For example, in DSSMs, by construction, noise in data is taken into account by the emission distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$, and is separated from the transition process $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})$. This helps the model learn the true hidden dynamics of the system that generates the observations (Chapter 4). However, because DSSMs and SVAEs all use at least one RNN, they need regularly sampled input

data. To create regularly sampled data from irregularly sampled data, there are two classes of solutions:

— *Interpolation*: use an external interpolation method, such as optimal interpolation (OI), linear interpolation, *etc.* to interpolate missing values. The interpolation is calculated just once (in the preprocessing step). This approach is more suitable for DSSMs, because DSSMs use the RNN only in the inference model, hence the effect of the interpolation errors will be less important. See Chapter 4 for an example.

— *Imputation*: if the optimisation method is an iterative method (*e.g.* EM or gradient-based), at iteration (i), we can used the current parameters $\{\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\phi}^{(i-1)}\}$ to run the generative model to generate the missing values, then used these values at the inputs to run the whole model and update $\{\boldsymbol{\theta}^{(i)}, \boldsymbol{\phi}^{(i)}\}$. We still need an interpolation method to initiate the inputs at iteration (0), however the effects of the interpolation will gradually faded after a few iterations. Because SVAEs models use RNNs in both the generating model and the inference model, this approach is more suitable than the interpolation approach. See Chapter 6 for an example.

For both approaches, the interpolated/imputed values should be used as inputs for the model only, they should not be taken into account when calculating the objective function.

2.5 Summary and discussion

In this chapter we have introduced the theory part of this thesis. We started with the basic form of LVMs, then led to their sequential versions: SSMs and RNNs. In general, to build an LVM, we have to first define the latent variable and the joint distribution based on prior knowledge of the considered problem, then design the corresponding generative model and a suitable inference model such that the objective function (usually the ELBO) is tractable. SSMs are structured stochastic models with some nice factorisation and independence properties, however, the amount of problems that classical SSMs can cover is quite limited because they rely on analytic solutions. RNNs are highly non-linear models that can capture long-term dependencies in data, however, because their hidden states are deterministic, RNNs can not model stochasticity. To remove the linearity bottleneck of SSMs, DSSMs parameterise three distributions (the transition distribution, the emission distribution and the inference distribution) by neural networks. On the other hand, SVAEs helps RNNs capture variation and uncertainty in data by adding stochastic components to the networks. In a layman's term, DSSMs integrate deep neural networks into classic SSMs

Table 2.1 – Comparison between a DSSM and an SVAE (SRNN).

	DSSM	**SRNN**
Base model	SSM	RNN
Transition distribution	$p_{\boldsymbol{\theta}}(\mathbf{z}_k\|\mathbf{z}_{k-1})$	$p_{\boldsymbol{\theta}}(\mathbf{z}_k\|\mathbf{z}_{k-1}, \mathbf{h}_k^f)$
Emission distribution	$p_{\boldsymbol{\theta}}(\mathbf{x}_k\|\mathbf{z}_k)$	$p_{\boldsymbol{\theta}}(\mathbf{x}_k\|\mathbf{z}_k, \mathbf{h}_k^f)$
Inference distribution	$q_{\boldsymbol{\phi}}(\mathbf{z}_k\|\mathbf{z}_{k-1}, \mathbf{h}_k^f, \mathbf{h}_k^b)$	$q_{\boldsymbol{\phi}}(\mathbf{z}_k\|\mathbf{z}_{k-1}, \mathbf{h}_k^b)$
Recurrence	$\mathbf{h}_k^f = f_{\boldsymbol{\phi}}^f(\mathbf{h}_{k-1}^f, \mathbf{x}_{k-1})$ $\mathbf{h}_k^b = f_{\boldsymbol{\phi}}^b(\mathbf{h}_{k+1}^b, \mathbf{x}_k)$	$\mathbf{h}_k^f = f_{\boldsymbol{\phi}}^f(\mathbf{h}_{k-1}^f, \mathbf{x}_{k-1})$ $\mathbf{h}_k^b = f_{\boldsymbol{\phi}}^b(\mathbf{h}_{k+1}^b, \mathbf{x}_k)$

and SVAEs modify RNNs to mimic SSMs. Table 2.1 shows the comparison between a DSSM and an SVAE (here is an SRNN). The main difference between DSSMs and SVAEs is the way we model the transition of the latent variable. In DSSMs, there is an autonomous process of the latent variable, *i.e.* they do not depend on the observations. In SVAEs, the next latent state depends not only the current state but also the current observation. Because of this difference in the dependency, the factorisation in the two classes of model are different. Depending on the application, one approach may perform better than the other. For example, to retrieve a physical process given a series of noisy observations, DSSMs are more suitable than SVAEs because true hidden process are autonomous, it follows physic laws and does not interfered by the errors in the measurement. To predict the position of a vessel, although the hidden states can model the moving patterns, the actual trajectory may be effected by environmental context, such as a strong wind or dense traffic, models that take into account the current position to predict be next position (SVAEs) may be a better choice.

In theory, neural networks could automatically learn very complex tasks, however, in practice they rarely do. Domain expertise, *i.e.* the understanding of the data and the process that we are modelling is very important. It helps pose the right hypotheses, and design the right network architecture. In the following parts of this thesis, we will present some models that we have built based on the philosophy of DSSMs and SVAEs for specific applications. In Chapter 4, we present a DSSM for learning dynamical systems from noisy and partial observations. In Chapter 6 we introduce an SVAE specifically designed for maritime traffic surveillance using AIS data. We also propose a novel methodology for anomaly detection using SVAEs, applied to AIS trajectories in Chapter 7 and to audio surveillance in Appendix A.

Part II

Variational Deep Learning for Dynamical System Identification

Everything that happens once can never happen again. But everything that happens twice will surely happen a third time.

Paulo Coelho

Chapter 3

Introduction to Dynamical Systems and Differential Equations

The first application of this thesis is the usage of VDL, and specifically DSSMs, for the identification of dynamical systems. We start by presenting some basic concepts of dynamical systems, from the notion of dynamics, differential equations, to some numerical methods for differential equations. We also give a brief introduction to learning dynamical system and current state-of-the-art methods for this topic. This chapter does not aim to introduce dynamical systems theory, but rather aims to provide a background to the content presented in Chapter 4. Readers who are interested in dynamical systems theory are referred to (Arrowsmith et al. 1990; Brin et al. 2002; Hirsch et al. 2012).

3.1 Dynamical systems and differential equations

Dynamical systems theory is the study of the evolution, which involves describing the processes in motion, predicting the states and understanding the limitations of systems (Arrowsmith et al. 1990; Brin et al. 2002; Hirsch et al. 2012). It is the core of many disciplines. For example, in geosciences, it provides the basis for the simulation of climate dynamics, short-term and medium-range weather forecast, short-term prediction of ocean and atmosphere dynamics, *etc.* In aerodynamics or in fluid dynamics, it is crucial for the design of aircrafts and control systems, for the optimisation of energy consumption, *etc.*

In medicine, it is used for disease modelling, or for understanding the mechanism of the brain, *etc.*

Mathematically, a continuous dynamical system can be described by a set of differential equations. Depending on the type and the stochastic nature of those equations, we may classify them as **ordinary differential equations** (ODEs), **stochastic differential equations** (SDEs) and **partial differential equations** (PDEs). In this thesis, we focus only on the first two classes.

Ordinary differential equations (ODEs): an ODE dynamical system is governed by an equation:

$$\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} = f\Big(\mathbf{z}_t, t, \mathbf{u}_t\Big) \tag{3.1}$$

where $\mathbf{z}_t \in \mathbb{R}^{d_z}$ is the state of the system, $f : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_z}$ is a *deterministic* function, called the *dynamical model*, t denotes time and $\mathbf{u}_t \in \mathbb{R}^{d_u}$ is the control input. In this thesis, we focus only on autonomous systems without control input, *i.e.* systems governed by the following equation:

$$\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} = f\Big(\mathbf{z}_t\Big). \tag{3.2}$$

Stochastic differential equations (SDEs): for a dynamical system governed by Eq. (3.2), the future states are deterministic and depend solely on the current state. Given $\mathbf{z}_t$, we can calculate the exact value of $\mathbf{z}_{t+\Delta t}$ at any time $t + \Delta t$ in the future. In many applications, however, the trajectories of the system do not in fact behave as predicted. It is reasonable to modify ODE to include the possibility of random effects, which gives us SDEs. An (Itô form) SDE can be written as follows:

$$\mathrm{d}\mathbf{z}_t = f\Big(\mathbf{z}_t\Big)\mathrm{d}t + g(\mathbf{z}_t)\mathrm{d}\mathbf{W}_t \tag{3.3}$$

where $f : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_z}$ is a *deterministic* function, called the *drift*, $g : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_z \mathrm{x} d_w}$ is another *deterministic* function, called the *diffusion* and $\mathrm{d}\mathbf{W}_t$ is a Wiener process.

Glossary of some terms in dynamical systems theory: we introduce here a brief and simple description of some basic terms in dynamical systems theory. This list contains only terms that are actively used in this thesis. The full glossary and the rigorous definition of those terms can be found in textbooks in on dymanical systems theory (Arrowsmith

et al. 1990; Brin et al. 2002; Zaslavsky et al. 2005; Hirsch et al. 2012; Lichtenberg et al. 2013)

— **Chaos**: a dynamical system is chaotic if it is highly sensitive to initial conditions, *i.e.* a small change in the initial condition will results in a significant different in future states.

— **Phase space**: The phase space is an d_z-dimensional abstract space in which all possible states of a system are represented.

— **Attractor**: an attractor of a dynamical system is a closed subset of the phase space to which the system evolves after a long enough time.

— **Lyapunov exponent**: in a chaotic system, two trajectories starting from two close initial conditions may diverge exponentially. The Lyapunov exponent is a measure of mean velocity of exponential divergence of two initially close trajectories. A d_z-dimensional dynamical system has d_z Lyapunov exponents. If a system is chaotic, its largest Lyapunov exponent is positive.

— **Lyapunov time**: the Lyapunov time of a system is a characteristic timescale, defined as the inverse of its largest Lyapunov exponent.

3.2 Examples of dynamical systems

Some of the most famous examples of dynamical systems are the Lorenz systems. In this section, we present three of them: the Lorenz-63 (L63) system (Lorenz 1963), the Lorenz-96 (L96) system (Lorenz 1996) and a stochastic Lorenz-63 (L63s) system (Chapron et al. 2018).

3.2.1 The Lorenz-63 system

The Lorenz-63 system (L63), named after Edward Lorenz, is a 3-dimensional dynamical system that model the atmospheric convection (Lorenz 1963). The L63 is governed by the following ODE:

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{z}_{t,1}}{\mathrm{d}t} &= \sigma\left(\mathbf{z}_{t,2} - \mathbf{z}_{t,1}\right) \\
\frac{\mathrm{d}\mathbf{z}_{t,2}}{\mathrm{d}t} &= \left(\rho - \mathbf{z}_{t,3}\right)\mathbf{z}_{t,1} - \mathbf{z}_{t,2} \\
\frac{\mathrm{d}\mathbf{z}_{t,3}}{\mathrm{d}t} &= \mathbf{z}_{t,1}\mathbf{z}_{t,2} - \beta\mathbf{z}_{t,3}
\end{aligned} \tag{3.4}$$

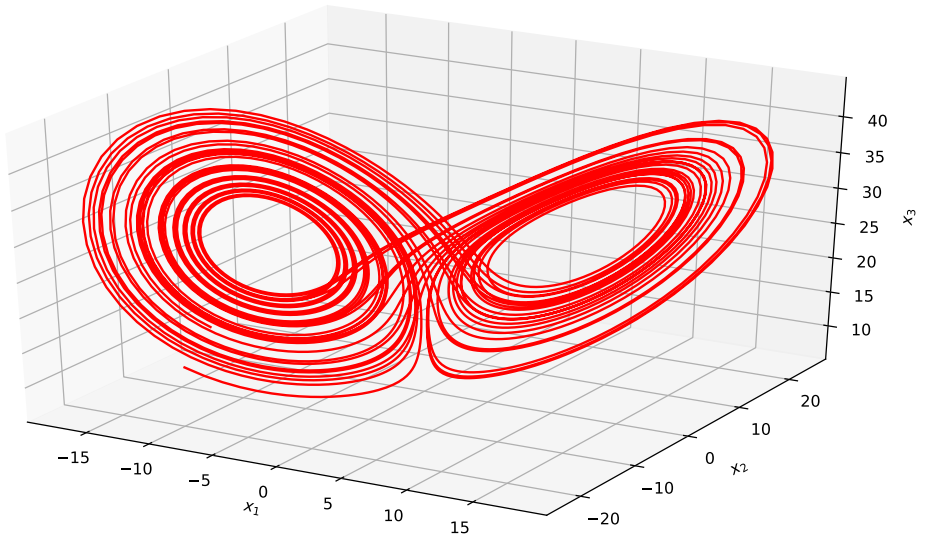


Figure 3.1 – The attractor of the Lorenz–63 system when $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$.

When $\sigma = 11$, $\rho = 28$ and $\beta = 8/3$, this system has a chaotic behavior, with the Lorenz attractor shown in Fig. 3.1.

Some characteristics of the L63 with the above set of parameters are as follows:

— The system is chaotic, a minor change in the initial condition will lead to a completely different trajectory in long term.

— The attractor of the L63 has a "butterfly form", the particles frequently change side of the attractor. The density of the particles in two sides of the attractor is also similar.

3.2.2 The Lorenz-96 system

The Lorenz-96 system (L96) (Lorenz 1996) is a periodic 40-dimensional dynamical system governed by the following ODEs:

For $i = 1, ..N_z$:

$$\frac{\mathrm{d}\mathbf{z}_{t,i}}{\mathrm{d}t} = (\mathbf{z}_{t,i+1} - \mathbf{z}_{t,i-2})\mathbf{z}_{t,i-1} - \mathbf{z}_{t,i} + F \tag{3.5}$$

with $N_z = 40$, $\mathbf{z}_{t,-1} = \mathbf{z}_{t,N_z-1}$, $\mathbf{z}_{t,0} = \mathbf{z}_{t,N_z}$ and $\mathbf{z}_{t,N_z+1} = \mathbf{z}_{t,1}$.

We choose $F = 8$ to have chaotic system. The Hovmøller plot of one trajectory of this system is shown in Fig. 3.2

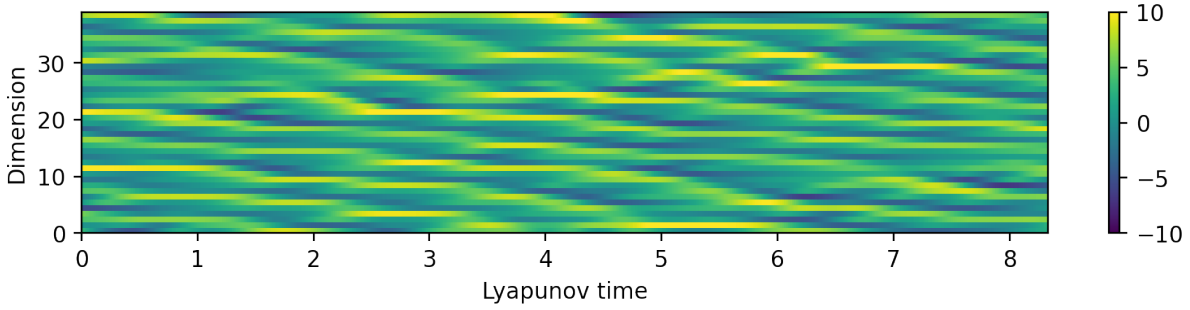


Figure 3.2 – Hovmøller plot of one trajectory of a L96.

3.2.3 The stochastic Lorenz-63 system

The stochastic Lorenz-63 system (L63s) is presented in (Chapron et al. 2018). It is a modified version of the L63 to model situations where the large-scale characteristics of a physical event may be changed because of accumulated perturbations in fine scales. The governing equations of the L63s are as follows:

$$\begin{aligned}
\mathrm{d}\mathbf{z}_{t,1} &= \left(\sigma\left(\mathbf{z}_{t,2}-\mathbf{z}_{t,1}\right)-\frac{4}{2\gamma}\mathbf{z}_{t,1}\right)\mathrm{d}t \\
\mathrm{d}\mathbf{z}_{t,2} &= \left(\left(\rho-\mathbf{z}_{t,3}\right)\mathbf{z}_{t,1}-\mathbf{z}_{t,2}-\frac{4}{2\gamma}\mathbf{z}_{t,2}\right)\mathrm{d}t+\frac{\rho-\mathbf{z}_{t,3}}{\gamma^{0.5}}\mathrm{d}B_t \\
\mathrm{d}\mathbf{z}_{t,3} &= \left(\mathbf{z}_{t,1}\mathbf{z}_{t,2}-\beta\mathbf{z}_{t,3}-\frac{8}{2\gamma}\mathbf{z}_{t,3}\right)\mathrm{d}t+\frac{\mathbf{z}_{t,2}}{\gamma^{0.5}}\mathrm{d}B_t
\end{aligned} \tag{3.6}$$

with B_t a Brownian motion.

In the L63s, the noise level is controlled by γ. For example, with $\sigma = 11$, $\rho = 28$ and $\beta = 8/3$ and $\gamma = 2$, the particles are easily trapped in one side of the attractor, as shown in Fig. 3.3.

3.3 Numerical methods for differential equations

Given the equations and the current state $\mathbf{z}_t$, solving a differential system means to calculate value of the state $\mathbf{z}_{t+\Delta t}$ of the system at time $t+\Delta t$. For ODE systems (Eq. (3.2)), this means to calculate the integral:

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \int_t^{t+\Delta t} f\Big(\mathbf{z}_u\Big)\mathrm{d}u. \tag{3.7}$$

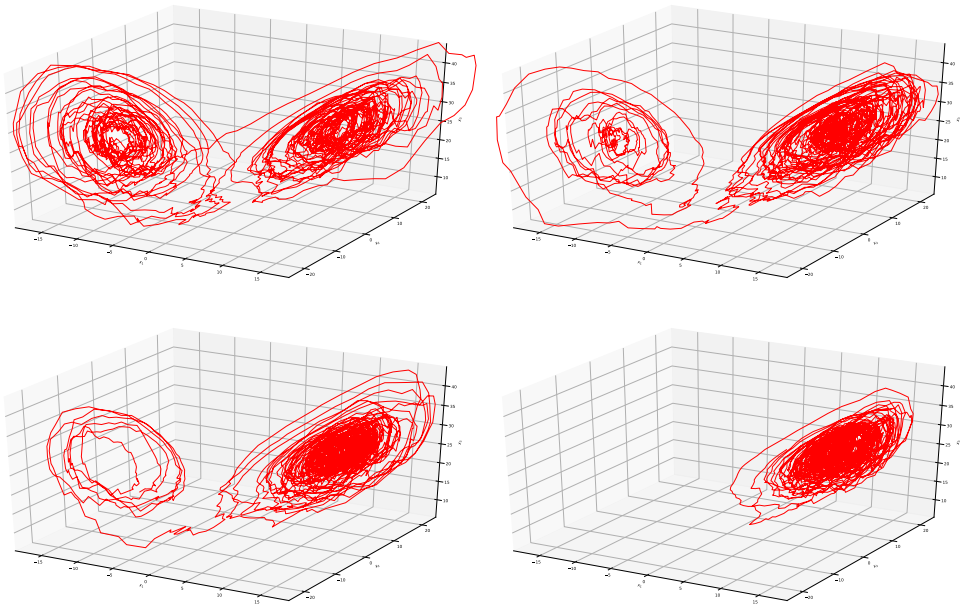


Figure 3.3 – Several attractors generated by the a L63s. These attractors are generated from the same initial condition. Because the system is stochastic, each runtime we obtain a different trajectory.

For SDE systems (Eq. (3.3)), this means to calculate the integral:

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \int_t^{t+\Delta t} f\big(\mathbf{z}_u\big)\mathrm{d}u + \int_t^{t+\Delta t} g\big(\mathbf{z}_u\big)\mathrm{d}\mathbf{W}_u. \tag{3.8}$$

Apart from some simple cases, such as linear dynamical systems, those differential equations do not have analytical solutions. We have to approximate the solutions using numerical methods, called "numerical integration". In this section, we present three examples of those methods: the **Euler method** and the **Runge-Kutta 4 method** for ODEs, and the **Euler-Maruyama method** for SDEs.

3.3.1 The Euler method

The simplest method for the numerical integration of ODEs is the **Euler method** (also called the **forward Euler method**). This method uses the standard forward derivative approximation as follows:

$$\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} \approx \frac{1}{\Delta t}\left[\mathbf{z}_{t+\Delta t} - \mathbf{z}_t\right]. \tag{3.9}$$

Applying this to Eq. (3.2) we obtain:

$$\frac{1}{\Delta t}\left[\mathbf{z}_{t+\Delta t} - \mathbf{z}_t\right] \approx f(\mathbf{z}_t). \tag{3.10}$$

For a small time step Δt, the Euler method takes this to be exact (Shampine 2018):

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t f(\mathbf{z}_t). \tag{3.11}$$

The Euler method is explicit and simple. It is a common choice for fast calculations of ODEs. However, applying the Taylor expansion of $\mathbf{z}_{t+\Delta t}$ around t:

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t \frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} + \frac{(\Delta t)^2}{2}\frac{\mathrm{d}^2\mathbf{z}_t}{\mathrm{d}t^2} + \mathcal{O}(\Delta t)^3), \tag{3.12}$$

we can see that the Euler method ignores the quadratic and higher-order terms. For more accurate solutions, we can extend it to produce Runge-Kutta methods.

3.3.2 The Runge-Kutta 4 method

The **Runge-Kutta 4 method** (RK4) is the most common method of the Runge-Kutta family. It is defined as follows (Shampine 2018):

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \frac{1}{6}\Delta t(k_1 + 2k_2 + 2k_4 + k_4), \tag{3.13}$$

$$k_1 = f(\mathbf{z}_t), \tag{3.14}$$

$$k_2 = f(\mathbf{z}_t + \Delta t\frac{k_1}{2}), \tag{3.15}$$

$$k_3 = f(\mathbf{z}_t + \Delta t\frac{k_2}{2}), \tag{3.16}$$

$$k_4 = f(\mathbf{z}_t + \Delta t k_3). \tag{3.17}$$

Applying the Taylor expansion, we can see that the truncation error of the RK4 is on the order of $\mathcal{O}((\Delta t)^5)$. The RK4 is one of the the most common numerical integration schemes for ODEs.

3.3.3 The Euler–Maruyama method

The **Euler–Maruyama method** is a method that uses Itô calculus for the numerical integration of SDEs. This method is defined as follows (Kloeden et al. 2013):

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t f(\mathbf{z}_t) + g(\mathbf{z}_t)\Delta \mathbf{W}_t. \tag{3.18}$$

where where $\Delta \mathbf{W}_t$ is a Gaussian random variable with expected value zero and variance Δt.

The Euler–Maruyama method can be roughly considered as the Euler method version for SDEs.

3.4 Learning dynamical systems

Classically, the derivations of governing equations are based on some prior knowledge of the intrinsic nature of the system (Lorenz 1963; Hilborn 2000; Sprott et al. 2003; Hirsch et al. 2012). The derived models can then be combined with the measurements (observations) to reduce errors, both in the model and in the measurements. This approach forms the discipline of Data Assimilation (DA). However, in many cases, the underlying dynamics of the system are unknown or only partially known, while a large number of observations are available. This has motivated the development of learning-based approaches (Brunton and Kutz 2019), where one aims at identifying the governing equations of a process from time series of measurements. Recently, the ever increasing availability of data thanks to developments in sensor technologies, together with advances in Machine learning (ML), has made this issue a hot topic.

Mathematically, learning a dynamical system means to retrieve the dynamics of this system from some observation datasets, that is to say identifying the governing equations f (and g for SDEs), given a series of observations $\mathbf{x}_{t_k}$:

$$\mathbf{x}_{t_k} = \Phi_{t_k}\Big(\mathcal{H}\Big(\mathbf{z}_{t_k}\Big) + \boldsymbol{\varepsilon}_{t_k}\Big) \tag{3.19}$$

where $\mathcal{H} : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_x}$ is the observation operator, usually known (d_x is the dimension of $\mathbf{x}_{t_k}$), $\boldsymbol{\varepsilon}_t \in \mathbb{R}^{d_x}$ is a zero-mean additive noise and $\{t_k\}_k$ refers to the time sampling, typically regular such that $t_k = t_0 + k.\delta$ with respect to a fine time resolution δ and a starting time t_0. The masking operator Φ_{t_k} accounts for the fact that observation $\mathbf{x}_{t_k}$ may not be

available at all time steps t_k ($\Phi_{t_k,j} = 0$ if the j^{th} variable of $\mathbf{x}_{t_k}$ is missing). For the sake of simplicity, from now on in this thesis, we use the notation $\mathbf{x}_k$ for $\mathbf{x}_{t_k}$ and $\mathbf{x}_{k+n}$ for $\mathbf{x}_{t_{k+n.\delta}}$.

One of the pioneering contributions in learning dynamical systems is the Sparse Identification of Nonlinear Dynamics (SINDy) presented in (Brunton, Proctor, et al. 2016). SINDy assumes that the governing equations of a dynamical model consist of only a few basic functions such as polynomial functions, trigonometric functions, exponential functions, etc. The method creates a dictionary of such candidates and uses sparse regression to retrieve the corresponding coefficients of each basic function. Under ideal conditions, SINDy can find the exact solution of Eq. (3.2). The key advantage of SINDy is the interpretability of its solutions, *i.e.* the parametric form of the governing equations can be recovered. Another advantage is that the solution comprises only a few terms, which improves the generalisation properties of the learnt models. However, SINDy requires the time derivative $\frac{d\mathbf{x}_t}{dt}$ to be observed. $\frac{d\mathbf{x}_t}{dt}$ might be highly corrupted by noise for noisy and partial observation datasets, which may strongly effect the performance of SINDy. Besides, it requires some prior knowledge about the considered system to create a suitable library of the basic functions.

Analog methods (Nagarajan et al. 2015; McDermott et al. 2016; Z. Zhao et al. 2016), including the Analog Data Assimilation (AnDA) presented in (Lguensat et al. 2017), propose a non-parametric approach for data assimilation. AnDA implicitly learns Eqs. (4.1) and (4.2) by remembering every seen pairs $\{state, successor\} = \{\mathbf{x}_k, \mathbf{x}_{k+1}\}$ and storing them in a catalog. To predict the evolution of a new query point $\mathbf{x}_k$, AnDA looks for k similar states in the catalog, the prediction is then a weighted combination of the corresponding successors of these states. The performance of this method heavily depends on the quality of the catalog. If the catalog contains enough data and the data are clean, AnDA provides a good and straightforward solution for data assimilation. However, since AnDA relies on a k-Nearest Neighbor (k-NN) approach, it may be strongly affected by noisy data especially when considering high-dimensional systems.

A number of neural-network-based (NN-based) methods have been introduced recently. These methods leverage deep neural networks as universal function approximators. They vary from direct applications of standard NN architectures, such as LSTMs in (Yeo et al. 2019), ResNets in (Qin et al. 2018), etc. to some more sophisticated designs, dedicated to dynamical systems and often referred to as Neural ODE schemes (R. T. Q. Chen et al. 2018; Fablet, Ouala, et al. 2018; Raissi, Perdikaris, and George Em Karniadakis 2018; Rubanova et al. 2019). The reservoir computing, whose idea is derived from Recurrent

Neural Networks (RNNs), used in (Pathak, Lu, et al. 2017) and (Pathak, Hunt, et al. 2018) can also be regarded as a NN-based model. As illustrated in (R. T. Q. Chen et al. 2018; Fablet, Ouala, et al. 2018; Raissi, Perdikaris, and George Em Karniadakis 2018; Rubanova et al. 2019), through the combination of a parametrisation for differential operator f and some predefined integration schemes (*e.g.*, explicit Runge-Kutta 4 scheme (RK4) in (Fablet, Ouala, et al. 2018), black-box ODE solvers in (R. T. Q. Chen et al. 2018; Rubanova et al. 2019)), the Neural ODE schemes provides significantly better forecasting performance than that of standard NN models, especially when dealing with chaotic dynamics. Powered by deep learning, these methods can successfully capture the dynamics of the system under ideal conditions (noise-free and regularly sampled with high frequency). However, they have the following limitations: i) the network requires fully-sampled data [1] and ii) when dealing with noisy observations, no regularisation techniques have been proved effective in preventing overfitting in dynamical system identification.

In the next chapter, we will present a variational deep learning framework that deals with those problems.

1. Latent ODE ((Rubanova et al. 2019) can apply to data sampled partially in time, however, data may be sampled partially in space also.

It has been said that something as small as the flutter of a butterfly's wing can ultimately cause a typhoon halfway around the world.

Chaos Theory

Chapter 4

DAODEN

[1] [2] In the Chapter 2, we introduced two classes of variational deep learning models for sequential data: DSSMs and SVAEs. In DSSMs, there is an independent process in the latent space, the observations are considered as noisy and potentially partial samples of this process. This modeling is very suitable for describing physical events, since the true hidden processes follow physical laws, and are independent from the measurements, which may be affected by errors in the sensors and/or by interference in the medium. In this chapter, we present a DSSM framework specifically designed for the data-driven recovery of the unknown governing equations of dynamical systems. This topic has recently received an increasing interest. However, the identification of the governing equations remains challenging when dealing with noisy and partial observations. Here, we address this challenge by proposing a DSSM framework. Within the proposed framework, we jointly learn an inference model to reconstruct the true states of the system the governing laws of these states from series of noisy and partial data. In doing so, this framework bridges classical data assimilation and state-of-the-art machine learning techniques. We also

1. This chapter is a modified version of paper (Duong Nguyen, Ouala, et al. 2020b)

2. This work was supported by public funds (Ministère de l'Education Nationale, de l'Enseignement Supérieur et de la Recherche, FEDER, Région Bretagne, Conseil Général du Finistère, Brest Métropole), Institut Mines Télécom, received in the framework of the VIGISAT program managed by "Groupement Bretagne Télédétection" (BreTel), Labex Cominlabs (grant SEACS), CNES (grant OSTST-MANATEE), Microsoft (AI EU Ocean awards) and by ANR (French Agence Nationale de la Recherche) under grants Melody and OceaniX. It benefited from HPC and GPU resources from Azure (Microsoft EU Ocean awards) and from GENCI-IDRIS (Grant 2020-101030).

We are thankful to Noura Dridi for helpful discussions and support.

demonstrate that it generalises state-of-the-art methods. Importantly, both the inference model and the governing model embed stochastic components to account for stochastic variabilities, model errors and reconstruction uncertainties. Various experiments on chaotic and stochastic dynamical systems support the relevance of our scheme w.r.t. state-of-the-art approaches.

4.1 Introduction

Recently, numerous methods have successfully captured the hidden dynamics of systems under ideal conditions, *i.e.* noise-free and high sampling frequency using a variety of data-driven schemes, including analog methods (Lguensat et al. 2017), sparse regression schemes (Brunton, Proctor, et al. 2016), reservoir computing (Pathak, Lu, et al. 2017; Pathak, Hunt, et al. 2018) and neural approaches (Fablet, Ouala, et al. 2018; Raissi, Perdikaris, and George Em Karniadakis 2018; Qin et al. 2018; Ayed et al. 2019; Vlachas et al. 2018). However, real life data are often corrupted by noise and/or observed partially, as for instance encountered in the monitoring of ocean and atmosphere dynamics from satellite-derived observation data (Pierce 2001; Johnson et al. 2005; Isern-Fontanet et al. 2014). In such situations, the above-mentioned approaches are most likely to fail to uncover unknown governing equations.

To address this challenge, we need to jointly solve the reconstruction of the hidden dynamics and the identification of governing equations. This may be stated within a data assimilation framework (Bocquet et al. 2019) using state-of-the-art assimilation schemes such as the Ensemble Kalman Smoother (EnKS) (Evensen and Leeuwen 2000). Deep learning approaches are also particularly appealing to benefit from their flexibility and computational efficiency. Here, we investigate a variational deep learning framework. More precisely, we state the considered issue as a variational inference problem with an unknown transition distribution associated with the underlying dynamical model. The proposed method generalises learning-based schemes such as (Fablet, Ouala, et al. 2018; Duong Nguyen, Ouala, et al. 2019; Duong Nguyen, Ouala, et al. 2020a; Brajard et al. 2019; Bocquet et al. 2020) and also explicitly relates to data assimilation formulations. Importantly, it can account for errors and uncertainties both within the dynamical prior and the inference model. Overall, our key contributions are:

— a general deep learning framework which bridges classical data assimilation and modern machine learning techniques for the identification of dynamical systems

from noisy and partial observations. This framework use variational inference and random-n-step-ahead forecasting, which can be considered as two complementary regularisation strategies to improve the learning of governing equations of dynamical systems;

— insights on the reason why many existing methods for learning dynamical systems do not work when the available data are not perfect, *i.e.* noisy and/or partial;

— numerical experiments with chaotic systems which support the relevance of the proposed framework to improve the learning of governing equations from noisy and partial observation datasets compared to state-of-the-art schemes;

— numerical experiments which demonstrate that our method can also capture the characteristics of dynamical systems where the stochastic factors are significant.

The chapter is organised as follows. In Section 4.2, we formulate the problem of learning non-linear dynamical systems. We review state-of-the-art methods and analyze their drawbacks in Section 4.3. Section 4.4 presents the details of the proposed framework, followed by the experiments and results in Section 4.5. We close the paper with conclusions and perspectives for future work in Section 4.6.

4.2 Problem formulation

Let us consider a dynamical system, described by an Ordinary Differential Equation (ODE) in Eq. (3.2). From Eq. (3.2) and (3.19), we derive a state space formulation:

$$\mathbf{z}_{k+n} = \mathcal{F}^n\Big(\mathbf{z}_k\Big) + \boldsymbol{\omega}_{k+n} \tag{4.1}$$

$$\mathbf{x}_k = \Phi_k\Big(\mathcal{H}\Big(\mathbf{z}_k\Big) + \boldsymbol{\varepsilon}_k\Big) \tag{4.2}$$

where $\mathbf{z}_{k+n}$ results from an integration of operator f from state $\mathbf{z}_k$:

$$\mathcal{F}^n\Big(\mathbf{z}_k\Big) = \mathbf{z}_k + \int_k^{k+n} f\Big(\mathbf{z}_u\Big)\mathrm{d}\mathbf{u}. \tag{4.3}$$

n is the number of timesteps ahead that $\mathcal{F}^n$ forecast, given the current state $\mathbf{z}_k$. $\mathcal{F}^n$ is hence called the n-step-ahead model. $\boldsymbol{\omega}_k \in \mathbb{R}^{d_z}$ is a zero-mean noise process, called the *model error*. $\boldsymbol{\omega}_k$ may come from the neglected physics, numerical approximations and/or modelling errors. $\boldsymbol{\varepsilon}_k$ is the *observation error* (or *measurement error*). Note that f is continuous, the discretisation only happens because we want to calculate the integral $\mathcal{F}^n$

over the interval $[t_k, t_{k+n}]$. Furthermore, as detailed later, the parametrisation of $\mathcal{F}^n$ may explicitly depend on n or not. In this thesis, to simplify the notation, $k+n$ includes both $k+1$ (*i.e.* $n=1$) and $k+n$ (*i.e.* $n \neq 1$). If we specify $k+1$ and $k+n$ in one sentence, it means $n \neq 1$ in this context.

Within this general formulation, the identification of governing equations f amounts to maximising the log likelihood $\ln p(\mathbf{x}_{0:T})$.

4.3 Related work

The identification of dynamical systems has attracted attention for several decades and closely relates to data assimilation (DA) for applications to geoscience. Proposed approaches typically consider some parametric model for operator $\mathcal{F}^n$, for example, a linear function in (Ghahramani and G. E. Hinton 1996) or Radial Basis Functions (RBFs) in (Ghahramani and Roweis 1999). While data assimilation mostly focuses on the reconstruction of the hidden dynamics given some observation series, a number of studies have investigated the situation where the dynamical prior is unknown. They typically learn the unknown parameters of the model using an iterative Expectation-Maximisation (EM) procedure. The E-step involves a DA scheme (*e.g.* the Kalman filter (Welch et al. 1995), the Extended Kalman filter (Hoshiya et al. 1984), the Ensemble Kalman filter (Evensen 2009), the particle filter (Doucet et al. 2009), etc.) to reconstruct the true states $\{\mathbf{z}_k\}$ from observations $\{\mathbf{x}_k\}$, whereas the M-step retrieves the parameters of $\mathcal{F}$ best describing the reconstructed state dynamics. Such methods address the fact that the observations may not be ideal. They may also account for model errors and uncertainties ($\boldsymbol{\omega}_k$ in Eq. (4.1)). However, since they rely on analytic solutions, the choices of the candidates for $\mathcal{F}$ are generally limited. For a comprehensive introduction as well as an analysis of the limitations of those methods, the reader is referred to (Voss et al. 2004).

Recently, the domain of dynamical systems identification has received a new wave of contributions. Advances in machine learning open new means for learning the unknown dynamics. As introduced in Section 3.4, numerous learning-based methods have been able to identify the governing laws of the systems, given noiseless and fully-observed data. However, they may not apply or fail when the observations are noisy and/or partial. Their learning step is stated as the minimisation of a short-term prediction error of the observed variables:

$$loss = \sum g(||\mathbf{x}_{k+n}^{pred} - \mathbf{x}_{k+n}||_2) \tag{4.4}$$

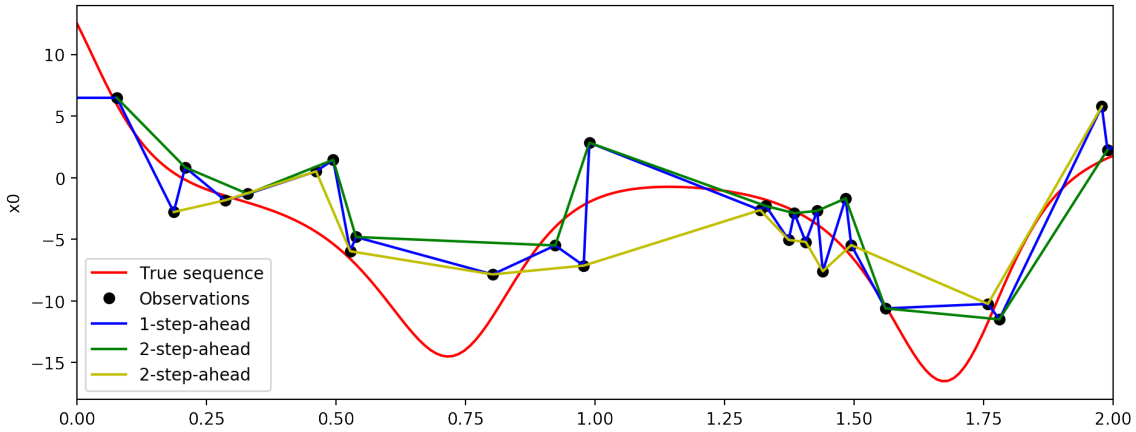


Figure 4.1 – Problems of learning dynamical systems from imperfect data. This figure plots the first component of the Lorenz-63 system, when the observation operator is the identity matrix. The observation is noisy, partial. If the learning algorithm is applied directly on the observations (black dots), which are noisy and partially sampled, and a linear interpolation is used to create regularly-sampled data, the dynamics seen by the network are the blue curve (for 1-step-ahead forecasting models) or the green and the yellow curves (for 2-step-ahead forecasting models, these two curves correspond to two possible starting points of the sequence) instead of the true dynamics (the red curve).

where $\mathbf{x}_{k+n}^{pred} = \mathcal{F}^n(\mathbf{x}_k)$ is the predicted observation at $k+n$ given the current observation $\mathbf{x}_k$, $||.||_2$ denotes the L2 norm, g is a function of $||\mathbf{x}_{k+n}^{pred} - \mathbf{x}_{k+n}||_2$. As shown in Fig. 4.1, with this family of cost functions, the model tends to overfit the observations (the blue curve or the green and yellow curves), instead of learning the true dynamics of the system (the red curve). Another reason why these methods fail is because they violate the Markovian property of the system. Note that the process of the true states $\mathbf{z}_{0:T}$ of the system is Markovian (*i.e.*, given $\mathbf{z}_t$, $\mathbf{z}_{k+1}$ does not depend on $\mathbf{z}_{0:k-1}$). However, when the data are damaged by noise, the process of the observations $\mathbf{x}_{0:T}$ is not Markovian. Given $\mathbf{x}_t$, we still need the information contained in $\mathbf{x}_{0:k-1}$ to predict $\mathbf{x}_{k+1}$. For this reason, applying Markovian architectures like SINDy (Brunton, Proctor, et al. 2016), AnDA (Lguensat et al. 2017), DenseNet (Raissi, Perdikaris, and George Em Karniadakis 2018) , BiNN (Fablet, Ouala, et al. 2018), etc. directly on the observations $\mathbf{x}_{0:T}$ would not succeed. Models with memory like LSTMs (Yeo et al. 2019) may capture the non-Markovian dynamics in the training phase, however, in the simulation phase, they still need the memory, which implies that the learnt dynamics do not have the Markovian property of the true dynamics of the system.

In this chapter, we consider a variational deep framework which derives from a varia-

tional inference for state space formulation (Eqs. (4.1), (4.2)). This framework accounts for uncertainty components in the dynamical prior as well as in the observation model. Similarly to DA schemes (Bocquet et al. 2019; Duong Nguyen, Ouala, et al. 2019; Duong Nguyen, Ouala, et al. 2020a; Brajard et al. 2019; Bocquet et al. 2020), it jointly solves for the reconstruction of the hidden dynamics and the identification of the governing equations. Importantly, it benefits from the computational efficiency and the modelling flexibility of deep learning frameworks for the specification of the dynamical prior and the inference model as well as for the use of a stochastic regularisation during the training phase through a randomised n-step-ahead prediction loss. The proposed framework generalises our recent works presented in (Duong Nguyen, Ouala, et al. 2019) and (Duong Nguyen, Ouala, et al. 2020a) and similar works, which have been developed concurrently in (Bocquet et al. 2019; Brajard et al. 2019) and (Bocquet et al. 2020). As detailed in the next section, (Bocquet et al. 2019; Brajard et al. 2019) and (Bocquet et al. 2020) may be regarded as specific instances of the proposed framework with some specific settings, such as constant model error covariance matrix (we relax this hypothesis), Ensemble Kalman Smoothers for the inference scheme (we exploit both strategies: Ensemble Kalman Smoothers and NN-based schemes), EM for the optimisation (we exploit both EM and gradient-based techniques).

4.4 Proposed framework

In this section, we detail the proposed variational deep learning framework for the data-driven identification of the governing equations of dynamical systems from noisy and partial observations. We first present the proposed framework based on variational inference. We then introduce the considered NN-based parametrisations for the dynamical prior and the inference model, along with the implemented learning scheme. We further discuss how the proposed framework relates to previous work.

4.4.1 Variational inference for learning dynamical systems

Given a series of observations $\mathbf{x}_{0:T} = \{\mathbf{x}_0, .., \mathbf{x}_k\}$, instead of looking for a model $\mathcal{F}^n$ that minimises a loss function in a family of short-term prediction error functions as in Eq. (4.4), we aim to learn operator $\mathcal{F}^n$ such that it maximises the log likelihood $\ln p(\mathbf{x}_{0:T})$ of the observed data. We assume that $\mathbf{x}_{0:T}$ are noisy and/or partial observations of the true states $\mathbf{z}_{0:T}$, like in Eqs. (4.1) and (4.2). We can derive the log-likelihood $\ln p(\mathbf{x}_{0:T})$ from

the marginalisation of $\ln p(\mathbf{x}_{0:T}, \mathbf{z}_{0:T})$ over $\mathbf{z}_{0:T}$:

$$\ln p(\mathbf{x}_{0:T}) = \ln \int p(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) d\mathbf{z}_{0:T} \tag{4.5}$$

As discussed in Chapter 2, with the exception of some simple cases, the integral in Eq. (4.5) is intractable because the posterior distribution $p(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ is intractable. To address this issue, Variational Inference (VI) approximates $p(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ by a distribution q which maximises the Evidence Lower BOund (ELBO) $\mathcal{L}(\mathbf{x}_{0:T}, p, q)$ as in Eq. (2.56) in Chapter 2.

Based on the state space formulation in Eqs. (4.1) and (4.2), we consider the following parametrisation for the joint likelihood $p(\mathbf{x}_{0:T}, \mathbf{z}_{0:T})$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}) p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) \tag{4.6}$$

$$p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_0) \prod_{k=1}^{n-1} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}) \prod_{k=n}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-n}) \tag{4.7}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) = \prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) \tag{4.8}$$

$$q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T}) = \prod_{k=0}^{T} q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T}) \tag{4.9}$$

with $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are the sets of parameters of p and q, respectively; $\mathbf{z}_k^f$ is the state forecast by $\mathcal{F}^1$ given $\mathbf{z}_{k-1}$ for $k = 1..n-1$, and by $\mathcal{F}^n$ given $\mathbf{z}_{k-n}$. $k = n..T$.

The distributions in Eqs. (4.7), (4.8) and (4.9), are respectively the classic distributions of a state space formulation: 1) the n-step ahead transition (or dynamic, or prior) distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$ (including $n = 1$); 2) the emission (or observation) distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$; and 3) the inference (or posterior) distribution $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})$. To better constrain the time consistency of the learnt dynamics, the considered dynamical prior embeds a n-step-ahead forecasting model. Given an initialisation $\mathbf{z}_0$, it first applies a one-step-ahead prior to propagate the initial state to the first n time steps. The application of the n-step-ahead prior then follows to derive the joint distribution over the entire time range $\{0, .., T\}$. This n-step-ahead prior is regarded as a mean to further regularise the time consistency of the learnt dynamical model.

By explicitly separating the transition, the inference and the generative processes, the proposed framework is fully consistent with the underlying state space formulation and the associated Markovian properties. Especially, the prior $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$ will embed a Markovian architecture; by contrast, the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})$ shall capture the non-

Markovian characteristics of the observed data. Given the learnt model, the generation of simulated dynamics only relies on the dynamical prior $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+1}|\mathbf{z}_k)$ to simulate state sequences, which conform to the Markovian property. Overall, for a given observation dataset, the learning stage comes to maximising Eq. (2.56) w.r.t. both $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, which comprise all the parameters of the inference and generative models, *i.e.* the parameters of $\mathcal{F}$, $\mathcal{H}$, $\boldsymbol{\omega}_k$ and $\boldsymbol{\epsilon}_k$.

So far we have introduced the general form of the proposed variational inference framework for learning dynamical systems from noisy and potentially partial observations. In the following sub-sections, we will analyse some specific instances of the proposed framework and provide insights into the associated implicit hypotheses behind methods in the literature.

4.4.2 Parametrisation of the generatvie model $p_{\boldsymbol{\theta}}$

Model $p_{\boldsymbol{\theta}}$ involves two sets of parameters: (i) $\boldsymbol{\theta}_z$—the parameters of the transition distributions $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$ and (ii) $\boldsymbol{\theta}_x$—the parameters of the emission distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$.

Regarding the later, similarly to (Brajard et al. 2019) and (Bocquet et al. 2020), we assume the observation noise to be a white noise process with a multivariate covariance $\mathbf{R}$ such that $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ is a conditional multivariate normal distribution:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) = \mathcal{N}(\mathcal{H}(\mathbf{z}_k), \mathbf{R}) \tag{4.10}$$

We may consider different experimental settings: with known or unknown observation operator $\mathcal{H}$.

Regarding the n-step-ahead dynamical prior $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k)$ (including $n = 1$), we consider a conditional Gaussian distribution where the mean path is driven by the the governing equation $\mathcal{F}^n$: $\mathbf{z}_{k+n} = \mathcal{F}^n(\mathbf{z}_k)$ and and the dispersion is represented by a covariance matrix $\mathbf{Q}_k$ (usually called the *model error covariance* in DA) (Evensen 2009). Any state-of-the-art architecture for learning dynamical systems can be used to model $\mathcal{F}$. Here, we consider NN-based methods associated with explicit integration schemes. To account for second-order polynomial model, as proposed in (Fablet, Ouala, et al. 2018), we consider a bilinear architecture to model f in Eq. (3.2) and an NN implementation of the RK4 integration scheme to derive the flow operator Eq. (4.1). Regarding the covariance dynamics, the covariance matrix $\mathbf{Q}_{k+n}$ is approximated by a diagonal matrix $diag\mathbf{d}_k^f$, with $\mathbf{d}_k^f$ the output

of a MultiLayer Perceptron (MLP):

$$\mathbf{d}_k^f = MLP^{var\_dyn}(\mathbf{z}_{k-n}, \mathcal{F}^n(\mathbf{z}_{k-n})) \tag{4.11}$$

4.4.3 Parametrisation of the inference model q_ϕ

There is no restriction for the parametrisation of posterior q_ϕ. However, the parametrisation clearly affects the performance of the overall optimisation. Here, we investigate two strategies for q_ϕ: 1) an Ensemble Kalman Smoother (EnKS) (Evensen and Leeuwen 2000) and 2) an LSTM Variational Auto Encoder (LSTM-VAE).

The former is a classic DA scheme that is widely used in many domains in which dynamical systems play an important role, for example in geosciences (Khare et al. 2008). We use the implementation presented in (Evensen and Leeuwen 2000). The latter is a modern NN architecture, which has been proven effective for modelling stochastic sequential data (J. Chung et al. 2015) (Fraccaro et al. 2016). The backbone of the LSTM-VAE is a bidirectional LSTM which captures the long-term correlations in data. Specifically, we parameterise the inference scheme as follows: The forward LSTM is given by:

$$\mathbf{h}_k^f = lstm(\mathbf{h}_{k-1}^f, MLP^{enc}(\mathbf{x}_{k-1}^f)) \tag{4.12}$$

and the backward LSTM is given by:

$$\mathbf{h}_k^b = lstm(\mathbf{h}_{k+1}^b, \mathbf{h}_k^f, MLP^{enc}(\mathbf{x}_k^f)) \tag{4.13}$$

where $\mathbf{h}_k^f$, $\mathbf{h}_k^b$ are the hidden states of the forward and the backward LSTM, respectively; $lstm$ is the recurrence formula of LSTM (Hochreiter et al. 1997); MLP^{enc} is an encoder parameterised by an MLP. We parameterise the posterior q_ϕ by a conditional Gaussian distribution with mean $\boldsymbol{\mu}_k^q$ and a diagonal covariance matrix $diag(\mathbf{d}_k^q)$:

$$q_\phi(\mathbf{z}_k) = \mathcal{N}(\boldsymbol{\mu}_k^q, diag(\mathbf{d}_k^q)) \tag{4.14}$$

$$\boldsymbol{\mu}_k^q, \mathbf{d}_k^q = MLP^{dec}(\mathcal{F}^n(\mathbf{z}_{k-n}), \mathbf{h}_k^f, \mathbf{h}_k^b) \tag{4.15}$$

with MLP^{dec} is a decoder parameterised by an MLP. Note that in Eq. (4.15), q_ϕ depends on $\mathbf{z}_k^f = \mathcal{F}^n(\mathbf{z}_{k-n})$. This idea is inspired by DA, where $\mathcal{F}^n(\mathbf{z}_{k-n})$ is analogous to the forecasting step and $q_\phi(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})$ is analogous to the analysis step, which depends on

the forecasting step. The whole model, called Data-Assimilation-based ODE Network (DAODEN) is illustrated in Fig. 4.2.

To our knowledge, DAODEN is the first stochastic end-to-end RNN-based model introduced for the identification of dynamical systems from noisy and partial observations. In this respect, the model used in (Yeo et al. 2019) is a purely deterministic RNN-based network.

However, similar architectures have been used in Natural Language Processing (NLP) such as the Variational Recurrent Neural Network (VRNN) presented in (J. Chung et al. 2015), the Sequential Recurrent Neural Network (SRNN) presented in (Fraccaro et al. 2016). Fig. 4.2 shows how DAODEN differs from those architectures. The main difference is that the transition $\mathbf{z}_k \rightarrow \mathbf{z}_{k+1}$ is independent of observation $\mathbf{x}_k$ (*i.e.* the dynamic is autonomous). Besides, the emission $\mathbf{z}_k \rightarrow \mathbf{x}_k$ is also independent of the historical state $\mathbf{z}_0, .., \mathbf{z}_{k-1}$. These differences relate to domain-related priors. In dynamical systems' theory and associated application domains such as geoscience, the underlying dynamics follow physical principles. Therefore, they are autonomous and are not affected by the measurements (the observations). As a consequence, $\mathbf{z}_{k+1}$ does not depend on $\mathbf{x}_{1:k-1}$ conditionally to $\mathbf{z}_k$. At a given time k, observation $\mathbf{x}_k$ is a measurement of state $\mathbf{z}_k$ of the system, this measurement does not depend on any other state $\mathbf{z}_{k' \neq k}$, *i.e.* given $\mathbf{z}_k$, $\mathbf{x}_k$ and $\mathbf{z}_{k'}$ are independent with any $k' \neq k$. For this reason, architectures used in NLP like VRNN, SRNN do not apply for dynamical system identification.

4.4.4 Objective functions

Following a variational Bayesian setting, the learning phase comes to minimising a loss given the negative of ELBO:

$$loss_{ELBO} = -\mathcal{L}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}}) \tag{4.16}$$

Instead of solving Eq. (4.16), one can solve its Maximum A Posteriori (MAP) solution by restricting $q_{\boldsymbol{\phi}}$ to Dirac distributions:

$$\mathcal{L}_{MAP} = \sum_{k=0}^{T} \ln p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k^*) + \ln p_{\boldsymbol{\theta}}(\mathbf{z}_0^*) + \sum_{k=1}^{n-1} \ln p_{\boldsymbol{\theta}}(\mathbf{z}_k^*|\mathbf{z}_{k-1}^*) + \sum_{k=n}^{T} \ln p_{\boldsymbol{\theta}}(\mathbf{z}_k^*|\mathbf{z}_{k-n}^*) \tag{4.17}$$

with $\mathbf{z}_k^* = \mathbb{E}\left[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{z}_k^f, \mathbf{x}_{0:T})\right]$ if $q_{\boldsymbol{\phi}}$ is parameterised by an EnKS and $\mathbf{z}_k^* = q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T}) =$

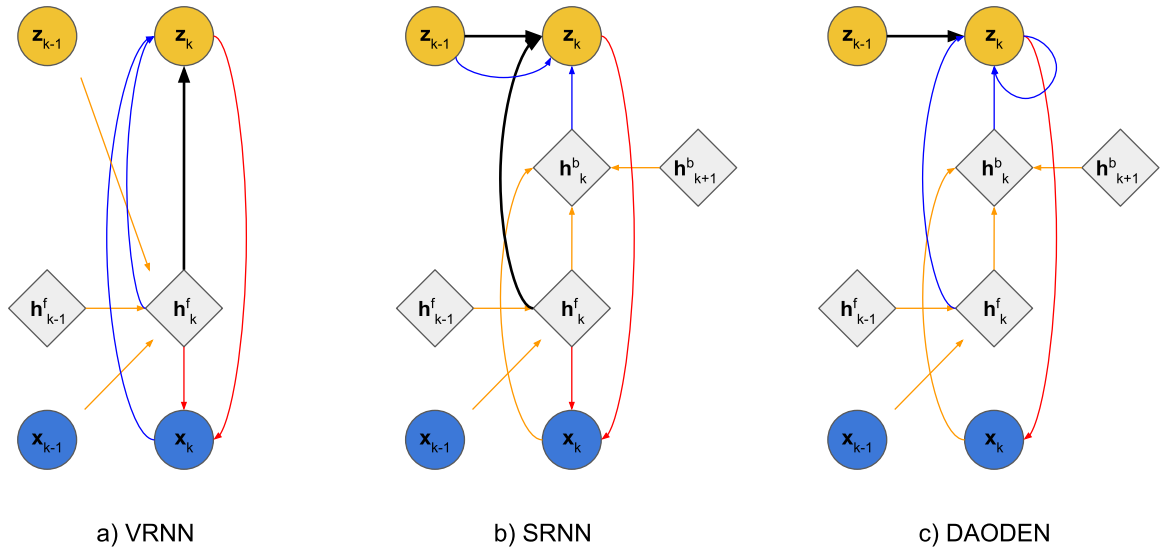


Figure 4.2 – Architecture of VRNN, SRNN and DAODEN when $n = 1$. We denote as $\mathbf{x}_k$ the observations, $\mathbf{z}_k$ the system's states, $\mathbf{h}_k^f$ the latent states of the forward LSTM and $\mathbf{h}_k^b$ the latent states of the backward LSTM. The black, red, blue and orange arrows denote respectively the transition of the system's states, the emission of the observations, the inference of the system's states and recurrence of the LSTMs, respectively. In VRNN (a) and SRNN (b), the dynamic $\mathbf{z}_k \rightarrow \mathbf{z}_{k+1}$ is not independent of the observation $\mathbf{x}_k$. The generation of the observation is also entangled with the recurrence of the LSTMs.

$\delta(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})$ if $q_{\boldsymbol{\phi}}$ is parameterised by a neural network. If we remove the covariance part in Eq. (4.15), the LSTM-VAE becomes an LSTM Auto Encoder (LSTM-AE):

$$\mathbf{z}_k^* = \boldsymbol{\mu}_k^q = dec(\mathcal{F}^n(\mathbf{z}_{k-1}), \mathbf{h}_k^f, \mathbf{h}_k^b) \tag{4.18}$$

The MAP loss function, which relates to the weak-constraint 4D-Var in DA (Courtier et al. 1994), is given by:

$$loss_{MAP} = -\mathcal{L}_{MAP}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}}) \tag{4.19}$$

This is the objective function used in (Bocquet et al. 2019; Brajard et al. 2019) and (Bocquet et al. 2020), with the assumption that $\mathbf{Q}_k$ is time invariant, *i.e.* $\mathbf{Q}_k = \mathbf{Q}$.

One may further assume that the covariance matrices of the transition distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_k^*|\mathbf{z}_{k-n}^*)$ and the covariance matrices of the observation distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k^*)$ are diagonal and constant, both in time and in space, Eq. (4.17) then becomes [3]:

$$\mathcal{L}_{determ} = -\lambda \sum_{k=0}^{T} ||\boldsymbol{\phi}_k(\mathcal{H}(\mathbf{z}_k^*)) - \mathbf{x}_k||_2^2 - \sum_{k=0}^{n-1} ||\mathcal{F}^1(\mathbf{z}_{k-1}^*) - \mathbf{z}_k^*||_2^2 - \sum_{k=0}^{T} ||\mathcal{F}^n(\mathbf{z}_{k-n}^*) - \mathbf{z}_k^*||_2^2 \tag{4.20}$$

The associated loss function is given by:

$$loss_{determ} = -\mathcal{L}_{determ}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q) \tag{4.21}$$

which is the objective function used in (Duong Nguyen, Ouala, et al. 2019) and (Duong Nguyen, Ouala, et al. 2020a). We may note that if $\mathbf{x}_k = \mathbf{z}_k$, (4.21) becomes the short-term prediction error widely used in the literature (Pathak, Lu, et al. 2017; Fablet, Ouala, et al. 2018; Qin et al. 2018; Pathak, Hunt, et al. 2018). In other words, (Pathak, Lu, et al. 2017; Fablet, Ouala, et al. 2018; Qin et al. 2018; Pathak, Hunt, et al. 2018) implicitly suppose that the observations are ideal.

4.4.5 Optimisation strategies

To learn parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ (*i.e.* the parameters of the generative and the inference models), there are two optimisation strategies: 1) alternatively optimise $\boldsymbol{\theta}$ then $\boldsymbol{\phi}$ (Expectation-Maximisation-like or EM-like) to minimise the loss function or 2) jointly

3. The derivation of (4.20) can be found in our previous paper (Duong Nguyen, Ouala, et al. 2019).

optimise the loss function over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

For models whose posterior $q_{\boldsymbol{\phi}}$ is implemented by an EnKS, since EnKS uses analytic formulas and the NN-based parametrisation of $p_{\boldsymbol{\theta}}$ is usually optimised by Gradient Descent (GD) techniques, we consider an alternated EM procedure as the optimisation strategy for the whole model. In the E-step, the EnKS computes the posterior $q_{\boldsymbol{\phi}}$, represented by an ensemble of states $\mathbf{z}_k^{(i)}$. Given this ensemble of states, the M-step minimises the loss function over $\boldsymbol{\theta}$ using a stochastic gradient descent algorithm.

For DAODEN settings, we can fully benefit from the resulting end-to-end architecture, as both the generative model $p_{\boldsymbol{\theta}}$ and the posterior $q_{\boldsymbol{\phi}}$ are parameterised by neural networks, to jointly optimise all model parameters using a stochastic gradient descent technique. The gradient descent technique may be regarded as a particular case of EM where the M-step takes only one single gradient step. For NN-based models, gradient descent strategies usually work better than EM (I. Goodfellow, Yoshua Bengio, et al. 2016).

4.4.6 Random-n-step-ahead training

Within the considered framework, we noted experimentally that the model may overfit the data, when the number of the forecasting steps is fixed. For example, if the observation operator $\mathcal{H}$ is an identity matrix, a possible overfitting situation is when the inference scheme also becomes an identify operator: $\mathbb{E}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_{0:T})\right] \rightarrow \mathbf{x}_k$. In such situations, the dynamics seen by the dynamical sub-modules would be the noisy dynamics.

To deal with these overfitting issues, we further exploit the flexibility of the proposed n-step-ahead dynamical prior during the training phase. For each mini-batch iteration in the training phase, we draw a random value of n between 1 and a predefined n-step-ahead_max. We then apply a gradient descent step with the sampled value of n. The resulting randomised training procedure is detailed in Alg. 1. This randomised procedure is regarded as a regularisation procedure to fit a time-consistent dynamical operator $\mathcal{F}^n$. We noted in previous works that neural ODE schemes may not distinguish well the dynamical operator from the integration scheme (Ouala, Pascual, et al. 2019). Here, through the randomisation of parameter n, we constrain the end-to-end architecture by applying it for different prediction horizons, which in turn constrains the identification of the dynamical model f. Asymptotically, the proposed procedure would be similar to a weighted sum of loss (4.16) computed for different values of n, which have been proposed for the data-driven identification of governing equations in the noise-free case (Rubanova et al. 2019).

Algorithm 1: Random-n-step-ahead training.

Result: The set of parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ of the learnt model.
Inputs: $\mathbf{x}_{0:T}$, $\mathbf{z}_0$, $\mathbf{R}$, the initial values of $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$, n-step-ahead_max, n_iteration_max;
iter = 0;
while *iter* < *n_iteration_max* **do**
  t = 0;
  n-step-ahead = randint(1,n-step-ahead_max);
  while $t < k - n$ **do**
    if $t <$ *n-step-ahead* -2 **then**
      $n = 1$;
    else
      $n = n$-step-ahead;
    $\mathbf{z}^f_{k+n} = \mathcal{F}^n(\mathbf{z}_k)$;
    $\mathbf{d}^f_{k+n} = MLP^{var\_dyn}(\mathbf{z}_k, \mathcal{F}^n(\mathbf{z}_k))$;
    $p_{\boldsymbol{\theta}}(\mathbf{z}_{k+n}|\mathbf{z}_k) = \mathcal{N}(\mathbf{z}^f_{k+n}, \mathbf{d}^f_{k+n})$;
    Calculate $q_{\boldsymbol{\phi}}(\mathbf{z}_{k+n}|\mathbf{z}^f_{k+n}, \mathbf{x}_{0:T})$;
    Sample $\mathbf{z}_{k+n} \sim q_{\boldsymbol{\phi}}(\mathbf{z}_{k+n}|\mathbf{z}^f_{k+n}, \mathbf{x}_{0:T})$;
    $p_{\boldsymbol{\theta}}(\mathbf{x}_{k+n}|\mathbf{z}_{k+n}) = \mathcal{N}(\mathcal{H}(\mathbf{z}_{k+n}), \mathbf{R})$;
  Calculate *loss*;
  Optimise *loss* w.r.t. $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$;

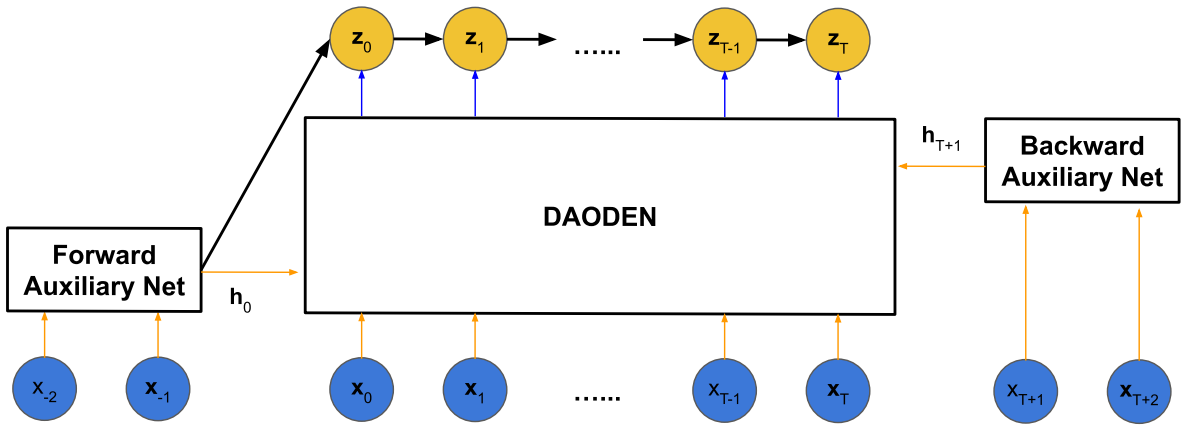


Figure 4.3 – Initialisation by optimisation. An auxiliary network is added for the initialisation of $\mathbf{x}_0$ and $\mathbf{h}_0$.

4.4.7 Initialisation by optimisation

In this section, we present the initialisation technique used for in the experiments in this chapter. Although this technique is not compulsory, it improve the stability of the training.

To calculate the state of the system at any given time k, we need both the true dynamics and the precise initial condition $\mathbf{z}_0$. If we use DAODEN, we also have to initialise $\mathbf{h}_0^f$ and $\mathbf{h}_{T+1}^b$. The common approach is "wash out" (Jaeger 2002), *i.e.* to initialise $\mathbf{h}_0^f$ and $\mathbf{h}_{T+1}^b$ to zeros or random values and run the LSTMs until the effect of the initial values disappears. However, this initialisation technique may not be suitable for learning dynamical systems, because during the wash out period, the network is not stable, especially when using an explicit integration scheme (here is the RK4). These instabilities may make the training fail. The value of the objective function also varies highly during this period, leading to an unreliable outcome of the final loss.

Sharing a similar idea with (Mohajerin et al. 2019) and (Rubanova et al. 2019), we use a different initialisation strategy. We add two auxiliary networks, a Forward Auxiliary Net to provide $\mathbf{h}_0$ and $\mathbf{z}_0$, and an Backward Auxiliary Net to provide $\mathbf{h}_{k+1}$ for the main model. Each auxiliary network is an LSTM. We use one segment at the beginning of the sequence and one segment at the end of the sequence as the inputs of these networks.

4.5 Experiments and results

In this section, we report numerical experiments to evaluate the proposed framework. We include a comparison with respect to state-of-the-art methods. Beyond the application to deterministic dynamics as considered in previous work (Duong Nguyen, Ouala, et al. 2019; Duong Nguyen, Ouala, et al. 2020a; Bocquet et al. 2019; Bocquet et al. 2020; Brajard et al. 2019; Pathak, Hunt, et al. 2018; Pathak, Lu, et al. 2017; Qin et al. 2018), we also investigate an application to stochastic dynamics and a reduced-order modelling, where the observation operator $\mathcal{H}$ is unknown. As case-study models, we focus on Lorenz-63 and Lorenz-96 dynamics, which provides a benchmarking basis w.r.t. previous work (Brunton and Kutz 2019; Lguensat et al. 2017; Fablet, Ouala, et al. 2018; Champion et al. 2019).

4.5.1 Benchmarking dynamical models

We report numerical experiments for three chaotic dynamical systems: a Lorenz 63 (L63) systems (Lorenz 1963), a Lorenz 96 (L96) system (Lorenz 1996) and a stochastic Lorenz 63 (L63s) system (Chapron et al. 2018).

Note that these models are chaotic, *i.e.* they are are highly sensitive to initial conditions such that a small difference in a state may lead to significant changes in future. Because of this chaotic nature, applying directly standard deep neural network architectures would not be successful.

We chose the L63 as a benchmarking system because of its famous butterfly attractor. The system involves 3-dimensional states, making it easy to visualise for a qualitative interpretation. Experiments on the L96 provides a means to evaluate how the proposed schemes can scale up to higher-dimensional systems. The last system—the L63s, is considered to show the benefit of stochastic architectures over deterministic ones.

For each system, we generated 200 sequences of length of 150 using 200 different initial conditions $\mathbf{z}_0$ with time step $\delta = 0.01$, $\delta = 0.05$ and $\delta = 0.01$ for the L63, L96 and L63s, respectively [4]. In total, the training set of each system comprises 30000 points in total. Those training sets are relatively small in comparison with those in (Champion et al. 2019) (512000 points) and (Brajard et al. 2019) (40000 points). Another setting when we generated only one long sequence of length of 4000 from one initial condition $\mathbf{z}_0$, then split it into smaller segments of length of 150 also gave similar results [5] (not reported in this

4. This is the setting used in (Champion et al. 2019)
5. This is the setting used in (Lguensat et al. 2017; Pathak, Hunt, et al. 2018), (Brajard et al. 2019)

chapter).

For the test sets, we generated 50 sequences of length of 150 using 50 different initial conditions $\mathbf{z}_0$ which are not observed in the training set. Let us recall that the true hidden states $\mathbf{z}_{0:T}$ of sequences are never used during the training phase, however, they are used in the test phase to give a quantitative evaluation.

As in (Duong Nguyen, Ouala, et al. 2019; Duong Nguyen, Ouala, et al. 2020a; Bocquet et al. 2019; Brajard et al. 2019; Bocquet et al. 2020), we first consider an experimental setting where $\mathcal{H}$ is an identity operator, and $\boldsymbol{\varepsilon}_k$ a zero-mean Gaussian white noise. We tested several signal-to-noise ratio values $r = \frac{std_\epsilon}{std_{\mathbf{z}}}$. Then we tested the proposed framework on a setting where $\mathcal{H}$ is unknown, as in (Champion et al. 2019).

4.5.2 Baseline schemes

In the reported experiments, we considered different state-of-the-art schemes for benchmarking purposes, namely the Analog Data Assimilation (AnDA) (Lguensat et al. 2017), the Sparse Identification of Nonlinear Dynamics (SINDy) (Brunton, Proctor, et al. 2016), the Bilinear Neural Network (BiNN) (Fablet, Ouala, et al. 2018), and the Latent ODE (Rubanova et al. 2019), the latter being among the state-of-the-art schemes in the deep learning literature. As explained earlier in this chapter, regardless of the network architecture, as long as the objective function does not take into account the fact that the observations are noisy and potentially partial, the method would not work. BiNN and Latent ODE embed the true solution of the L63 and the L96. Under ideal conditions, they should work as well as other NN-based ODE model, such as those in (Raissi, Perdikaris, and George Em Karniadakis 2018; Qin et al. 2018; Yeo et al. 2019), etc. The difference between BiNN and Latent ODE is BiNN uses an explicit integration scheme (the RK4), while Latent ODE uses a black-box ODE solver. Latent ODE also uses an additional network to infer the initial condition $\mathbf{z}_0$.

Since VRNN (J. Chung et al. 2015) and SRNN (Fraccaro et al. 2016) are not designed for dynamical system identification (no autonomous dynamics in the hidden space), we do not consider these architectures in this chapter.

Table 4.1 – Implementations of the proposed framework.

Model name	$p_\theta(\mathbf{z}_{k+n}\|\mathbf{z}_k)$	$q_\phi(\mathbf{z}_k\|\mathbf{z}_k^f, \mathbf{x}_{0:T})$	Objective function	Optimiser
BINN_EnKS	BiNN	EnKS	Eq. (4.21)	EM
DAODEN_determ	BiNN	LSTM-AE	Eq. (4.21)	GD
DAODEN_MAP	BiNN	LSTM-AE	Eq. (4.19)	GD
DAODEN_full	BiNN	LSTM-VAE	Eq. (4.16)	GD

4.5.3 Instances of the proposed framework

We synthesise in Table. 4.1 the different configurations of the proposed framework that we implemented in our numerical experiments. We may point out that BiNN_EnKS configuration is similar to (Bocquet et al. 2020). All configurations use a BiNN with a fourth-order Runge-Kutta scheme to parameterise $\mathcal{F}$. As presented above, other architectures can also be used to parameterise $\mathcal{F}^n$, we choose BiNN to highlight the performance of learning dynamical systems with and without inference schemes (by comparing the performance of BiNN and that of models following the proposed framework). The parameters of each model are presented in the Appendices. We provide the code that can reproduce the result in this chapter: https://github.com/CIA-Oceanix/DAODEN. Interested users are highly encouraged to try those models above on different dynamical systems or to replace the dynamical sub-module by different learning methods to see the improvement of its performance on noisy and partial observations.

In this chapter, unless specified otherwise the n-step-ahead_max parameter was set to 4 for DAODEN models and 1 for baseline models (1-step-ahead is the default setting in the original papers of those methods). As in (Bocquet et al. 2020), for BiNN_EnKS, we suppose that we know $\mathbf{R}$. However, for DAODEN, we do not need the exact value of $\mathbf{R}$, when using a fixed value of $\mathbf{R}$ that was from 1 to 2 times larger than the true value of $\mathbf{R}$, the results were similar.

4.5.4 Evaluation metrics

We evaluate both the short-term and long-term performance of the learnt models using the following metrics:

(Bocquet et al. 2020)

— The Root Mean Square Error (RMSE) of the short-term forecast at $t_n = t_0 + n.\delta$:

$$e_n = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(\mathbf{z}_k^{pred} - \mathbf{z}_k^{true})^2} \tag{4.22}$$

with $\mathbf{z}_k^{pred} \triangleq \mathcal{F}^k(\mathbf{z}_0)$ and $\mathbf{z}_0$ is the first state of each sequence in the test set.

— The reconstruction capacity given the observations, denoted as rec:

$$rec = \sqrt{\frac{1}{T}\sum_{k=0}^{T}(\mathbf{z}_k^{*} - \mathbf{z}_k^{true})^2} \tag{4.23}$$

with $\mathbf{z}_k^{*} = \mathbb{E}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})\right]$.

— The first time (in Lyapunov unit) when the RMSE reaches half of the standard deviation of the true system, denoted as $\pi_{0.5}$.

— The capacity to maintain the long-term topology of the system, evaluated via the first Lyapunov exponent λ_1 calculated in a forecasting sequence of length of 20000 time steps, using the method presented in (Wolf et al. 1985). The true λ_1 of the L63 is 0.91 and the true λ_1 of the L96 is 1.67.

For each metric, we compute the average of the results on 50 sequences in the test set.

As Lorenz dynamics may interpreted in terms of geophysical dynamics, we may also give some physical interpretation to the considered metrics. For example, in geosciences, for experiments on the L96 system with δ=0.05 (correspond to 6 hours in real-world time), e_4 would relate to the precision of a weather forecast model for the next day, $\pi_{0.5}$ indicates how long the forecast is still meaningful, λ_1 indicates whether a model can be used for long-term forecast such as the simulation of climate change and rec indicates the ability of a model to reconstruct the true states of a system when the observations are noisy and partial, such as reconstructing the sea surface condition from satellite images.

4.5.5 L63 case-study

In this section we report the results for the L63 case-study. We first assess the identification performance on noisy but complete observations (*i.e.* $\boldsymbol{\phi}_k$ is an identity matrix at all time steps) of the L63 system, then address cases where the observations are sampled partially, both in time and in space.

Table 4.2 shows the performance of the considered model on noisy L63 data. We compare the performance of the 4 proposed models with the baselines', w.r.t the short-term prediction error and the capacity to maintain the long-term topology. All the models based on the proposed framework outperform the baselines by a large margin. This asserts the ability of the proposed framework to deal with noisy observations. In Fig. 4.4 we show the first component of a L63 sequence in the test set reconstructed by the inference scheme of DAODEN_determ. q_ϕ is expected to infer a mapping that converts data from the corrupted observation space (black dots) to the true space of the dynamics (the red curve). In this space, data-driven methods can successfully learn the governing equations of the system. The reconstructed sequence is very close to the true sequence.

At first glance, we can see that no model is better than all the others in all 4 criteria. This is aligned with the findings in (Fablet, Drumetz, et al. 2020). BiNN_EnKS and DAODEN_full have very good forecasting score, however, the performance of BiNN_EnKS in reconstructing the true states is not as well as DAODEN models. The dynamics learnt by DAODEN models are also more synchronised to the true dynamics (indicated by $\pi_{0.5}$) than those learnt by BiNN_EnKS. This might suggest that NN-based models (here are LSTM-AE and LSTM-VAE) can be an alternative for classic inference schemes like EnKS, which are among the state-of-the-art methods in data assimilation (Lahoz et al. 2010).

In Fig. 4.5, we show the attractors generated by the learnt models. AnDA is more suitable for data assimilation than for forecasting. When the noise level is small (r=8.5% and r=16.7%), SINDy and BiNN can still capture the dynamics of the system. When the noise level is significant (r=33.3% and r=66.7%), the attactors generated by SINDy and BiNN are distorted, which indicates that the learnt models are not valid for long-term simulations. On the other hand, all the models of the proposed framework successfully reconstructed the butterfly topology of the attractor, even when the noise level is high.

In real life applications, we cannot always measure a process regularly with a high sampling frequency. Hence, we address here the problem of learning dynamical systems from not only noisy but also partial observations [6]. Specifically, we consider a case study where the noisy L63 data are sampled partially, both in time and in space, with a missing rate of 87.5% (see Fig. 4.6). For this configuration, baseline schemes do not apply. We report in Table. 4.3 and Fig. 4.7 the performance of the different configurations of the

6. The term "partial" in this context means the observations are not complete at every time step. Some components of the observations may be missing, in both spatial and temporal dimensions; however, all the components of states of the system are seen at least once. For the cases where some components of the systems are never observed, please refer to (Ayed et al. 2019; Ouala, Duong Nguyen, Drumetz, et al. 2020)

Table 4.2 – Performance of models trained on noisy L63 data. For each index, the best score is marked in **bold** and the second best score is marked in *italic*.

Model		r 8.5%	16.7%	33.3%	66.7%
AnDA	e_4	0.351±0.184	0.777±0.350	1.683±0.724	3.682±1.346
	rec	0.416±0.019	0.941±0.037	2.134±0.076	4.876±0.168
	$\pi_{0.5}$	0.820±0.480	0.380±0.172	0.249±0.174	0.104±0.116
	λ_1	26.517±7.665	27.146±42.927	76.267±28.150	127.047±0.881
SINDy	e_4	0.068±0.052	0.149±0.106	0.311±0.196	0.694±0.441
	$\pi_{0.5}$	0.490±0.261	0.165±0.085	0.077±0.049	0.034±0.034
	λ_1	0.898±0.008	0.840±0.035	0.840±0.035	nan±nan
BiNN	e_4	0.045±0.030	0.119±0.085	0.283±0.185	0.684±0.408
	$\pi_{0.5}$	3.608±1.364	2.053±0.666	0.975±0.488	0.308±0.125
	λ_1	0.900±0.011	0.868±0.010	0.122±0.208	-0.422±0.047
Latent-ODE	e_4	0.051±0.027	0.062±0.034	*0.065±0.042*	*0.213±0.084*
	$\pi_{0.5}$	2.504±1.332	2.336±1.472	2.852±1.352	2.118±1.129
	λ_1	0.892±0.018	0.877±0.018	0.885±0.015	0.675±0.027
BiNN_EnKS	e_4	**0.019±0.016**	**0.024±0.023**	**0.037±0.024**	0.276±0.160
	rec	0.323±0.024	0.431±0.042	0.598±0.093	1.531±0.332
	$\pi_{0.5}$	2.807±1.128	3.004±1.355	2.996±1.641	*2.081±1.214*
	λ_1	0.856±0.031	0.869±0.024	0.826±0.065	0.868±0.014
DAODEN_determ	e_4	0.049±0.031	0.056±0.034	0.077±0.048	0.268±0.201
	rec	0.216±0.125	0.269±0.110	**0.448±0.199**	*0.873±0.216*
	$\pi_{0.5}$	*3.519±1.282*	*3.488±1.327*	**3.470±1.562**	1.803±1.104
	λ_1	0.882±0.036	0.895±0.021	0.911±0.013	0.793±0.021
DAODEN_MAP	e_4	0.038±0.027	0.038±0.038	0.101±0.070	0.233±0.088
	rec	*0.209±0.096*	**0.234±0.065**	0.525±0.253	**0.817±0.330**
	$\pi_{0.5}$	3.271±1.270	3.219±1.260	2.993±1.413	**2.650±1.382**
	λ_1	0.860±0.047	0.876±0.029	0.916±0.012	0.920±0.008
DAODEN_full	e_4	*0.023±0.015*	*0.027±0.016*	0.072±0.045	**0.187±0.127**
	rec	**0.178±0.050**	*0.258±0.066*	*0.469±0.168*	1.003±0.380
	$\pi_{0.5}$	**3.533±1.139**	**3.496±1.215**	*3.426±1.512*	1.897±0.918
	λ_1	0.869±0.036	0.858±0.028	0.881±0.024	0.884±0.013

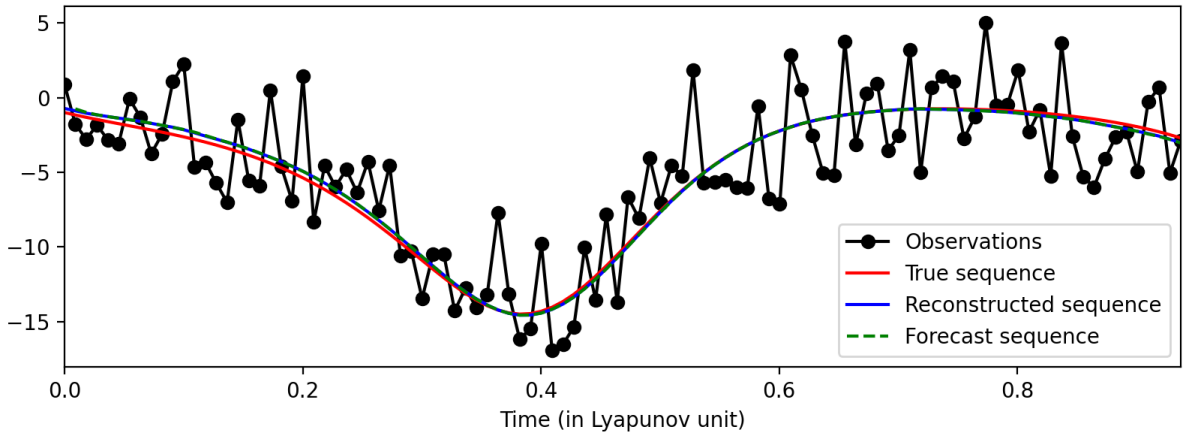


Figure 4.4 – An example of the the first dimension of the L63 system reconstructed by the inference module of DAODEN_determ, $r = 33\%$. Given the noisy observations (black dots), the inference module $q_\phi(\mathbf{z}_k|\mathbf{z}_k^f, \mathbf{x}_{0:T})$ reconstructs a clean sequence of the hidden state (blue curve), which is very close to the true unknown dynamic (red curve). Given this sequence, the transition network (BiNN) can successfully learn the governing laws of the system, as it can do under ideal conditions. The green dash shows the forecast $\mathbf{z}_{k+1}^* = \mathcal{F}^1(\mathbf{z}_k^*)$ given the mean $\mathbf{z}_k^*$ of q_ϕ.

proposed framework. If the noise level is not significantly high (r=33.3% or r=66.7%), all the models are able to capture the dynamical characteristics of the data. When the noise level is small, BiNN_EnKS tends to perform better than DAODEN. However, when the data are awash with noise, BiNN_EnKS does not work well anymore. On the other hand, DAODEN models, especially DAODEN_full work well in these cases. This may come from the capacity of LSTM architectures to capture long-term correlations in data.

4.5.6 L96 case-study

In this section we present experiments on a L96 system. The objective is to assess how the proposed framework applies in higher-dimensional spaces. We choose the deterministic and the full version of DAODEN as the candidate models. The results of models trained on noisy observations are shown in Table. 4.4. DAODEN models outperforms state-of-the-art methods both in terms of short-term prediction and long-term topology. In Fig. 4.8 we show the error between the true sequence and the sequence generated by the DAODEN_determ learnt on noisy observation with $r = 19.4\%$. Both sequences have the same starting point.

Table 4.3 – Performance of models trained on noisy and partial L63 data. The data are observed partially, both in time and in space, with a missing rate of 87.5%. For each index, the best score is marked in **bold** and the second best score is marked in *italic*.

Model		r 8.5%	16.7%	33.3%	66.7%
BiNN_EnKS	e_4	*0.129±0.081*	**0.143±0.065**	0.350±0.204	*0.973±0.649*
	rec	**0.721±0.204**	**1.062±0.401**	2.342±1.622	6.675±1.410
	$\pi_{0.5}$	1.873±1.034	**2.146±1.048**	*1.616±1.042*	*0.290±0.153*
	λ_1	0.801±0.016	0.782±0.012	0.304±0.147	-1.588±0.009
DAODEN_determ	e_4	0.135±0.082	0.170±0.105	*0.290±0.202*	25.034±19.821
	rec	1.300±1.525	1.448±1.332	1.985±1.474	4.222±2.191
	$\pi_{0.5}$	2.399±1.360	*2.140±1.110*	1.441±0.823	0.022±0.087
	λ_1	0.905±0.014	0.888±0.013	0.809±0.018	-0.011±0.014
DAODEN_MAP	e_4	0.175±0.119	0.325±0.235	0.459±0.343	9.105±7.136
	rec	1.352±0.997	1.705±1.434	*1.972±1.247*	*3.704±1.180*
	$\pi_{0.5}$	**2.628±1.448**	1.706±1.125	1.505±0.949	0.064±0.216
	λ_1	0.894±0.010	0.844±0.016	0.736±0.017	0.453±0.030
DAODEN_full	e_4	**0.089±0.062**	*0.158±0.104*	**0.162±0.104**	**0.254±0.142**
	rec	*1.052±0.612*	*1.268±0.718*	**1.685±0.928**	**2.725±1.356**
	$\pi_{0.5}$	*2.590±1.193*	1.943±0.904	**1.984±0.949**	**1.347±1.014**
	λ_1	0.892±0.011	0.846±0.013	0.859±0.013	0.720±0.019

Table 4.4 – Performance of models trained on noisy L96 data. For each index, the best score is marked in **bold**.

Model		r 19.4%	38.8%
AnDA	e_4	0.582±0.106	1.140±0.174
	$\pi_{0.5}$	1.491±0.481	0.768±0.281
	λ_1	53.362±0.734	92.733±0.883
SINDy	$e4$	0.309±0.048	0.767±0.117
	$\pi_{0.5}$	0.628±0.166	0.150±0.047
	λ_1	1.444±0.048	1.316±0.045
BiNN	e_4	0.310±0.046	0.788±0.112
	$\pi_{0.5}$	2.503±0.565	1.111±0.274
	λ_1	1.409±0.019	1.041±0.016
DAODEN_determ	e_4	**0.048±0.006**	0.157±0.022
	$\pi_{0.5}$	**4.790±0.960**	**3.178±0.779**
	λ_1	1.624±0.022	1.601±0.023
DAODEN_full	e_4	0.067±0.014	**0.145±0.030**
	$\pi_{0.5}$	4.076±1.084	3.146±0.962
	λ_1	1.543±0.026	1.348±0.020

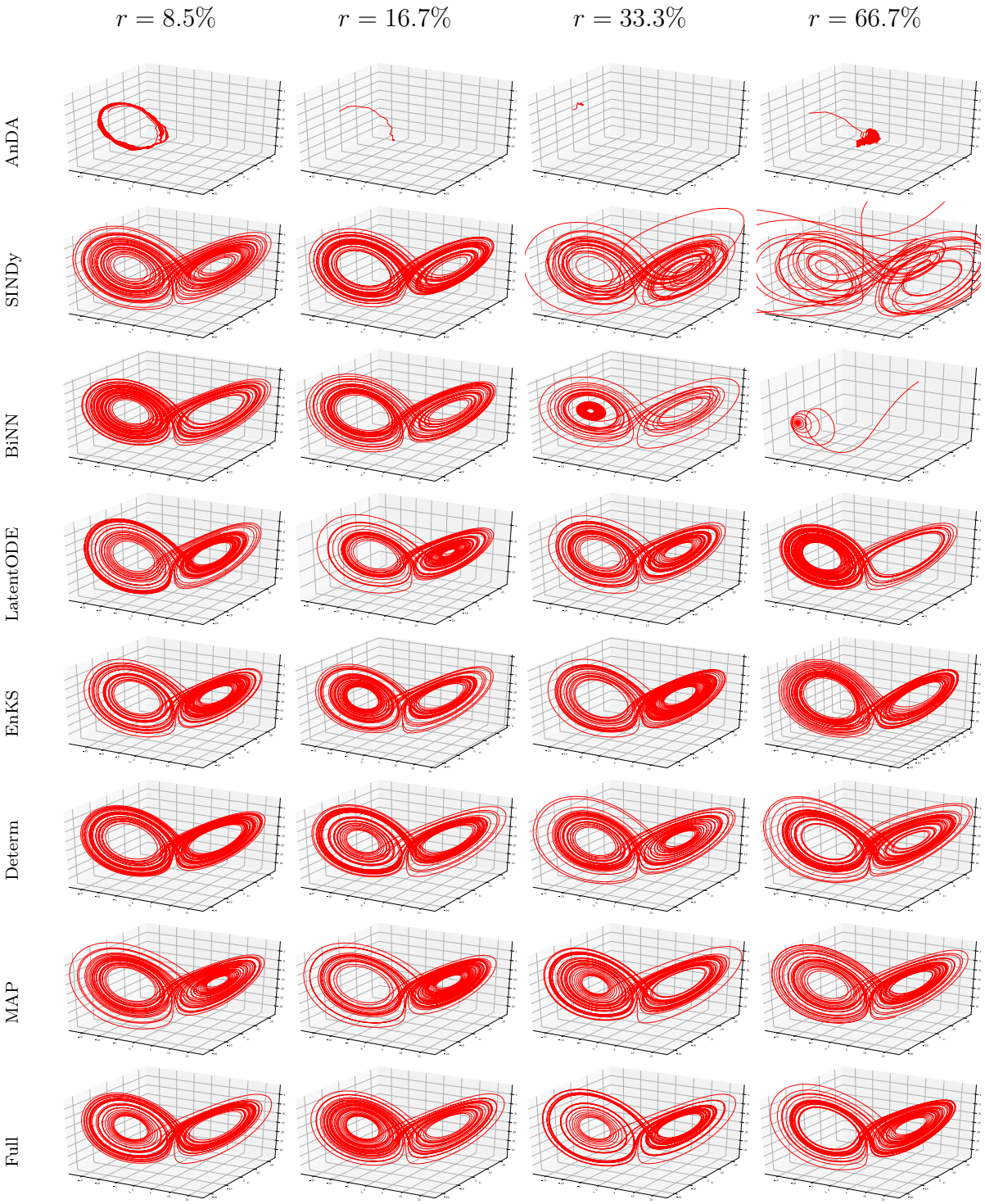


Figure 4.5 – Attactors generated by models trained on noisy data.

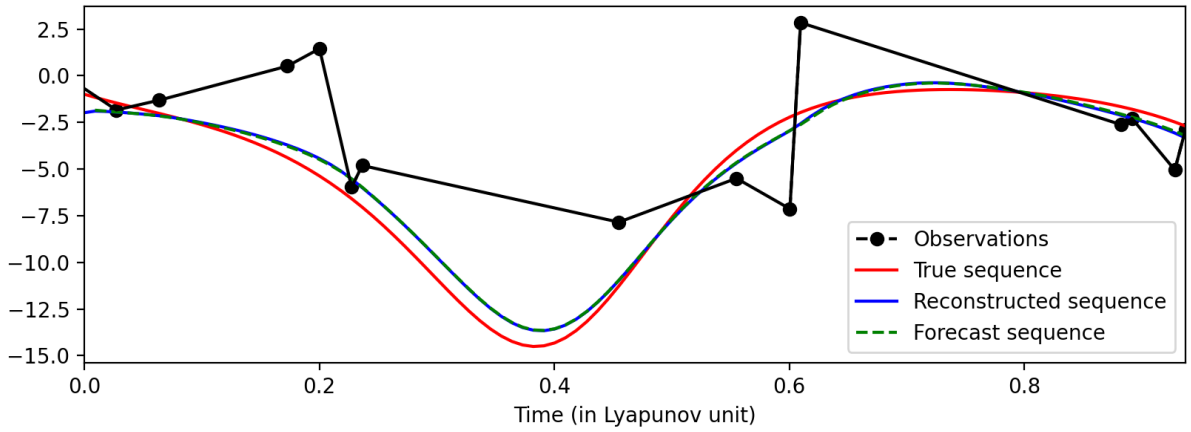


Figure 4.6 – An example of the the first dimension of the L63 system reconstructed by the inference module of DAODEN_determ trained on noisy and partial data. The observations are noisy ($r = 33\%$) and observed partially with a missing rate of 87.5%.

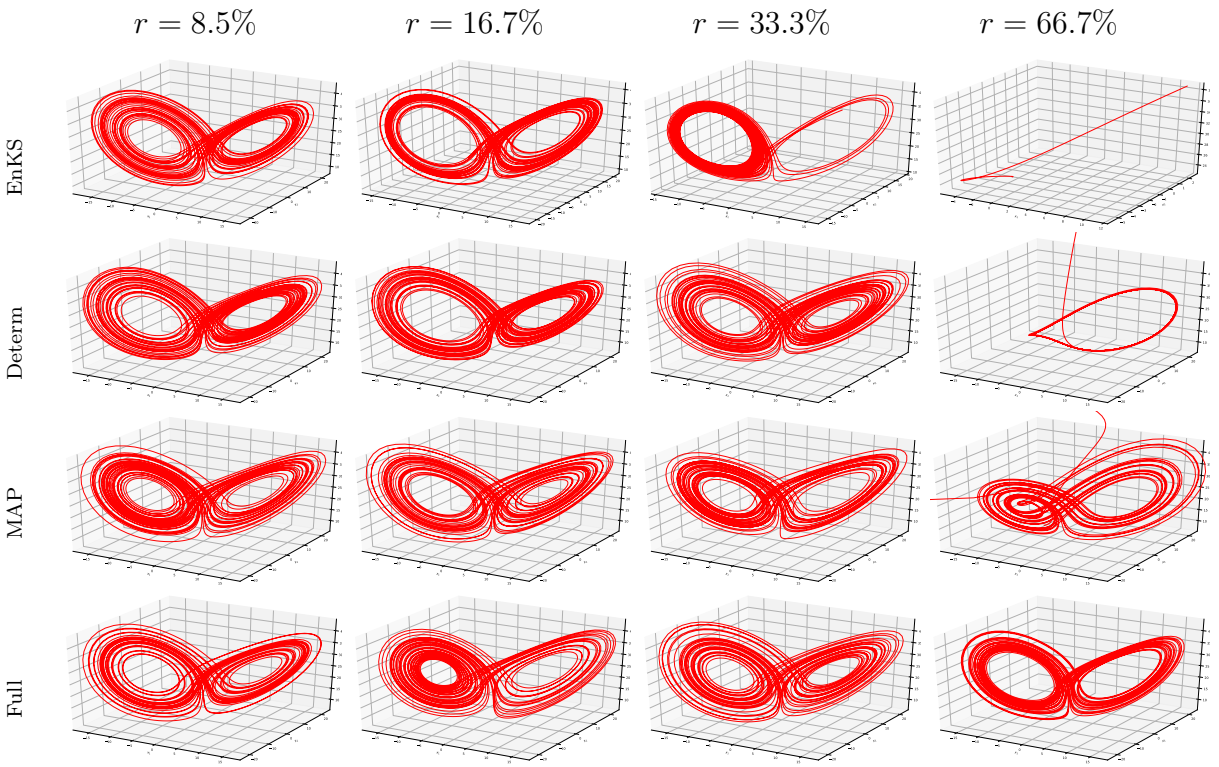


Figure 4.7 – Attractors generated by models trained on noisy and partially observed data.

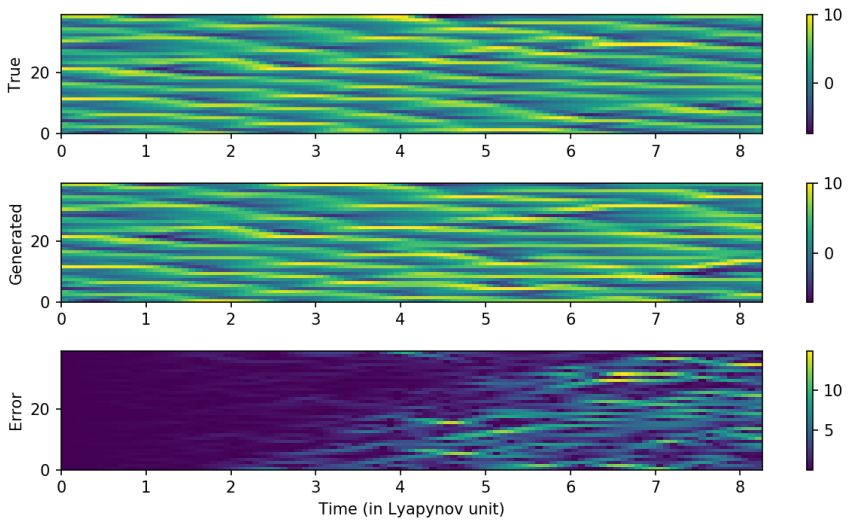


Figure 4.8 – The true L96 sequence (top), the sequence generated by the model trained on noisy data with $r = 19.4\%$ (middle) and the error between the true and the generated sequence (bot).

4.5.7 L63s case-study

Whereas most related work is designed for ODE only, (*i.e.* the governing equations are deterministic), the proposed framework accounts for stochastic perturbations, hence it can apply to Stochastic Differential Systems (SDEs). Using the stochastic Lorenz-63 system (L63s) presented in (Chapron et al. 2018), we illustrate in this experiment the ability of DAODEN_full scheme to infer stochastic governing equations from noisy observation data. We may recall that DAODEN_full scheme embeds a parametric form of the covariance of perturbation $\boldsymbol{\omega}_t$ given by ((4.1)). Note that this parametrisation is consistent with the true parametrisation for L63s (Chapron et al. 2018).

Here, we ran experiments similar to those Section 4.5.5 using L63s datasets with an additive Gaussian noise with $r = 33.3\%$. We then run the identification of the governing equations using both a deterministic parametrisation (*e.g.*, BiNN_EnKS and DAODEN_determ) and the fully-stochastic scheme DAODEN_full. For weak stochastic perturbations (typically, γ larger than 8.0 in Eq. (3.6) in Appendix 3.2.3), deterministic models like BiNN_EnKS or DAODEN_determ can still be able to capture the dynamics of the system (not reported in this chapter). However, when $\boldsymbol{\omega}_t$ plays in important role in controlling the large-scale statistical characteristics of the system, deterministic models fail as illustrated in Fig. 4.9 for L63s dynamics with $\gamma = 5.0$. By contrast, the fully-stochastic

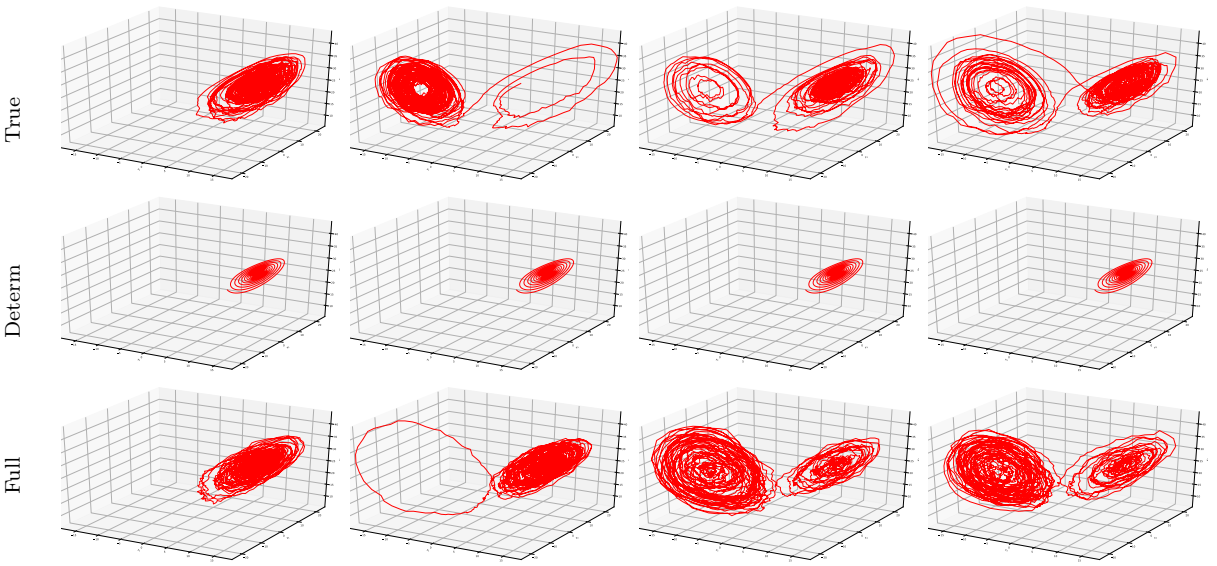


Figure 4.9 – Several attractors generated by the true L63s models (top), by DAODEN_determ (middle) and by DAODEN_full (bottom). The true L63s and DAODEN_full system are stochastic, hence each runtime we obtain a different sequence, even with the same initial condition. The models were trained on noisy observations with $r = 33.3\%$.

model successfully uncover the stochastic dynamics in both situations. In Fig. 4.9 top, we depict four different L63s trajectories from the same initial conditions. Due to the stochastic perturbation, the trajectories may strongly differ but all show a wide spreadout within the attractor. When considering a deterministic model (Fig. 4.9 middle), the four trajectories are strictly similar as there is no stochastic perturbation. Besides, the deterministic model simulates trajectories trapped on one side of the attractor, which cannot reproduce the spread of the true model. As illustrated in Fig. 4.9 bottom, DAODEN_full scheme succeed in capturing this stochastic patterns by embedding the stochastic factors of the system in the dispersion matrix $\mathbf{Q}_t$. Using a Monte Carlo technique, as presented in Alg. 2, to forecast the state of the dynamics, we can obtain sequences with similar characteristics to the true L63s system.

4.5.8 Dealing with an unkown observation operator

In previous experiments, the observation operator $\mathcal{H}$ was known. We may also address the situation where it is unknown. It may for instance refer to reduced-order modelling when one looks for a lower-dimensional representation of a higher-dimensional dynamical system.

Algorithm 2: Generate stochastic sequence

Result: A sequence $\mathbf{S}$ of length N, generated by the model $\{\mathcal{F}, MLP^{var\_dyn}\}$, starting form the initial condition $\mathbf{x}_0$.

Inputs: N, $\mathcal{F}$, MLP^{var_dyn}, $\mathbf{x}_0$;

$\mathbf{x} = \mathbf{x}_0$;

$\mathbf{S} = list()$;

t = 0;

while $t < N$ **do**

  $\mu = \mathcal{F}^1(\mathbf{x})$;

  $\mathbf{d}^{dyn} = MLP^{var\_dyn}(\mathbf{x})$;

  $\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{d}^{dyn}\mathbf{I})$;

  $\mathbf{S}$.append($\mathbf{x}$);

As case-study, we consider an experimental setting with Lorenz-63 dynamics similar to (Champion et al. 2019). The 128-dimensional observation space derives from a 3-dimensional space, where the system is governed by L63 ODE, according to a polynomial of $\mathbf{z}_t$ and $\mathbf{z}_t^3$ with six spatial modes of Legendre coefficients (for details, see (Champion et al. 2019)). Whereas noise-free cases were considered in (Champion et al. 2019), we report here experiments with a Gaussian additive noise with r=19.4%. Fig. 4.10 shows the observations in a high-dimensional space. The inference scheme in (Champion et al. 2019) is an NN-based encoder, this architecture does not take into account the sequential correlations in the data, hence when the observations are noisy, it can not apply (because $p(\mathbf{z}_t|\mathbf{x}_t)$ is intractable). Moreover, (Champion et al. 2019) supposes that the time derivative $\frac{d\mathbf{x}_t}{dt}$ is observed. This assumption may not be true for many real-life systems. Our model, on the other hand, uses a state space assimilation formulation. The inference scheme in our model is a sequential model, and we do not need the time derivative of the data, though it could be accounted for in the observation model.

The unknown observation operator $\mathcal{H}$ was parameterised by the same MLP architecture as the one used in (Champion et al. 2019). We run this experiment with DAODEN_determ. Fig. 4.11 shows that the proposed framework successfully captures the low-dimensional attractor of the observed high-dimensional observation sequences. This is further supported by the first Lyapunov exponent of the learnt model $\lambda_1 = 0.92$, which is close to the true value (0.91). Because there are several possible solutions for this problem (any affine transformation of the original L63 is a solution), the coordinates of the learnt system are different, however, the topology is well captured.

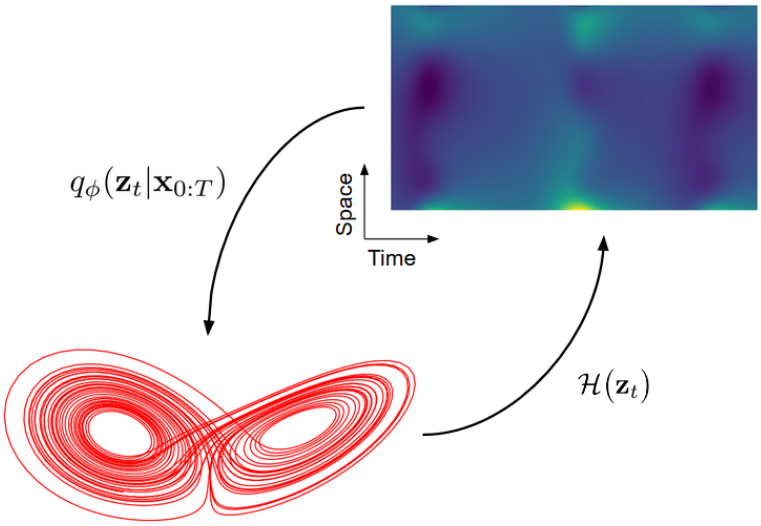


Figure 4.10 – Higher-dimensional Legendre observations governed by lower-dimensional L63 dynamics. Following (Champion et al. 2019), the observations (top right) are in a 128-dimensional space, while L63 dynamics (bottom left) are in a 3-dimensional space. The observation operator involves a non-linear mapping according to Legendre polynomials (Champion et al. 2019).

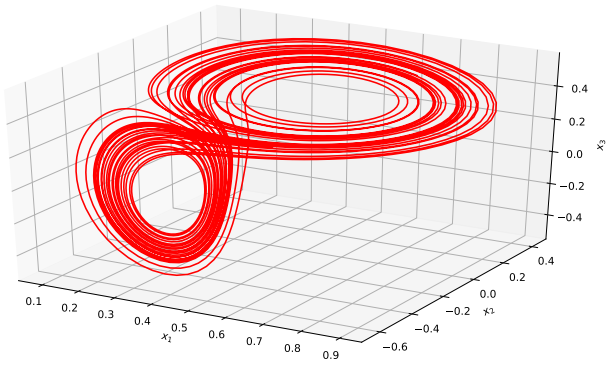


Figure 4.11 – Low-dimensional attractor generated by the proposed model trained from noisy higher-dimensional Legendre observations of L63 dynamics. This attractor recovers the topology of L63 dynamics. We let the reader refer to the main text for details on this experiment.

4.6 Conclusions

This chapter introduces a novel deep learning scheme for the identification of governing equations of a given system from noisy and partial observation series. We combine a Bayesian formulation of the data assimilation with state-of-the-art deep learning architectures. Compared with related work (Brajard et al. 2019; Bocquet et al. 2020), we account for stochastic dynamics rather than only deterministic ones and derive an end-to-end architecture using a variational deep learning model, which fully conforms to the state space formulation considered in data assimilation. Through numerical experiments for chaotic and stochastic dynamics, we have demonstrated that we can extend the observation configurations where we can recover hidden governing dynamics from noisy and partial data w.r.t. the state-of-the-art, including for high-dimensional systems governed by lower-dimensional dynamics.

Beyond the generalisation of previous work through a variational Bayesian formulation, the proposed framework involves two key contributions w.r.t. state-of-the-art data assimilation schemes. We first show that neural network architectures bring a new means for the parametrisation of both the dynamical models and the inference scheme. Especially, our experiments support the relevance of LSTM-based architectures as alternatives to state-of-the-art data assimilation schemes such as Ensemble Kalman methods (Evensen and Leeuwen 2000). Future work shall further explore these aspects and could benefit from the resulting end-to-end architecture to improve reconstruction performance (Fablet, Drumetz, et al. 2020).

For deep learning practitioners, our experiments point out that assimilation schemes and random n-step-ahead forecasting can be considered as regularisation techniques to prevent overfitting. We have also shown that the stochastic implementation of the proposed framework can capture characteristics of stochastic dynamical systems from noisy data. These results open new research avenues for dealing with real dynamical systems, for which the stochastic perturbations often play a significant role in driving long-term patterns.

From a practical point of view, the results showed in this paper suggests that although some models might be able to discover the governing equations of an unknown dynamical system when the data are not corrupted, one should incorporate those models with data assimilation schemes to account for that fact that the model may contain error, and the data are not perfect. Other results also support the use of NN-based method for the identification of dynamical systems.

Part III

Variational Deep Learning for Maritime Traffic Surveillance

Oh, I'm sailing away, my own true love.
I'm sailing away in the morning.
Is there something I can send you from across the sea?
From the place where I'll be landing?

Bob Dylan

Chapter 5

Introduction to the Automatic Identification System

In the world of a globalised economy, maritime surveillance is a vital demand. Currently being the most efficient long-distance transporting method, sea transport is carrying about 90% of the world trade (IMO 2020). With the persistent growth of maritime traffic, safety and security are key issues. Besides, the real-time delivery of maritime situation maps is also necessary for a variety of activities: fishing activities control, smuggling detection, EEZ intrusion detection, transshipment detection,maritime pollution monitoring, *etc*. Among many other technologies that have been developed, the **automatic identification system** (AIS) is one of the most important sources on information. In this chapter, we will introduce what AIS is, the potential of AIS for maritime domain awareness and challenges working with AIS. This chapter provides the context for the work in Chapter 6 and Chapter 7.

5.1 The automatic identification system

The **automatic identification system** (AIS) is an identifying and locating system installed on board of vessels to self-report the static information of the ship and the dynamic information of the voyage. According the International Maritime Organisation's (IMO) International Convention for the Safety of Life at Sea (SOLAS), (SOLAS 1974), all

international vessels with 300 or more gross tonnage, and all passenger ships regardless of size have to equip AIS transceivers.

AIS transmits the following information:

— The **static information** of the vessel (sent every 6 minutes and on request):
  + The Maritime Mobile Service Identity (MMSI) number, which is a nine-digit number for identifying a ship. All AIS electronic devices on board of a vessel use one MMSI, this number is assigned by the appropriate authorities in the country of registration, and can be recognised internationally. The format of MMSI numbers is MIDXXXXXX, where the first three digits are the maritime identification digits (MID), identifying the the nationality, and the last six digits are the unique identification of the vessel;
  + The International Maritime Organisation (IMO) number, assigned by the to the hull of each ship. The format of IMO numbers is three letters "IMO" followed by a seven-digit unique number. While the MMSI of a vessel can be changed (when the vessel is registered with another country for example), the IMO number is permanent. However, AIS is using the MMSI, not the IMO number, as the unique identification of a vessel;
  + The name and the call sign of the vessel;
  + The length and the beam;
  + The type of the vessels;
  + *etc.*

— The **voyage related information** (sent every 6 minutes, when data is changed, or on request)
  + The vessel's draught;
  + The type of cargo;
  + The destination and the estimated time of arrival (ETA);
  + *etc.*

— The **dynamic information** (every two seconds to a few minutes):
  + The precise current position (longitude, latitude) of the vessel;
  + The current **speed over ground** (SOG);
  + The current **course over ground** (COG) and the heading of the vessel.

  + The **rate of turn** (ROT) (if available);
  + The navigation status;
  + *etc.*

— Short safety related messages: free format short text messages with important navigational safety related information. are shown in an extra window.

The information above is broadcast in the **very high frequency** (VHF) mobile maritime band. AIS allows vessels to detect and display other vessels in their vicinity and helps maritime authorities to track vessel movements in order to monitor maritime traffic. The system was developed in the 1990s, and was originally designed as a high intensity, short-range network, meaning vessels send their AIS signal to other AIS receivers within the VHF range (about 10–20 nautical miles). For this reason, it could be used terrestrially only (called *T-AIS*). However, as maritime international transport and trading has become more and more popular, AIS has evolved to be detected by satellites (since 2008). Satellite-based AIS, called *S-AIS*, uses **time-division multiple access** (TDMA) radio access to transmit signal. S-AIS, in complement with T-AIS, provide relatively full coverage of the globe, as shown in Figure 5.1.

5.2 AIS applications

The original purpose of AIS is collision avoidance. However, thanks to its information richness, AIS has been exploited in a lot of applications (Iphar et al. 2019):

— **Collision avoidance**: at sea, information about the position and the movement of others vessels in the vicinity is crucial for vessels to avoid collision. AIS can be used, in complement with other sources of information such as visual observation, audio exchanges or radar, to give a better real time picture of the sea situation. Moreover, AIS has also integrated a collision alarm system that predicts the movements of vessels based on the current positions and rises an alarm if there is possibly a collision will happen.

— **Fishing control**: about 31 percent of the seafood flowing through the global market is illegal, unreported or unregulated; and at least 50 percent of the amount of fish being taken has been underestimated (Agnew et al. 2009). Fishing control, therefore, becomes crucial. Many countries, including the United States, EU member, *etc.* have used AIS to monitor fishing activities along their coast line. The positions of the

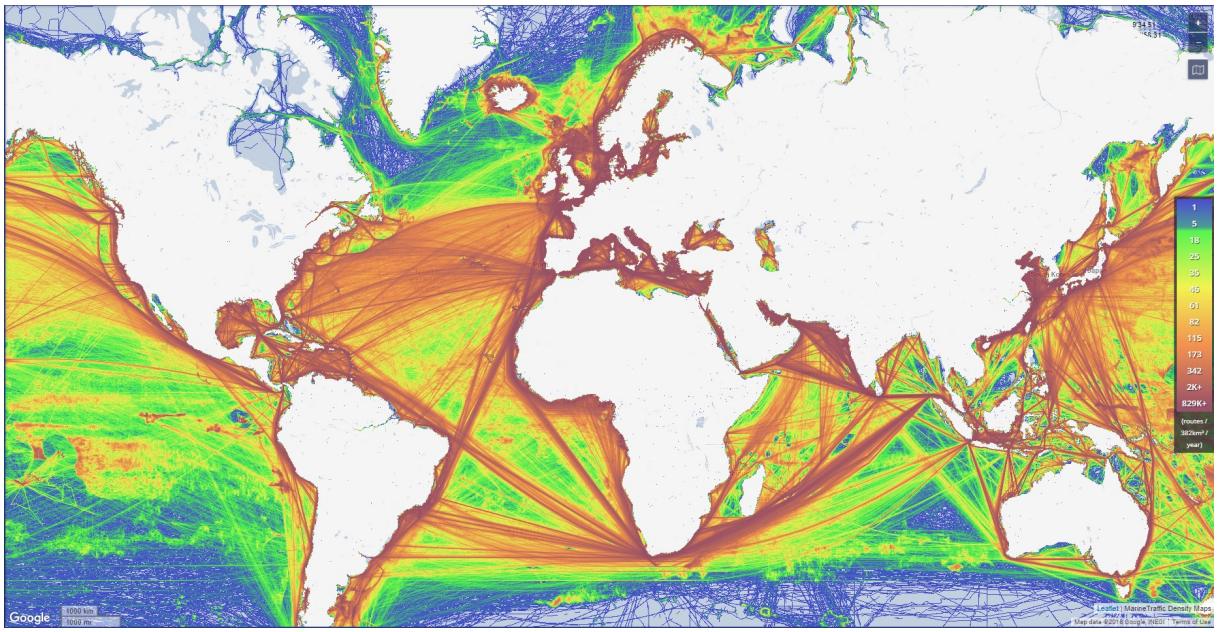


Figure 5.1 – An AIS density map in a global scale (image from MarineTraffic).

ships are relayed back to the government agencies. Any suspicious behavior detected such as the presence of a fishing ship in a non-fishing region, the discontinuation of the AIS track of a vessel, *etc.* will raise an alarm then the law enforcement vessels will examine those ships.

— **Maritime security**: the main strength of AIS in maritime domain awareness is the capability of identifying a threat as early and as distant as possible. As AIS provides the precise position of vessels, it can be used to detect exclusive economic zone (EEZ) intrusion or potential dangerous actions at the beginning, gives the coast guards enough time to find the optimal solution and to prevent the threat from happening.

— **Fleet and cargo tracking, route planning**: thanks to AIS, vessels now can know exactly where they are, whap happen in the vicinity, what is the optimal energy consumption rout, the estimated time of arrival.

— **Aids to navigation**: AIS aids to navigation stations, acting as a modern lighthouse, can broadcast their positions and some additional information (weather, see state, *etc.*) to help vessels navigate safely. This information is useful for safety navigation.

5.3 Challenges working with AIS

Although in theory, AIS is a rich, fine-grained source of information of maritime traffic in a global scale, exploiting AIS at its full potential is challenging:

— Every day, there are more than 500 millions AIS messages transmitted (Perobelli 2016). This amount of data quickly overwhelms human capacity to process AIS data manually (see Chapter 6 and Chapter 7).

— AIS data are unreliable. Some attributes in AIS, such as the navigation status, destination, *etc.* are set manually. Most of the time, they are (intentionally or unintentionally) incorrect. Even the attributes that are measured automatically by the sensors such as the positions (latitude, longitude), the SOG, the COG, *etc.* can also be spoofed (Iphar et al. 2020).

— AIS data are noisy and are sampled irregularly. Noise may come from the errors in the measurements of the sensors, or from interference in the transmitting channel, or because the data are spoofed. Usually, the interval between two consecutive AIS messages is irregular.

— AIS data can be interrupted, because the vessel enter a zone not covered by AIS, or because the transponders are switched off intentionally.

— No metadata are available for AIS.

In the next chapter, we will present a model that tackles those problems. Specifically, in Chapter 6, we introduce a multitask deep learning architecture that can handle massive, noisy and irregularly sampled AIS data to perform important maritime traffic surveillance tasks such as trajectory reconstruction, vessel type identification, anomaly detection, *etc.*

So we beat on, boats against the current, borne back ceaselessly into the past.

F. Scott Fitzgerald

Chapter 6

MultitaskAIS

[1] [2] As presented above, in a world of global trading, maritime safety, security and efficiency are crucial issues. In this chapter, we present our work under the context of SESAME initiative [3], which aims at developing new solutions for management and analysis of maritime satellite data. Specifically, we propose a multi-task deep learning framework for vessel monitoring using Automatic Identification System (AIS) data streams. We combine recurrent neural networks with latent variable modelling and an embedding of AIS messages to a new representation space to jointly address key issues to be dealt with when considering AIS data streams: massive amount of streaming data, noisy data and irregular time-sampling. We demonstrate the relevance of the proposed deep learning framework on real AIS datasets for a three-task setting, namely trajectory reconstruction, anomaly detection and vessel type identification.

1. This chapter is a modified version of paper (Duong Nguyen, Vadaine, et al. 2018)

2. This work was supported by public funds (Ministère de l'Education Nationale, de l'Enseignement Supérieur et de la Recherche, FEDER, Région Bretagne, Conseil Général du Finistère, Brest Métropole) and by Institut Mines Télécom, received in the framework of the VIGISAT program managed by "Groupement Bretagne Télédétection" (BreTel). It benefited from HPC and GPU resources from Azure (Microsoft EU Ocean awards) and from GENCI-IDRIS (Grant 2020-101030).

We acknowledge the support of DGA (Direction Générale de l'Armement) and ANR (French Agence Nationale de la Recherche) under reference ANR-16-ASTR-0026 (SESAME initiative), grants Melody and OceaniX.

We also would like to thank MarineCadastre for the Gulf of Mexico AIS dataset and Collecte Localisation Satellites as well as Erwan Guegueniat for the Brittany AIS dataset.

3. http://recherche.imt-atlantique.fr/sesame

6.1 Introduction

Over the last decades, the development of terrestrial networks and satellite constellations of Automatic Identification System (AIS) has opened a new era in maritime traffic surveillance. Every day, AIS provides on a global scale hundreds of millions of messages (Perobelli 2016), which contain ships' identifier, their Global Positioning System (GPS) coordinates, their speed, course, etc. The potential of this massive amount of data is clearly of interest if tools and models provide means to efficiently extract, detect and analyze relevant information from these data streams. However, current operational systems, which strongly rely on human experts, can only deal with a limited fraction of AIS data.

Thus, the development of AI-based systems is a critical challenge. Beyond the volume of streaming data to be dealt with, there are two other key issues make it difficult to design these types of systems: noise patterns exhibited by AIS data and the irregular time-sampling.

Both are very common in AIS and make the direct application of state-of-the-art supervised machine learning models, including deep learning ones poorly adapted. This chapter addresses these issues and explores deep learning models and architectures, and more specifically Recurrent Neural Networks (RNNs) to develop an automatic system that can process and detect, extract and characterise useful information in AIS data streams for maritime surveillance. More specifically, our key contributions are three-fold:

— The design of a novel big-data-compliant unsupervised architecture which automatically learns and extracts useful information from noisy and partial AIS data streams on a regional scale;

— The joint exploitation of this architecture as a basis for specific tasks using mathematically-sound statistical models, namely trajectory reconstruction and forecasting, maritime route estimation, vessel type identification, detection of abnormal vessel behaviours, *etc.*;

— The demonstration of the proposed approach's relevance on real regional datasets off Brittany coast and in the Gulf of Mexico, significantly more complex than case-studies addressed in previous work.

This chapter is organised as follows: in Section 6.2, we review the state-of-the-art methods in AIS-based maritime surveillance. The proposed method is detailed in Section 6.3. We present experiments in Section 6.4, and further discuss the main features and performance of our approach in Section 6.5. Finally, conclusions and perspectives for future

work are presented in Section 6.6.

6.2 Related work

In this section, we review the related works in the field of AIS-based maritime traffic surveillance, especially regarding trajectory reconstruction and forecasting and anomaly detection.

Trajectory reconstruction and forecasting: For simplicity purpose, we use here the term "trajectory reconstruction" to refer to both trajectory reconstruction and trajectory forecasting. Early efforts for trajectory reconstruction include linear interpolation, curvilinear interpolation (Best et al. 1997) and their improvements (Perera et al. 2012; Schubert et al. 2008). They rely on a physical model of the movement $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t * \mathbf{x}_k^{'}$ (where $\mathbf{x}_k$ is the position of vessel at the time k, $\mathbf{x}_k^{'}$ is the deviation of $\mathbf{x}_k$, usually the SOG and the COG). More sophisticated methods suppose that vessel trajectories follow a distribution and learn it from historical data (Millefiori et al. 2016; G. Pallotta et al. 2014). Currently, state-of-the-art methods for trajectory reconstruction (F. Mazzarella, V. F. Arguedas, et al. 2015; Hexeberg et al. 2017; Coscia et al. 2018) use the following typical three-step approach: i) the first step involves a clustering method, *e.g.* TRACLUS (Lee et al. 2007) or TREAD (Giuliana Pallotta et al. 2013) to cluster historical motion data into route patterns, ii) the second one assigns the vessel to be processed to one of these clusters iii) the third one interpolates or predicts the vessel trajectory based on the route pattern of the assigned cluster.

Anomaly detection: Some models detect abnormal behaviours by defining them explicitly (Holst et al. 2016; Gaspar et al. 2016). These types of models are usually limited themselves by their own definitions, and can not handle all the complex phenomenons observed at sea. To overcome those drawbacks, other methods detect anomalies implicitly by creating normalcy models, then consider trajectories or trajectory segments that do not suit these models as abnormal. In (Rhodes et al. 2005), Rhodes divided the map into small zones and used Normalcy Box to detect abnormal vessel speed in each zone. **gaussian mixture models** (GMMs), **kernel density estimation** (KDE) were explored in (Laxhammar 2008; Ristic et al. 2008). More sophisticated methods have used time series analysis techniques, such as **Gaussian process** (Kowalska et al. 2012; Will et al. 2011), or **Bayesian networks** (BNs) (Johansson et al. 2007; Mascaro et al. 2014) to capture the sequential structure of AIS streams. All these models share the same basic idea: in a

small region, vessels should perform similar behaviours.

All models and approaches reviewed for trajectory reconstruction and anomaly detection present three main drawbacks:

— They depend on strong priors and can hardly capture all the heterogeneous characteristics of AIS data as well as the varieties of vessels' behaviours. Near-far, fast-slow, etc. are relative definitions and are difficult to be implemented. Almost all current models work only for cargo and tanker vessels on specific high-traffic maritime routes. However, more sophisticated models and relaxed assumptions are required to address the range of vessel types and vessel behaviours revealed by AIS streams on a regional or global scale.

— Most if not all methods exploit at some point a clustering. They typically assume that in specific areas, all vessels tend to perform similar behaviours, and then use clustering methods (Kmeans, DBSCAN, etc.) to find those behaviours. For example, for trajectory reconstruction issues, each cluster is a maritime route (Giuliana Pallotta et al. 2013); in anomaly detection, each cluster is a speed mode (G. Pallotta et al. 2014; Rhodes et al. 2005), etc. We believe that such clustering steps result in information losses. By contrast, we argue that continuous latent states should be preferred to address the complexity of AIS data streams.

— Current methods do not explicitly address the irregular time-sampling of AIS streams. Non-sequential methods (Rhodes et al. 2005) do not take it into account and sequential ones (F. Mazzarella, V. F. Arguedas, et al. 2015) assume they are provided with regularly-sampled streams, which is not true or may result in the creation of artificial, possibly erroneous AIS positions if interpolation techniques are used as a pre-processing step.

As detailed hereafter, we develop a novel multi-task deep learning framework to address these issues and demonstrate its relevance from experiments on a real AIS dataset on a regional scale.

6.3 Proposed multi-task VRNN model for AIS data

As sketched in Fig. 6.1, we propose a general multi-task neural-network-based model for the analysis of AIS data streams. Received AIS messages are regarded as irregular noisy observations of the true hidden states - called regimes; these regimes themselves

may correspond to specific activities (*e.g.* under way using engine, at anchor, fishing, etc.). The key component of our model is the **Embedding block**, which converts noisy and irregularly-sampled AIS data to consistent and regularly-sampled hidden regimes. This Embedding block relies on a VRNN (J. Chung et al. 2015) and operates at a 10-minute time scale. Higher-level blocks are task-specific submodels, addressing at different time-scales (*e.g.* daily, monthly,...) the detection of abnormal behaviours, the automatic identification of vessel types, vessel position prediction, the identification of maritime routes, *etc.*

6.3.1 A latent variable model for vessel behaviours

Through a VRNN architecture (see Section 2.4.2 for details), we introduce hidden regimes (latent variables $\mathbf{z}_k$ of the VRNN) as a data representation [4] that captures the true maneuvers of vessels (*natural clustering*). Hidden regimes can be regarded as the "roots" of AIS messages. They govern how the vessel moves. From the point of view of higher levels (task-specific layers), hidden regimes provide the necessary information for their task (*hierarchical organisation of explanatory factors* and *shared factors across tasks*). They disentangle the underlying information of AIS data (*simplicity of factors dependencies*). For example, saying "this vessel is performing a fishing maneuver" is much more informative than saying "the speed of this vessel is high".

It is important to note that the hidden regimes are not clusters of AIS messages, because the act of assigning data to group would cause information loss. We share the same idea with (Diederik P. Kingma and Welling 2013), that latent variables (hidden regimes in this case) are continuous and there are no simple interpretations of these dimensions.

The introduction of hidden regimes brings us two key benefits: an efficient encoding of AIS datasets and a regularly-sampled sequential representation. Regarding the first aspect, state-of-the-art systems such as TREAD (Giuliana Pallotta et al. 2013) have to store all the AIS messages in the training set, which is updated incrementally new AIS messages. Therefore, data volume to be handled for the test phase increases rather linearly with the area of the **region of interest** (ROI) and the duration of the considered time period. This may prevent such systems from scaling up to regional or global scales. By contrast, once the VRNN is trained, all the knowledge gained from a given AIS dataset is encoded by the characteristics of the hidden regimes, more precisely the fitted conditional distributions $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})$. Therefore, for the application of

4. Here we use the criteria defined in (Y. Bengio et al. 2013) to evaluate this representation, readers are encouraged to read (Y. Bengio et al. 2013) for additional information.

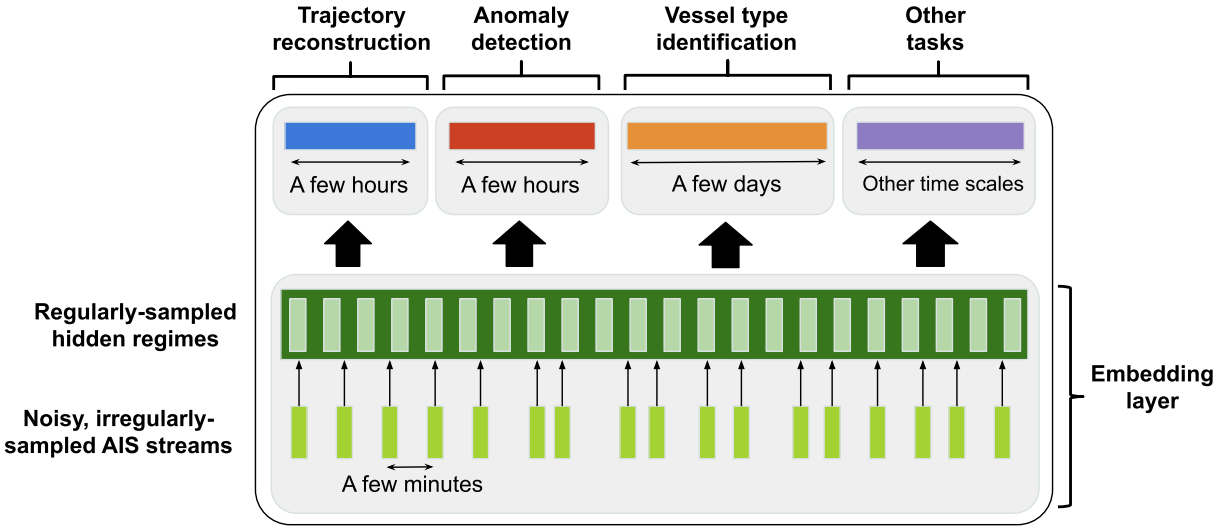


Figure 6.1 – Proposed VRNN architecture.

a trained model, there is no need to access the training dataset. This dataset may only be of interest to retrain or fine-tune a given model. It may be noted that the complexity of the representation of the hidden regimes (*i.e.*, the associated number of parameters) does not depend on the training data volume. For instance, in the considered experiments, for a dataset of more than 2.10^8 AIS messages (each message contains several attributes), the fitted hidden regime representation involves about 5.10^6 parameters. The second important feature is the mapping of an input space consisting of a noisy irregularly-sampled time series to a novel regularly-sampled sequential representation which naturally accounts for the different sources of uncertainties exhibited by AIS datasets. Hence, the proposed architecture embeds somehow a time regularisation of the input data and does not require the definition of ad hoc denoising and interpolation pre-processing steps, which prove difficult due to the variabilities to be dealt with (*i.e.*, duration of the missing data segments, noise patterns, inhomogeneous space-time variabilities, etc.). From a mathematical point of view, the considered model naturally embeds these issues through the time propagation of the approximate posterior distribution $q_\phi(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{0:k-1})$. Overall, this regularly-sampled sequential representation makes feasible the design of classic architectures on top of the embedding layer to deal with task-specific issues as detailed in Sections 6.3.4, 6.3.5 and 6.3.6.

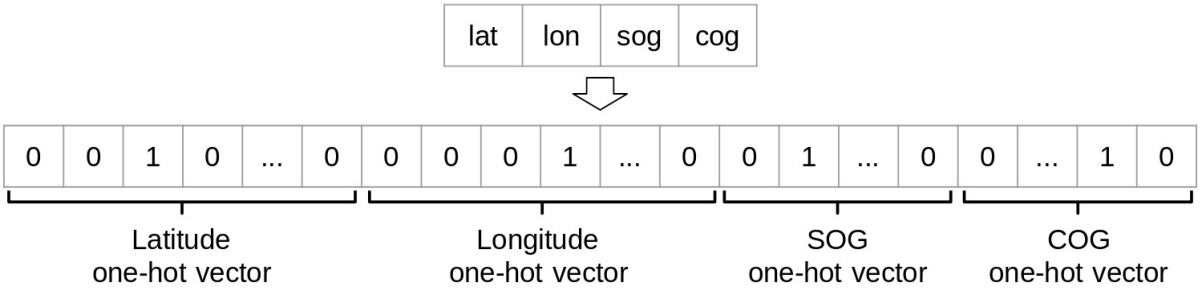


Figure 6.2 – "Four-hot" vector.

6.3.2 "Four-hot" representation of AIS messages

Instead of presenting AIS messages directly by their 4-D real-value vector: like methods in the literature (G. Pallotta et al. 2014):

$$\mathbf{x}_k = [lat_k, lon_k, SOG_k, COG_k]^T, \tag{6.1}$$

we apply a bucketing technique to introduce a novel representation of AIS data: the "four-hot encoding" (Fig. 6.2). This representation, inspired by the one-hot encoding in language modelling, is created by concatenating the one-hot vectors of 4 attributes in AIS message: latitude coordinate, longitude coordinate, SOG and COG. To create the one-hot vector of an attribute, we simply divide the entire value range of this attribute into $N_{attribute\_i}$ equal-width bins.

The "four-hot encoding" not only brings us the benefits of bucketised representation but also provides a more structured representation to learn trajectory spatial patterns as illustrated in Section 6.5. Our four-hot representation shares similarities with (Jiang et al. 2017). However, in (Jiang et al. 2017), the authors explained their representation as a transformation from feature space to semantic space based on the smoothness prior assumption. They argued that the continuous values of features did not matter, the explanatory factors were the semantic interpretation presented in their discrete vector of these values. We, on the other hand, consider the "four-hot encoding" as a presentation that can i) accelerate the calculation of neural networks (similar to one-hot encoding), ii) disentangle some explanatory factors of input features (see Section 6.5). The semantic space in our architecture is the space of hidden regimes.

The implicit reduction of the precision of the AIS position and velocity features may be regarded as a drawback of the four-hot representation. We however argue that for the

targeted applications there is no need for the embedding block to provide precise numerical features. For example, a speed of 12 knots and a speed of 12.1 knots do not mean any difference in our context.

6.3.3 Embedding block

The embedding block is a VRNN (J. Chung et al. 2015), where $\mathbf{x}_k$ is the "four-hot encoding" of AIS message and $\mathbf{z}_k$ is the concatenation of the hidden state of the network and the latent variable at the time k. This layer works at a 10-minute time scale (*i.e.* we downsample AIS data stream to a resolution of 10 minutes) and learns the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:k})$ (via the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1})$, the emission distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})$ and the approximate posterior distribution $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{0:k-1})$).

After being trained, the embedding layer consistently generates regularly time-sampled hidden regime series. This series is used as input to task-specific submodels as sketched in Fig.6.1.

6.3.4 Trajectory reconstruction submodel

The Embedding block is naturally a generative model, so the construction of a vessel trajectory estimator/predictor on top of this block is relatively direct. We follow the philosophy of (F. Mazzarella, V. F. Arguedas, et al. 2015). In this approach, one infers maritime contextual information, which is used to enhance the prediction/estimation. The contextual information in (F. Mazzarella, V. F. Arguedas, et al. 2015) was inferred by TREAD (Giuliana Pallotta et al. 2013), which means that each vessel would be assigned to a predefined route. By contrast, we avoid such a hard assignment to a predefined behavioural cluster. We benefit from the richer contextual representation inferred by the Embedding block. Formally, the proposed trajectory reconstruction model is stated as the inference of the posterior $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{0:k-1})$ and the sampling-resampling from the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}|\mathbf{x}_{0:k}, \mathbf{z}_{0:k}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_{k+1}|\mathbf{x}_{0:k}, \mathbf{z}_{0:k+1}) p_{\boldsymbol{\theta}}(\mathbf{z}_{k+1}|\mathbf{x}_{0:k}, \mathbf{z}_{0:k}) d\mathbf{z}_{k+1}$ (all learnt by the Embedding block) using a particle filter (Maddison et al. 2017).

6.3.5 Abnormal behaviour detection submodel

The second specific task on top of the Embedding block is the detection of abnormal behaviours. It comes to define a normalcy model to detect the (unlikely) anomalies w.r.t.

this model. As a direct by-product of the trained Embedding block, we can evaluate the likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:k})$ of any input AIS sequence $\mathbf{x}_{0:k}$ using a marginalisation w.r.t. the hidden regimes. A series of AIS messages with a very low likelihood w.r.t. a given threshold may be regarded as being unlikely for model $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:k})$ and hence as abnormal.

One may however consider context-aware detection rules. For example on maritime routes, vessels' behaviours are roughly identical, which leads to high values for the likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:k})$. In other regions, the variety of vessel types and activities results in much more complex mixtures of behaviours and much lower likelihood values for the normalcy model. The selection of a global threshold over an entire region may not be as appropriate. To address these issues, we introduce an *a contrario* detector (Ammar et al. 2013) [5]. It works at a 4-hour time scale and addresses the early detection of abnormal vessel behaviours. We divide the map into small cells C_i. In each cell, we calculate the mean $\mathbf{m}_i$ and the standard deviation $\mathbf{std}_i$ of the $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1})|_{\mathbf{x}_k \in C_i}$ using the tracks in the validation set. Any evolution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1})$ at timestep k of an AIS track will be considered as an abnormal evolution if its log-likelihood is much lower than the distribution of other log-likelihoods in the same cell. The *a contrario* detection detects if an arbitrary segment is abnormal based on the number of abnormal evolutions in this segment and its length.

6.3.6 Vessel type identification submodel

The third task addressed by our model is the identification of the vessel type from its AIS-derived trajectory data. It may be noted that the vessel type should be one of the attributes included in AIS messages. However, not all vessels send their static messages. Some may even send on purpose a false vessel type in AIS messages. A Vessel type identification submodel is then an important tool to detect suspicious behaviours.

Different types of vessels usually perform specific behaviours, which may differ among others in terms of geographical zones, speed patterns, etc. For example, tankers normally follow maritime routes (usually straight lines between two maritime waypoints (Giuliana Pallotta et al. 2013)), their average speed is relatively low, about 12-15 knots, whereas passenger ships have relatively high average speed, about 20-25 knots. If a vessel declares itself as type "A" but performs a maneuver of type "B", it is likely that it may carry out illegal activities.

In this study, we design a Vessel type identification submodel at a 1-day time scale.

5. We let the reader refer to (Ammar et al. 2013) for a detailed description of the *a contrario* setting.

This submodel explores a Convolutional Neural Network (CNN). The input of this CNN is a HxD matrix, whose columns are the hidden regimes (dimension H), and D is the number of timesteps in one day (144 in this case). Because the hidden regime is regularly time-sampled, this configuration applies directly.

6.4 Experiments and Results

We implemented the proposed framework for a three-task model, addressing respectively vessel trajectory reconstruction, abnormal behaviour detection and vessel type identification, in the Gulf of Mexico and the abnormal behaviour detection off Brittany coast in the Ushant zone [6]. The Ushant water is the entrance to English channel, this region is interesting to maritime surveillance because of its separation scheme and the heavy traffic there. The Gulf of Mexico is relatively large compared to the case-study regions considered in previous studies (Giuliana Pallotta et al. 2013; Laxhammar 2008; Kowalska et al. 2012). This region involves multiple vessel types and activities of vessels. It comprises big ports, fishing zones, oil platforms and dense maritime routes. Overall we considered AIS data from January to March 2017 off Brittany coast in the Ushant zone (2,021,236 AIS messages) and from January to March 2014 in the Gulf of Mexico (180,344,817 AIS messages).

6.4.1 Preprocessing

For the pre-processing step, first, infeasible speed or infeasible position messages were removed from the set. To handle the problem of very long sequence when working with RNNs, we split vessels' tracks into subtracks of from 4 to 24 hours. From now on, in this chapter, vessel tracks refer to such subtracks. We also removed tracks whose speed is smaller than 0.1 knots for more than 80% of the time (at anchor or moored vessels).

One objective of the proposed architecture is to deal with irregularly-sampled data. However, we need regularly sampled data to train the model first. In a layman's term, the Embedding block must see how regularly sampled AIS tracks should be and learn their characteristics, after that (after being trained), it with generate regularly-sampled data from irregularly-sampled ones. Therefore, for the training set, we only chose tracks whose the maximum time interval between two successive received AIS messages is 1 hour, then used constant velocity model to create regularly time-sampled AIS tracks at 10-minute

6. The Tensorflow code and the datasets are available at https://github.com/dnguyengithub/MultitaskAIS

Table 6.1 – Log likelihoods of the Embedding block with different dimension settings (Gulf of Mexico dataset).

Hidden regime dimension	Number of parameters	Log likelihood on training set	Log likelihood on test set
200	1 605 402	-7.592710	-7.678684
400	**5 129 202**	**-6.557936**	**-7.520255**
500	7 611 102	-6.130078	-7.690255

time scale. By doing this, the intervals between two successive AIS messages are small enough that the errors in the estimation of the constant velocity model do not effect our model too much.

6.4.2 Embedding block calibration

We implemented the Embedding block by a VRNN whose the RNN is a single-layer LSTM, distributions $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{h}_k)$, $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k)$, $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_k, \mathbf{h}_k)$ are fully connected networks with one hidden layer of the same size of the LSTM's. $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{h}_k)$ is binomial, $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k)$ and $q_{\boldsymbol{\phi}}(\mathbf{z}_k|\mathbf{x}_k, \mathbf{h}_k)$ are Gaussians. The network was trained with stochastic gradient descent using Adam optimiser (Diederik P. Kingma and Ba 2015), learning rate of 0.0003.

There is a trade-off between the resolutions of AIS features and the size of the network when choosing the length of the "four-hot encoding". If the resolutions are too high, the "four-hot" vector will be too long, requires a big hardware memory and computational power; if the resolutions are too low, we lose information. We set here the resolution of the latitude and longitude coordinate at about 1 km, the resolution of SOG at 1 knot and the resolution of COG at 5°. These resolutions are fine enough for almost all the maritime safety, security and efficiency tasks. For example, with this setting, the uncertainty zone of vessel's position is about 1kmx1km, small enough for position-related tasks.

The choice of the dimension of hidden regime effects the modelling capacity of the Embedding block. As shown in Table 6.1, if the latent size is too small, the model can not capture all the variations of AIS data. In contrast, if the latent size is too big, the model becomes too bulky and overfitting. For the rest of this chapter, we set the latent size at 400 for tests on the Gulf of Mexico dataset and at 100 for tests on the Brittany dataset (Ushant water).

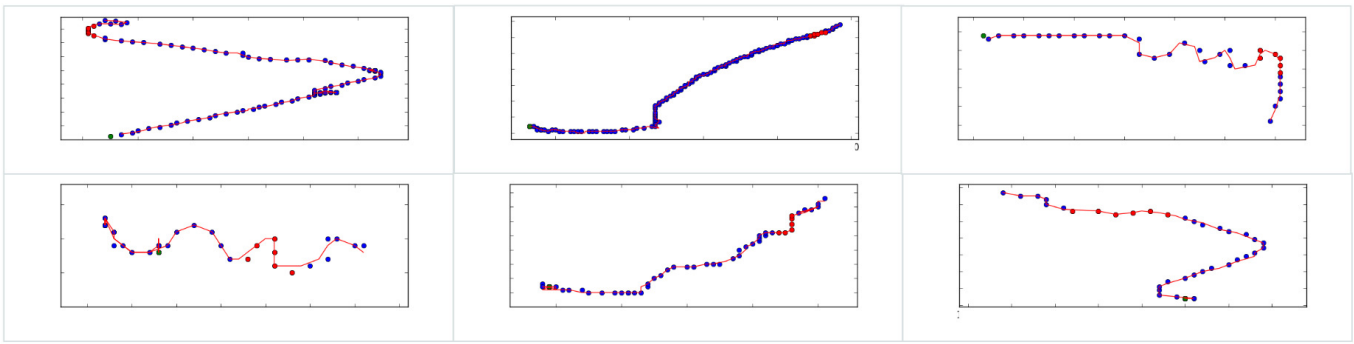


Figure 6.3 – Trajectory reconstruction examples using the proposed model. Blue dots: received AIS messages; red dots: missing AIS messages; red lines: trajectories reconstructed by our model.

6.4.3 Vessel trajectory construction

We deleted a 2-hour segment from each AIS track then used the Vessel trajectory construction layer to reconstruct this segment. The maritime contextual information learnt by the Embedding block gave the model the ability to reconstruct some complex trajectories like those on the top right and bottom left of Fig. 6.3. These constructions can not be achieved by interpolation methods such as linear or spline interpolation.

The performance of this layer depends strongly on the maritime contextual information extracted by the Embedding layer. If the extraction is good, the model can predict complicated patterns like those shown in Fig. 6.3. However, in zones whose the vessel density is low, or in zones where the behaviours of vessels are too complicated for the Embedding layer to learn, the construction layer completely fails to estimate the positions of vessels. In these cases, we use constant velocity method. The switch between particle method and constant velocity method is automatic, because the model knows when the Embedding layer can not extract the maritime contextual information (based on the value of the probability $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1})$).

6.4.4 Abnormal behaviour detection

We divided each dataset into 3 sets: a training set to train the model, a validation set to calculate the the *mean* and *std* of the log probability, and a test set to test the anomaly detection. The proportion of the 3 sets was 60/30/10. Although the training sets were used for learning the normalcy model, we did not do data cleaning, *i.e.* the training sets themselves may contain abnormal trajectories. Our framework relies on probabilistic

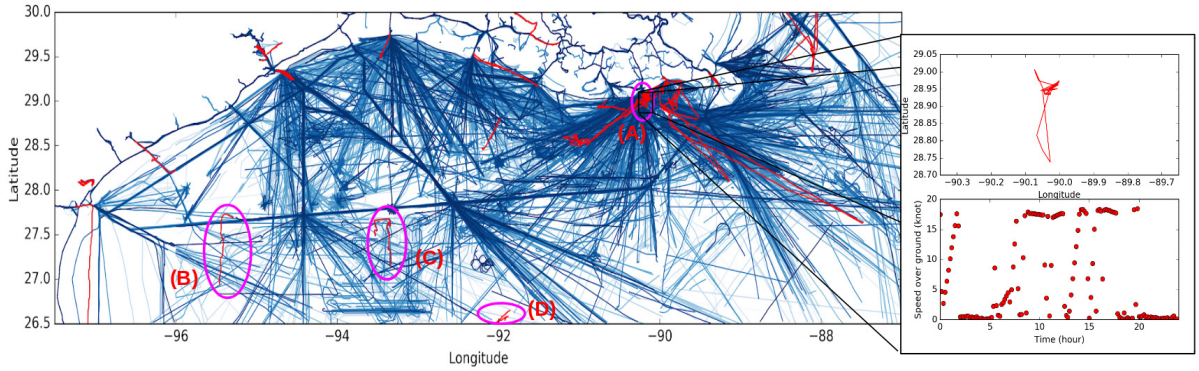


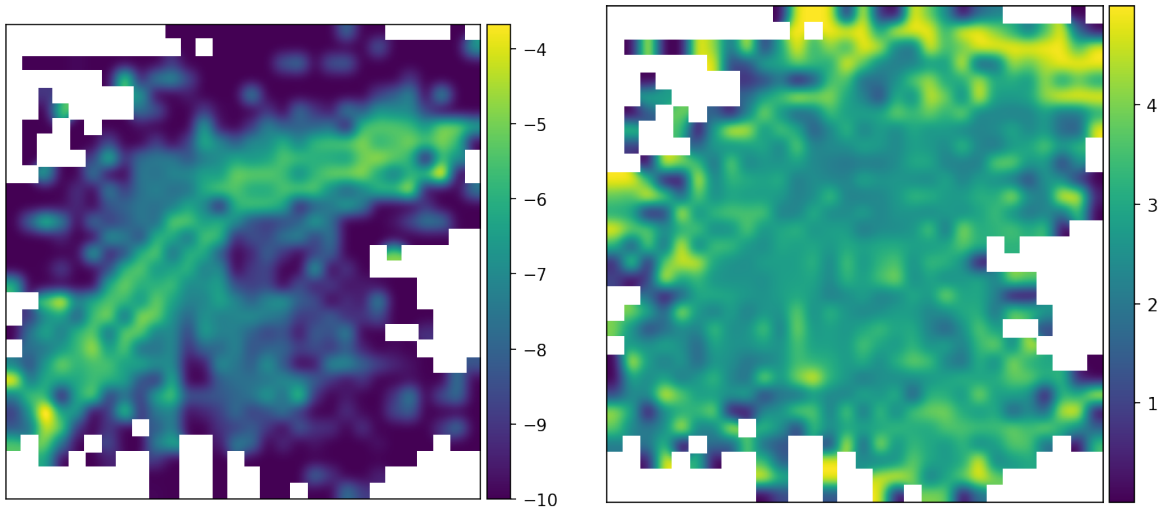
Figure 6.4 – Detection of abnormal behaviours using global thresholding (Gulf of Mexico dataset). Blue: tracks in the training set (which itself may contain abnormal tracks); red: abnormal tracks detected in the test set. We highlight four examples: a track with an abnormal speed pattern **(A)** ii), two tracks with abnormal trajectory shapes from others' in the same region **(B,C)** iii) a track in a low-density area (abnormal zone) **(D)**.

models and implicitly assumes that abnormal trajectories are rare events, that is to say that the probability mass at these trajectories would be very low.

We report the outcome of the anomaly detection submodel when using global threshold detection on the Gulf of Mexico dataset in Fig. 6.4. A track will be detected as abnormal if its shape is unusual, its speed pattern is rare, or it appears in an abnormal region, etc. Each type of these anomalies corresponds to a signature of trajectory data, like geographical pattern, geometric pattern, speed and course distribution, etc. These signatures will be presented in Section 6.5.

For the *a contrario* detection, we split the ROI into small cells of 10kmx10km. The maps of the mean and the standard deviation of the log-likelihood on the Ushant dataset are shown in Fig. 6.5. We can see that the log-likelihood strongly depends on geographical region, global thresholding would not work. On the mean map, there are some lines/curves of high value, they are the maritime routes. On the maritime routes, the vessel density is high, vessels performs simple and similar maneuvers, so the model can learn these patterns easily. On the other hand, in regions where the vessel density is low, or the behaviours of vessels are too complicated, the identification of abnormal behaviours appears more complex and may require larger training datasets. Detection examples and their corresponding interpretation are shown in Fig. 6.6.

To evaluate the consistency of the *a contrario* model, we tested this detector for simulated abnormal examples. We translated a normal track out of maritime routes to simulate the divergence from a given route (zone (A)) and translated circle-shaped tracks in



(a) Mean of the log-likelihood in each cell (b) Std of log-likelihood in each cell

Figure 6.5 – Maps of the mean and the std of the log-likelihood of the trained model in each cell (Brittany dataset).

zone (B) to zone (C) in Fig. 6.7 to verify that some specific patterns of vessels' maneuvers should appear in their specific zones. Experiment shows that the model can detect the divergences if the distance to the maritime route is far enough (10km) and it detects 9 over 13 circle-shaped tracks in zone (C).

In comparison to methods in the literature, our method has several benefits:

— It can detect abnormal patterns that are detected in state-of-the-art methods, such as the double-U-turn detection reported in Fig. 6.6f and also illustrated in (Giuliana Pallotta et al. 2013).

— Methods like those in (Giuliana Pallotta et al. 2013) and (Mascaro et al. 2014) first assign a track to a maritime route, then compare the similarity between this track with the those in the corresponding route to decide whether this track is normal. However, it is very difficult to link tracks like the one in Fig. 6.6d to a maritime route. Therefore, our model which does not require the prior identification of maritime routes appears more generic and robust.

— Our model relaxes strong assumptions. In (Giuliana Pallotta et al. 2013), the authors assumed that the probability of observing a feature vector ($[lon_k, lat_k, SOG_k, COG_k]^T$ in their case) of a vessel at the time k, given its position and assigned route was independent. This assumption neglects the fact that AIS streams provide sequential

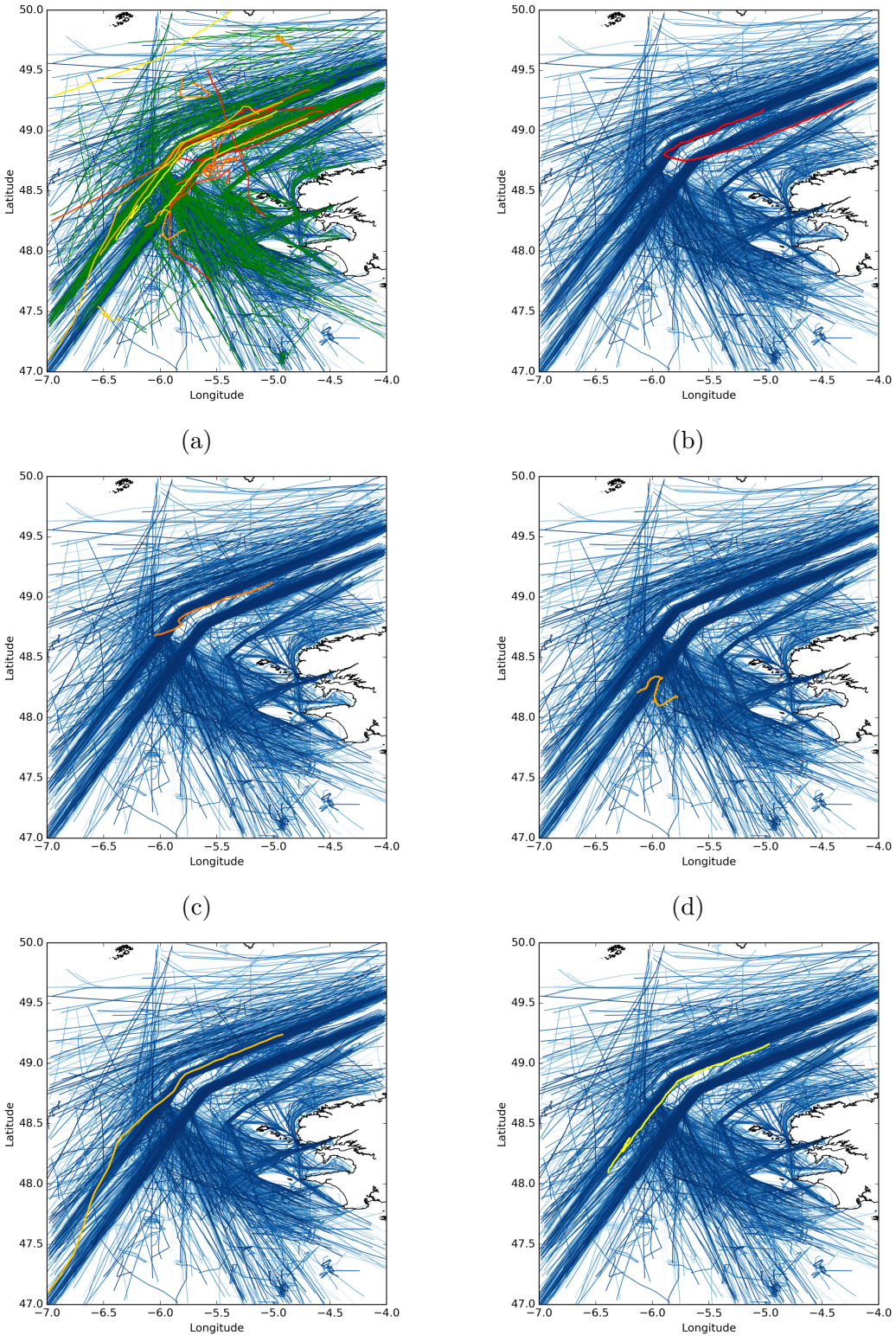


Figure 6.6 – Abnormal tracks detected by the proposed *a contrario* model (Brittany dataset). (a) All tracks detected in the test set; blue: tracks in the training set; green: normal tracks in the test set; other colors: abnormal tracks in the test set. (b) Abnormal U-turn. (c-d) Divergences from maritime route. (e) Abnormal route change. (f) Abnormal double-U-turn.

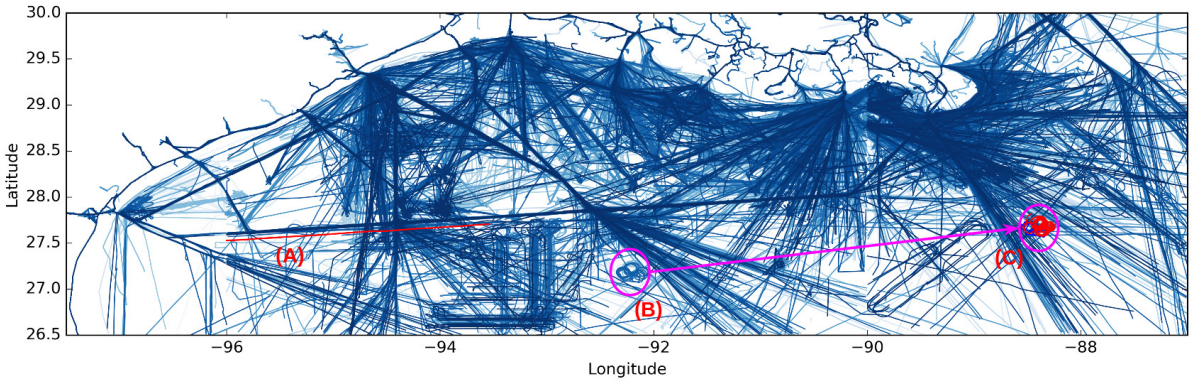


Figure 6.7 – Example of the *a contrario* anomaly detection on simulated dataset (Gulf of Mexico dataset). The circle-shaped tracks in zones **(C)** were simulated by translating from **(B)**; **(A)** is a detection of a divergence from maritime route.

data, feature vectors of a vessel's track are related to this vessel and interdependent. For instance, for such approaches the two branches of the "U" in Fig. 6.6b are normal.

— Methods in the literature do not deal with irregularly time-sampling problem. For example, model in (Giuliana Pallotta et al. 2013) used sliding window to avoid incomplete tracks, and processed only the most recent points of the partially observed tracks. The vessel in Fig. 6.6b can outsmart this model by switching off its AIS transponder when performing the U-turn (which lasts about 30 minutes).

— In a complicated region like the Gulf of Mexico, all the methods based on DBSCAN (F. Mazzarella, V. F. Arguedas, et al. 2015; Giuliana Pallotta et al. 2013) cannot apply since DBSCAN fails to extract effective waypoints. As shown in Fig. 6.7, in this area, vessels do not enter or exit the ROI at some specific zones, in consequence, DBSCAN can not detect entry and exist waypoints; beside that, a lot of vessels stop at sea for purposes (fishing for example), leads to false stationary waypoint detection by DBSCAN.

6.4.5 Vessel type identification

We tested the Vessel type identification submodel with a set of 1800 AIS tracks of 4 types of vessels: cargo, passenger, tanker and tug.

We compared the performance of our model with the one of other types of neural networks: CNNs and LSTMs (which are currently the state-of-the-art for time series classification). To simulate the missing data phenomenon in AIS streams, we deleted a

Table 6.2 – Classification results.

Model	Precision	Recall	F1-score
LSTM	47.51%	64.11%	52.08%
LSTM_4-hot	**88.04%**	87.16%	87.43%
CNN	83.83%	84.06%	83.75%
VRNN-CNN	*88.00%*	***87.67%***	***87.72%***

2-hours segment in each AIS track. Constant velocity model was used to fill the missing points for CNN model. We tested the LSTM networks with and without the "four-hot encoding" layer to show the benefit of this presentation. For each type of architecture we tried several configurations and report the best result.

The results are shown in Table. 6.2. First, the poor performance of LSTMs without the "four-hot encoding" layer shows the relevance of this presentation for disentangling the explanatory information in continuous feature spaces of AIS messages' attributes. Second, we can see that the proposed model achieved comparable performances with those of the state-of-the-art methods. It is because the embedding layer can provide a solid regular series of hidden regimes despite irregular time sampling in AIS streams. In addition to the slight improvement of the classification performance (from 87.43% to 87.72%), the proposed model also significantly reduces storage redundancies and computational requirements when doing each task separately, which is highly beneficial in an AIS big data context.

6.5 Insights on the considered approach

In this section, we further discuss the key features of the considered approach with respect to state-of-the-art approaches. Overall, AIS vessel tracks (and trajectory data in general) may be characterised according to the following features:

— Time evolutions in terms of vessel position, speed and course;

— Geographical patterns (where is the vessel?);

— Geometric patterns (what is the shape of the track?);

— Speed and course distributions;

— Spatial-temporal patterns, called *"phase patterns"* (moves fast in specific zones and slowly in other zones for example).

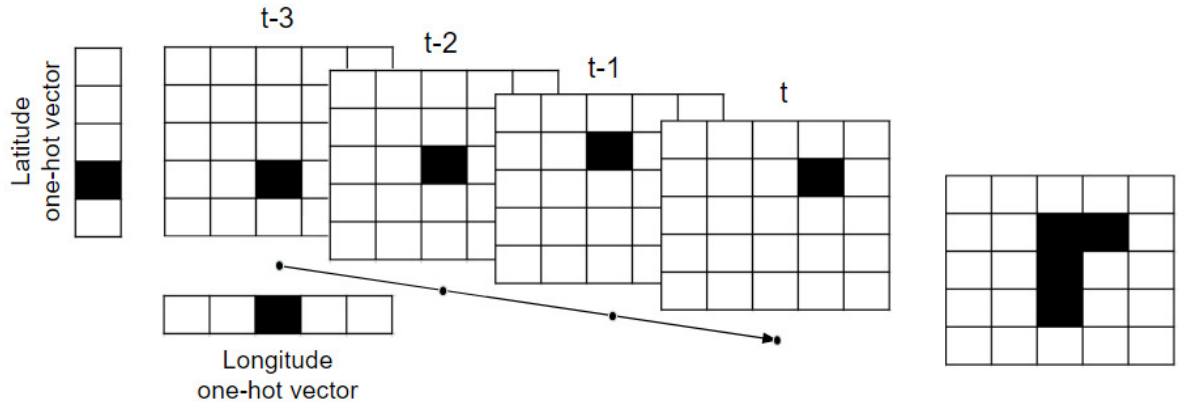


Figure 6.8 – Geometric patterns appear by summing up one-hot vectors of latitude and longitude coordinates.

We discuss below how our approach addresses the learning of these key features.

In time series, different features change at different temporal and spatial scales (Y. Bengio et al. 2013). The proposed model learns these features from different points of view at different scales. At micro-scales, it learns the evolutions of the trajectories, *e.g.* with this historical information, in 10 minutes, vessel "V" seems to appear in zone "Z", maintain its speed around "S" knots. These evolutions are modelled by the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{0:k-1})$. At macro-scales, the model tends to learn the patterns of the entire AIS tracks.

Viewing the "four-hot" representation as an image-based representation of a track seems relevant to understand how our model can learn complex space-time patterns. More precisely, the one-hot vectors of the latitude/longitude coordinates of AIS messages indicate the rows/columns of the pixels in the image, respectively. Hence, if we cumulate these two one-hot vectors over a given time period, we build an image-based representation, which describes the geometric pattern of the vessel track (Fig. 6.8).

The proposed model is not translation-invariant and can learn spatial patterns and the geographical distributions of vessel tracks, *i.e.* a given type of tracks should appear in zone "A" and not in an other zone "B". The phase patterns, on the other hand, reflect the correlations between temporal features and spatial ones. One typical example of phase pattern in trajectory data is the speed-position correlation, *e.g.* the average speed of vehicles on highway is higher than the one in urban area. Methods that use only the positions (longitude and latitude coordinates) to model trajectory and consider the speed as the first-order derivative of the positions can not capture this information. For example,

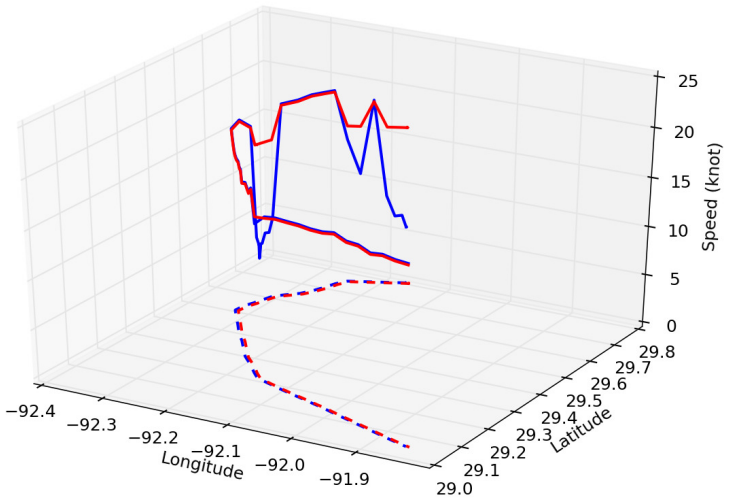


Figure 6.9 – Illustration of phase patterns. We report the two examples of two AIS tracks of the processed dataset (red and blue). The solid lines are 3D curves (latitude, longitude and speed time series) reflect the phase pattern, whereas the dash curves (latitude and longitude time series only) reflect the associated 2D geometric patterns, which can not reveal the observed phase patterns.

the two tracks depicted in Fig. 6.9 are two examples from the processed dataset. They are similar in terms of spatial patterns, but different in terms of phase patterns, space-speed time series. Despite inter-individual variabilities, these two tracks exhibit in some regions low vessel speed and high vessel speed in other regions.

These different aspects are similar to the wave-particle duality in physics, where the patterns correspond to the wave properties and the evolutions correspond to the particle properties.

6.6 Conclusions and perspectives

In this chapter, we proposed a novel deep-learning-based scheme for maritime traffic surveillance from AIS data streams. Stated within a probabilistic framework using Variational RNN, our approach overcomes strong limitations of state-of-the-art methods to jointly address multi-task issues, namely abnormal behaviour detection, trajectory reconstruction and vessel type identification, on a regional scale, that is to say for datasets of spatially-heterogeneous datasets of tens or hundreds of millions of AIS data. More precisely, we tackled three main drawbacks of state-of-the-art approaches:

— First, we relax strong assumptions usually considered such as a finite number of behavioural categories (or hidden regimes) (Holst et al. 2016; Gaspar et al. 2016).

— Second, by using VRNN, we can capture the maritime contextual information while avoiding problems that may be encountered if doing clustering.

— Third, the Embedding block in our model can deal with noise and irregularly time-sampling of AIS data streams. Besides, the Embedding block also results in an efficient compression of the behavioural information conveyed in data, which avoids making accessible the entire training dataset for the operational use of the trained model. This appears critical for an operational big-data-compliant AIS system.

We also discussed the key aspects of the considered trajectory data representation, which is embedded in the considered VRNN framework.

Beyond benchmarking issues for large-scale datasets, including the evaluation of the ability of the proposed approach to scale up to global AIS data streams, the fusion with other sources of information available in the maritime domain could be a promising solution. Weather and ocean conditions, such as sea surface winds and currents, are two important factors that effect the behaviours of vessels. The exploitation of such variables should further constrain the considered VRNN framework and improve its representativity. The inference of behavioural models in low-density areas might require specific investigations in future studies, for instance some type of regularisation.

There are three kinds of lies: lies, damned lies, and statistics.

Mark Twain

Chapter 7

GeoTrackNet

[1] [2] In the previous chapter we presented MultitaskAIS—a multitask NN-based model for maritime traffic surveillance using AIS data. We demonstrated the ability of MultitaskAIS to handle noisy and irregularly sampled data as well as the computational benefit of this architecture for multiple tasks in maritime surveillance. This chapter focuses on detailing the most important task: anomaly detection. This task-specific submodel—referred to as GeoTrackNet—exploits state-of-the-art neural network schemes to learn a probabilistic representation of AIS tracks and *a contrario* detection to detect abnormal events. The neural network provides a new means to capture complex and heterogeneous patterns in vessels' behaviours, while the *a contrario* detector takes into account the fact that the learnt distribution may be location-dependent. Several experiments with different settings

1. This chapter is a modified version of papers (Duong Nguyen, Vadaine, et al. 2019) and (Duong Nguyen, Simonin, et al. 2020)

2. This work was supported by public funds (Ministère de l'Education Nationale, de l'Enseignement Supérieur et de la Recherche, FEDER, Région Bretagne, Conseil Général du Finistère, Brest Métropole) and by Institut Mines Télécom, received in the framework of the VIGISAT program managed by "Groupement Bretagne Télédétection" (BreTel). It benefited from HPC and GPU resources from Azure (Microsoft EU Ocean awards) and from GENCI-IDRIS (Grant 2020-101030). We acknowledge the support of DGA (Direction Générale de l'Armement) and ANR (French Agence Nationale de la Recherche) under reference ANR-16-ASTR-0026 (SESAME initiative), grants Melody and OceaniX. It benefited from HPC and GPU resources from Azure (Microsoft EU Ocean awards) and from GENCI-IDRIS (Grant 2020-101030).

The dataset used in this chapter is provided by Collecte Localisation Satellites (CLS) and Erwan Guegueniat.

We are thankful to Iraklis Varlamis and Mohammad Etemad for enlightening discussions on the implementation of DBSCAN.

on a real AIS dataset demonstrate the relevance of the proposed method compared with state-of-the-art schemes.

7.1 Introduction

In maritime domain awareness, anomaly detection is one of the most important tasks, since anomalies usually involve accidents (loss of navigation, damages in engine, etc.) or illegal activities (smuggling, illegal transshipment, etc.). Initially designed for collision avoidance, the Automatic Identification System (AIS) has quickly become the main source of information for maritime surveillance thanks to its information richness. Roughly speaking, AIS messages contain the identity (the MMSI number), the GPS coordinates (latitude, longitude), the current speed (Speed Over Ground–SOG) and course (Course Over Ground–COG), as well as other information about the vessel and the voyage. A series of AIS messages gives the trajectory of the vessel. The potential of AIS is enormous, however, it is not fully utilised. AIS data are awash in noise, besides that, the massive amount of data quickly overwhelms human processing capacity. This emphasises the need for a system that can automatically analyse and arise an alarm whenever there is an abnormal event. However, since AIS was originally created for collision avoidance only, no metadata (quality, reliability, uncertainty, etc.) are available, making the detection of anomalies from AIS a very difficult task. Morever, AIS data in particular, and trajectory data in general, have some specific characteristics that other types of data do not: geographical features, temporal correlations, geographical-temporal features. For these reasons, anomaly detection methods used in other domains such as network traffic analysis or cybersecurity (Nanduri et al. 2016; Radford et al. 2018) do not apply. We may also emphasise that there are no representative groundtruth datasets for this task, hence, supervised learning strategies for anomaly detection as in (Song et al. 2018; Bouritsas et al. 2019) do not apply either.

Here, we present GeoTrackNet—a new approach for maritime trajectory-based anomaly detection [3] using a probabilistic RNN-based (Recurrent Neural Network) representation of AIS tracks and *a contrario* detection. this chapter is an extended version of our previous work in (Duong Nguyen, Vadaine, et al. 2018). The first step in GeoTrackNet is to build a normalcy model that represents the characteristics of AIS tracks. At sea, either being

3. The detection presented here is trajectory-based, i.e. we focus on the behaviours of vessels. Point-based methods, where the detection is focused on AIS signal, are out of scope of this chapter.

enforced by law or for optimisation issues (e.g. optimal fuel consumption, safety purposes, optimal patterns for fishing, etc.), vessels follow some specific patterns, and we expect to learn these patterns from data (F. Mazzarella, Vespe, et al. 2014; Bomberger et al. 2006; Giuliana Pallotta et al. 2013; V. Fernandez Arguedas et al. 2018; Dobrkovic et al. 2018; Duong Nguyen, Vadaine, et al. 2018). In this work, we exploit variational sequential latent models, specifically the Variational Recurrent Neural Networks (VRNNs) (J. Chung et al. 2015) to create a probabilistic representation of vessels' movement patterns. RNNs have been famous for their ability to capture long-term correlation in time series (here AIS tracks), VRNNs are an extension of RNNs where stochastic factors are added to improve the networks' capacity of modelling the data variations and uncertainties. This architecture is one of the state-of-the-art methods for text, speech and music analysis and generation (J. Chung et al. 2015; Fraccaro et al. 2016; Maddison et al. 2017). Besides the quality of AIS signals, which may depend on the metocean conditions as well as interferences in dense traffic areas, vessel trajectory data may also reflect sea surface and wind conditions. These different sources of variations beyond the behavioural patterns of the vessels make anomaly detection in AIS data streams a particularly challenging task. In this context, VRNNs emerge as a promising candidate for AIS series modelling. In the proposed scheme, given the learnt representation of the movement patterns of vessels, a "geospatial *a contrario*" detector evaluates how likely an AIS track segment is to state the detection of abnormal patterns. This detector exploits a geospatial prior depending on the location-dependent complexity of the patterns observed in the considered dataset. This prior also accounts for the strong geographical variations of vessels' occurrences and movement patterns.

Our contributions are as follows:

— We propose a new representation of AIS messages for deep neural networks. This representation aims to highlight the specific route-related characteristic of trajectory data.

— We propose a new method to build a normalcy model for AIS trajectories. This method relies on VRNNs, which can capture the variations and uncertainties in AIS tracks to create a probabilistic representation of vessels' trajectories. Concretely, this method is the Embedding layer of MultitaskAIS presented in Chapter 6.

— We highlight the fact that vessels' behaviours are geospatially-dependent, hence the model representing AIS trajectories is also geographically-dependent, hence the model representing AIS trajectories hall also be geospatially-dependent. We propose a new anomaly detection method based on this argument.

— We demonstrate the relevance of the proposed scheme with respect to state-of-the-art approaches on a real dataset comprising more than 4.2 million AIS messages.

The paper is organised as follows. In Section 7.2, we give an overview of related work, and analyze the drawbacks of those models. The details of the proposed approach are presented in Section 7.3. Section 7.4 demonstrates the relevance of GeoTrackNet by experiments on real-life data. Conclusions, remaining challenges and future lines of work are discussed in Section 7.5.

7.2 Related work

Recently, there has been a large number of publications related to maritime anomaly detection using AIS. Among them, we can cite (Rhodes et al. 2005; Bomberger et al. 2006; Laxhammar 2008; Ristic et al. 2008; Giuliana Pallotta et al. 2013; Mascaro et al. 2014; d'Afflisio et al. 2018; Kawaguchi 2018; Forti et al. 2019; Varlamis et al. 2019) and references in (Tu et al. 2017; Riveiro et al. 2018). Those methods can be categorised into two groups: rule-based anomaly detection and learning-based anomaly detection.

The former group defines the abnormal behaviours explicitly and uses a set of rules to state the detection. A large list of such rules can be found in (Kazemi et al. 2013). The advantage of this approach is its interpretability. However, it is difficult to define an exhaustive list of abnormal behaviours, and some terminologies such as fast/slow are relative and are hard to implement in operational systems, which may lower their usefulness.

The latter group uses historical data to learn the implicit detection rules. Since no representative groundtruth data are available for maritime anomaly detection, learning-based anomaly detection schemes cannot apply supervised methods like in (Song et al. 2018; Bouritsas et al. 2019). Unsupervised learning methods are then preferred (Rhodes et al. 2005; Bomberger et al. 2006; Giuliana Pallotta et al. 2013; V. Fernandez Arguedas et al. 2018; Forti et al. 2019; Varlamis et al. 2019; L. Zhao et al. 2019). Learning frameworks provide us means to overtake the limitations associated with the definition of an exhaustive list of normal/abnormal behaviours. Given the lack of labeled data for the anomalous class, unsupervised schemes naturally arise as the relevant learning strategies. Due to its flexibility and its ability to apply on a large scale, this second category of approaches has become the dominant approach in maritime anomaly detection (Laxhammar 2008; Giuliana Pallotta et al. 2013; Varlamis et al. 2019; Forti et al. 2019).

Learning-based methods consist of two main stages: i) learning a normalcy model, ii) detecting deviations from the normalcy. In the first stage, density-based spatial clustering techniques, especially DBSCAN (Ester et al. 1996), have been very popular (Giuliana Pallotta et al. 2013; Coscia et al. 2018; d'Afflisio et al. 2018; Varlamis et al. 2019). Typically, DBSCAN is applied to cluster the critical points of AIS tracks into so-called Waypoints (WPs): ENs—where vessels enter the Region of Interest (ROI), EXs—where vessels exit the ROI, and POs—where vessels stop. From these WPs, these approaches build a graph whose nodes are the WPs and edges are the maritime routes. Using a probabilistic setting, e.g., Kernel Density Estimation (KDE) (Giuliana Pallotta et al. 2013), Gaussian Mixture Models (GMM) (Laxhammar 2008), multiple Ornstein-Uhlenbeck (OU) processes (Forti et al. 2019), a normalcy model is fitted for each edge. The next stage aims to evaluate how likely a new AIS track is in order to state the detection of abnormal tracks. This is typically achieved by applying a threshold on the distance to the centroid feature vector representing the route (Varlamis et al. 2019) or on the probability of the AIS track given the normalcy model (Giuliana Pallotta et al. 2013), or through an adaptive hybrid Bernoulli filter (Forti et al. 2019).

In all of the above mentioned methods, the extraction of WPs is critical. However, the considered clustering techniques, such as DBSCAN, may be sensitive to hyper-parameters. Different settings may lead to very different outcomes. Moreover, it is not always possible to link a track to an edge of the normalcy graph, i.e. we can not assign the beginning point and the end point of a track to any WP. This is a common problem of any method based on a clustering step. Another important limitation of the above mentioned approaches is that they apply to cargo and tanker vessels, and may not apply to other vessel types, for instance, fishing vessels whose AIS patterns do not involve route-like patterns. As AIS metadata may not be reliable, dealing with all vessel types in operational systems would require additional preprocessing steps to filter out vessels' types.

Although over the last decade, deep learning has achieved very impressive results in many complicated tasks and has become the state-of-the-art approach in many domains (LeCun et al. 2015), AIS-based maritime surveillance is not one of them. Popular network architectures for time series modelling and analysis such as Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), etc. may hardly model the dynamics of AIS trajectories because the data are noisy and may be effected by external factors (e.g. metocean conditions). Another issue is that those methods assume the performance of the learnt normalcy model is geospatially-homogeneous. However, in some areas, there

are a lot of vessels and their behaviours are similar, the maneuvering patterns in these areas can be learnt easily. By contrast, other areas may involve much less training data and/or highly-complex and multi-modal patterns, which result in poor performance of clustering-based normalcy models and of the associated anomaly detection schemes. The application of the same anomaly detection policy (threshold, filter) in these two types of areas does not seem relevant. Another important limitation of the above mentioned approaches is that they apply to cargo and tanker vessels but may not apply to other vessel types, for instance, fishing vessels whose AIS patterns do not involve route-like patterns. As AIS metadata may not be reliable, dealing with all vessel types in operational systems would require additional preprocessing steps to filter out vessels' types.

In this chapter, we present a new method, referred to as GeoTrackNet that tackles those problems by exploiting advances in probabilistic neural network representations for time series analysis and an *a contrario* detection framework for maritime anomaly detection from AIS data streams. Our method provides a new means to address key issues of state-of-the-art approaches, both in terms of the extraction and representation of the normalcy and of the detection of deviations from the normalcy for all types of vessels.

7.3 Proposed Approach

In this section, we present the details of the proposed approach. GeoTrackNet relies on the architecture of the Embedding layer we introduced for the MultitaskAIS network presented in (Duong Nguyen, Vadaine, et al. 2018). We first introduce this architecture, then detail the formulation of the proposed anomaly detection method.

7.3.1 Data representation

As discussed in the previous chapter, the most common way to represent an AIS message is using a 4-D real-valued vector (two dimensions for the position and the other two for the velocity, e.g. $[lat, lon, SOG, COG]^T$) (Giuliana Pallotta et al. 2013; d'Afflisio et al. 2018; Forti et al. 2019; Üney et al. 2019). We argue that this representation is not suitable for neural-network-based methods, because it is difficult for a neural network to disentangle the underlying geospatial meaning of these numbers. Instead, we represent each AIS point by a "four-hot" vector (Section 6.3.2). A "four-hot" representation is a concatenated vector of the one-hot vectors of the latitude coordinate, longitude coordinate,

SOG and COG.

In addition to the classically-expected benefits of bucketing representation (Y. Bengio et al. 2013), the "four-hot" vectors help disentangle the geometric features as well as the phase (time-space) patterns of AIS tracks. For example, Fig. 6.8 shows how this representation accentuates the geometric feature of an AIS track. Similarly, the phase feature appears when we sum up the one-hot vectors of the latitude, longitude coordinate and the speed in the resulting 3-D space (see Section 6.3.2). We also expect that during the learning process, the "four-hot" representation enforces route-related characteristics of trajectory data in general, and of AIS data in particular. More precisely, the model shall learn that some vessels should follow some specific routes, and hence detects as abnormal any vessel deviating from the maritime route that it is on. As an illustration, Fig. 7.1 shows how the "four-hot" representation can help the model detect abnormal movements deviating from maritime routes.

The hyper-parameters are the resolution of each bin in the one-hot vectors. If the resolution is too high, the whole network becomes too bulky and requires a high computational resource to run, and may also lead to overfitting. If the resolution is too low, we may lose critical information. For anomaly detection, we may not need very accurate position and velocity features. For example, a speed of 10 knots or 10.1 knots is not expected to make any difference in the context of anomaly detection. Overall, our experiments suggest that the resolutions of 0.01° for longitude and latitude, 1 knot for SOG and 5° for COG work well most of the time.

7.3.2 Probabilistic Recurrent Neural Network Representation of AIS Tracks

In this section we summary the probabilistic neural network representation of AIS tracks presented in Section 6.3.1. However, we use a different derivation which would clarify some terms used in the next sections of this chapter.

For any contiguous AIS track [4], we can always apply an interpolation and sampling technique to create a sequence of T variables: $\mathbf{x}_{0:T} = \{\mathbf{x}_k\}_{,k=0:T}$, with $\mathbf{x}_k$ is the "four-hot" vector representation of AIS messages presented in Section 7.3.1. The objective is to learn

4. A contiguous AIS track is a track whose the time gap between any two successive messages is smaller than a threshold, here 2h.

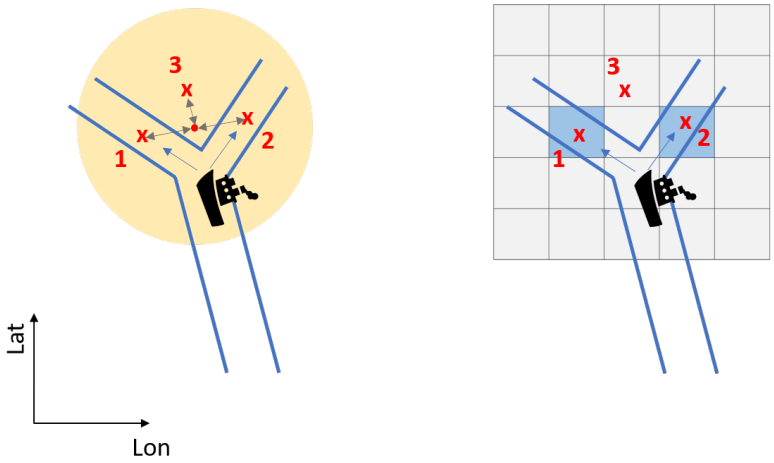


Figure 7.1 – Continuous real-valued representation (left) vs. "four-hot" representation (right) of AIS messages in the considered learning-based setting. For the sake of simplicity, SOG and COG are not considered here. Assume that there is a maritime route (depicted by blue lines), and at the junction, half of the vessels in the historical dataset turned left (to position $\mathbf{x}^1$) and half turned right (to position $\mathbf{x}^2$), but none of them went straight ahead (to position $\mathbf{x}^3$). ***Left***: If vessel positions are represented by real-valued vectors and the dynamics of vessels are modeled by Gaussian distributions, at the next timestep, the abnormal position $\mathbf{x}^3$ would yield a better score than the actual normal positions $\mathbf{x}^1$ and $\mathbf{x}^2$, because $\mathbf{x}^3$ is closer to the red dot—the center of the Gaussian distribution (depicted by the yellow circle). ***Right***: If vessels positions are represented by "four-hot" vectors and the dynamics of vessels are modeled by multivariate Bernoulli distributions, at the next timestep, the model would give higher probability values to the two blue "'bins" only, and position $\mathbf{x}^3$ would be very unlikely compared with positions $\mathbf{x}^1$ and $\mathbf{x}^2$.

a distribution that maximise the log likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$, which can factorise as:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}_0) \sum_{k=0}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}). \tag{7.1}$$

Recently, time series modelling and analysis has experienced the emergence of Recurrent Neural Networks (RNNs) as the state-of-the-art approach in many tasks (LeCun et al. 2015; I. Goodfellow, Yoshua Bengio, et al. 2016). RNNs assume that at a given time k, the relevant historical information of $\mathbf{x}_{0:t-1}$ can be encoded in a deterministic hidden state $\mathbf{h}_k$: $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:t-1}) = p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$. The dynamics of the series are modelled by a deterministic differentiable function f: $\mathbf{h}_k = f(\mathbf{x}_{k-1}, \mathbf{h}_{k-1})$. f is usually parameterised by LSTMs (Hochreiter et al. 1997) or GRUs (Junyoung Chung et al. 2015). The initial condition $\mathbf{h}_0$ is commonly set to $\mathbf{0}$. Eq. (7.1) becomes:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = \sum_{k=0}^{T} \log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k). \tag{7.2}$$

The fact that f is deterministic makes RNNs hardly capable of capturing all the variabilities and uncertainties in data. In our context, f can be interpreted as a model of the maneuvering patterns of vessels from AIS tracks. Associated uncertainties may come from AIS data streams themselves as well as their discretisation using "four-hot" vectors. Variations in AIS data streams may relate to vessel types, weather conditions, AIS message corruption, etc.

To account for such uncertainties, probabilistic RNNs relate to the introduction of latent stochastic variables, denoted as $\mathbf{z}_k$, which follow a prior distribution:

$$\mathbf{z}_k \sim p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k). \tag{7.3}$$

The dynamics and the emission distribution become:

$$\mathbf{h}_k = f(\mathbf{x}_{k-1}, \mathbf{z}_{k-1}, \mathbf{h}_{k-1}), \tag{7.4}$$

$$\mathbf{x}_k \sim p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{h}_k). \tag{7.5}$$

At each time step k, the joint probability of $\mathbf{x}_k$ and $\mathbf{z}_k$ can factorise as:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_k, \mathbf{z}_k|\mathbf{h}_k) = p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k, \mathbf{h}_k) p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k). \tag{7.6}$$

Hence, $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ can be obtained by integrating out $\mathbf{z}_k$ from Eq. (7.6):

$$p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k) = \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_k,\mathbf{h}_k)}\left[p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k,\mathbf{h}_k)p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k)\right]. \tag{7.7}$$

As discussed in Chapter 2, this integral is usually intractable. Variational approaches propose that instead of maximising $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$, we maximise a lower bound of this distribution, called the Evidence Lower BOund (ELBO), by using an approximation $q(\mathbf{z}_k|\mathbf{x}_k,\mathbf{h}_k)$ of the true posterior distribution $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{x}_k,\mathbf{h}_k)$ (J. Chung et al. 2015; Bishop 2006):

$$\mathcal{L}(\mathbf{x}_k|\mathbf{h}_k,p_{\boldsymbol{\theta}},q_{\boldsymbol{\phi}}) = \mathbb{E}_{q(\mathbf{z}_k|\mathbf{x}_k,\mathbf{h}_k)}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k,\mathbf{h}_k)\right] - \mathrm{KL}\left[q(\mathbf{z}_k|\mathbf{x}_k,\mathbf{h}_k)||p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k)\right]. \tag{7.8}$$

Overall, given the neural network parametrisation for function f, the emission distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k,\mathbf{h}_k)$ and the approximated posterior distribution $q(\mathbf{z}_k|\mathbf{x}_k,\mathbf{h}_k)$, the training step comes to maximise Eq. (7.2) where the term $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ is approximated by $\mathcal{L}(\mathbf{x}_k|\mathbf{h}_k,p_{\boldsymbol{\theta}},q_{\boldsymbol{\phi}})$. This maximisation is implemented using a stochastic gradient ascent technique. The details of the considered neural network parametrisations for the different building blocks of the model (using LSTMs) are presented in Section 7.4.

7.3.3 *A contrario* detection

Once distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$ is learnt, we can simply apply a "global thresholding" rule to state the detection, i.e. AIS tracks whose $\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) < \varepsilon$ are flagged as abnormal, like in our work in Appendix A . However, as discussed in Section 6.3.5, vessels' behaviours vary significantly depending on the considered geographical areas. In some areas, AIS tracks may involve multimodal but well-defined patterns and the learnt model can precisely captures these patterns. As a result, normal AIS tracks shall be associated with high probability values, whereas tracks will low probability values shall relate to unusual and possibly abnormal ones. In other areas, because of the variabilities of vessels' behaviours, limited amount of AIS data and/or a lower capacity of the model to represent AIS tracks, the learnt model may result in low probability values whatever the tracks. In such cases, the use of a global thresholding approach might lead to poorly relevant detection results.

To address these issues, we introduce a new detection method, referred to as "geospatial *a contrario*" detection. It takes into account the geospatially-heterogeneous performance of the learnt model. We rely on the division of the ROI into a grid. Let us denote by $l^{C_i}_{\mathbf{x}_k}$ the

log probability $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ of AIS messages in a small geographical cell C_i (i.e., $\mathbf{x}_k \in C_i$) and p^{C_i} the distribution of $l^{C_i}_{\mathbf{x}_k}$:

$$l^{C_i}_{\mathbf{x}_k} \sim p^{C_i}. \tag{7.9}$$

An AIS message in cell C_i is considered as abnormal if its log probability is smaller than the lowest $\frac{1}{p}$-quantile of p^{C_i}.

$$\mathbf{x}_k \textit{ is abnormal} \Leftrightarrow p^{C_i}(\mathrm{L} < l^{C_i}_{\mathbf{x}_k}) < p. \tag{7.10}$$

That means, if we randomly sample $l^{C_i}_{\mathbf{x}_k}$ from p^{C_i} (note that p^{C_i} is the distribution of variable $l^{C_i}_{\mathbf{x}_k}$, and not $\mathbf{x}_k$), the probability that "$\mathbf{x}_k$ is abnormal" is p.

Assuming that the event "$\mathbf{x}_k$ is abnormal" of each AIS message $\mathbf{x}_k$ in an AIS track $\mathbf{x}_{0:T}$ is independent, the probability that "at least k out of n AIS messages in an AIS segment of length n (denoted $\mathbf{x}_{k:k+n}$) of this track are abnormal" is a tail of a Binomial distribution:

$$\mathcal{B}(n,k,p) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \tag{7.11}$$

The *a contrario* detection (Desolneux et al. 2008) detects whether such an AIS segment is abnormal based on the Number of False Alarms (NFA), defined as:

$$\mathrm{NFA}(n,k,p) = N_s \mathcal{B}(n,k,p), \tag{7.12}$$

where $N_s = \frac{T(T+1)}{2}$ is the number of all possible segments. For example, if $T = 3$, there are 6 possible segments: 3 segments of length 1, 2 segments of length 2 and 1 segment of length 3. If the NFA of a track segment is smaller than a predefined threshold ε, this segment will be considered as abnormal and an AIS track is abnormal if at least one of its segment is abnormal.

$$\mathbf{x}_{0:T} \textit{ is abnormal.} \Leftrightarrow \exists (n,k), \mathrm{NFA}(n,k,p) < \varepsilon. \tag{7.13}$$

The threshold ε is the allowed expectation of "false alarm", that means, i.e., if we run the detector on a series of random $l^{C_i}_{\mathbf{x}_k}$ $1/\varepsilon$ times, there will be 1 segment flagged as abnormal. Interested readers are referred to (Desolneux et al. 2008) for more details. To implement this *a contrario* scheme, we use two approaches to model distribution p^{C_i}: i) a simple Gaussian approximation and ii) a Kernel Density Estimation (KDE) (Rosenblatt 1956),(Parzen 1962).

7.4 Experiments and results

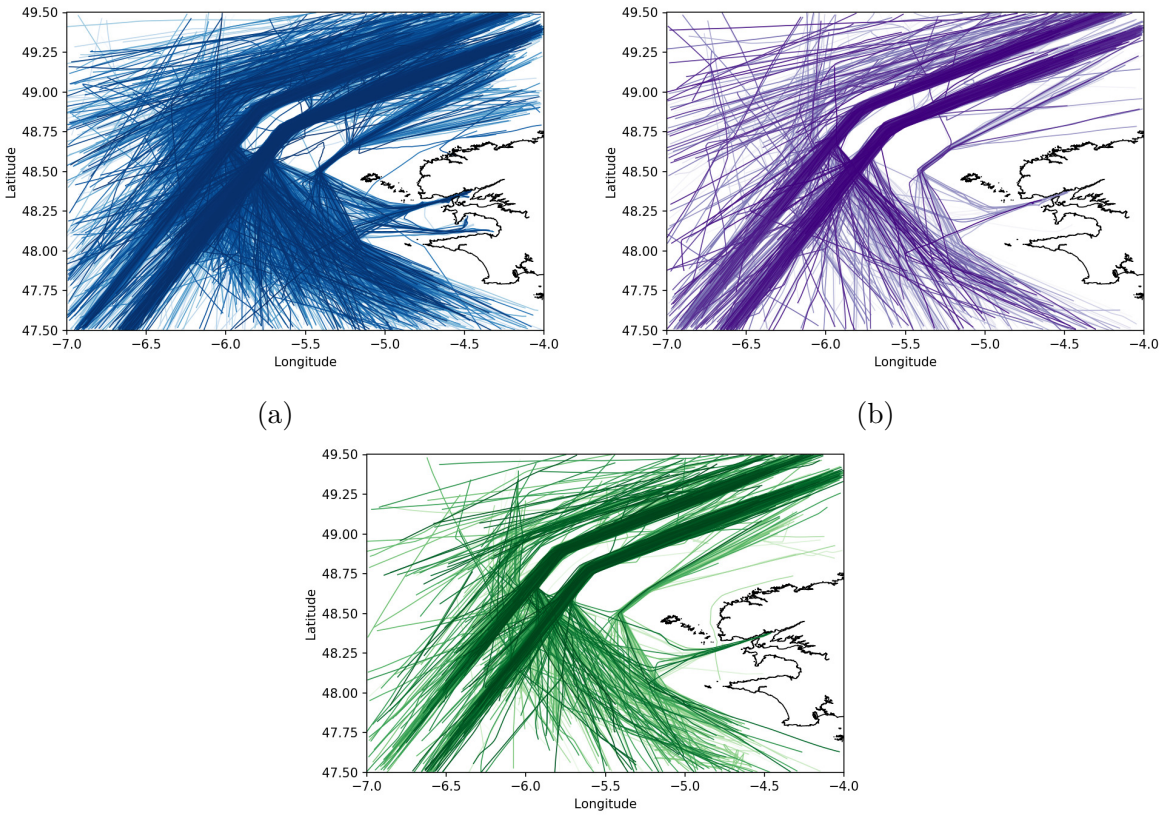
7.4.1 Experimental set-up

Datasets: We tested our model on AIS data received by an AIS station located in Ushant. The ROI was a rectangle from (47.5°N, 7.0°W) to (49.5°N, 4.0°W). The data were collected from January to March 2017 and from July to September 2017. In each period, there are more than 4.2 million AIS messages. For each period, we divided the data into three sets: a training set, from the first day to the 10th of the last month of this period (e.g. from January 1 to March 10); a validation set, from the 11th of the last month to the 20th of the last month (e.g. from March 11 to March 20) and a test set, from the 21st of the last month to the last day of this period (e.g. from March 20 to March 31). The basic idea behind this experimental setting is that for an operational application, we use historical data to train the model (i.e. to learn $p(\mathbf{x}_{1:T})$), then apply this model to current data. The validation sets are used to check for overfitting and for the estimation of distribution p^{C_i}. Fig. 7.2 shows an illustration of the training set, the validation set and the test set of the period from January to March 2017.

Preprocessing: *GeoTrackNet* can process AIS streams in real-time. In real-time operational applications, whenever an AIS message arrives, it will be grouped into a track keyed by the MMSI. The detection starts if the track is long enough to be meaningful, here greater or equal to 4 hours. The system incrementally updates the tracks by adding arriving AIS messages and discarding old data. The implementation and the performance of the online detection version of *GeoTrackNet* can be found in (Duong Nguyen, Simonin, et al. 2020). Those technical details are out of scope of this chapter. Here, for the sake of simplicity, we present the offline version of *GeoTrackNet*.

We removed erroneous position or speed messages in the considered AIS data streams. The SOG was truncated to 30 knots. Discontiguous voyages (voyages that have the maximum interval between two successive AIS messages longer than a threshold, here 2 hours) were split into contiguous ones. We re-sampled all voyages to a resolution of 10 minutes (i.e. , $\{k+1\} - \{k\} = 10\text{mins}$) using a linear interpolation. Very long voyages were split into smaller tracks from 4 to 24 hours each.

Neural Network architectures: for the model reported in this chapter, the resolutions of the latitude, longitude, SOG and COG were set to 0.01°(about 1km), 0.01°, 1 knot and 5°, respectively. We modelled f by a LSTM with one single hidden layer of size 100 for datasets comprising only cargo and tanker vessels, and of size 120 for datasets comprising



(a) (b)

(c)

Figure 7.2 – All AIS tracks in the dataset from January 1 to March 31, 2017. (a) training set; (b) validation set; (c) test set.

all types of vessels. $\mathbf{z}_k$ was real-valued vectors of the same size of the hidden layer of the LSTM. $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{h}_k)$ and $q(\mathbf{z}_k|\mathbf{x}_k, \mathbf{h}_k)$ were two Gaussian distributions parameterised by two fully connected networks with one hidden layer of size 100. $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k, \mathbf{z}_k)$ is a multivariate Bernoulli distribution parameterised by a fully connected network with one hidden layer of size 100. The network was trained using Adam optimiser (Diederik P. Kingma and Ba 2015) with a learning rate of 0.0003.

***A contrario* detection**: for the *a contrario* detector, we chose $p = 0.1$. ε was initially set at a high value (in order to flag many tracks as abnormal), then was gradually decreased to reduce the number of false positives while keeping all the true detections.

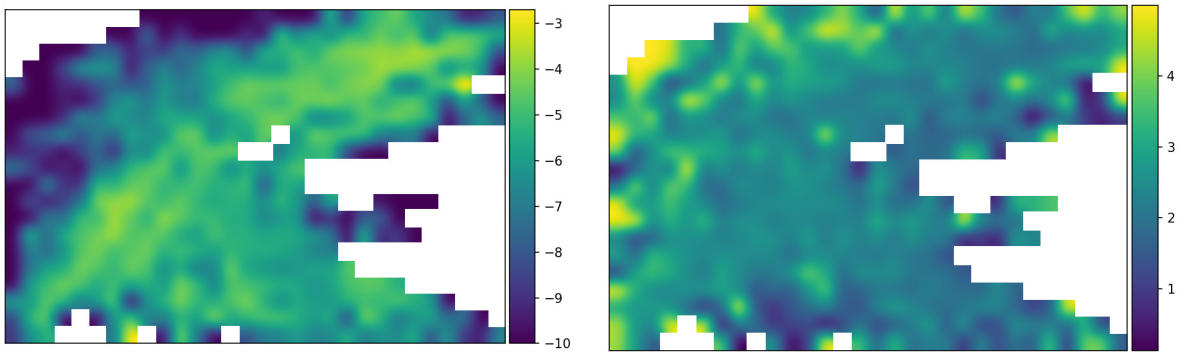
The code, as well as the data that can replicate the results in this chapter are available at: https://github.com/CIA-Oceanix/GeoTrackNet

Baseline: We used the Traffic Route Extraction and Anomaly Detection (TREAD) method, presented in (Giuliana Pallotta et al. 2013; V. Fernandez Arguedas et al. 2018) as the baseline. This model supposes that vessels following the same route have similar velocity in each small area. The hyper-parameters were set at the values suggested by (Giuliana Pallotta et al. 2013) and (Varlamis et al. 2019) ($minPts = 10$, $eps = 2000$, the radius of each small area is 3km). We also included state-of-the-art NN models for sequential data, namely LSTMs (Marchi, Vesperini, Eyben, et al. 2015; Marchi, Vesperini, Weninger, et al. 2015) and VRNNs (D. Nguyen et al. 2019; Su et al. 2019)

Evaluation method: As no reference groundtruth dataset is available, a quantitative benchmarking synthesis in terms of accuracy or false alarm rate is not feasible. We rather analyse the different types of anomalies identified by different models. Besides, a more thorough analysis has been performed for *GeoTrackNet* through an inspection of each detected anomaly by AIS experts.

7.4.2 Experiments and results

Basic case study: For this test, we trained the model on the training set and evaluated the performance on the corresponding test set of each period. The dataset comprises only cargoes and tankers. Fig. 7.3 shows the mean and the standard deviation of distributions p^{C_i}. As expected, in some regions, there are many vessels and the learnt model fits well the data with a mono-modal or multimodal distribution, such that the values of $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ are high. There are also regions where $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ is low on average. If an AIS track results in a low log probability in these regions, we do not know whether this track is unusual or the model does not fit well the data. Applying a "global thresholding" rule like



(a) (b)

Figure 7.3 – The "geospatial performance" map displaying the mean (a) and the standard deviation (b) of the Gaussian approximation of distributions p^{C_i} from AIS messages in the validation set from January to March, 2017. On maritime routes, there are many vessels, mainly cargoes and tankers, their movement patterns can be learnt easily, $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ is usually high and its variation is small. On the other hand, some areas depict few vessels or vessels' behaviours are too complicated for the model to learn, $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ is usually low and highly variable. Blank regions are regions where we do not apply the detection (e.g., land areas or regions where we do not have enough data).

in (D. Nguyen et al. 2019) would lead to a bad outcome, as shown in Fig. 7.4d, where all the detections are in low log likelihood regions. By contrast, the proposed *a contrario* detector compares $\log p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k)$ of an AIS message $\mathbf{x}_k$ with those in the same area, if it is significantly smaller than the others, then $\mathbf{x}_k$ is regarded as abnormal. The results are shown in Fig. 7.4. Most of the time, the model using Gaussian distribution approximation and the one using KDE gives similar outcomes. The proposed model can detect both: i) space-wise (geometric and geographic) anomalies, when vessels deviate from maritime routes, perform unusual turns, etc. and ii) phase-wise (kinetic) anomalies, when vessels have abnormal evolution in speed and course (e.g. unusual slowing down, sudden changes in speed, etc.). as shown in Fig. 7.5. Among those 25 tracks flagged as abnormal in Fig. 7.4f, AIS experts reported only one (the dark yellow track turning north at (49°N, 5°W)) as a false alarm. We suspect this detection relates to the low number of AIS tracks in this area in the training set as this area is outside of the coverage zone of the terrestrial AIS station located in Ushant. Additional experiments reported in (Duong Nguyen, Simonin, et al. 2020) support this statement as the model trained with a larger training set (comprising both terrestrial AIS and satellite AIS) does not flag this AIS track as abnormal.

Regarding LSTM and VRNN models (Fig. 7.4a and Fig. 7.4b, respectively), the

performance does not appear very relevant. They flag many normal tracks as abnormal. For example, in both figures, the tracks along the 6.0°W longitude line are usual tracks. In Fig. 7.4b, the yellow, orange and red tracks departing from Brest (48.4°N, 4.5°W) are normal tracks (except the red track in Fig. 7.5e).

When comparing our approach to TREAD (Giuliana Pallotta et al. 2013), we note that some types of anomaly are detected by both approaches, like the double U-turn, abnormal turns, or abnormal speeds, as shown in Fig. 7.4c and Fig. 7.4f. Since TREAD compares the velocity of a vessel with the average of vessels on the same route to state the detection, this method is sensitive to vessels' speed. TREAD considers all vessels that move slower or faster than others as abnormal. This may lead to some unwanted results, when the statistical anomaly is not suspicious, like the one in Fig. 7.6a. This vessel was flagged by TREAD because it moved too fast. However, it may not involve any suspicious activity. On the other hand, *GeoTrackNet* focuses more on sudden changes in speed of vessels, see Fig. 7.6d for an example. This detection is relevant because this vessel my encounter an engine failure.

The detection of abnormal tracks which do not follow any maritime route like those in Fig. 7.5a and Fig. 7.5e is a key advantage of *GeoTrackNet* over DBSCAN-based models. Because those tracks can not be mapped to any maritime route, DBSCAN-based methods have two options, either flag all of them as abnormal or do not monitor them. Since the number of those tracks is high, typically from 10% to 60% of the total tracks in the ROI, (Giuliana Pallotta et al. 2013) (see Fig. 7.7), neither of these options is relevant for maritime surveillance.

Relevance of the "four-hot" representation: to demonstrate the relevance of the "four-hot" representation, we tested the proposed model without the "four-hot" representation. The result is shown in Fig. 7.8a. The model fails to detect small, yet very unusual deviations from the common behaviours, such as the double U-turn in Fig. 7.5d, or the abnormal turns of the red track in Fig. 7.5b. We also tested *GeoTrackNet* with different resolutions of the "four-hot vector". In general, *GeoTrackNet* is relatively robust to the considered resolutions for the latitude, longitude, SOG and COG. The performance of the model was consistent when we increased or decreased the resolutions of the latitude and the longitude by a factor of 2. When we increased or reduced those resolutions by a factor of 5, the detection started changing. The results with those settings are shown in Fig. 7.8b and Fig. 7.8c. When the resolution is too fine, the amount of information that the model has to learn is too much. For example, a spatial resolution of 0.002° means that the model

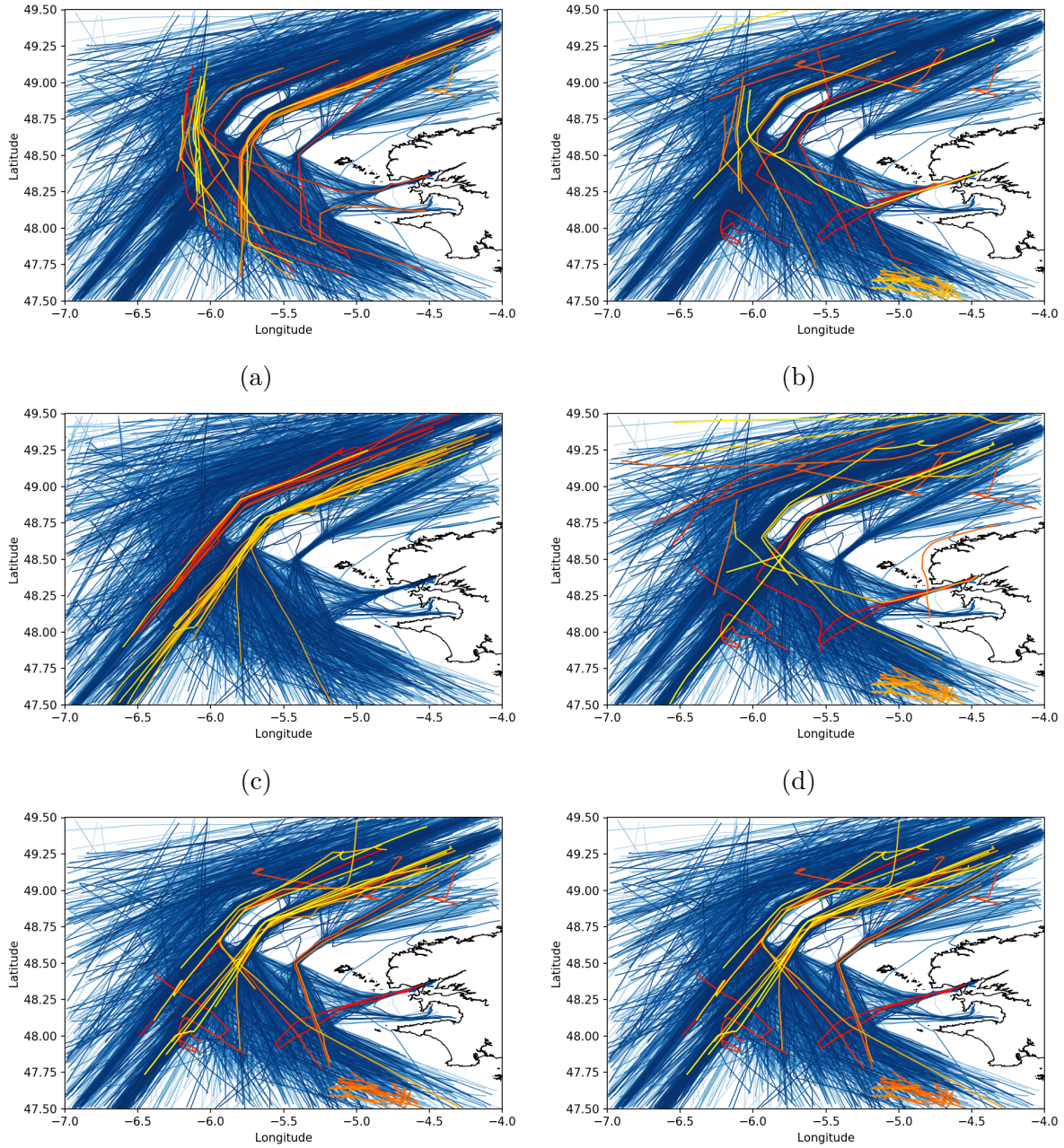


Figure 7.4 – Abnormal tracks detected by different models (the dataset comprises only cargo and tanker vessels, from January to March 2017). Blue: tracks in the training set; other colors: abnormal tracks in the test set (the colors of abnormal tracks were chosen randomly). (a) LSTM; (b) VRNN, (c) TREAD (a DBSCAN-based method introduced in (Giuliana Pallotta et al. 2013));. (d) *GeoTrackNet* without the *a contrario* detector (i.e. using a "global thresholding" rule); (e) *GeoTrackNet*, approximating each p^{C_i} by a Gaussian distribution; and (f) *GeoTrackNet*, approximating each p^{C_i} by KDE.

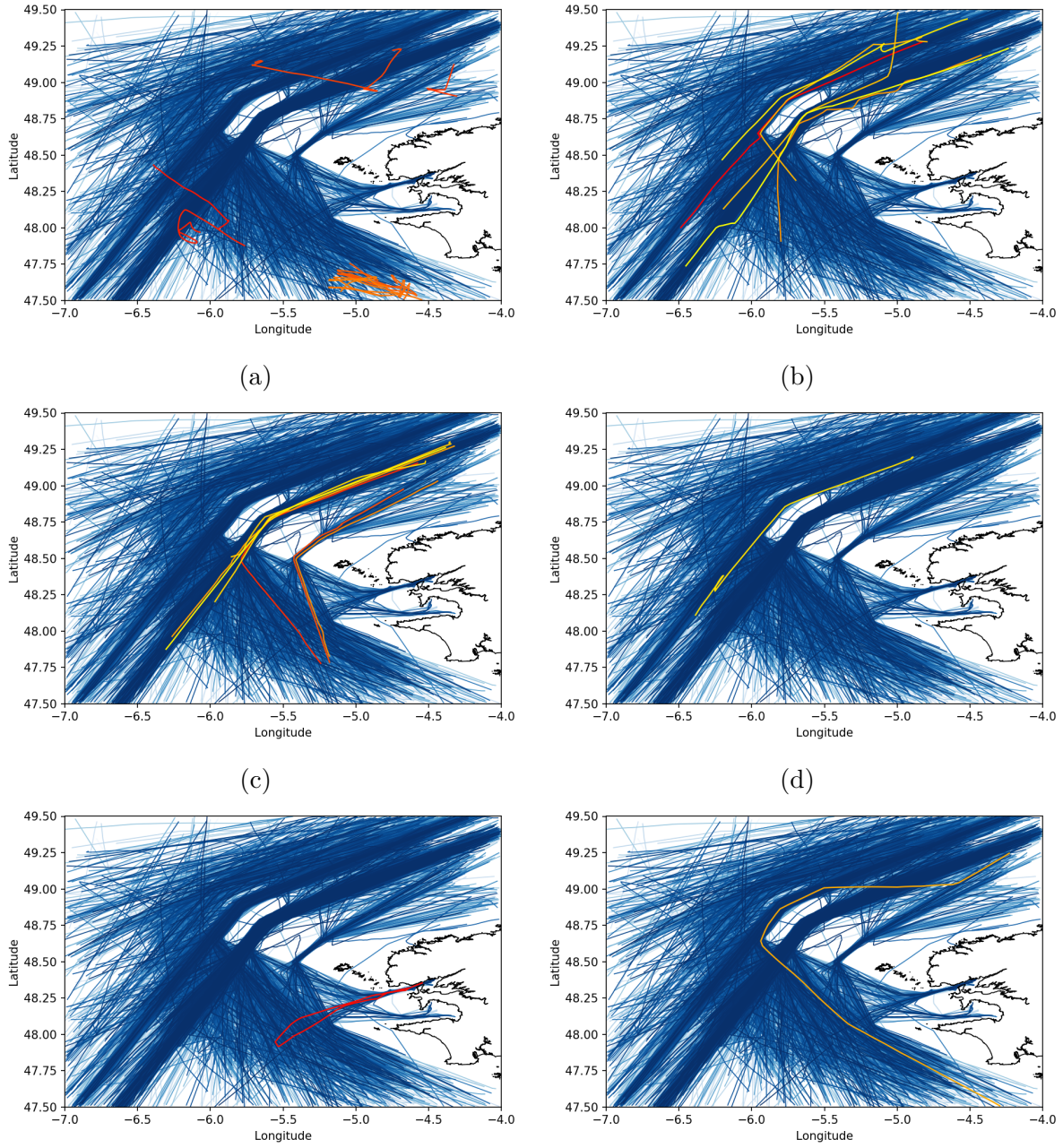


Figure 7.5 – Examples of anomalies detected by KDE *GeoTrackNet*. (a) Vessels following abnormal routes. DBSCAN-based methods can not apply to these tracks because they can not be assigned to any common maritime route. (b) Geometrically or geographically abnormal tracks (e.g., deviating from maritime routes, unusual turns, etc.). (c) Abnormal speed tracks (e.g. suspiciously slowing down in a maritime route). (d) Double U-turns. (e) A cargo vessel steamed to sea then went back to the departing port. (f) Each segment of this track is normal, however, it is unusual that a vessel follows this path. *GeoTrackNet* can detect this track because it has a memory (the memory of its LTSM).

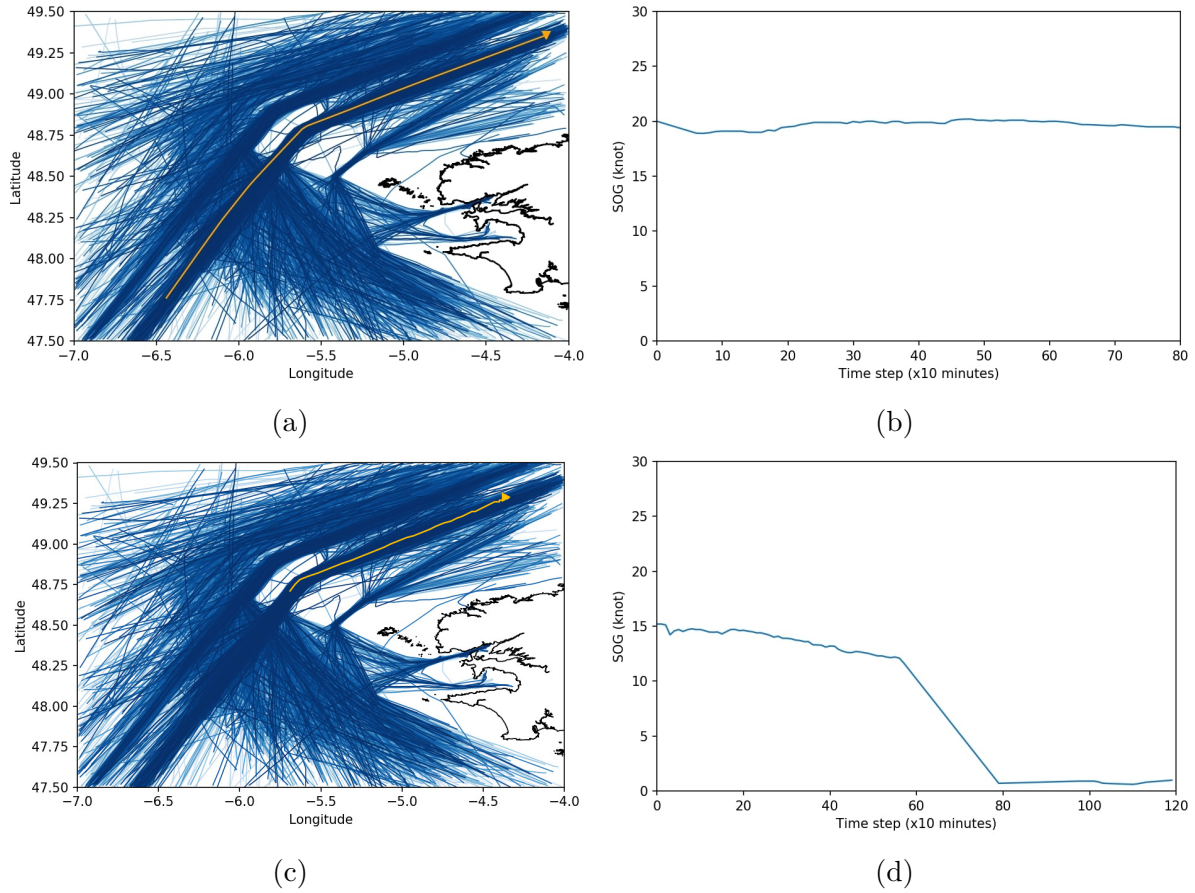


Figure 7.6 – Examples of tracks with abnormal speed patterns detected by TREAD and *GeoTrackNet*. (a) An example of a track flagged as abnormal by TREAD and the associated speed pattern (b). The speed of vessels along this route typically varies between 10 and 18 knots while this vessel was moving at around 19 to 20 knots. (c) An example of a track flagged as abnormal by KDE *GeoTrackNet* and the associated speed pattern (d). It involves a sudden slowing-down which may relate to engine problems or abnormal sea/traffic conditions.

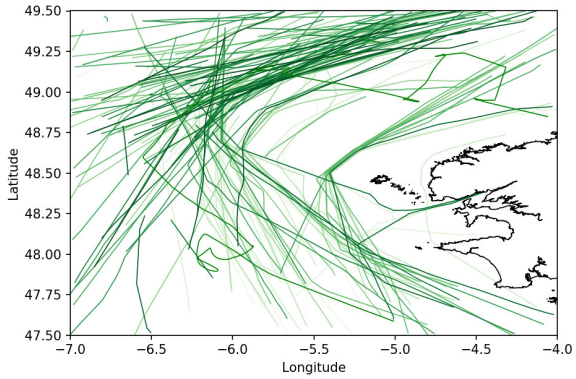


Figure 7.7 – AIS tracks that cannot be mapped to maritime routes, hence cannot be monitored by DBSCAN-based methods. In the test set that comprises only cargo and tanker vessels (from March 21 to March 31, 2017), such tracks account for 13% of all AIS tracks.

has to be able to predict the next position a vessel in 10 minutes (the time resolution of the model) with a tolerance of only 200 meters. On the other hands, if the resolution is too coarse, the information available to the model may not be enough to characterise the movement patterns. For example, a spatial resolution of 0.05° means that two positions within a radius of 5 kilometers are not distinguishable.

Vessel types: Another advantage of*GeoTrackNet* is the possibility of applying to any type of vessels. The first step of DBSCAN-based methods is to cluster AIS tracks into maritime routes and learn the signature of each route. Hence, those methods can only apply to vessels that follow maritime routes, i.e. cargo and tanker vessels. By contrast, our method does not impose any hypothesis of this type, so it can apply to any type of vessels. We tested our model on a dataset that comprises all kinds of vessels, the results are shown in Fig. 7.10. Since the number of vessels of other types than cargo and tanker is significant, applying the surveillance on all types of vessels is of interest. However, this is a difficult task. Unlike cargo and tanker vessels, some other types, for example fishing vessels, have very complicated moving patterns, the model can hardly learn all of them. Even when the model is able to capture all the dynamics of AIS tracks, unexpected results are still inevitable, when the statistical anomalies are actually not suspicious (see Fig. 7.10a). There is a trade-off between the monitoring capacity and the performance. When monitoring all types of vessels, it is possible that in a small area, there are some patterns that can be learnt and others that can not. The distribution p^{C_i} is not unimodal anymore. Hence, it cannot be approximated by a Gaussian distribution (see Fig. 7.9). This explains

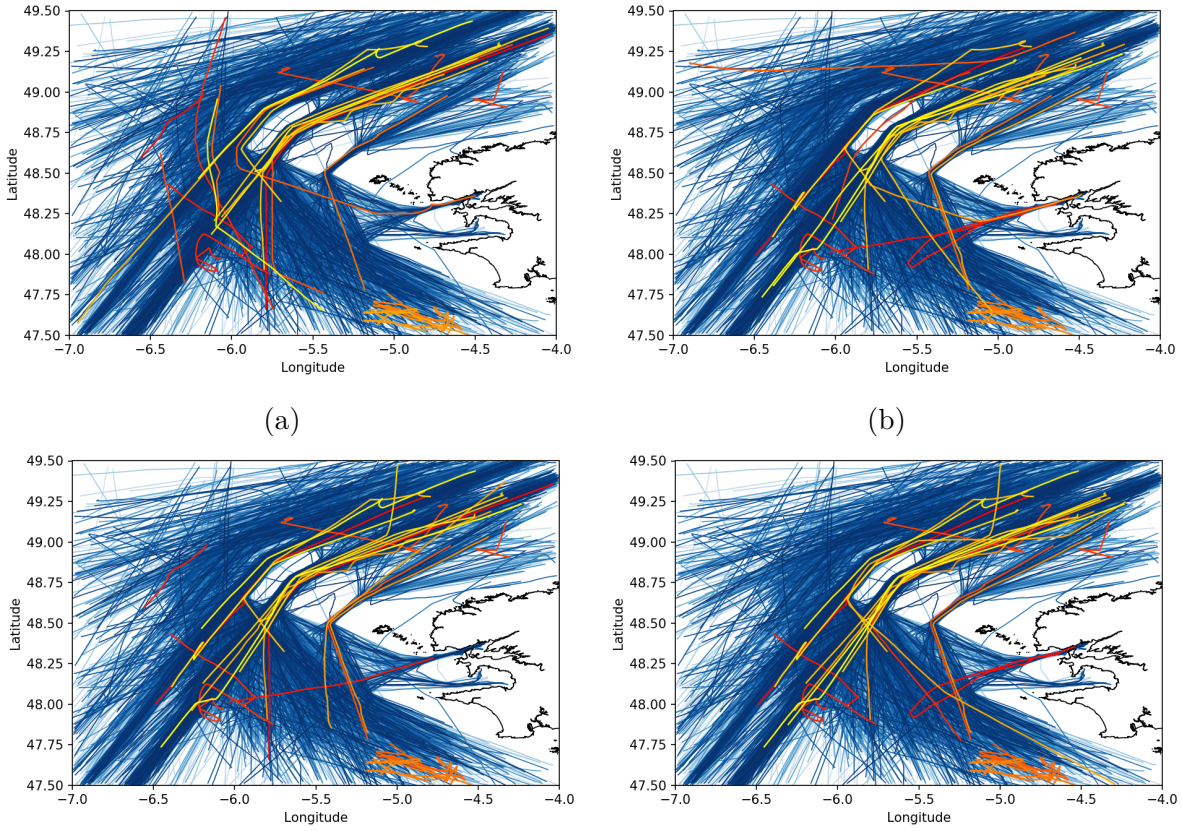
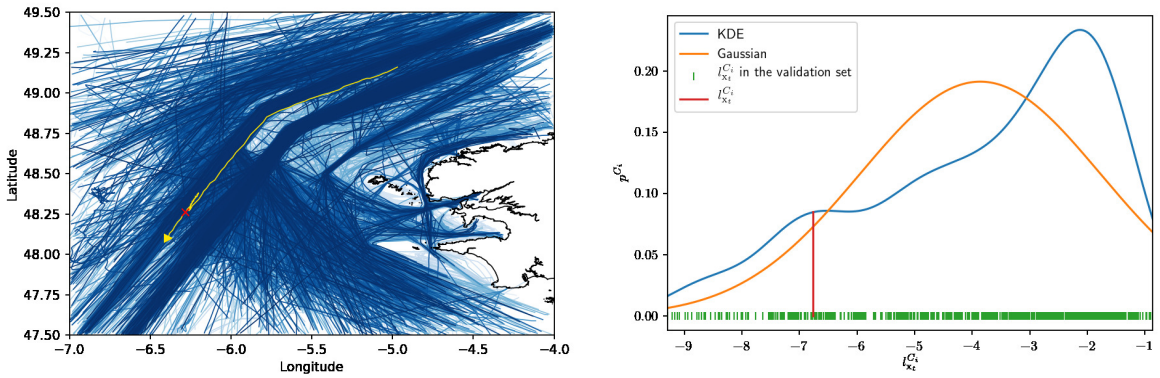


Figure 7.8 – Illustration of the relevance of the "four-hot" representation. (a) Abnormal tracks detected by a model without the "four-hot" representation; (b) Abnormal tracks detected by a *GeoTrackNet* model with the resolutions of the latitude, longitude, SOG and COG set to 0.002°(=0.2 times the reference setting), 0.002°, 1 knot and 5°, respectively; (c) Abnormal tracks detected by a *GeoTrackNet* model with the resolutions of the latitude, longitude, SOG and COG set to 0.05°(=5 times the reference setting), 0.05°, 1 knot and 5°, respectively. (d) The reference result, the resolutions of the latitude, longitude, SOG and COG were set to 0.01°, 0.01°, 1 knot and 5°, respectively.



(a) (b)

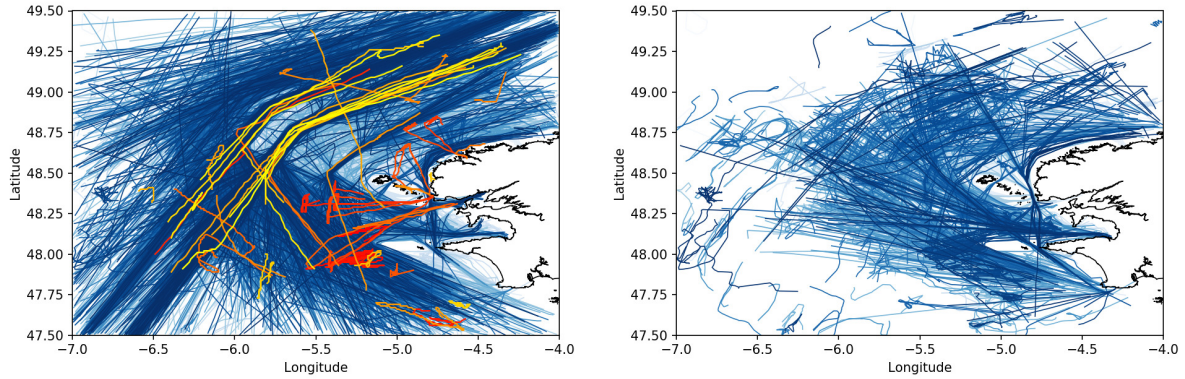
Figure 7.9 – Comparison between the Gaussian approximation and KDE for distribution p^{C_i}. (a) a track detected as abnormal by KDE *GeoTrackNet*, and not by Gaussian *GeoTrackNet* when the dataset comprises all types of vessels. (b) p^{C_i} of the area around the point "x" in (a). $p^{C_i}_{KDE}(\mathrm{L} < l^{C_i}_{\mathbf{x}_k}) = 0.128$ while $p^{C_i}_{Gauss}(\mathrm{L} < l^{C_i}_{\mathbf{x}_k}) = 0.082$. Overall, when the data comprises all types of vessels, p^{C_i} is not unimodal and KDE shall be preferred.

why the non-parametric density estimation using KDE gives better outcomes in those cases.

Hereafter in this chapter, unless specified otherwise, the reported results are the results of KDE *GeoTrackNet*. Results similar to those reported above for a dataset from July to September 2017 and from January to March 2018 can be found for models learnt for these periods.

Seasonal effects: We conducted additional experiments to demonstrate the consistency of *GeoTrackNet*. In this test, the models learnt from the training set of one period were evaluated on the test set of another period [5]. Table 7.1 shows the average log likelihood on different test sets of models trained on data from January 1 to March 10, 2017. The test sets are data from the 21st to the end of the corresponding month. Seasonal effects are small for cargo and tanker vessels. Over seasons, most of the changes are in speed. While for other types of vessels, especially for fishing vessels, the behaviours change completely. That explains why the log likelihood of the model trained on all vessels, from January 1 to March 10, 2017 is considerably low on the test set of September 2017. As shown in Fig. 7.11, between winter and summer, the fishing patterns are very different. A model trained on data in one season may not apply to data in another season. These experiments suggest considering season-specific models and/or training a general model which also takes into

5. In real-life applications, we always train the model on recent data. This setting is just to test the consistency of the model



(a) (b)

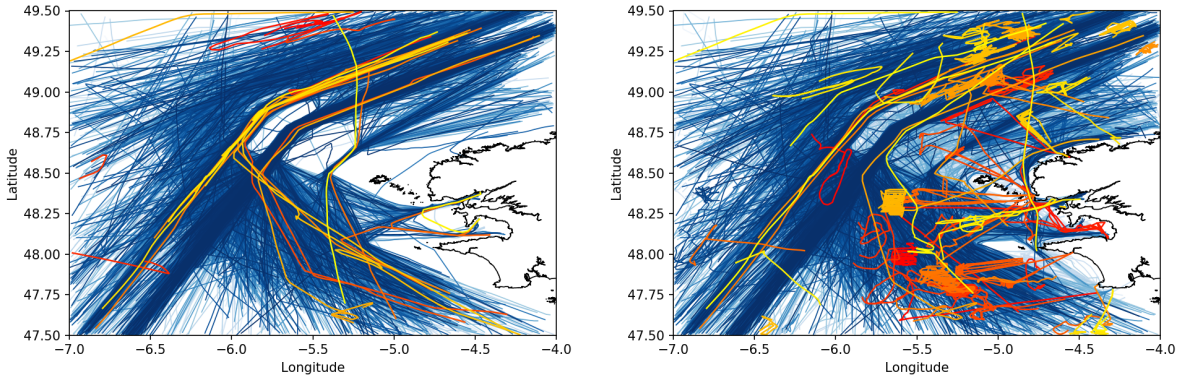
Figure 7.10 – Anomaly detection examples of KDE *GeoTrackNet* with AIS data comprising all vessel types from January to March 2017. (a) AIS tracks that are flagged as abnormal by KDE *GeoTrackNet*. Some tracks are statistically abnormal, however, their behaviours are not suspicious. For examples, the red tracks that steamed from land are fishing vessels went fishing; they were detected as abnormal because there are not enough similar AIS tracks in the training set. (b) AIS tracks of fishing vessels in the training set (about 13% of tracks in the training set).

Table 7.1 – Average log likelihood of *GeoTrackNet* for different test sets when trained on AIS data from Jan 1 to Mar 10, 2017.

Test set	Cargoes and tankers	All types
March 2017	-5.83	-6.53
September 2017	-5.93	-7.43
March 2018	-5.84	-6.76

account a seasonal information.

AIS memory requirements: In operational mode, one question arises is how long we should keep the past data of each AIS track. In the offline version of *GeoTrackNet*, this quantity is the maximum duration L_{max} of each track. Fig. 7.12 shows the results of the detection when we split long voyages into small tracks from 4h to: (a) 8h and (b) 16h. Discarding old AIS messages may save memory resources of the system, however, in some cases, we have to observe the track long enough to recognise the anomaly. For example, the voyage of the cargo vessel in Fig. 7.5e was not detected if the maximum duration of each track is 8h. This is because without knowing the other parts, each segment of this voyage is normal. For dataset presented in this chapter, $L_{max} = 16h$ and $L_{max} = 24h$



(a) (b)

Figure 7.11 – Anomaly detection examples of the model trained on data from January 1 to March 20, 2017 and tested on data from July 21 to September 30, 2017. (a) When the data comprise only cargo and tanker vessels. (b) When the data comprise all kind of vessels.

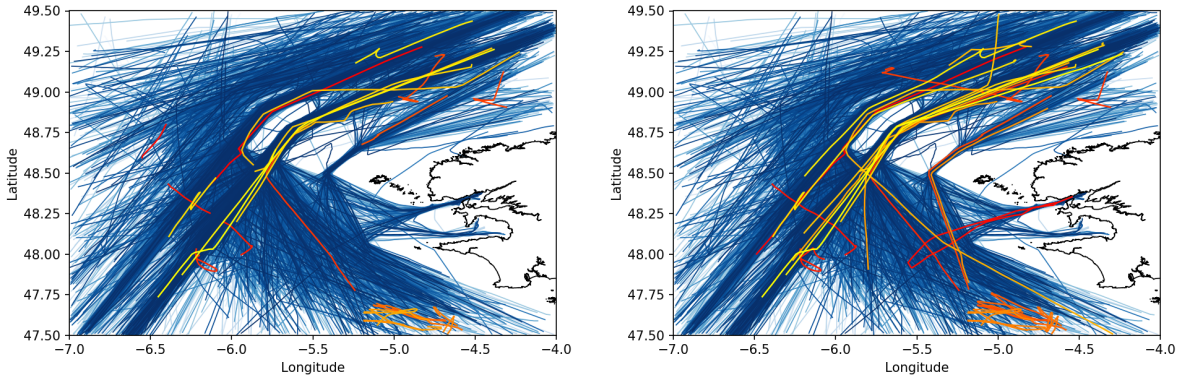
give the same outcomes. We chose $L_{max} = 24h$ in our experiments as our computational resources could store and process the resulting datasets.

7.5 Conclusions and future work

We introduced a new approach for maritime anomaly detection using AIS data. To our knowledge, this is the first model which relies on a normalcy model of AIS tracks using a deep learning generative scheme. The proposed model is novel, both in the way the normalcy model is built and the way deviations from the normalcy are evaluated. More precisely, we exploit Variational Recurrent Neural Networks to represent AIS tracks probabilistically using an original four-hot encoding of AIS data. Once the approximate distribution of the data is learnt, a geospatial *a contrario* detector is used to evaluate how likely an AIS track is. This detector takes into account the fact that the performance of the learning is geographically dependent. The general idea is that if an AIS message has its log probability lower than other messages' in the same region, it should be flagged as abnormal. An AIS track is abnormal if there are many abnormal messages in this track.

The key features of the proposed approach are as follows:

— It requires a minimal prior knowledge about the data. The model can be applied in different regions without major modifications.



(a) (b)

Figure 7.12 – Effect of the size of the historical data. (a) The maximum duration of each track is 8h; (b) The maximum duration of each track is 16h. If the system does not keep the track long enough, some anomalies may be missed.

— It does not require important hyperparameters such as the number of points in a cluster when using DBSCAN, the number of modes in mixture models, etc.

— We can control the percentage of the activities expected to be flagged as abnormal by simply changing the value of ε in Eq. (7.13).

— DBSCAN-based models cannot monitor AIS tracks that do not follow maritime routes. Fig. 7.7 and Fig. 7.10b show that the number of those tracks are significant [6]. Our method applies to all AIS tracks in the processed area.

— The proposed model can detect both geometric/geographic and speed-related anomalies.

— The nature of VRNN provides an additional means to condition the output onto external forcing variables or other sources of information. Hence, our model could further benefit from complementary information such as weather conditions, ocean current situations, etc. Mathematically, it comes to modelling $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{1:t-1}) = p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k, \mathbf{u}_t)$ with $\mathbf{u}_t$ the forcing variables and additional information.

— It is worth noting that anomaly detection is one task (and the most important one) in maritime surveillance. A model that can be integrated into a bigger system would optimise computational and storage resources. In the preliminary version of this work (Duong Nguyen, Vadaine, et al. 2018), we showed the proposed NN architecture to

6. The original paper (Giuliana Pallotta et al. 2013) reported the fraction of processable AIS messages varied from 40 to 95%

be generic and relevant to address other tasks besides anomaly detection such as vessel type recognition and trajectory interpolation. We let the reader to (Duong Nguyen, Vadaine, et al. 2018) for additional information. Regarding computational requirements, the resolution of *GeoTrackNet* is 10 minutes, i.e. the system keeps only one AIS message each 10 minutes. This reduces significantly the amount of data to process and store (by convention, the transmit rate of dynamic AIS message is from every few seconds to every few minutes (IMO 2017)). Once the model is learnt, we do not need to store the training dataset. For example, the training set used in this chapter from January 1 to March 10, 2017 comprises about 3.3 million AIS messages, which amounts to ∼450MB in *.csv format. The learnt model (i.e., VRNN weights) can be embedded into ∼40MB in Tensorflow format, which is relatively small. We may also point the development of a stream-based version *GeoTrackNet* (Duong Nguyen, Simonin, et al. 2020) supports its relevance for a real-time implementation within a big data and distributed system.

Although deep learning has recently grown extremely fast and has become the state-of-the-art approach in many domains (LeCun et al. 2015), its achievements in MDA are surprisingly limited. To the best of our knowledge, this work is the first one that applies unsupervised deep neural networks to maritime anomaly detection. This work opens new avenues to explore new research directions to complement and/or outperform DBSCAN-based approaches.

As any unsupervised learning-based model, the proposed approach detects events that are statistically unusual. These events may not involve suspicious actions. Ongoing experiments involve analyses by experts to evaluate the consistency of the detections w.r.t. operational requirements. In this respect, the creation of a reference groundtruth dataset would be highly beneficial to advance the state-of-the-art and make benchmarking experiments quantitative. This is however a complex task that would require a large collaborative effort. A more thorough study of the relationship between the resolution of the "four-hot" vector and the corresponding detection results could facilitate the hyper-parameters selection process when applying the model in different zones. The proposed neural network representation provides a flexible and powerful means to learn the distribution of AIS tracks, yet uninterpretable. The model is more suitable for a computer-assisted system (where the final decision is still on the human operator) than a fully automatic system.

We may emphasise that this representation is also of interest for other tasks, e.g., AIS

track interpolation, vessel type identification, as shown in Chapter 6. Future work might benefit from such multi-task settings.

Part IV

Closing

Great men are not born great, they grow great.

Mario Puzo

Chapter 8

Conclusions

8.1 Conclusions

This thesis has studied advances in variational deep learning for time series modelling and analysis. We have introduced the motivation,the formulation and presented different VDL architectures. Variational inference frames the inference as an optimisation problem. VDL leverages neural network and gradient-based optimisation to increase the capacity of the model. Current state-of-the-art VDL methods for time series modelling can be categorised into two classes: DSSM and SVAE. DSSM is an improved version of classical SSM. By using neural networks to parameterise the distributions, DSSM overcomes the difficulties of non-linearity. SVAE extends RNN by adding stochastic components to overcome the limit of deterministic transitions. Both are powerful for capturing long-term dependencies in highly-nonlinear, noisy and irregularly sampled time series. However, depending on specific applications, one may be more suitable than the other.

We have proposed novel VDL methods for two specific applications: dynamical system identification and maritime traffic surveillance. Specifically:

— We have proposed a general deep learning framework—called DAODEN—for learning chaotic and potentially stochastic dynamical systems. This framework uses a deep state space model formulation to retrieve the unknown differential equations that govern the data in the training set. By bridging classical data assimilation and modern machine learning techniques (deep learning), DAODEN can significantly improve the

performance of current state-of-the-art learning models under imperfect conditions, *i.e.* noisy and partial observation. Furthermore, because DAODEN embeds stochastic components to account for stochastic variabilities, model errors and reconstruction uncertainties, this framework can apply stochastic dynamical systems.

— We have proposed a deep learning architecture—called MultitaskAIS—for maritime surveillance using AIS data. The key component of MultitaskAIS is a VRNN, which embeds the information in the "four-hot" vectors of AIS trajectories in series of regular latent states. Many task-specific submodels can be built on top of this layer. We have demonstrated that MultitaskAIS could achieve state-of-the-art performance on three tasks: trajectory reconstruction, vessel type identification anomaly detection, while significantly reducing storage and computational needs. The most important submodel in MultitaskAIS is the anomaly detection model—referred as GeoTrackNet. This model leverages the probabilistic representation given by the VRNN, and uses an geo-spatial *a contrario* detector to detect abnormal vessels' behaviours. The *a contrario* detector in GeoTrackNet takes into account the fact that AIS data are location-dependent, hence the performance of the VRNN is also location-dependent. GeoTrackNet is the first successful application of DL in maritime traffic anomaly detection. The model is under consideration for deployment in a commercialised big data platform.

Through the work done in this thesis, we have experienced that although in theory big neural networks can archive any complex task, in reality it's barely the case. Domain expertise is crucial. We have to find a way to encode prior knowledge of the problem of interest to the network to obtain the desired outcomes. For example, the prior knowledge encoded in GeoTrackNet is the fact that AIS trajectory data are location-dependent.

8.2 Open questions and future work

Research is a long journey. The results presented in this thesis have set some steps forward for the considered problems, however, they also raise a new set of open (and probably more difficult) questions that remain for future explorers. We divide them in two three topics: i) VDL for time series modelling and analysis in general; ii) VDL for learning dynamical systems and iii) VDL for maritime surveillance using AIS.

VDL for time series modelling and analysis:

— In DSSM and SVAE, the choice of the parameterisation (*e.g.* number of layers, the type of activation function, *etc.*) for the transition, the emission and the approximate inference distribution is crucial. Is there a way to quickly identify which setting is suitable for a specific task?

— For time series analysis, understanding the stochasticity of the data is very important. For a new problem, how can we verify which stochastic components come from the observation operator (the emission distribution), and which come from the intrinsic nature of the dynamics of the data (the transition distribution). In classical data assimilation, this problem is stated as model errors and observation errors estimation (Li et al. 2009; Gershgorin et al. 2010). Could we adapt those ideas to NN-based models?

— The core of variational inference is to approximate the likelihood of the observed data by an lower bound. How to know which lower bound is good? The work presented in (Ma et al. 2019; L. Chen et al. 2018) can be considered for future investigation.

— In this thesis we focused on parametric distributions, non-parametric variational inference (Gershman et al. 2012) could also be an alternative.

— Almost all of current VDL models approximate the inference distribution for real-valued data by an Gaussian. This would not be efficient if the true prior distribution is multimodal or does not have a "bell" shape. Hence, to model non-Gaussian events (*e.g.* rainfall, extreme weather, *etc.*), we have to use non-Gaussian variational inference, as in (Ma et al. 2019; Qiu et al. 2018).

— For the problems considered in this thesis, the states and the observations are 1-dimensional vectors. Passing to high dimensional space (for example, when the observations are 2-D images) would required further studies to reduce the complexity of the representation and the computation.

— In this thesis, we used LSTMs to capture long-term correlations in data. However, if we have prior knowledge of the "length" of the temporal dependencies, parallel architectures such as transformer (Vaswani et al. 2017) will accelerate the computational time.

VDL for learning dynamical systems:

— We tested DAODEN models on already-known dynamical systems. The characteristics, the long-term topology of the benchmarked systems have been well studied.

Although this information is not used in the training and the validation phases, it is required to evaluate how good a learnt model is. For a completely unknown system, how could we validate a learnt model? Especially when common criteria such as the prediction error are not effective for stochastic systems.

— We have examined cases where the observations are partial in the sense that some components of the observations may be missing, in both spatial and temporal dimensions, however, all the components of the system's states are seen at least once. There are also situations where some components of the systems are never observed. For those case, we have to exploit augmented states (Abarbanel et al. 1994; Robinson 2005; Ayed et al. 2019; Ouala, Duong Nguyen, Herzet, et al. 2019).

— The key idea of SINDy (Brunton, Proctor, et al. 2016) is the sparsity hypothesis. Similarly to (Dremeau et al. 2012), we can incorporate this idea with with DL to create a "sparse network" for the identification of dynamical systems.

— It is crucial to enforce physical constraints in order to obtain physically-meaningful learnt model. Preliminary work in (Cockburn et al. 1990; Raissi, Perdikaris, and George E. Karniadakis 2019; Bézenac et al. 2019) could be investigated in future studies.

— The idea of presenting the dynamics of time series by ODEs or SDEs opens new means to tackle the issue of irregular sampling: using continuous representation of neural networks to handle sporadic observations (De Brouwer et al. 2019).

VDL for maritime surveillance using AIS:

— AIS is a self-reporting system. Although MultitaskAIS and GeoTrackNet could handle missing data, if a vessel turns off its AIS signal for long period, AIS-based models can not apply. AIS on-off detection models, such as those in (Fabio Mazzarella et al. 2017; Kontopoulos et al. 2020) are required for maritime surveillance.

— Although the performance of GeoTrackNet is very impressive, the detected AIS tracks were validated by independent AIS experts, however, we do not know what the detector may miss. A labeled, well-prepared data would be highly beneficial for the community to push forward recent advances.

Appendices

People generally see what they look for, and hear what they listen for.

Harper Lee

Appendix A

Variational Deep Learning for Acoustic Anomaly Detection

[1] In this chapter, we adapt Recurrent Neural Networks with Stochastic Layers, which are the state-of-the-art for generating text, music and speech, to the problem of acoustic novelty detection. By integrating uncertainty into the hidden states, this type of network is able to learn the distribution of complex sequences. Because the learnt distribution can be calculated explicitly in terms of probability, we can evaluate how likely an observation is then detect low-probability events as novel. The model is robust, highly unsupervised, end-to-end and requires minimum preprocessing, feature engineering or hyperparameters tuning. An experiment on a benchmark dataset shows that our model outperforms the state-of-the-art acoustic novelty detectors.

A.1 Introduction

Audio processing in general, and acoustic novelty detection in particular has attracted significant attention recently. A number of studies have used acoustic data to detect abnormal events, mostly for surveillance purposes, such as human fall detection (Salman

1. This work was conducted during the stay of Duong Nguyen at the Institute for Big Data Analytics, Dalhousie University, Canada. It was supported by the UBL Mobility Fund and the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors would like to thank A3Lab for the dataset.

Khan et al. 2015), abnormal jet engine vibration detection (Clifton et al. 2015), hazardous events detection (Ntalampiras et al. 2011).

The main challenge of novelty detection is we do not have a large amount of novel events to learn their characteristics, while the normal set is usually very big and contains a large amount of uncertainty. The common approach is to use unsupervised methods to learn the normality model, then consider events that do not fit this model as abnormal (novel). Most of these systems use Gaussian Mixture Model (GMM) or Hidden Markov Model (HMM) (Kumar et al. 2005; Ntalampiras et al. 2011; Atrey et al. 2006). Bayesian Networks have also been explored (Zajdel et al. 2007; Giannakopoulos et al. 2010). Recently, advances in deep learning (LeCun et al. 2015), especially in Recurrent Neural Networks (RNNs) and their extensions (Long Short-Term Memory — LSTM (Sak et al. 2014), Gated Recurrent Unit — GRU (Junyoung Chung et al. 2015)) have opened new venues for acoustic modelling. In (Marchi, Vesperini, Eyben, et al. 2015), the authors employed LSTMs to create an AutoEncoder (AE) to model normal sounds and detect abnormal sounds using the reconstruction errors. This idea has been extended in (Principi et al. 2017) by applying an adversarial training protocol.

However, acoustic signals are stochastic. RNN-based networks, whose hidden states are deterministic, can hardly capture all the variations in the data. Recent efforts to improve the modelling capacity of RNNs by including stochastic factors in their hidden states have shown impressive results, especially for generating text, music and speech (Bayer et al. 2014; J. Chung et al. 2015; Fraccaro et al. 2016).

In this chapter, we adapt these models to create an unsupervised acoustic novelty detector. Our approach performs an end-to-end learning of a probabilistic representation of acoustic signals. Given this representation, we can evaluate how likely an observation and state the detection of novel events as the detection of observations with a low probability. We argue that this model is robust, highly unsupervised, end-to-end and requires minimum preprocessing, feature engineering or hyperparameter tuning. Our empirical evaluation on a dataset for novel event detection in audio data shows that the proposed model outperforms the state-of-the-art.

The paper is organised as follows: in Section A.2, we present the details of the proposed approach; we compare the model with state-of-the-art methods to point out its advantages in Section A.3; the experiment and results are shown in Section A.4; finally in Section A.5 we give conclusions and some perspectives for future work.

A.2 The proposed approach

A.2.1 Recurrent Neural Networks with Stochastic Layers (RNNSLs)

For time series modelling, the two most common approaches are State Space Models (SSMs) and Recurrent Neural Networks (RNNs). SSMs such as Kalman filters (Brown et al. n.d.) and particle filters (Doucet et al. 2009) have been explored for a long time and are the state-of-the-art model-driven schemes thanks to their ability to model stochasticity. However, these models are limited by their mathematical assumptions (for example, Kalman filters assume the data generating process is Gaussian). RNNs, on the other hand, have attracted a lot of attentions recently by their capacity to represent long-term dependencies in time series (LeCun et al. 2015). The main drawback of RNNs is that their hidden states are deterministic, making them unable to capture all the stochastic components of the data. A number of efforts have been made to bring together the power of SSMs and RNNs (Bayer et al. 2014; J. Chung et al. 2015; Fraccaro et al. 2016; R. G. Krishnan et al. 2017): Recurrent Neural Networks with Stochastic Layers (RNNSLs).

RNNSLs aim to learn the distribution p, which can be factored through time, over a sequence of T observed random variables $\{\mathbf{x}_t\}_{,t=1..T}$:

$$p(\mathbf{x}_{1:T}) = \prod_{t=1}^{T} p_t(\mathbf{x}_t|\mathbf{x}_{<t}), \tag{A.1}$$

where $\mathbf{x}_{<t}$ denotes $\mathbf{x}_{1:t-1}$.

Following an SSM formulation, we assume that the data generation process of $\mathbf{x}_{1:T}$ relies on a sequence of T latent random variables $\{\mathbf{z}_t\}_{,t=1..T}$. At each time step t, the joint distribution $p_t(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}\mathbf{z}_{<t})$ can be factorised into:

$$p_t(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}\mathbf{z}_{<t}) = p_t(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t})p_t(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}), \tag{A.2}$$

where $\mathbf{z}_{\leq t}$ denotes $\mathbf{z}_{1:t}$. In other words, each time step of the network is an autoencoder, conditionally to the historical information.

Depending on the stochastic nature of the considered data, the emission distribution $p_t(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t})$ may be highly nonlinear. However, this nonlinearity usually leads to the intractability of the inference distribution $p_t(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$. The most common solution to overcome this obstacle is the variational approach (J. Chung et al. 2015; Fraccaro et al. 2016), which introduces an approximation $q_t(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ of the posterior distribution

$p_t(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ then estimates $p_t(\mathbf{x}_t|\mathbf{x}_{<t})$ by the Evidence Lower BOund (ELBO) $\mathcal{L}(\mathrm{x}, p_t, q_t)$:

$$\log p_t(\mathbf{x}_t|\mathbf{x}_{<t}) \geq \mathcal{L}(\mathrm{x}, p_t, q_t) = \mathbb{E}_{\mathbf{z}_t \sim q_t}\Big[\log p_t(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t})\Big] - \mathrm{KL}\Big[q_t(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})||p_t(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})\Big] \tag{A.3}$$

where $\mathrm{KL}\Big[q_t||p_t\Big]$ is the Kullback-Leibler divergence between two distributions q_t and p_t.

There are several types of RNNSLs, differing in the way that they model the structure of the latent space. The most common types are Variational Recurrent Neural Networks (VRNNs) (J. Chung et al. 2015), Stochastic Recurrent Neural Networks (SRNNs) (Fraccaro et al. 2016) and Deep Kalman Filters (DKFs) (R. G. Krishnan et al. 2017). We experimented most of these types, however, in this chapter, for simplicity purposes, we only report the VRNNs, introduced by Chung et al. (J. Chung et al. 2015).

In VRNNs, the historical information $(\mathbf{x}_{<t}, \mathbf{z}_{<t})$ is encoded by the dynamics of the hidden states of their RNN (LSTM) $\mathbf{h}_t = h(\mathbf{x}_{t-1}, \mathbf{z}_{t-1}, \mathbf{h}_{t-1})$. More precisely, it involves the parameterisation of the following distributions, namely the emission distribution $p_t(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t}) = p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{h}_t)$, the prior distribution $p_t(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}) = p(\mathbf{z}_t|\mathbf{h}_t)$ and the variational posterior distribution $q_t(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}) = p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{h}_t)$ as neural networks. Here, we consider fully connected networks with Gaussian formulation of these three distributions. For more details of VRNNs, please refer to (J. Chung et al. 2015).

A.2.2 RNNSLs for Acoustic Novelty Detection

RNNSLs were initially designed for generating text, music, speech. They are currently the state-of-the-art in these domains (J. Chung et al. 2015; Fraccaro et al. 2016; Maddison et al. 2017). The interesting point of this type of models in comparison to other state-of-the-art methods like Wavenet (Oord et al. 2016) is that these models calculate the distribution $p(\mathbf{x}_{1:T})$ explicitly, so that after learning this distribution from the training set, we can evaluate the probability for each new sequence. The idea of using RNNSLs for novelty detection was first introduced in (Duong Nguyen, Vadaine, et al. 2018) for the detection of abnormal behaviors of vessels, we adapt this model to novelty detection in acoustic data.

Here, an acoustic signal is modelled as a time series $\{\mathbf{x}_t\}_{,t=1..T}$ where $\mathbf{x}_t$ can be a chunk of n samples of the waveform, or n frequency bins in a spectrogram at a given time t. A RNNSL first learns the distribution over $\mathbf{x}_{1:T}$ in the training set, which may or may not contain some abnormal sequences. Then, for any new acoustic signal, we can evaluate its

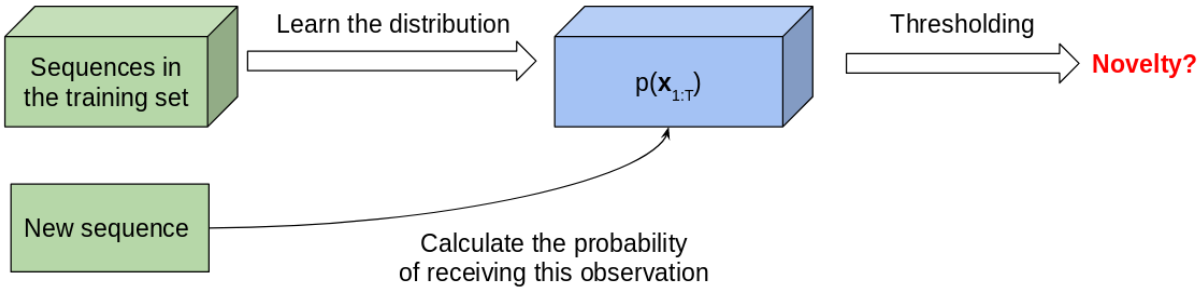


Figure A.1 – Architecture of the proposed RNNSL-based novelty detector.

log-probability. If this log-probability is smaller than a threshold, the sequence will be considered as abnormal (or novel), as illustrated in Fig. A.1.

To choose the threshold, we create a validation set, which again may or may not contain some abnormal sequences and compute the mean μ_{valid} and the standard deviation σ_{valid} of the log-probability of the sequences in this set. The value of the threshold is then chosen as: $\theta = \mu_{valid} - \alpha * \sigma_{valid}$. α is usually chosen as 3.

The training set and the validation set may contain some abnormal sequences. However, since RNNSLs are probabilistic models, they will eventually ignore these "outliers" (this conjecture is confirmed experimentally). This property helps to reduce data cleaning efforts.

A.3 Related work

A number of researches have explored deep neural networks to detect novelty in acoustic surveillance. We point out here the advantages of our model over those used in (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017), which are currently the state-of-the-art methods.

Both (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017) used RNNs (LSTMs in particular) as an AutoEncoder (AE) which can reconstruct the original signal from a compressed representation (Compression AutoEncoders — CAEs) or from a corrupted version of it (Denoising AutoEncoders — DAEs). However, as discussed in (J. Chung et al. 2015; Fraccaro et al. 2016; R. G. Krishnan et al. 2017), the fact that the hidden states of RNNs are deterministic reduces their capacity to capture all data variations, especially for data that contain high levels of randomness.

Moreover, the detection criterion used in (Marchi, Vesperini, Eyben, et al. 2015) is

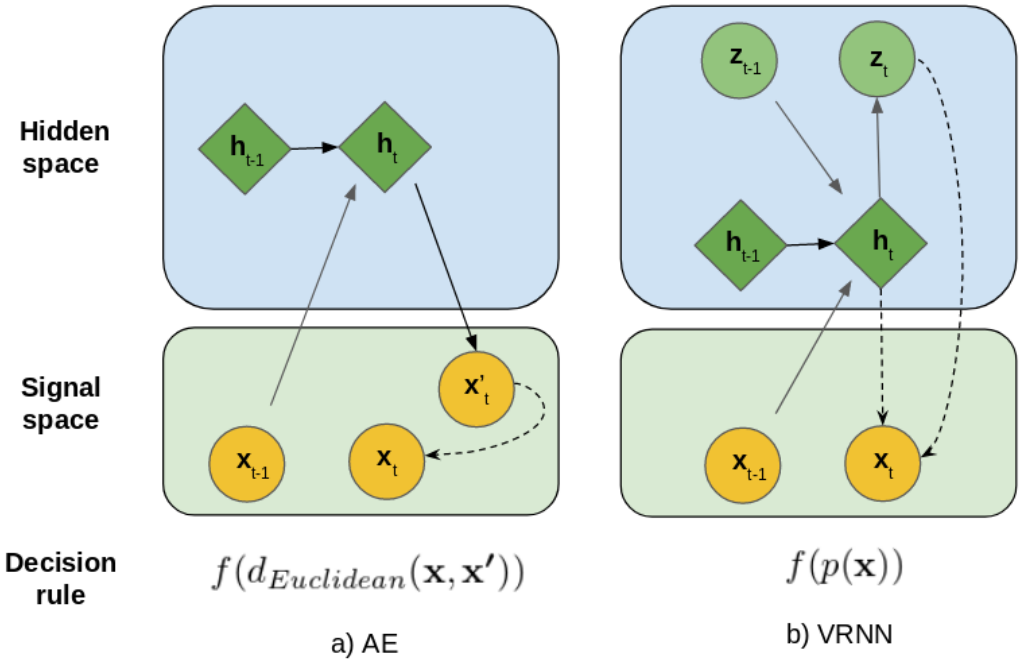


Figure A.2 – Architecture and decision rule of the proposed model (VRNN) in compared to previously proposed AE-based models. $\mathbf{x}_t$ is the original signal at the given time step t, $\mathbf{h}_t$ is the hidden state of the RNN (LTSM), $\mathbf{z}_t$ is the latent stochastic state, $\mathbf{x}'_t$ is the reconstructed output of the AE. The solid arrows denote the calculation processes, while the dashed arrows show how the cost function is calculated. We use the same notation as (Fraccaro et al. 2016), circles for stochastic factors, diamonds for deterministic factors.

the Euclidean distance between the original input and the reconstructed output of the autoencoder. This criterion is very sensitive to noise. (Principi et al. 2017) addressed this drawback by using an adversarial strategy, however, the ultimate idea is also to compare the original input and the reconstructed output from the autoencoder. By contrast, our method detects novel events by directly evaluating the probability of the received signal. Besides the improved detection criterion, the architecture of our model is also more robust to noise (Duong Nguyen, Vadaine, et al. 2018).

These differences are sketched in Fig. A.2. The hidden space of our model has stochastic factors, which help to increase modelling capacity. The decision rule of our model is a function of the distribution learnt by the network, making the model more robust to noise.

The selection of the thresholding value for novelty detection is another important difference compared to previous work. The approach in (Marchi, Vesperini, Eyben, et al. 2015) is not fully unsupervised, because it needs some information about the proportion of

abnormal events in the data. Our method, in contrast, only uses the information from the training set and the validation set to chose the threshold, without any prior knowledge of the annotations, based on a statistically-sound criterion, *i.e.* the false alarm rate.

A.4 Experiment and Result

A.4.1 Dataset

We tested our model [2] on the same dataset used in (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017), which is part of the PASCAL CHiME speech separation and recognition challenge dataset (Barker et al. 2013). The original dataset contains 7 hours of in-home environment recordings with two children and two adults performing common activities, such as talking, eating, playing and watching television. The author of (Marchi, Vesperini, Eyben, et al. 2015) took a part of those recordings and created a dataset for acoustic novelty detection (100 minutes for the training set and 70 minutes for the test set). In the new dataset, the sounds of the PASCAL CHiME are considered as background, the test set was generated by digitally adding abnormal sounds like alarms, falls, fractures (breakages of objects), screams. The details of the dataset were presented in (Marchi, Vesperini, Eyben, et al. 2015).

A.4.2 Experimental Setup

In order to use the models in (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017) as baselines, we set up our model to have the same evaluation metric that was used in those papers. However, instead of transforming the data to mel spectrograms like in (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017), we worked directly with the waveform (end-to-end model). The dataset was recorded by a binaural microphone at a sample rate of 16kHz. We converted each audio to 1 channel and then split it into sequences of 160-dimensional frames, each frame corresponds to 0.01s, as in (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017). (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017) evaluated the detection at each frame instead of at the whole sequence, so we also applied the thresholding step to each $\log p(\mathbf{x}_t|\mathbf{x}_{<t})$, instead of $\log p(\mathbf{x}_{1:T})$.

2. The code is available at https://github.com/dnguyengithub/AudioNovelty

Table A.1 – Detection result, in comparison with state-of-the-art methods.

Method	**Online Processing**	**Precision**	**Recall**	**F1 score**
GMM	Yes	99.1	87.8	89.4
HMM	Yes	94.1	88.9	91.1
LSTM-CAE	Yes	91.7	86.6	89.1
BLSTM-CAE	No	93.6	89.2	91.3
LSTM-DAE	Yes	94.2	90.6	92.4
BLSTM-DAE	No	94.7	92.0	93.4
Adversarial AE	?	?	?	93.3
VRNN	Yes	95.4	91.8	**93.6**
VRNN*	Yes	95.4	92.8	**94.1**

We tested different topologies of VRNN, with the latent size of 64, 80, 160 and 200. The models were trained using Adam optimiser (Diederik P. Kingma and Ba 2015), with a learning rate of $3e-5$.

A.4.3 Results

Different configurations gave different log-likelihoods on the dataset, however the final detection results were quite similar. We report here only one of the topologies, which gave the best result: VRNN with 160 latent units (the models with 80 hidden units also gave similar results). We compare the performance of our model with the result of GMM, HMM, those in (Marchi, Vesperini, Eyben, et al. 2015) (LSTM-based CAE, LSTM-based DAE) and in (Principi et al. 2017) (Adversarial AE). The result is shown in Table A.1. [3] Besides choosing the threshold automatically as discussed in Section A.2, we also used the same technique as in (Marchi, Vesperini, Eyben, et al. 2015) to chose the optimal threshold value, denoted as **VRNN***.

Our method not only outperformed the state-of-the-art methods, but also has the ability to work online, which is highly beneficial for real-time surveillance. Models that use bidirectional LSTM (BLSTM-CAEs, BLSTM-DAEs) can not reach online processing the because a look-ahead buffer is required. The online processing ability of Adversarial AEs depends on the structure that they use (LSTM or BLSTM).

3. The values in Table A.1 are from (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017; Principi et al. 2017) did not show the precision and recall of their model

Table A.2 – Robustness test.

SNR	Precision	Recall	F1 score
5dB	96.0	91.2	93.6
10dB	96.1	91.9	94.0
15dB	96.1	92.1	94.0

When investigating the cases where the proposed model misdetected the novelty, we found that actually the model could detect all the novel events, however, the way the detection was evaluated reduced the accuracy. As in (Marchi, Vesperini, Eyben, et al. 2015) and (Principi et al. 2017), the detection was evaluated at each time step of 0.01s. Our model has a memory effect (the memory of its LSTM cells), so it tends to merge the abnormal events that are very close to each other, as shown in Fig. A.3. In other cases, the model missed a part of the sound, especially for the tail of the fractures, as shown in Fig. A.4. These sounds have a long tail which is gradually submerged in the background. These misdetections are not detrimental in real life applications, because we are more interested in whether or not there is a novel event than on how long the event is.

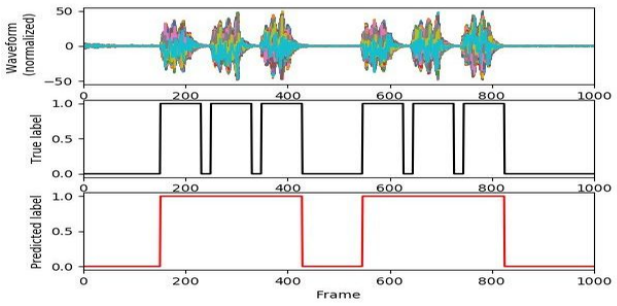


Figure A.3 – An example where the novelty events were merged. This figure shows the waveform of two alarms, each alarm consists of there "beeps", our model considered this "beep beep beep" as one event, while the annotation made by the authors of (Marchi, Vesperini, Eyben, et al. 2015) separates these "beeps".

We also conducted a robustness test where we added Gaussian noise to the test set. The additive noise is unknown by the model. This is a common scenario in audio surveillance, when the background environment changes (*e.g.* because of winds) or when noise appears in the electronic system. Table A.2 shows the performance of the proposed approach (with optimal threshold) on the corrupted test sets with different level of Signal to Noise Ratio (SNR). Thanks to the nature of VRNNs and the improved detection criterion, our model is robust to noise.

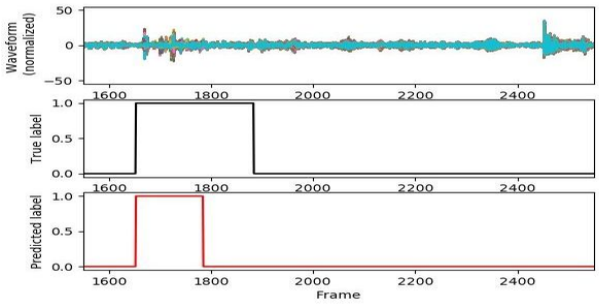


Figure A.4 – An example where the model missed a part of the novelty event. This figure shows the waveform of the sound of a fracture of a dish. The tail of the sound is very mall and gradually becomes submerged in the background.

A.5 Conclusions and perspectives

We have presented a novel unsupervised end-to-end approach for acoustic novelty detection. This approach exploits RNNs with stochastic layers, which are the state-of-the-art frameworks for time series modelling. Given the learnt probabilistic representations, novelty detection can be stated as a classic statistical test, which fully accounts for the stochasticity of the considered acoustic datasets. Reported experiments on a benchmarked dataset showed that the model outperforms the state-of-the-art detectors (Marchi, Vesperini, Eyben, et al. 2015; Principi et al. 2017).

The dataset used in this chapter is quite simple, the novel events in it are quite easy to be detected. Future work could involve applying this model to more complex signals, *e.g.* underwater acoustic signals which depict even greater variabilities. The impact of the threshold is also being studied to obtain better threshold selection rule.

Appendix B

Extended Abstract/Résumé Étendu

[1] Au cours de la dernière décennie, le monde a été témoin d'un développement incroyable de l'apprentissage profond (LeCun et al. 2015). L'apprentissage automatique en général, et l'apprentissage profond en particulier, ont récemment révolutionné de nombreux domaines de recherche et d'application (I. Goodfellow, Yoshua Bengio, et al. 2016; He et al. 2015; Mikolov et al. 2013; Ravì et al. 2017; Min et al. 2017; Goh et al. 2017; Kamilaris et al. 2018). Cependant, la majorité des applications pratiques de l'apprentissage automatique utilisent des méthodes d'apprentissage supervisées, qui apprennent un mappage d'une variable d'entrée à une variable de sortie à l'aide d'un ensemble de données étiquetées. Les données étiquetées sont rares et coûteuses à obtenir, contrairement à la grande quantité de données non étiquetées qui peuvent être collectées à un coût relativement faible. Un axe majeur de la recherche récente sur l'apprentissage automatique est donc le développement de méthodes d'apprentissage non supervisé (Diederik P. Kingma and Welling 2013; Rezende and Mohamed 2015; Vacar et al. 2019), qui utilisent les données disponibles non étiquetées. Dans ce context, un modèle devrait pouvoir décrire la structure sous-jacente des données, *e.g.* des motifs, des corrélations statistiques ou des structures causales. Cette thèse se concentre sur l'étude d'apprentissage profond pour un type particulier de données qui évolue dans le temps, appelées séries temporelles. Plus précisément, nous développons une famille de modèles séquentiels qui utilise l'inférence variationnelle pour appendre la dynamique cachée

1. Ph.D. theses completed in a French institution written in another language are required to provide an extended abstract in French/Les thèses de doctorat accomplies dans un établissement français mais rédigées dans une autre langue devraient être accompagnées d'un résumé étendu en français.

des données observée. Ces données peuvent être bruitées et échantillonnées de manière irrégulière. Nous combinons des architectures d'apprentissage profond avec des modèles classiques probabilistes de séries temporelles et intégrons les connaissances préalables du domaine pour créer: i) un nouveau cadre d'apprentissage des systèmes dynamiques, et ii) un nouveau modèle d'apprentissage profond pour la surveillance maritime à l'aide de données AIS (système d'identification automatique-automatic identification system en anlais).

Les contributions sont les suivantes:

— Nous introduisons deux classes de modèles à variables latentes profonds pour les données séquentielles: les modèles d'espace d'état profond (DSSMs—deep state space models en anglais) et les auto-encodeurs variationnels séquentiels (SVAEs—sequential variational autoencoders en anglais). Nous présentons les dérivations de ces modèles, à partir des des méthodes séquentielles classiques: les modèles espace d'états (SSMs—state space models en anglais) et les réseaux de neurones récurrents (RNNs—recurrent neural networks en anglais). Nous analysons les avantages et les inconvénients de chaque méthode, puis montrons comment DSSMs et SVAEs aident ces modèles à devenir plus expressifs et évolutifs en les combinant avec l'apprentissage profound.

— Nous présentons un DSSM, appelé DAODEN (data-assimilation-based ordinary differential equations network en anglais), spécialement conçu pour l'apprentissage des systèmes dynamiques. Ce modèle utilise des architectures de réseau de neurones pour modéliser la dynamique des systèmes d'équations différentielles ordinaires (ODEs—ordinary differential equation en anglais) et éventuellement des systèmes d'équations différentielles stochastiques (SDEs—stochastic differential equations en anglais). DAODEN contient deux composants clés : un modèle d'inférence qui imite les méthodes classiques d'assimilation de données pour reconstruire les vrais états du système à partir d'observations bruitées et potentiellement partielles, et un modèle génératif qui utilise la représentation des reseaux de neurones des systèmes dynamiques pour récupérer la dynamique sous-jacente de ces états. Par conséquent, par construction, DAODEN peut obtenir des performances comparables à celles des modèles entraînés sur des observations idéales, même lorsque le modèle est entraîné sur des données fortement endommagées.

— Nous présentons une architecture d'apprentissage profond, appelée MultitaskAIS, pour la surveillance du trafic maritime à l'aide de données AIS. Le cœur de cette

architecture est un SVAE, qui convertit les messages AIS bruités et irrégulièrement échantillonnés en séries d'états cachés propres et régulièrement échantillonnés de la trajectoire du navire. Ces états peuvent ensuite être utilisés pour des sous-modèles spécifiques à une tâche (tels que la reconstruction de trajectoire, l'identification du type de navire, la détection d'anomalies). Les expériences montrent que MultitaskAIS peut atteindre des performances de pointe sur ces tâches, tout en utilisant un stockage et des besoins de calcul nettement inférieurs. Parmi ces tâches, la plus importante est la détection d'anomalies. Nous introduisons GeoTrackNet, un détecteur géospatialisé qui utilise l'apprentissage profond variationnel pour construire une représentation probabiliste des trajectoires AIS, puis détecter les anomalies en jugeant la probabilité de cette trajectoire. Ce détecteur prend en compte le fait que les données AIS sont géographiquement hétérogènes. Des expériences sur des données réelles affirment la pertinence de la méthode proposée.

B.1 Apprentissage profond variationnel pour la modélisation et l'analyse de séries temporelles

Lorsque nous surveillons ou suivons un processus, les séquences des observations obtenues sont généralement corrélées dans le temps. Ce type de donnée est appelé série temporelle. La modélisation des séries temporelles est une tâche difficile, parce que la plupart du temps, nous ne connaissons pas les lois qui définissent la dynamique du processus considéré. Ces lois peuvent être hautement non linéaires, chaotiques et/ou stochastiques. De plus, les données que nous obtenons peuvent ne pas être les vrais états du processus, mais plutôt les observations/mesures bruitées et partielles. Au cours des dernières années, l'apprentissage profond variationnel séquentiel est apparu comme une approche très prometteuse pour la modélisation et l'analyse de séries temporelles (LeCun et al. 2015; R. G. Krishnan et al. 2017; J. Chung et al. 2015; Fraccaro et al. 2016). Cette approche combine la modélisation probabiliste et l'apprentissage profond (généralement des réseaux basés sur RNNs) pour construire des modèles expressifs de grande capacité qui peuvent capturer les stochasticités, les variations, les incertitudes et les corrélations à long terme dans les données. Dans ce chapitre, nous présenterons la motivation, la formulation et les applications de cette approche. Le contenu présenté ici est la partie théorique des applications dans les sections suivantes.

B.1.1 Modèles de variable latente pour la modélisation et l'analyse de séries temporelles

Etant donné une séries d'observations $\mathbf{x}_{0:T}$, l'objectif est d'apprendre un modèle $p_{\boldsymbol{\theta}}$ paramètrisé par un ensemble de paramètres $\boldsymbol{\theta}$, qui maximise la vraisemblance $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})$. On suppose que le processus de génération de données de $\mathbf{x}_{0:T}$ dépend d'une séquence de variables latentes $\mathbf{z}_{0:T}$. La distribution conjointe $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}, \mathbf{x}_{0:T})$ peut être factorisée en:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}). \tag{B.1}$$

La log vraisemblance de l'observation peut être obtenue en marginalisant les variables latentes:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}, \mathbf{z}_{0:T}) \mathrm{d}\mathbf{z}_{0:T} = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}) \mathrm{d}\mathbf{z}_{0:T}. \tag{B.2}$$

En général on ne peut pas calculer cette intégrale. Les approches variationnelles (Diederik P. Kingma and Welling 2013; J. Chung et al. 2015; Fraccaro 2018) proposent au lieu de maximiser cette vraisemblance, on maximise une borne inférieure, appelée ELBO (evidence lower bound en anglais), en approximant la vraie distribution postérieure $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ par une distribution variationnella distribution à posteriorie $q(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$:

$$\mathcal{L}(\mathbf{x}_{0:T}, p_{\boldsymbol{\theta}}, q_{\boldsymbol{\phi}}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})} \left[p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})\right] - \mathrm{KL}\left[q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})||p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})\right] \leq \log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}). \tag{B.3}$$

Habituellement, on impose quelques hypothèses pour factoriser $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})$, $q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ and $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})$. En fonction de ces hypothèses et de la factorisation, on obtient différents modèles pour la modélisation de séries temporelles. En général, ils peuvent être classés en deux classes: modèle espace d'états (SSMs) et réseau neuronal récurrent (RNNs).

Modèles espace d'états (SSMs)

Dans SSMs, on suppose que i) le processus de $\mathbf{z}_k$ est markovien, et ii) étant donné $\mathbf{z}_k$, $\mathbf{x}_k$ ne dépend pas d'autres états ou observations. Avec ces hypothèses, $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})$ et

$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})$ peuvent être factorisées comme suit:

$$p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{z}_0) \prod_{k=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}), \tag{B.4}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T}) = \prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k). \tag{B.5}$$

On obtient la forme générale de SSM:

$$\mathbf{z}_k \sim p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1}) \tag{B.6}$$

$$\mathbf{x}_k \sim p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k) \tag{B.7}$$

avec $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})$ est la **distribution de transition** qui modélise l'évolution temporelle de $\mathbf{z}_k$ et $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$ est la **distribution d'émission** qui présente l'opérateur d'observation.

Lorsque la transition et l'émission sont linéaires et gaussiennes, le filtre de Kalman (Kalman 1960) fournit une solution mathématiquement élégante pour le problème d'inférence. Cependant, lorsque la transition et/ou l'émission n'est plus linéaire et gaussienne, la distibution à posteriori ne peut plus être calculée. Nous devons effectuer des approximations. Pour les cas où la transition et l'émission peuvent être décrites par des fonctions différentiables, le **filtre de Kalman étendu** (EKF—extended Kalman filter en anglais) (Smith et al. 1962) approche la distribution à posteriori par une linéarisation de $p_{\boldsymbol{\theta}}(\mathbf{z}_k|\mathbf{z}_{k-1})$ et $p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{z}_k)$. Le **filtre particulaire** (PF—particle filter en anglais) (Doucet et al. 2009) a une approche différente. Cette méthode utilise un "échantillonnage séquentiel important" (important sampling en anglais) pour approximer récursivement $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:k}|\mathbf{x}_{0:k})$, étant donné $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:k-1}|\mathbf{x}_{0:k-1})$. Dans le filtre particulaire, une distribution est représentée par un ensemble de particules. Le **filtre de Kalman d'ensemble** (EnKF—ensemble Kalman filter en anglais) (Evensen 2003) fait le lien entre l'idée du filtre de Kalman et du filtre particulaire en supposant que les distributions représentées par les particules sont gaussiennes.

Réseaux de neurones récurrents (RNNs)

Au lieu d'imposer la propriété markovienne à $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})$ et $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})$, les RNNs supposent qu'au temps k, toutes les informations historiques peuvent être codées dans une

variable déterministe $\mathbf{h}_k$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p_{\boldsymbol{\theta}}(\mathbf{x}_0) \prod_{k=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{x}_{0:k-1}) = \prod_{k=0}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_k|\mathbf{h}_k). \tag{B.8}$$

Et $\mathbf{h}_k$ peut être mise à jour en utilisant les informations de l'état précédent $\mathbf{h}_{k-1}$ et de l'observation précédente $\mathbf{x}_{k-1}$:

$$\mathbf{h}_k = h_{\boldsymbol{\theta}}(\mathbf{h}_{k-1}, \mathbf{x}_{k-1}), \tag{B.9}$$

avec $h_{\boldsymbol{\theta}}$ est une fonction non-linéaire différentiable, comme celles de (Elman 1990). Pour pouvoir capturer les corrélations temporelles dans les données, la fonction $h_{\boldsymbol{\theta}}$ doit être de grande capacité. De nos jours, on utilise généralement des extensions des RNNs, telles que LSTMs (long short-term memory en anglais) (Hochreiter et al. 1997) et GRUs (gated recurrent units en anglais) (Junyoung Chung et al. 2015). Comme $\mathbf{h}_k$ est déterministe, l'inférence $p_{\boldsymbol{\theta}}(\mathbf{h}_{0:T}|\mathbf{x}_{0:T})$ peut être calculée. Cependant, RNNs ne peuvent pas capturer tous les variabilités et les incertitudes dans les données.

B.1.2 Modèles probabilistes séquentiels profonds

Les SSMs sont des modèles stochastiques structurés avec de belles propriétés de factorisation et d'indépendance, cependant, le nombre de problèmes que les SSMs classiques peuvent couvrir est assez limité, car ils reposent sur des solutions analytiques. Les RNNs sont des modèles hautement non-linéaires qui peuvent capturer les dépendances à long terme dans les données, toutefois, vu que leurs états cachés sont déterministes, les RNN ne peuvent pas modéliser la stochasticité. Nous pouvons fusionner les idées des SSMs et des RNNs pour construire de meilleurs modèles de séries temporelles, qui héritent des avantages et surmontent les faiblesses des méthodes originales. On peut formuler deux approches : i) DSSMs, qui sont des extensions de SSMs, et ii) SVAEs, qui sont des extensions de RNNs.

Pour construire un DSSM, on parametrise les trois distributions $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{z}_{0:T})$, $p_{\boldsymbol{\theta}}(\mathbf{z}_{0:T})$ et $q_{\boldsymbol{\phi}}(\mathbf{z}_{0:T}|\mathbf{x}_{0:T})$ dans SSMs par des réseaux de neurones. Sur la base de l'expertise du domaine, on choisit les architectures adaptées. Il n'y a pas d'architectures universelles qui fonctionnent bien dans tous les cas. Pour construire un SVAE, on augmente l'espace latent en ajoutant une varibale aléatoire $\mathbf{z}_k$. La récurrence du modèle devient donc stochastique. Deux exemples populaires de modèles de ce type sont VRNN (variational recurrent neural network en anglais) (J. Chung et al. 2015) et SRNN (stochastic recurrent neural network

en anglais) (Fraccaro et al. 2016). En termes simples, les DSSM intègrent des réseaux de neurones profonds dans les SSMs classiques et les SVAEs modifient les RNNs pour imiter les SSMs. La principale différence entre les DSSM et les SVAE est la façon dont nous modélisons la transition de la variable latente. Dans les DSSMs, il existe un processus autonome de la variable latente, *i.e.* ils ne dépendent pas des observations. Dans les SVAEs, le prochain état latent dépend non seulement de l'état actuel, mais également de l'observation actuelle. En raison de cette différence de dépendance, la factorisation dans les deux classes de modèle est différente. Selon l'application, une approche peut fonctionner mieux que l'autre. Par exemple, pour modéliser un processus physique à partir d'une série d'observations bruitées, les DSSMs sont plus adaptés que les SVAEs, car les véritables processus cachés sont autonomes, ils suivent des lois physiques et ne sont pas interférés par les erreurs de mesure. Pour prédire la position d'un navire, bien que les états cachés puissent modéliser les modèles de déplacement, la trajectoire réelle peut être affectée par le contexte environnemental, tel qu'un vent fort ou un trafic dense, des modèles qui prennent en compte la position actuelle pour prédire la prochaine positions (SVAEs), peut être un meilleur choix.

Dans cette thèse, nous présenterons des modèles que nous avons construits sur la base de la philosophie des DSSMs et SVAEs pour des applications spécifiques. Dans le chapitre 4, nous présentons un DSSM pour l'apprentissage de systèmes dynamiques à partir d'observations bruitées et partielles. Dans le chapitre 6, nous présentons un SVAE spécialement conçu pour la surveillance du trafic maritime à l'aide des données AIS. Nous proposons également une nouvelle méthode de détection d'anomalies à l'aide des SVAEs, appliquée aux trajectoires AIS dans le chapitre 7 et à la surveillance audio dans l'annexe A.

B.2 VDL pour l'identification de systèmes dynamiques

Les systèmes dynamiques sont un excellent moyen de décrire les systèmes physiques, les systèmes biologiques, tout ce qui change dans le temps. Ils sont au cœur de nombreux domaines, tels que les géosciences, la dynamique des fluide et la dynamique aérodynamique (Arrowsmith et al. 1990; Brin et al. 2002; Hirsch et al. 2012).

Un système dynamique peut etre décrit par plusieurs types de représentations, on peut citer par exemple les équations différentielles ordinaires (ODE—ordinary differential equation en anglais), les équations différentielles stochastiques (SDE—stochastic differential

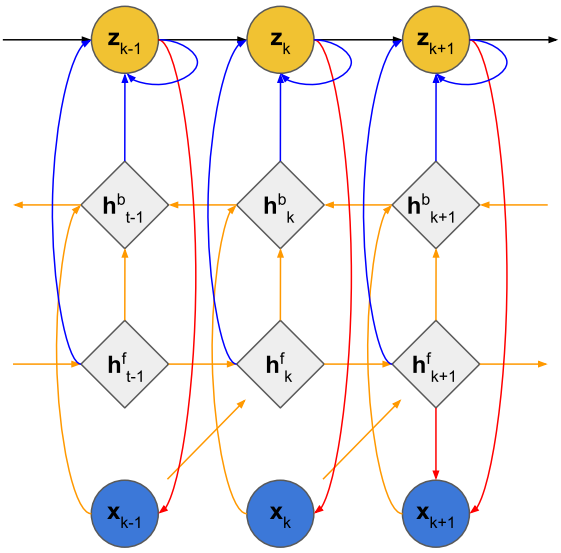


Figure B.1 – Modèle graphique de DAODEN. Le modèle génératif comprend la transition (flèches noires) et l'émission (flèches rouges). Le modèle d'inférence contient la récurrence (flèches jaunes) et l'inférence (flèches bleues).

equation en anglais), ou les équation aux dérivées partielles (PDE—partial differential equation en anglais) comme différents outilles mathématiques pouvant décrire un système qui évolue dans le temps. Ici, on considère une ODE:

$$\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} = f\Big(\mathbf{z}_t\Big) \tag{B.10}$$

où $\mathbf{z}_t \in \mathbb{R}^{d_z}$ est l'état du système, $f : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_z}$ est une fonction *déterministe*, appelée le *modèle dynamique* et t désigne le temps.

L'objectif de l'identification des systèmes dynamiques est de récupérer la dynamique de ce système à partir de certains jeux de données d'observation, c'est-à-dire d'identifier les équations gouvernantes f, compte tenu d'une série d'observations $\mathbf{x}_{0:T}$:

$$\mathbf{x}_k = \Phi_k\Big(\mathcal{H}\Big(\mathbf{z}_k\Big) + \boldsymbol{\varepsilon}_k\Big) \tag{B.11}$$

où $\mathcal{H} : \mathbb{R}^{d_z} \longrightarrow \mathbb{R}^{d_x}$ est l'opérateur d'observation, généralement connu, $\boldsymbol{\varepsilon}_k \in \mathbb{R}^{d_x}$ est un bruit additif de moyenne nulle et k se réfère au temps échantillonnage, typiquement régulier tel que $k = t_0 + k.\delta$ par rapport à une résolution temporelle fine δ. L'opérateur Φ_k tient

compte du fait que l'observation $\mathbf{x}_k$ peut être indisponible au pas de temps k ($\Phi_{k,j} = 0$ si la variable j^{th} de $\mathbf{x}_k$ est manquante).

En utilisant une formulation espace d'états, nous rencadrons le problème d'apprentissage à partir de données bruitées et/ou partielles comme un problème d'assimilation de données, avec un modèle dynamique inconnu. Nous proposons un DSSM, appelé DAODEN, spécialement conçu pour cette tâche. L'architecture de ce modèle est illustrée à la Fig. B.1. Cette architecture comprend deux sous-modules: un module dynamique, désigné par des flèches noires et un module d'inférence, désigné par les flèches bleues et oranges. Le module d'inférence fonctionne comme un schéma d'assimilation de données. Il construit les vrais états du système $\mathbf{z}_{0:T}$ à partir de la série d'observations $\mathbf{x}_{0:T}$, qui peut être bruitées et partielles. Etant donné ces états, le module dynamique exploite une architecture de pointe de réseaux de neurones pour l'identification de systèmes dynamiques afin de récupérer les équations gouvernantes. En utilisant une formulation d'espace d'états, le cadre proposé intègre des composantes stochastiques pour tenir compte des variabilités stochastiques, des erreurs de modèle et des incertitudes de reconstruction.

Nous avons testé nos modèles sur un système de Lorenz-63 (L63), un systèm de Lorenz-96 (L96) et un système de Lorenz-63 stochastiques (L63s). Les observations sont bruitées avec différent niveaux de bruit $r = std_{noise}/std_{signal}$. Nous comparons les performances des 4 modèles proposés avec celles des références: AnDA (analog data assimilation en anglais) (Lguensat et al. 2017), SINDy (sparse identification of nonlinear dynamics en anglais) (Brunton, Proctor, et al. 2016), BiNN (bi-linear neural network en anglais) (Fablet, Ouala, et al. 2018) et Latent-ODE (Rubanova et al. 2019). Les résultats sont évalués selon quatre critères: i) l'erreur de prédiction après 4 pas temps e_4, ii) La première fois où l'erreur de prédiction atteint la moitié de l'écart type du signal $\pi_{0.5}$, iii) L'erreur de reconstruction rec, et iv) la capacité à conserver la topologie à long terme du système décrite par le premier exposant de Lyapunov λ_1.

Tel que dans le Tableau. B.1, tous les modèles proposés surpassent largement les méthodes de références. Cela affirme la capacité de la méthode proposée à traiter des observations bruitées.

D'autres expériences sur des données L63 partiellement observées, sur des données L96 bruitées et sur des données L63 bruitées démontrent que la méthode proposée peut s'appliquer avec des données partielles, de grande dimension ou stochastiques.

Table B.1 – Performances des modèles entraînés sur des données L63 bruitées. BiNN_EnKS, DAODEN_determ, DAODEN_MAP et DAODEN_full sont des versions différents du modèle proposé.

Model		r 8.5%	16.7%	33.3%	66.7%
AnDA	e_4	0.351±0.184	0.777±0.350	1.683±0.724	3.682±1.346
	rec	0.416±0.019	0.941±0.037	2.134±0.076	4.876±0.168
	$\pi_{0.5}$	0.820±0.480	0.380±0.172	0.249±0.174	0.104±0.116
	λ_1	26.517±7.665	27.146±42.927	76.267±28.150	127.047±0.881
SINDy	e_4	0.068±0.052	0.149±0.106	0.311±0.196	0.694±0.441
	$\pi_{0.5}$	0.490±0.261	0.165±0.085	0.077±0.049	0.034±0.034
	λ_1	0.898±0.008	0.840±0.035	0.840±0.035	nan±nan
BiNN	e_4	0.045±0.030	0.119±0.085	0.283±0.185	0.684±0.408
	$\pi_{0.5}$	3.608±1.364	2.053±0.666	0.975±0.488	0.308±0.125
	λ_1	0.900±0.011	0.868±0.010	0.122±0.208	-0.422±0.047
Latent-ODE	e_4	0.051±0.027	0.062±0.034	*0.065±0.042*	*0.213±0.084*
	$\pi_{0.5}$	2.504±1.332	2.336±1.472	2.852±1.352	2.118±1.129
	λ_1	0.892±0.018	0.877±0.018	0.885±0.015	0.675±0.027
BiNN_EnKS	e_4	**0.019±0.016**	**0.024±0.023**	**0.037±0.024**	0.276±0.160
	rec	0.323±0.024	0.431±0.042	0.598±0.093	1.531±0.332
	$\pi_{0.5}$	2.807±1.128	3.004±1.355	2.996±1.641	*2.081±1.214*
	λ_1	0.856±0.031	0.869±0.024	0.826±0.065	0.868±0.014
DAODEN_determ	e_4	0.049±0.031	0.056±0.034	0.077±0.048	0.268±0.201
	rec	0.216±0.125	0.269±0.110	**0.448±0.199**	*0.873±0.216*
	$\pi_{0.5}$	*3.519±1.282*	*3.488±1.327*	**3.470±1.562**	1.803±1.104
	λ_1	0.882±0.036	0.895±0.021	0.911±0.013	0.793±0.021
DAODEN_MAP	e_4	0.038±0.027	0.038±0.038	0.101±0.070	0.233±0.088
	rec	*0.209±0.096*	**0.234±0.065**	0.525±0.253	**0.817±0.330**
	$\pi_{0.5}$	3.271±1.270	3.219±1.260	2.993±1.413	**2.650±1.382**
	λ_1	0.860±0.047	0.876±0.029	0.916±0.012	0.920±0.008
DAODEN_full	e_4	*0.023±0.015*	*0.027±0.016*	0.072±0.045	**0.187±0.127**
	rec	**0.178±0.050**	*0.258±0.066*	*0.469±0.168*	1.003±0.380
	$\pi_{0.5}$	**3.533±1.139**	**3.496±1.215**	*3.426±1.512*	1.897±0.918
	λ_1	0.869±0.036	0.858±0.028	0.881±0.024	0.884±0.013

B.3 VDL pour la surveillance du trafic maritime

Dans l'application précédente, nous avons utilisé des DSSMs. L'autre approche, les SVAEs, sera exploitée dans la deuxième application: la surveillance du trafic maritime à l'aide des données AIS.

L'AIS est un système de communication conçu pour les navires. L'objectif initial de l'AIS est d'éviter les collisions. Cependant, grâce à sa richesse d'information, l'AIS est rapidement devenu l'une des sources d'information les plus importantes dans le trafic maritime. En gros, à partir des données AIS, nous pouvons obtenir l'identité du navire, la position actuelle, la vitesse actuelle, le cap actuel et d'autres informations du navire ainsi que du voyage. Les informations fournies par l'AIS sont utiles pour de nombreuses tâches de surveillance maritime, telles que la prédiction de la trajectoire des navires, l'identification des routes maritimes, l'identification du type de navire, la detection d'anomalies, *etc.* Cependant, il est difficile d'exploiter efficacement les données AIS, car i) la quantité de données est massive, nous ne pouvons pas traiter les données AIS manuellement, ii) aucun jeu de données de vérité terrain n'est disponible, ce qui nous empêche d'utiliser des méthodes supervisées, et iii) les données AIS sont inondées de bruit et peuvent être échantillonnées irrégulièrement.

Dans ce travail de thèse, nous abordons ces problèmes en exploitant un SVAE pour développer un système automatique qui peut traiter et détecter, extraire et caractériser des informations utiles dans les données AIS pour la surveillance maritime.

L'architecture de MultitaskAIS présentée dans la Fig. B.2. Le noyau du modèle est un VRNN, qui apprend une distribution probabiliste décrivant les trajectoires AIS dans l'ensemble d'apprentissage. D'autres sous-modules spécifiques à une tâche telle que la reconstruction de trajectoire, la détection d'anomalies, l'identification du type de navire, *etc.* sont construits sur cette couche.

L'un des avantages de cette architecture est que la quantité massive de données AIS peut être intégrée dans un nombre beaucoup plus restreint de paramètres du VRNN. Le VRNN peut également prendre en compte les variations et les incertitudes des données AIS. Ces informations sont utiles pour les sous-modules supérieurs. Nous avons testé le modèle sur un véritable ensemble de données AIS contenant plus de 4,2 millions de messages AIS. MultitaskAIS atteint des performances identiques ou meilleures que les méthodes de pointe dans toutes les tâches.

Le sous-module de détection d'anomalies de MultitaskAIS s'appelle GeoTrackNet. Il

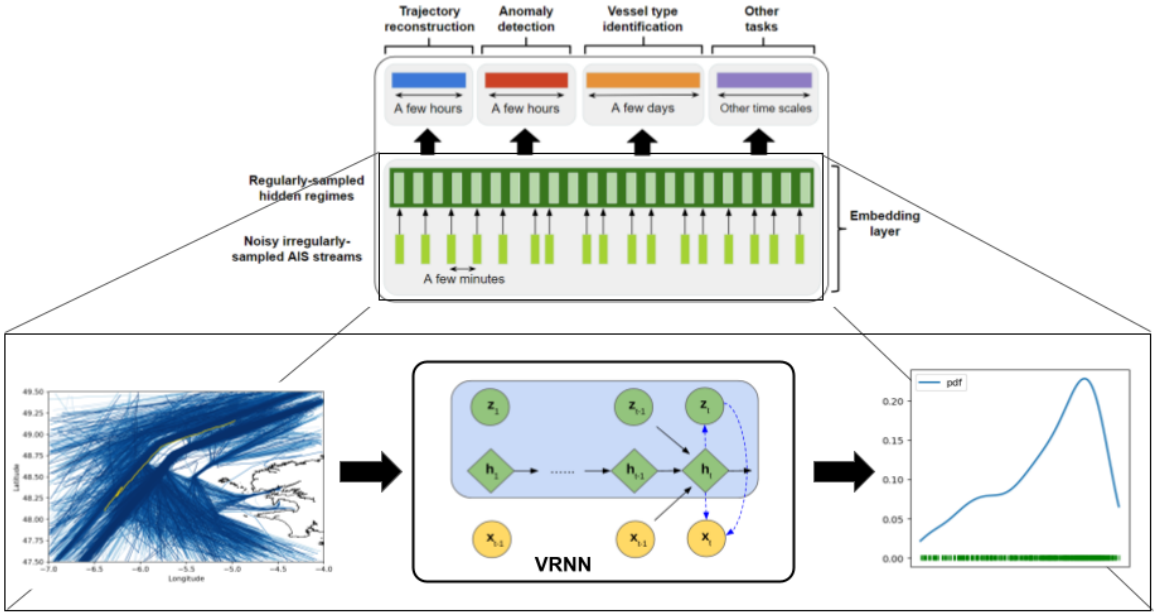


Figure B.2 – Architecture de MultitaskAIS.

s'agit d'un détecteur *"a contrario"* qui prend en compte le fait que les données AIS sont géospatialement dépendantes, de sorte que les performances de la distribution (apprise par le VRNN) est géospatialement dépendantes aussi. GeoTrackNet divise la ROI (region of interest en anglais) en petites cellules, et considère tout message AIS qui a une probabilité relativement plus faible que les autres messages AIS dans la même cellule comme anormal. Toute trajectoire comportant de nombreux messages AIS anormaux sera considérée comme anormale.

La Fig. B.3 montre des trajectoires anormales détectées par GeoTrackNet autour d'Ouessant, de janvier à mars 2017. Le modèle proposé peut détecter à la fois: i) des anomalies spatiales (géométriques et géographiques), lorsque les navires dévient des routes maritimes, effectuent des virages inhabituels, *etc.* et ii) des anomalies de phase (cinétiques), lorsque les navires ont une évolution anormale de leur vitesse et de leur cap (par exemple, ralentissement inhabituel, changements brusques de vitesse, etc.).

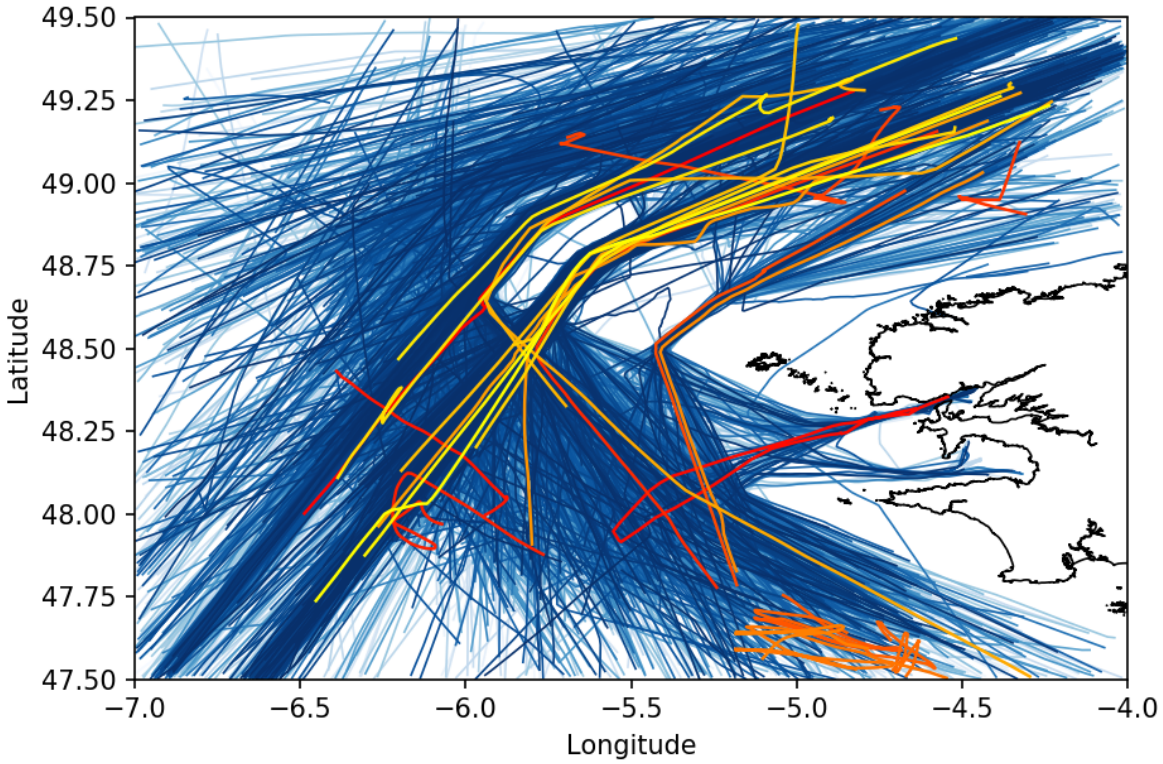


Figure B.3 – Trajectoires anormales détectées par GeoTrackNet. Bleu: trajectoires dans la base d'apprentissage; autres couleurs: trajectoires anormales dans la base de test.

B.4 Conclusion et perspectives

B.4.1 Conclusion

Dans cette thèse on a étudié l'apprentissage profond variationnel (VDL) pour la modélisation et l'analyse de séries temporelles. Les méthodes VDL de pointe actuelles pour la modélisation de séries temporelles peuvent être classées en deux classes: DSSM et SVAE. DSSM est une version améliorée du SSM classique. En utilisant des réseaux de neurones pour paramétrer les distributions, le DSSM surmonte les difficultés de non-linéarité. SVAE étend RNN en ajoutant des composants stochastiques pour surmonter la limite des transitions déterministes. Les deux sont puissants pour capturer les dépendances à long terme dans des séries temporelles hautement non linéaires, bruitées et échantillonnées de manière irrégulière. Cependant, selon les applications spécifiques, l'une peut être plus appropriée que l'autre.

Nous avons proposé de nouvelles méthodes VDL pour deux applications spécifiques : l'identification dynamique des systèmes et la surveillance du trafic maritime. Plus précisément :

— Nous avons proposé une méthode générale d'apprentissage profond—appelé DAODEN—pour l'apprentissage de systèmes dynamiques chaotiques et potentiellement stochastiques. Cette méthode utilise une formulation de modèle d'espace d'état profond pour récupérer les équations différentielles inconnues qui régissent les données de l'ensemble d'apprentissage. En combinant l'assimilation des données classiques et les techniques modernes d'apprentissage automatique (apprentissage profond), DAODEN peut considérablement améliorer les performances des modèles d'apprentissage de pointe actuelle dans des conditions imparfaites, c'est-à-dire des observations bruitées et partielles. De plus, étant donné que DAODEN intègre des composants stochastiques pour tenir compte des erreurs modèle et des incertitudes de reconstruction, cette méthode peut appliquer à des systèmes dynamiques stochastiques.

— Nous avons proposé une architecture d'apprentissage profond—appelée MultitaskAIS—pour la surveillance maritime utilisant les données AIS. Le composant clé de MultitaskAIS est un VRNN. Ce modèle apprend une distribution probabiliste décrivant les trajectoires AIS dans l'ensemble d'apprentissage. De nombreux sous-modèles spécifiques aux tâches peuvent être construits au-dessus de cette couche. Nous avons démontré que MultitaskAIS pouvait atteindre des performances de pointe sur trois tâches: reconstruction de trajectoire, détection d'anomalies d'identification du type

de navire, tout en réduisant considérablement les besoins de stockage et de calcul. Le sous-modèle le plus important de MultitaskAIS est le modèle de détection d'anomalies—appelé GeoTrackNet. Ce modèle exploite la représentation probabiliste donnée par le VRNN et utilise un détecteur géospatial *a contrario* pour détecter les comportements anormaux des navires. Le détecteur *a contrario* de GeoTrackNet prend en compte le fait que les données AIS sont géospatialement dépendantes, de sorte que les performances du VRNN dépendent également géospatialement. GeoTrackNet est la première application réussie de DL dans la détection d'anomalies du trafic maritime.

B.4.2 Perspectives

Les résultats présentés dans cette thèse ont établi des avancées pour les problèmes considérés, mais ils soulèvent également un nouvel ensemble de questions ouvertes (et probablement plus difficiles) qui restent pour les futurs explorateurs. Concernant la partie théorique :

— Pour les problèmes considérés dans cette thèse, les états et les observations sont des vecteurs unidimensionnels. Le passage dans un espace dimensionnel élevé (par exemple, lorsque les observations sont des images 2D) nécessiterait des études complémentaires pour réduire la complexité de la représentation et du calcul.

— Dans cette thèse, nous avons utilisé les LSTMs pour capturer des corrélations à long terme dans les données. Cependant, si nous avons une connaissance préalable de la "longueur" des dépendances temporelles, des architectures parallèles telles que transformer (Vaswani et al. 2017) accéléreront le temps de calcul.

Concernant la partie d'identification de systèmes dynamiques :

— Nous avons examiné des cas où les observations sont partielles en ce sens que certaines composantes des observations peuvent manquer, tant dans les dimensions spatiales que temporelles, cependant, toutes les composantes des états du système sont vues au moins une fois. Il existe également des situations où certains composants des systèmes ne sont jamais observés. Pour ces cas, nous devons exploiter des états augmentés (Abarbanel et al. 1994; Robinson 2005; Ayed et al. 2019; Ouala, Duong Nguyen, Herzet, et al. 2019).

— Il est crucial d'appliquer des contraintes physiques afin d'obtenir un modèle appris physiquement significatif. Les travaux préliminaires dans (Cockburn et al. 1990;

Raissi, Perdikaris, and George E. Karniadakis 2019; Bézenac et al. 2019) pourraient être étudiés dans de futures études.

Concernant la partie de surveillance du trafic maritime:

— L'AIS est un système d'auto-déclaration. Bien que MultitaskAIS et GeoTrackNet puissent gérer les données manquantes, si un navire coupe son signal AIS pendant une longue période, les modèles basés sur l'AIS ne peuvent pas s'appliquer. Les modèles de détection AIS on-off, tels que ceux de (Fabio Mazzarella et al. 2017; Kontopoulos et al. 2020) sont nécessaires pour la surveillance maritime.

— Bien que les performances de GeoTrackNet soient prometteus et que les trajectoires AIS détectées ont été validées par des experts AIS indépendants, nous ne savons pas ce que le détecteur peut manquer. Des données étiquetées et bien préparées seraient très utiles pour que la communauté fasse progresser les progrès récents.

Bibliography

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard (2016), « Tensorflow: A system for large-scale machine learning », *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283.

Abarbanel, Henry DI, T. A. Carroll, L. M. Pecora, J. J. Sidorowich, and L. Sh Tsimring (1994), « Predicting physical variables in time-delay embedding », *Physical Review E* 49.3, p. 1840.

Agnew, David J., John Pearce, Ganapathiraju Pramod, Tom Peatman, Reg Watson, John R. Beddington, and Tony J. Pitcher (Feb. 2009), « Estimating the Worldwide Extent of Illegal Fishing », *PLOS ONE* 4.2, Publisher: Public Library of Science, e4570.

Ammar, M. and S. Le Hegarat-Mascle (Dec. 2013), « An A-Contrario Approach for Object Detection in Video Sequence », *International Journal of Pure and Apllied Mathematics* 89.2.

Arguedas, V. Fernandez, G. Pallotta, and M. Vespe (Mar. 2018), « Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring », *IEEE Transactions on Intelligent Transportation Systems* 19.3, pp. 722–732.

Arrowsmith, David K., Colin M. Place, and C. H. Place (1990), *An introduction to dynamical systems*, Cambridge university press.

Atrey, P.K., N.C. Maddage, and M.S. Kankanhalli (2006), « Audio Based Event Detection for Multimedia Surveillance », *2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings*, vol. 5, Toulouse, France: IEEE, pp. V–813–V–816.

Ayed, Ibrahim, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari (Feb. 2019), « Learning Dynamical Systems from Partial Observations », *arXiv:1902.11136 [physics]*, arXiv: 1902.11136.

Barbos, Andrei-Cristian, Francois Caron, Jean-François Giovannelli, and Arnaud Doucet (2017), « Clone MCMC: Parallel High-Dimensional Gaussian Gibbs Sampling », *Advances in Neural Information Processing Systems 30*, ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Curran Associates, Inc., pp. 5020–5028.

Barker, Jon, Emmanuel Vincent, Ning Ma, Heidi Christensen, and Phil Green (May 2013), « The PASCAL CHiME speech separation and recognition challenge », *Computer Speech & Language*, Special Issue on Speech Separation and Recognition in Multisource Environments 27.3, pp. 621–633.

Bayer, Justin and Christian Osendorfer (Nov. 2014), « Learning Stochastic Recurrent Networks », *arXiv:1411.7610 [cs, stat]*, arXiv: 1411.7610.

Bengio, Y., A. Courville, and P. Vincent (Aug. 2013), « Representation Learning: A Review and New Perspectives », *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1798–1828.

Best, R. A. and J. P. Norton (July 1997), « A new model and efficient tracker for a target with curvilinear motion », *IEEE Transactions on Aerospace and Electronic Systems* 33.3, pp. 1030–1037.

Bézenac, Emmanuel de, Arthur Pajot, and Patrick Gallinari (Dec. 2019), « Deep learning for physical processes: incorporating prior scientific knowledge », *Journal of Statistical Mechanics: Theory and Experiment* 2019.12, p. 124009.

Biancamaria, Sylvain, Dennis P. Lettenmaier, and Tamlin M. Pavelsky (2016), « The SWOT mission and its capabilities for land hydrology », *Surveys in Geophysics* 37.2, pp. 307–337.

Bishop, Christopher (2006), *Pattern Recognition and Machine Learning*, Information Science and Statistics, New York: Springer-Verlag.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (2017), « Variational inference: A review for statisticians », *Journal of the American statistical Association* 112.518, pp. 859–877.

Bocquet, Marc, Julien Brajard, Alberto Carrassi, and Laurent Bertino (July 2019), « Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models », *Nonlinear Processes in Geophysics* 26.3, pp. 143–162.

— (2020), « Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization », *Foundations of Data Science* 2.1, arXiv: 2001.06270, pp. 55–80.

Bomberger, N. A., B. J. Rhodes, M. Seibert, and A. M. Waxman (July 2006), « Associative Learning of Vessel Motion Patterns for Maritime Situation Awareness », *2006 9th International Conference on Information Fusion*, pp. 1–8.

Bottou, Léon, Frank E. Curtis, and Jorge Nocedal (2018), « Optimization methods for large-scale machine learning », *Siam Review* 60.2, pp. 223–311.

Bouritsas, Giorgos, Stelios Daveas, Antonios Danelakis, and Stelios CA Thomopoulos (2019), « Automated Real-time Anomaly Detection in Human Trajectories using Sequence to Sequence Networks », *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, pp. 1–8.

Brajard, J., A. Carrassi, M. Bocquet, and L. Bertino (2019), « Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model », *Geoscientific Model Development Discussions* 2019, pp. 1–21.

Bresnahan, Philip J., Taylor Wirth, Todd Martz, Kenisha Shipley, Vicky Rowley, Clarissa Anderson, and Thomas Grimm (2020), « Equipping smart coasts with marine water quality IoT sensors », *Results in Engineering* 5, p. 100087.

Brin, Michael and Garrett Stuck (2002), *Introduction to dynamical systems*, Cambridge university press.

Brown, Robert Grover and Patrick Y C Hwang (n.d.), « Introduction to Random Signals and Applied Kalman Filtering » (), p. 3.

Brunton, Steven L. and J. Nathan Kutz (2019), *Data-driven science and engineering: Machine learning, dynamical systems, and control*, Cambridge University Press.

Brunton, Steven L., Joshua L. Proctor, and J. Nathan Kutz (Apr. 2016), « Discovering governing equations from data by sparse identification of nonlinear dynamical systems », *Proceedings of the National Academy of Sciences* 113.15, pp. 3932–3937.

Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov (Nov. 2016), « Importance Weighted Autoencoders », *arXiv:1509.00519 [cs, stat]*, arXiv: 1509.00519.

Champion, Kathleen, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton (Mar. 2019), « Data-driven discovery of coordinates and governing equations », *Proceedings of the National Academy of Sciences* 116, pp. 22445–22451.

Chapron, B., P. Dérian, E. Mémin, and V. Resseguier (2018), « Large-scale flows under location uncertainty: a consistent stochastic framework », *Quarterly Journal of the Royal Meteorological Society* 144.710, pp. 251–260.

Chen, Liqun, Chenyang Tao, Ruiyi Zhang, Ricardo Henao, and Lawrence Carin Duke (2018), « Variational inference and model selection with generalized evidence bounds », *International conference on machine learning*, pp. 893–902.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud (June 2018), « Neural Ordinary Differential Equations », *arXiv:1806.07366 [cs, stat]*, arXiv: 1806.07366.

Chen, Tianqi, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang (2015), « Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems », *arXiv preprint arXiv:1512.01274*.

Chung, J., K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio (June 2015), « A Recurrent Latent Variable Model for Sequential Data », *Advances in neural information processing systems*, pp. 2980–2988.

Chung, Junyoung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio (2015), « Gated Feedback Recurrent Neural Networks », *International Conference on Machine Learning*, p. 9.

Clifton, D. A. and L. Tarassenko (Aug. 2015), « Novelty detection in jet engine vibration spectra », *International Journal of Condition Monitoring* 5, pp. 2–7.

Cockburn, Bernardo, Suchung Hou, and Chi-Wang Shu (1990), « The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case », *Mathematics of Computation* 54.190, pp. 545–581.

Coscia, P., P. Braca, L. M. Millefiori, F. A. N. Palmieri, and P. Willett (2018), « Multiple Ornstein-Uhlenbeck Processes for Maritime Traffic Graph Representation », *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1.

Courtier, P., J.-N. Thépaut, and A. Hollingsworth (1994), « A strategy for operational implementation of 4D-Var, using an incremental approach », *Quarterly Journal of the Royal Meteorological Society* 120.519, pp. 1367–1387.

d'Afflisio, E., P. Braca, L. M. Millefiori, and P. Willett (July 2018), « Maritime Anomaly Detection Based on Mean-Reverting Stochastic Processes Applied to a Real-World Scenario », *2018 21st International Conference on Information Fusion (FUSION)*, pp. 1171–1177.

d'Afflisio, E., P. Braca, L. M. Millefiori, and P. Willett (Dec. 2018), « Detecting Anomalous Deviations From Standard Maritime Routes Using the Ornstein–Uhlenbeck Process », *IEEE Transactions on Signal Processing* 66.24, pp. 6474–6487.

De Brouwer, Edward, Jaak Simm, Adam Arany, and Yves Moreau (Nov. 2019), « GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series », *Advances in Neural Information Processing Systems*, arXiv: 1905.12374.

Dechter, Rina (1986), « Learning while searching in constraint-satisfaction problems ».

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977), « Maximum likelihood from incomplete data via the EM algorithm », *Journal of the Royal Statistical Society, Series B* 39.1, pp. 1–38.

Desolneux, Agnés, Lionel Moisan, and Jean-Michel Morel (2008), *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, ed. by S. S. Antman, L. Sirovich, J. E. Marsden, and S. Wiggins, vol. 34, Interdisciplinary Applied Mathematics, New York, NY: Springer New York.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018), « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding », *arXiv:1810.04805 [cs]*, arXiv: 1810.04805.

Dobrkovic, Andrej, Maria-Eugenia Iacob, and Jos van Hillegersberg (Mar. 2018), « Maritime pattern extraction and route reconstruction from incomplete AIS data », *International Journal of Data Science and Analytics* 5.2, pp. 111–136.

Doucet, Arnaud and Adam M Johansen (2009), « A tutorial on particle filtering and smoothing: Fifteen years later », *Handbook of nonlinear filtering* 12.656-704, p. 3.

Dremeau, A., C. Herzet, and L. Daudet (July 2012), « Boltzmann Machine and Mean-Field Approximation for Structured Sparse Decompositions », *IEEE Transactions on Signal Processing* 60.7, pp. 3425–3438.

Duchi, John, Elad Hazan, and Yoram Singer (2011), « Adaptive subgradient methods for online learning and stochastic optimization. », *Journal of machine learning research* 12.7.

Elman, Jeffrey L. (1990), « Finding structure in time », *Cognitive science* 14.2, pp. 179–211.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu (1996), « A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise », *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, Portland, Oregon: AAAI Press, pp. 226–231.

Esteva, Andre, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean (Jan. 2019), « A guide to deep learning in healthcare », *Nature Medicine* 25.1, pp. 24–29.

Evensen, Geir (Nov. 2003), « The Ensemble Kalman Filter: theoretical formulation and practical implementation », *Ocean Dynamics* 53.4, pp. 343–367.

— (Aug. 2009), *Data Assimilation: The Ensemble Kalman Filter*, Google-Books-ID: 2_zaTb_O1AkC, Springer Science & Business Media.

Evensen, Geir and Peter Jan van Leeuwen (June 2000), « An Ensemble Kalman Smoother for Nonlinear Dynamics », *Monthly Weather Review* 128.6, pp. 1852–1867.

Fablet, Ronan, Lucas Drumetz, and Francois Rousseau (June 2020), « Joint learning of variational representations and solvers for inverse problems with partially-observed data », *arXiv:2006.03653 [cs, eess, stat]*, arXiv: 2006.03653.

Fablet, Ronan, Said Ouala, and Cédric Herzet (Sept. 2018), « Bilinear Residual Neural Network for the Identification and Forecasting of Geophysical Dynamics », *2018 26th European Signal Processing Conference (EUSIPCO)*, ISSN: 2219-5491, pp. 1477–1481.

Féron, Olivier, François Orieux, and Jean-François Giovannelli (Mar. 2016), « Gradient Scan Gibbs Sampler: An Efficient Algorithm for High-Dimensional Gaussian Distributions », *IEEE Journal of Selected Topics in Signal Processing* 10.2, pp. 343–352.

Flory, Paul J. (1942), « Thermodynamics of high polymer solutions », *The Journal of chemical physics* 10.1, pp. 51–61.

Forti, N., L. M. Millefiori, P. Braca, and P. Willett (May 2019), « Anomaly Detection and Tracking Based on Mean–Reverting Processes with Unknown Parameters », *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8449–8453.

Fraccaro, Marco (2018), *Deep Latent Variable Models for Sequential Data*, DTU Compute.

Fraccaro, Marco, Sø ren Kaae Sø nderby, Ulrich Paquet, and Ole Winther (2016), « Sequential Neural Models with Stochastic Layers », *Advances in Neural Information Processing Systems*, Curran Associates, Inc., pp. 2199–2207.

Gaspar, Philippe, Rémy Lopez, Marza Marzuki, Ronan Fablet, Philippe Gros, Jean-Michel Zigna, and Gaetan Fabritius (July 2016), « Analysis of Vessel Trajectories for Maritime Surveillance and Fisheries Management », *Maritime Knowledge Discovery and Anomaly Detection Workshop*, Joint Research Centre, ISPRA, Italy.

Geman, Stuart and Donald Geman (1984), « Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images », *IEEE Transactions on pattern analysis and machine intelligence* 6, pp. 721–741.

Gershgorin, Boris, John Harlim, and Andrew J. Majda (2010), « Test models for improving filtering with model errors through stochastic parameter estimation », *Journal of Computational Physics* 229.1, pp. 1–31.

Gershman, Samuel, Matt Hoffman, and David Blei (2012), « Nonparametric variational inference », *arXiv preprint arXiv:1206.4665*.

Ghahramani, Zoubin and Geoffrey E. Hinton (1996), *Parameter estimation for linear dynamical systems*, tech. rep., Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science.

Ghahramani, Zoubin and Sam T. Roweis (1999), « Learning Nonlinear Dynamical Systems Using an EM Algorithm », *Advances in Neural Information Processing Systems 11*, ed. by M. J. Kearns, S. A. Solla, and D. A. Cohn, MIT Press, pp. 431–437.

Giannakopoulos, Theodoros, Alexandros Makris, Dimitrios Kosmopoulos, Stavros Perantonis, and Sergios Theodoridis (2010), « Audio-Visual Fusion for Detecting Violent Scenes in Videos », *Artificial Intelligence: Theories, Models and Applications*, vol. 6040, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 91–100.

Goh, Garrett B., Nathan O. Hodas, and Abhinav Vishnu (2017), « Deep learning for computational chemistry », *Journal of Computational Chemistry* 38.16, pp. 1291–1307.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2014), « Explaining and harnessing adversarial examples », *arXiv preprint arXiv:1412.6572*.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016), *Deep learning*, MIT press.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014), « Generative adversarial nets », *Advances in neural information processing systems*, pp. 2672–2680.

He, K., X. Zhang, S. Ren, and J. Sun (Dec. 2015), « Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification », *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034.

Hexeberg, S., A. L. Flåten, B. O. H. Eriksen, and E. F. Brekke (July 2017), « AIS-based vessel trajectory prediction », *2017 20th International Conference on Information Fusion (Fusion)*, pp. 1–8.

Hilborn, Robert C (2000), *Chaos and nonlinear dynamics: an introduction for scientists and engineers*, Oxford University Press on Demand.

Hinton, Geoffrey (2012), *Lecture 6e rmsprop: Divide the gradient by a running average of its recent magnitude.*

Hirsch, Morris W, Stephen Smale, and Robert L Devaney (2012), *Differential equations, dynamical systems, and an introduction to chaos*, Academic press.

Hochreiter, Sepp and Jürgen Schmidhuber (1997), « Long short-term memory », *Neural computation* 9.8, pp. 1735–1780.

Hoffman, Matt, David M. Blei, Chong Wang, and John Paisley (June 2012), « Stochastic Variational Inference », *arXiv:1206.7051 [cs, stat]*, arXiv: 1206.7051.

Holst, Anders, Peter Ryman, and Anders Linse (July 2016), « Stattistical Anomaly Detection for Maritime Surveillance and Monitoring », *Maritime Knowledge Discovery and Anomaly Detection Workshop*, Joint Research Centre, ISPRA, Italy.

Hoshiya, Masaru and Etsuro Saito (1984), « Structural identification by extended Kalman filter », *Journal of engineering mechanics* 110.12, pp. 1757–1770.

Huggins, Maurice L. (1941), « Solutions of long chain compounds », *The Journal of chemical physics* 9.5, pp. 440–440.

IMO (2020), *IMO profile.*

— (2017), *International Convention for the Safety of Life at Sea (SOLAS), 1974.*

Iphar, Clément, Cyril Ray, and Aldo Napoli (June 2019), « Uses and Misuses of the Automatic Identification System », *OCEANS 2019 - Marseille*, pp. 1–10.

— (June 2020), « Data integrity assessment for maritime anomaly detection », *Expert Systems with Applications* 147, p. 113219.

Isern-Fontanet, J. and Erwan Hascoët (2014), « Diagnosis of high-resolution upper ocean dynamics from noisy sea surface temperatures », *Journal of Geophysical Research: Oceans* 119.1, pp. 121–132.

Jaeger, Herbert (2002), *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*, vol. 5, GMD-Forschungszentrum Informationstechnik Bonn.

Jiang, Xiang, Erico N. de Souza, Ahmad Pesaranghader, Baifan Hu, Daniel L. Silver, and Stan Matwin (May 2017), « TrajectoryNet: An Embedded GPS Trajectory Representation for Point-based Classification Using Recurrent Neural Networks », *arXiv:1705.02636 [cs]*, arXiv: 1705.02636.

Johansson, F. and G. Falkman (Dec. 2007), « Detection of vessel anomalies - a Bayesian network approach », *Sensor Networks and Information 2007 3rd International Conference on Intelligent Sensors*, pp. 395–400.

Johnson, C., N. K. Nichols, and B. J. Hoskins (2005), « Very large inverse problems in atmosphere and ocean modelling », *International journal for numerical methods in fluids* 47.8-9, pp. 759–771.

Kalman, Rudolph Emil (1960), « A new approach to linear filtering and prediction problems ».

Kamilaris, Andreas and Francesc X. Prenafeta-Boldú (Apr. 2018), « Deep learning in agriculture: A survey », *Computers and Electronics in Agriculture* 147, pp. 70–90.

Karras, Tero, Samuli Laine, and Timo Aila (2019), « A style-based generator architecture for generative adversarial networks », *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4401–4410.

Kawaguchi, Yohei (Sept. 2018), « Anomaly Detection Based on Feature Reconstruction from Subsampled Audio Signals », *2018 26th European Signal Processing Conference (EUSIPCO)*, Rome: IEEE, pp. 2524–2528.

Kazemi, Samira, Shahrooz Abghari, Niklas Lavesson, Henric Johnson, and Peter Ryman (Oct. 2013), « Open data for anomaly detection in maritime surveillance », *Expert Systems with Applications* 40.14, pp. 5719–5729.

Khare, Shree P., Jeffrey L. Anderson, Timothy J. Hoar, and Douglas Nychka (Jan. 2008), « An investigation into the application of an ensemble Kalman smoother to high-dimensional geophysical systems », *Tellus A: Dynamic Meteorology and Oceanography* 60.1, pp. 97–112.

Kingma, Diederik P. and Jimmy Ba (2015), « Adam: A Method for Stochastic Optimization », *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kingma, Diederik P. and Max Welling (Dec. 2013), « Auto-Encoding Variational Bayes », *arXiv:1312.6114 [cs, stat]*, arXiv: 1312.6114.

Kingma, Durk P., Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling (2014), « Semi-supervised learning with deep generative models », *Advances in neural information processing systems*, pp. 3581–3589.

Kloeden, Peter E. and Eckhard Platen (2013), *Numerical solution of stochastic differential equations*, vol. 23, Springer Science & Business Media.

Kontopoulos, Ioannis, Konstantinos Chatzikokolakis, Dimitris Zissis, Konstantinos Tserpes, and Giannis Spiliopoulos (2020), « Real-time maritime anomaly detection: detecting intentional AIS switch-off », *International Journal of Big Data Intelligence* 7.2, pp. 85–96.

Kowalska, K. and L. Peel (July 2012), « Maritime anomaly detection using Gaussian Process active learning », *2012 15th International Conference on Information Fusion*, pp. 1164–1171.

Krishnan, R. G., U. Shalit, and D. Sontag (Feb. 2017), « Deep Kalman Filters », *AAAI Conference on Artificial Intelligence*.

Krishnan, Rahul G., Uri Shalit, and David Sontag (Sept. 2016), « Structured Inference Networks for Nonlinear State Space Models », *arXiv:1609.09869 [cs, stat]*, arXiv: 1609.09869.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012), « ImageNet Classification with Deep Convolutional Neural Networks », *Advances in Neural Information Processing Systems 25*, ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Curran Associates, Inc., pp. 1097–1105.

Kumar, P., A. Mittal, and P. Kumar (Dec. 2005), « A Multimodal Audio Visible and Infrared Surveillance System (MAVISS) », *2005 3rd International Conference on Intelligent Sensing and Information Processing*, pp. 151–156.

Lahoz, Boris Khattatov William and Richard Menard (2010), *Data assimilation*, Springer.

Landau, Lev Davidovich (1937), « On the theory of phase transitions. I. », *Zh. Eksp. Teor. Fiz.* 11, p. 19.

Laxhammar, R. (June 2008), « Anomaly detection for sea surveillance », *2008 11th International Conference on Information Fusion*, pp. 1–8.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (May 2015), « Deep learning », *Nature* 521.7553, pp. 436–444.

Lee, Jae-Gil, Jiawei Han, and Kyu-Young Whang (2007), « Trajectory Clustering: A Partition-and-group Framework », *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, New York, NY, USA: ACM, pp. 593–604.

Lguensat, Redouane, Pierre Tandeo, Pierre Ailliot, Manuel Pulido, and Ronan Fablet (Oct. 2017), « The Analog Data Assimilation », *Monthly Weather Review* 145.10, pp. 4093–4107.

Li, Hong, Eugenia Kalnay, and Takemasa Miyoshi (2009), « Simultaneous estimation of covariance inflation and observation errors within an ensemble Kalman filter », *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 135.639, pp. 523–533.

Lichtenberg, Allan J. and Michael A. Lieberman (2013), *Regular and chaotic dynamics*, vol. 38, Springer Science & Business Media.

Locatello, Francesco, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem (2019), « Challenging common assumptions in the unsupervised learning of disentangled representations », *international conference on machine learning*, pp. 4114–4124.

Lorenz, Edward N. (Mar. 1963), « Deterministic Nonperiodic Flow », *Journal of the Atmospheric Sciences* 20.2, pp. 130–141.

— (1996), « Predictability: A problem partly solved », *Seminar on predictability*, vol. 1.

Ma, Zhanyu, Jiyang Xie, Yuping Lai, Jalil Taghia, Jing-Hao Xue, and Jun Guo (2019), « Insights into multiple/single lower bound approximation for extended variational inference in non-Gaussian structured data modeling », *IEEE Transactions on Neural Networks and Learning Systems*.

Maddison, Chris J., Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh (May 2017), « Filtering Variational Objectives », *Advances in Neural Information Processing Systems*, pp. 6576–6586.

Marchi, E., F. Vesperini, F. Eyben, S. Squartini, and B. Schuller (Apr. 2015), « A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks », *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1996–2000.

Marchi, E., F. Vesperini, F. Weninger, F. Eyben, S. Squartini, and B. Schuller (July 2015), « Non-linear prediction with LSTM recurrent neural networks for acoustic novelty detection », *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7.

Marcus, Gary (Jan. 2018), « Deep Learning: A Critical Appraisal », *arXiv:1801.00631 [cs, stat]*, arXiv: 1801.00631.

Mascaro, Steven, Ann E. Nicholso, and Kevin B. Korb (Jan. 2014), « Anomaly detection in vessel tracks using Bayesian networks », *International Journal of Approximate Reasoning*, Applications of Bayesian Networks 55.1, Part 1, pp. 84–98.

Mazzarella, F., V. F. Arguedas, and M. Vespe (Oct. 2015), « Knowledge-based vessel position prediction using historical AIS data », *2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6.

Mazzarella, F., M. Vespe, D. Damalas, and G. Osio (July 2014), « Discovering vessel activities at sea using AIS data: Mapping of fishing footprints », *17th International Conference on Information Fusion (FUSION)*, pp. 1–7.

Mazzarella, Fabio, Michele Vespe, Alfredo Alessandrini, Dario Tarchi, Giuseppe Aulicino, and Antonio Vollero (2017), « A novel anomaly detection approach to identify intentional AIS on-off switching », *Expert Systems with Applications* 78, pp. 110–123.

McDermott, Patrick L. and Christopher K. Wikle (2016), « A model-based approach for analog spatio-temporal dynamic forecasting », *Environmetrics* 27.2, pp. 70–82.

Metz, Luke, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein (2018), « Meta-learning update rules for unsupervised representation learning », *arXiv preprint arXiv:1804.00222*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean (2013), « Distributed representations of words and phrases and their compositionality », *Advances in neural information processing systems*, pp. 3111–3119.

Millefiori, L. M., P. Braca, K. Bryan, and P. Willett (Oct. 2016), « Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction », *IEEE Transactions on Aerospace and Electronic Systems* 52.5, pp. 2313–2330.

Min, Seonwoo, Byunghan Lee, and Sungroh Yoon (Sept. 2017), « Deep learning in bioinformatics », *Briefings in Bioinformatics* 18.5, pp. 851–869.

Mohajerin, Nima and Steven L. Waslander (Nov. 2019), « Multistep Prediction of Dynamic Systems With Recurrent Neural Networks », *IEEE Transactions on Neural Networks and Learning Systems* 30.11, pp. 3370–3383.

Murphy, Kevin P. (2012), *Machine learning: a probabilistic perspective*, MIT press.

Nagarajan, Badrinath, Luca Delle Monache, Joshua P. Hacker, Daran L. Rife, Keith Searight, Jason C. Knievel, and Thomas N. Nipen (Dec. 2015), « An Evaluation of Analog-Based Postprocessing Methods across Several Variables and Forecast Models », *Weather and Forecasting* 30.6, pp. 1623–1643.

Nanduri, Anvardh and Lance Sherry (2016), « Anomaly detection in aircraft data using Recurrent Neural Networks (RNN) », Ieee, pp. 5C2–1.

Neal, Radford M. and Geoffrey E. Hinton (1998), « A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants », *Learning in Graphical Models*, ed. by Michael I. Jordan, NATO ASI Series, Dordrecht: Springer Netherlands, pp. 355–368.

Nguyen, D., O. S. Kirsebom, F. Frazão, R. Fablet, and S. Matwin (May 2019), « Recurrent Neural Networks with Stochastic Layers for Acoustic Novelty Detection », *ICASSP*

2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 765–769.

Nguyen, Duong, Said Ouala, Lucas Drumetz, and Ronan Fablet (Mar. 2019), « EM-like Learning Chaotic Dynamics from Noisy and Partial Observations ».

— (May 2020a), « Assimilation-Based Learning of Chaotic Dynamical Systems from Noisy and Partial Data », *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, pp. 3862–3866.

— (Sept. 2020b), « Variational Deep Learning for the Identification and Reconstruction of Chaotic and Stochastic Dynamical Systems from Noisy and Partial Observations ».

Nguyen, Duong, Matthieu Simonin, Guillaume Hajduch, Rodolphe Vadaine, Cédric Tedeschi, and Ronan Fablet (2020), « Detection of Abnormal Vessel Behaviors from AIS data using GeoTrackNet: from the Laboratory to the Ocean », *21st IEEE International Conference on Mobile Data Management (MDM)*.

Nguyen, Duong, Rodolphe Vadaine, Guillaume Hajduch, René Garello, and Ronan Fablet (Oct. 2018), « A Multi-task Deep Learning Architecture for Maritime Surveillance using AIS Data Streams », *2018 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*.

— (Dec. 2019), « GeoTrackNet-A Maritime Anomaly Detector using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection », *arXiv:1912.00682 [cs, stat]*, arXiv: 1912.00682.

Ntalampiras, S., I. Potamitis, and N. Fakotakis (Aug. 2011), « Probabilistic Novelty Detection for Acoustic Surveillance Under Real-World Conditions », *IEEE Transactions on Multimedia* 13.4, pp. 713–719.

Oord, Aaron van den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu (Sept. 2016), « WaveNet: A Generative Model for Raw Audio », *arXiv:1609.03499 [cs]*, arXiv: 1609.03499.

Ouala, Said, Duong Nguyen, Lucas Drumetz, Bertrand Chapron, Ananda Pascual, Fabrice Collard, Lucile Gaultier, and Ronan Fablet (2020), « Learning Latent Dynamics for Partially-Observed Chaotic Systems », *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.

Ouala, Said, Duong Nguyen, Cédric Herzet, Lucas Drumetz, Bertrand Chapron, Ananda Pascual, Fabrice Collard, Lucile Gaultier, and Ronan Fablet (July 2019), « Learning Ocean Dynamical Priors from Noisy Data Using Assimilation-Derived Neural Nets »,

IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, ISSN: 2153-7003, pp. 9451–9454.

Ouala, Said, Ananda Pascual, and Ronan Fablet (May 2019), « Residual Integration Neural Network », *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, pp. 3622–3626.

Pajot, Arthur, Emmanuel de Bezenac, and Patrick Gallinari (Sept. 2018), « Unsupervised Adversarial Image Reconstruction ».

Pallotta, G., S. Horn, P. Braca, and K. Bryan (July 2014), « Context-enhanced vessel prediction based on Ornstein-Uhlenbeck processes using historical AIS traffic patterns: Real-world experimental results », *17th International Conference on Information Fusion (FUSION)*, pp. 1–7.

Pallotta, Giuliana, Michele Vespe, and Karna Bryan (June 2013), « Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction », *Entropy* 15.6, pp. 2218–2245.

Parzen, Emanuel (1962), « On estimation of a probability density function and mode », *The annals of mathematical statistics* 33.3, pp. 1065–1076.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer (2017), « Automatic differentiation in pytorch ».

Pathak, Jaideep, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott (Jan. 2018), « Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach », *Physical Review Letters* 120.2, p. 024102.

Pathak, Jaideep, Zhixin Lu, Brian R. Hunt, Michelle Girvan, and Edward Ott (Dec. 2017), « Using Machine Learning to Replicate Chaotic Attractors and Calculate Lyapunov Exponents from Data », *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.12, arXiv: 1710.07313, p. 121102.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014), « Glove: Global vectors for word representation », *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Perera, L. P., P. Oliveira, and C. Guedes Soares (Sept. 2012), « Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction », *IEEE Transactions on Intelligent Transportation Systems* 13.3, pp. 1188–1200.

Perobelli, Nicola (June 2016), *MarineTraffic - A day in numbers*.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018), « Deep contextualized word representations », *arXiv preprint arXiv:1802.05365*.

Pierce, David W. (2001), « Distinguishing coupled ocean–atmosphere interactions from background noise in the North Pacific », *Progress in Oceanography* 49.1-4, pp. 331–352.

Principi, E., F. Vesperini, S. Squartini, and F. Piazza (May 2017), « Acoustic novelty detection with adversarial autoencoders », *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3324–3330.

Qin, Tong, Kailiang Wu, and Dongbin Xiu (Nov. 2018), « Data Driven Governing Equations Approximation Using Deep Neural Networks », *arXiv:1811.05537 [cs, math, stat]*, arXiv: 1811.05537.

Qiu, Lin, Sheng Gao, Qinjie Lyu, Jun Guo, and Patrick Gallinari (Feb. 2018), « A novel non-Gaussian embedding based model for recommender systems », *Neurocomputing*, Recent Advances in Machine Learning for Non-Gaussian Data Processing 278, pp. 144–152.

Rabiner, Lawrence R. (1989), « A tutorial on hidden Markov models and selected applications in speech recognition », *Proceedings of the IEEE* 77.2, pp. 257–286.

Radford, Benjamin J., Leonardo M. Apolonio, Antonio J. Trias, and Jim A. Simpson (Mar. 2018), « Network Traffic Anomaly Detection Using Recurrent Neural Networks », *arXiv:1803.10769 [cs]*, arXiv: 1803.10769.

Rainforth, Tom, Adam R. Kosiorek, Tuan Anh Le, Chris J. Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh (Feb. 2018), « Tighter Variational Bounds are Not Necessarily Better », *arXiv:1802.04537 [cs, stat]*, arXiv: 1802.04537.

Raissi, Maziar, Paris Perdikaris, and George E. Karniadakis (2019), « Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations », *Journal of Computational Physics* 378, pp. 686–707.

Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis (Jan. 2018), « Multistep Neural Networks for Data-driven Discovery of Nonlinear Dynamical Systems », *arXiv:1801.01236 [nlin, physics:physics, stat]*, arXiv: 1801.01236.

Ravì, Daniele, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang (Jan. 2017), « Deep Learning for Health Informatics », *IEEE Journal of Biomedical and Health Informatics* 21.1, pp. 4–21.

Rezende, Danilo Jimenez and Shakir Mohamed (2015), « Variational inference with normalizing flows », *arXiv preprint arXiv:1505.05770*.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (June 2014), « Stochastic Backpropagation and Approximate Inference in Deep Generative Models », *International Conference on Machine Learning*, PMLR, pp. 1278–1286.

Rhodes, B. J., N. A. Bomberger, M. Seibert, and A. M. Waxman (Oct. 2005), « Maritime situation monitoring and awareness using learning mechanisms », *MILCOM 2005 - 2005 IEEE Military Communications Conference*, 646–652 Vol. 1.

Ristic, B., B. La Scala, M. Morelande, and N. Gordon (June 2008), « Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction », *2008 11th International Conference on Information Fusion*, pp. 1–7.

Riveiro, Maria, Giuliana Pallotta, and Michele Vespe (2018), « Maritime anomaly detection: A review », *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 8.

Robinson, James C. (2005), « A topological delay embedding theorem for infinite-dimensional dynamical systems », *Nonlinearity* 18.5, p. 2135.

Rosenblatt, Murray (1956), « Remarks on some nonparametric estimates of a density function », *The Annals of Mathematical Statistics*, pp. 832–837.

Rubanova, Yulia, Ricky T. Q. Chen, and David K Duvenaud (2019), « Latent Ordinary Differential Equations for Irregularly-Sampled Time Series », *Advances in Neural Information Processing Systems 32*, ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, Curran Associates, Inc., pp. 5320–5330.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein (2015), « Imagenet large scale visual recognition challenge », *International journal of computer vision* 115.3, pp. 211–252.

Sak, Hasim, Andrew Senior, and Francoise Beaufays (2014), « Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling », *Fifteenth annual conference of the international speech communication association*, p. 5.

Salman Khan, Muhammad, Miao Yu, Pengming Feng, Liang Wang, and Jonathon Chambers (May 2015), « An unsupervised acoustic fall detection system using source separation for sound interference suppression », *Signal Processing*, Machine learning and signal processing for human pose recovery and behavior analysis 110, pp. 199–210.

Schubert, R., E. Richter, and G. Wanielik (June 2008), « Comparison and evaluation of advanced motion models for vehicle tracking », *2008 11th International Conference on Information Fusion*, pp. 1–6.

Sendra, Sandra, Lorena Parra, Jaime Lloret, and José Miguel Jiménez (July 2015), *Oceanographic Multisensor Buoy Based on Low Cost Sensors for Posidonia Meadows Monitoring in Mediterranean Sea*, Research Article.

Shampine, Lawrence F. (2018), *Numerical solution of ordinary differential equations*, Routledge.

Shu, Kai, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu (2017), « Fake news detection on social media: A data mining perspective », *ACM SIGKDD explorations newsletter* 19.1, pp. 22–36.

Smith, Gerald L., Stanley F. Schmidt, and Leonard A. McGee (1962), *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*, National Aeronautics and Space Administration.

SOLAS (1974), *The International Convention for the Safety of Life at Sea (SOLAS)*.

Song, Li, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai, and Chuan Shi (2018), « Anomalous trajectory detection using recurrent neural network », *International Conference on Advanced Data Mining and Applications*, Springer, pp. 263–277.

Sprott, Julien Clinton and Julien C Sprott (2003), *Chaos and time-series analysis*, vol. 69, Citeseer.

Su, Ya, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei (July 2019), « Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network », *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, New York, NY, USA: Association for Computing Machinery, pp. 2828–2837.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014), « Sequence to sequence learning with neural networks », *Advances in neural information processing systems*, pp. 3104–3112.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus (Feb. 2014), « Intriguing properties of neural networks », *arXiv:1312.6199 [cs]*, arXiv: 1312.6199.

Tu, Enmei, Guanghao Zhang, Lily Rachmawati, Eshan Rajabally, and Guang-Bin Huang (2017), « Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey », *IEEE Transactions on Intelligent Transportation Systems*.

Üney, M., L. M. Millefiori, and P. Braca (May 2019), « Data Driven Vessel Trajectory Forecasting Using Stochastic Generative Models », *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8459–8463.

Vacar, Cornelia and Jean-François Giovannelli (Mar. 2019), « Unsupervised joint deconvolution and segmentation method for textured images: a Bayesian approach and an advanced sampling algorithm », *EURASIP Journal on Advances in Signal Processing* 2019.1, p. 17.

Varlamis, Iraklis, Konstantinos Tserpes, Mohammad Etemad, Amílcar Soares Júnior, and Stan Matwin (2019), « A Network Abstraction of Multi-vessel Trajectory Data for Detecting Anomalies. », *EDBT/ICDT Workshops*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin (2017), « Attention is All you Need », *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008.

Vlachas, Pantelis R., Wonmin Byeon, Zhong Y. Wan, Themistoklis P. Sapsis, and Petros Koumoutsakos (May 2018), « Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks », *Proceedings. Mathematical, Physical, and Engineering Sciences* 474.2213.

Voss, Henning U., Jens Timmer, and Jürgen Kurths (June 2004), « Nonlinear dynamical system identification from uncertain and indirect measurements », *International Journal of Bifurcation and Chaos* 14.06, pp. 1905–1933.

Welch, Greg and Gary Bishop (1995), « An introduction to the Kalman filter ».

Will, J., L. Peel, and C. Claxton (2011), « Fast Maritime Anomaly Detection using KD Tree Gaussian Processes », *2nd IMA Conference on Maths in Defence*.

Wolf, Alan, Jack B. Swift, Harry L. Swinney, and John A. Vastano (July 1985), « Determining Lyapunov exponents from a time series », *Physica D: Nonlinear Phenomena* 16.3, pp. 285–317.

Wu, CF Jeff (1983), « On the convergence properties of the EM algorithm », *The Annals of statistics*, pp. 95–103.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean (Sept. 2016), « Google's Neural Machine

Translation System: Bridging the Gap between Human and Machine Translation », *arXiv:1609.08144 [cs]*, arXiv: 1609.08144.

Yeo, Kyongmin and Igor Melnyk (Jan. 2019), « Deep learning algorithm for data-driven simulation of noisy dynamical system », *Journal of Computational Physics* 376, pp. 1212–1231.

Yin, Zhichao and Jianping Shi (2018), « Geonet: Unsupervised learning of dense depth, optical flow and camera pose », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1983–1992.

Zajdel, W., J.D. Krijnders, T. Andringa, and D.M. Gavrila (Sept. 2007), « CASSANDRA: audio-video sensor fusion for aggression detection », *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, London, UK: IEEE, pp. 200–205.

Zaslavsky, George M. and Georgij Moiseevič Zaslavskij (2005), *Hamiltonian chaos and fractional dynamics*, Oxford University Press on Demand.

Zhao, Liangbin and Guoyou Shi (2019), « Maritime Anomaly Detection using Density-based Clustering and Recurrent Neural Network », *The Journal of Navigation* 72.4, pp. 894–916.

Zhao, Zhizhen and Dimitrios Giannakis (Aug. 2016), « Analog forecasting with dynamics-adapted kernels », *Nonlinearity* 29.9, pp. 2888–2939.

Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G. Lowe (2017), « Unsupervised learning of depth and ego-motion from video », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858.

Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros (2017), « Unpaired image-to-image translation using cycle-consistent adversarial networks », *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Apprentissage Variationnel Profond pour la Modélisation et l'Analyse des Séries Temporelles, Applications à l'Identification de Systèmes Dynamiques et à la Détection d'Anomalies de Trafic Maritime.

Mot clés : apprentissage profond, inférence variationnelle, identification de systèmes dynamiques, détection d'anomalies, AIS, surveillance de trafic maritime.

Résumé : Ce travail de thèse se focalise sur une classe de méthodes d'apprentissage profond, probabilistes et non-supervisées qui utilisent l'inférence variationnelle pour créer des modèles évolutifs de grande capacité pour les séries temporelles. Nous présentons deux classes d'apprentissage variationnel profond, puis nous les appliquons à deux problèmes spécifiques liés au domaine maritime.

La première application est l'identification de systèmes dynamiques à partir de données bruitées et partiellement observées. Nous introduisons un cadre qui fusionne l'assimilation de données classique et l'apprentissage profond moderne pour retrouver les équations différentielles qui contrôlent la dynamique du système. En utilisant une formulation d'espace d'états, le cadre proposé intègre des composantes stochastiques pour tenir compte des variabilités stochastiques, des erreurs de modèle et des incertitudes de reconstruction.

La deuxième application est la surveillance du trafic maritime à l'aide des données AIS. Nous proposons une architecture d'apprentissage profond probabiliste multitâche pouvant atteindre des performances très prometteuses dans différentes tâches liées à la surveillance du trafic maritime, telles que la reconstruction de trajectoire, l'identification du type de navire et la détection d'anomalie, tout en réduisant considérablement la quantité de données à stocker et le temps de calcul. temps. Pour la tâche la plus importante - la détection d'anomalie, nous introduisons un détecteur géospatialisé qui utilise l'apprentissage profond variationnel pour construire une représentation probabiliste des trajectoires AIS, puis détecter les anomalies en jugeant la probabilité de cette trajectoire.

Title: Variational Deep Learning for Time Series Modelling and Analysis, Applications to Dynamical System Identification and Maritime Traffic Anomaly Detection

Keywords: deep learning, variational inference, dynamical systems identification, anomaly detection, AIS, maritime traffic surveillance

Abstract: This thesis work focuses on a class of unsupervised, probabilistic deep learning methods that use variational inference to create high capacity, scalable models for time series modelling and analysis. We present two classes of variational deep learning, then apply them to two specific problems related to the maritime domain.

The first application is the identification of dynamical systems from noisy and partially observed data. We introduce a framework that merges classical data assimilation and modern deep learning to retrieve the differential equations that control the dynamics of the system. Using a state space formulation, the proposed framework embeds stochastic components to account for stochastic variabilities, model errors and reconstruction uncertainties.

The second application is maritime traffic surveillance using AIS data. We propose a multitask probabilistic deep learning architecture can achieve state-of-the-art performance in different maritime traffic surveillance related tasks, such as trajectory reconstruction, vessel type identification and anomaly detection, while reducing significantly the amount data to be stored and the calculation time. For the most important task—anomaly detection, we introduce a geospatial detector that uses variational deep learning to builds a probabilistic representation of AIS trajectories, then detect anomalies by judging how likely this trajectory is.