

CRANFIELD UNIVERSITY

CHENGYAO SUN

EXPLAINABLE ARTIFICIAL INTELLIGENCE FOR
TIME SERIES MODELLING AND CAUSAL INFERENCE

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING
PhD in Aerospace

Doctor of Philosophy
Academic Year: 2019–2022

Supervisors: Prof. Weisi Guo and Dr. Yang Xing
December 2022

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING

PhD in Aerospace

Doctor of Philosophy

Academic Year: 2019–2022

CHENGYAO SUN

Explainable Artificial Intelligence for Time Series Modelling
and Causal Inference

Supervisors: Prof. Weisi Guo and Dr. Yang Xing
December 2022

This thesis is submitted in partial fulfilment of the
requirements for the degree of Doctor of Philosophy.

© Cranfield University 2022. All rights reserved. No part of
this publication may be reproduced without the written
permission of the copyright owner.

Abstract

Due to the concern over the trust issue of black-box artificial intelligence (AI) in daily human life, “right to an explanation” requirement is hence proposed for any algorithm in Europe general data protection regulation (GDPR). However, the framework of explainable AI (XAI) is still in its infancy. Thus, this thesis aims to propose a framework of explainable AI for time series (X-AI4TS) in neural modelling and causal inference.

In order to achieve the trustworthy usage of algorithm, at the first stage, this thesis embed the priori knowledge in time series and break it down into neuron-based dynamic system with domain expertise for explainable modelling. Then secondly, this thesis address the explainability in causal inference steps of the neuron-based model — (i) Identifying the “hard to learn” features for neurons; (ii) Revealing and visualizing the inner operations within neurons; (iii) Inferring the original posteriori multimodal distribution via neurons.

In addition, the aforementioned methodologies are formulated into an X-AI4TS framework and applied to a real business case in semiconductor fabrication for an integrated implementation. A data-driven neuron-based time-series forecasting digital twin is generated, under my proposed framework, for a dynamic manufacturing system. The results offer the diverse explainability to different users (i.e., end-users, data scientist and AI expert) and show the landing potential in dynamic engineering time series application.

Keywords

Explainable AI; Time Series Modelling; Causal Inference

Contents

Abstract	iii
Contents	iv
Acknowledgements	viii
1 Introduction	1
1.1 Background	1
1.1.1 XAI Research Directions: Integrated and Post-hoc for Local and Global Interpretability	2
1.1.2 Different Modes of Generating Explainability	3
1.1.2.1 Structural Mode	4
1.1.2.2 Statistical Mode	5
1.1.2.3 Human Perception Mode	6
1.2 Aims and Objectives	6
1.3 Thesis Structure	7
2 Papers — Time Series Modelling	8
2.1 Climate-Conflict Causality Measures	8
2.1.0 Abstract	8
2.1.1 Introduction	9
2.1.1.1 Literature Review	9
2.1.1.2 Causality Methods Review	9
2.1.2 Data and Toy Validation Model Descriptions	12
2.1.2.1 Data Descriptions	12
2.1.2.2 Assumptions and Testing via Validation Toy Model	12
2.1.3 Results and Validation	13
2.1.3.1 Experiment Setups	13
2.1.3.2 Validation using Theoretical Model	14
2.1.3.3 Overall Causality Results by Region with PI, CCM and CMI	15
2.1.3.4 LSTM Causality Results	20
2.1.3.5 Temporal Lag and Further Questions	20
2.1.3.6 Weaknesses in Work	21
2.1.4 Conclusions and Next Steps	21
2.1.4.1 Conclusion	21
2.1.4.2 Next Step	22

2.2	Revealing the Excitation Causality between Climate and Political Violence via a Neural Forward-Intensity Poisson Process	23
2.2.0	Abstract	23
2.2.1	Introduction	23
2.2.1.1	Background	24
2.2.1.2	Challenges of State-of-the-Art Models	25
2.2.1.3	Motivation for a Neural Forward-Intensity Model	25
2.2.1.4	Contribution and Novelty	27
2.2.2	Algorithm Formulation and Validation	27
2.2.2.1	End-to-End Learning Framework	28
2.2.2.2	Model Scope Validation	29
2.2.3	Implementation and Results	30
2.2.3.1	Evaluation Metrics	30
2.2.3.2	Data Description	31
2.2.3.3	Results and Discussion	32
2.2.4	Conclusion and Future Work	36
3	Papers — Explainable Neural Inference	38
3.1	Approximate Symbolic Explanation for Neural Network Enabled Water-Filling Power Allocation	38
3.1.0	Abstract	38
3.1.1	Introduction	39
3.1.1.1	Multi-Channel Power Allocation	39
3.1.1.2	Legal and Safety Motivations to Explainable AI (XAI)	40
3.1.1.3	Contribution	41
3.1.2	System Setup	42
3.1.2.1	Water-Filling Power Allocation	42
3.1.2.2	Neural Network Power Allocation	42
3.1.3	Symbolic Explainability via Meijer G-function	43
3.1.4	Conclusions and Future Work	45
3.2	Scalable Partial Explainability in Neural Networks via Flexible Activation Functions	46
3.2.0	Abstract	46
3.2.1	Introduction	47
3.2.1.1	Motivation and Related Work	47
3.2.1.2	Novelty and Contribution	49
3.2.2	System Model	50
3.2.2.1	Neural Network Topology	50
3.2.2.2	Activation Function	51
3.2.2.3	Back-Propagation for Control Points Tuning	54
3.2.2.4	Symbolic Model and Visualized Explainability	55
3.2.3	Case Study and Result Discussion	55
3.2.3.1	Results Overview	56
3.2.3.2	Model Interpretation	56
3.2.3.3	Scalability, Expressive Power and Explainability	59
3.2.4	Conclusion and Future Work	59

3.3	Forecasting Wireless Demand with Extreme Values using Feature Embedding in Gaussian Processes	61
3.3.0	Abstract	61
3.3.1	Introduction	61
3.3.1.1	State-of-the-Art	62
3.3.1.2	Novelty and Contribution	64
3.3.2	System Model	64
3.3.2.1	Data Decomposition	65
3.3.2.2	Priori Gaussian Process Model	66
3.3.2.3	Feature Embedding Kernel Function	67
3.3.2.4	Posteriori Prediction	69
3.3.3	Experimental Results and Discussion	71
3.3.3.1	Data Source	71
3.3.3.2	Explainable Feature Matrix for FE-GP	71
3.3.3.3	Performance Metrics	72
3.3.3.4	Results Analysis	72
3.3.3.5	Uncertainty Quantification	73
3.3.3.6	Training Process	74
3.3.4	Conclusion and Future Work	75
3.4	Scarce Data Driven Deep Learning of Drones via Generalized Data Distribution Space	77
3.4.0	Abstract	77
3.4.1	Introduction	77
3.4.1.1	Related work	80
3.4.1.2	Innovation: GAN-TDA framework	82
3.4.2	Method	86
3.4.2.1	Step 1) - Learn data distribution by GAN	86
3.4.2.2	Step 2) - Latent feature maps extraction and TDA	87
3.4.2.3	TDA interpretation	90
3.4.2.4	Experimental setup	90
3.4.2.5	The dataset and hardware	91
3.4.2.6	DCGAN settings	92
3.4.2.7	Validation settings	93
3.4.3	Results	94
3.4.3.1	TDA result	95
3.4.3.2	Discriminator result	95
3.4.4	Conclusion	97
4	Overall Discussion with Implementation	99
4.1	XAI4TS Framework	99
4.2	Implementation in Smart Manufacturing	100
4.2.1	Introduction	100
4.2.1.1	Domain Knowledge	100
4.2.1.2	Background and Motivation	101
4.2.1.3	Contribution and Novelty	102
4.2.2	Time Series Modelling	102

4.2.2.0	Notation	102
4.2.2.1	Fabrication Dynamic	103
4.2.2.2	Neural Modeling	104
4.2.2.3	Symbolic Boundary Condition	105
4.2.3	Explainable Neuron	106
4.2.3.1	Targeted Data Collection	106
4.2.3.2	Explainable Neuron Twins	106
4.2.3.3	Probabilistic Inference	107
4.2.4	Results	107
4.2.4.1	GAN-TDA	107
4.2.4.2	Symbolic Approximation	108
4.2.4.3	Multimodal Distribution Inference	108
4.2.4.4	Time Series Forecasting	109
4.3	Overall Discussion	110
5	Conclusions and Future Work	111
	References	113

Acknowledgements

I spent most of my PhD time under the period of Covid-19 pandemic, which got a lot of things off track and difficult for me. But now I have made it through, and sincerely want express my gratitude to people who let me successfully get here.

Above all, I would like to thank my supervisor Prof. Weisi Guo, who always gives the strongest support to my PhD research and life during last three years. To me, he is more than a supervisor and definitely acts as a crucial role in my values so far. I am grateful for each open communication with Prof. Guo.

Then, I would like to thank my partner Bailu Jin and my cat Juju, who have been keeping me company and giving emotional encouragement.

In addition, I would like to thank my team member Chen Li, who is my best friend in UK and one of my co-authors for two major papers. In collaboration, he had inspired me with active brainstorm.

Last but not least, I would like to thank my family for consistent understanding and support in my student life until now.

Chapter 1

Introduction

1.1 Background

The intensive application of artificial intelligence (AI) techniques in various fields of human life, especial in safety related industry (e.g., auto-pilot, medicine design and etc.), has raise the attention of trust issues regarding both AI reliability and its provenance of reasoning, because most AI models are well-known as black-boxes. Then, “right to an explanation” is hence proposed for any algorithm in Europe general data protection regulation (GDPR). However, the framework of explainable AI (XAI) is still in its infancy.

Generally, the research direction of explainability can be classified as using either a ”local” or ”global” approach, with either ”Integrated” or ”Post-hoc” interpretability design [22]. Modes of analysis define the ways to explain ML models, classified into structural mode, abstract statistical mode and human perception mode. All the above designs contribute to the following purpose. During the data biography from input to output via a decision module, a sufficient explainable AI model must have the ability to introduce and explain itself completely in purpose, observed data, processing logic and decision, as I define the composition of explainability into the following units:

1. The ability to explain purpose: what problem the ML model trying to solve.
2. The ability to explain input data: where does the observed data originate from

(provenance), the data format, quality and transparency.

3. The ability to explain processing logic: clarify the features extracted from observed data, and explain which parts play the most significant rule in decision-making.
4. The ability to explain decision: what is the output prediction and decision, and their real-world impact, such as reward distribution in action space for Deep-Q-learning and resulting state feedback.
5. The ability to explain problem: explain how the AI is solving a problem.

1.1.1 XAI Research Directions: Integrated and Post-hoc for Local and Global Interpretability

I envisage that a request for explanation maybe demanded either post-event or continuously during reasoning (operations) - see Fig. 1.1a. In either case, the request should specify the direction or granularity in which the reasoning needs to be made (e.g., at the local or global reasoning level, and at the post-hoc or integrated level. The methods to satisfy these demands will depend on several factors related to the algorithm and other factors discussed later on in the paper. For now, I detail these directions.

Integrated interpretability directly extracts explanations from the ML model structures and processing logic [88]. The structural complexity of ML models limit current integrated interpretability to only models with low complexity, e.g. Bayesian inference based on low-dimensional decision rules. Simple classifiers like Naive Bayes, physics-informed regression models and decision trees can directly explain their predictions and also expose potential biases (e.g., impact of uncertainty in marginals of Bayesian inference). With rising ML model complexity, higher ML precision leads to greater difficulty in interpreting models. For example, deep neural networks (DNN) and support vector machines (SVM) can not directly explain processing logic due to complex multi-layer connections or hyperplanes.

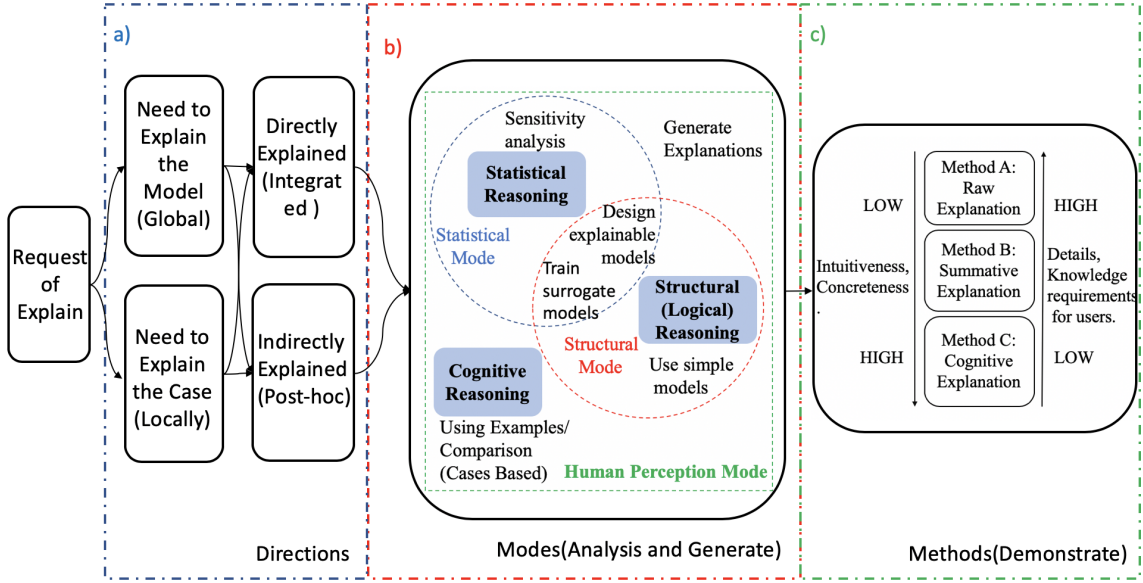


Figure 1.1: Demonstration of a) Directions, b) Modes, c) Methods and their relationships

Post-hoc interpretability twin-system approach proposes to explain the AI model by using high transparency (white-box) systems (like decision trees, linear models) to mimic/mirror the AI model after training. The twin-system can roughly explain black-box models (e.g. DNN, DQN) [22] at the risk of poor approximation or over-fitting to a particular training set.

Local and global explanation indicates the granularity scope. Local explanation examines an individual prediction while global explanation explains the operational logic of the entire ML model behaviour [21]. For example, complex service slice handover decisions between BSs can be explained by local explanations (e.g., the mobility and data demand of one service), but can also be explained by the overall handover policy that governs the process.

1.1.2 Different Modes of Generating Explainability

The modes indicate methods to extract and generate the compositions of explainability from learning models, guides the turning of learning model into explainable learning model (Integrated), or the analysed explanation based on the output of models (post-doc)

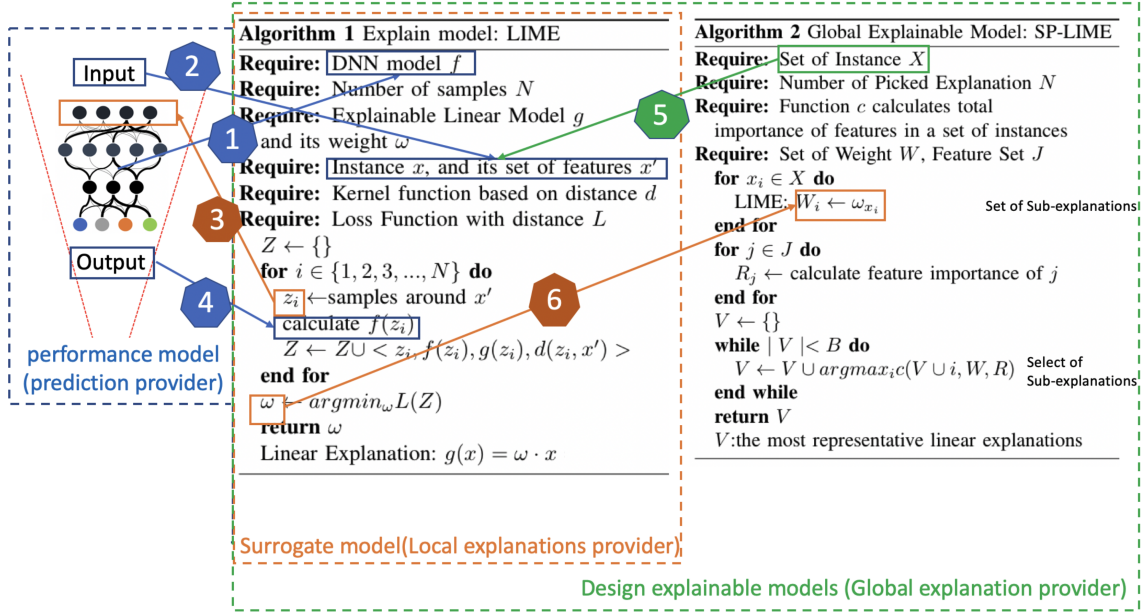


Figure 1.2: Demonstration of Surrogate Model and Design Explainable Model: 1) performance model provide LIME (surrogate explanation) for RRM by DNN. 2) Based on instance x_i , export LIME features. 3) A set of generated instance Z around x_i by LIME uploaded to DNN. 4) The output of DNN based on input Z is sent back to LIME. 5) After the work of LIME, SP-LIME (Design Explainable Model) select the set of instances required for linear explanations by LIME. 6) LIME pass back explanations for each instance to SP-LIME. In this case, SP-LIME needs to be implemented after the implementation of LIME, using explanations from LIME.

as shown in b), Fig. 1.1.

1.1.2.1 Structural Mode

Use simple models: low complexity in structure can generate logical explanations at the perceptual level to be accepted by users, the backtracking of the decision-making process from decision tree can directly generate explanation. Symbolic classification can also relate to physical laws or well-established optimisation results (e.g. water-filling or channel inversion).

Design explainable models: the machine learning models are reconstructed by interactive sections that process sub-tasks respectively from the overall prediction task, and each section and interaction could be inherent architecturally explainable. Proposed in [107],

Sub-modular Pick Local Interpretable Model-Agnostic Explanations (SP-LIME) generates reliable non-redundant explanations globally by using a set of representative-enough instances from LIME (a surrogate model introduced later) see - Fig. 1.2; it iterates to greedily find the explainable set by the softmax of instance coverage. This indicates the overall importance of features that appeared in the representative instance, then return the representative set to users, in order to let users chose interested cases themselves and track the raw data.

1.1.2.2 Statistical Mode

Sensitivity analysis: the gradient of input features respect to a label can give out which part contains the most influential information while doing decision. For instance, Layer-Wise Relevance Propagation (LRP) allocates each input with a relevance score quantified by how much this input contributed to the prediction; DeepLIFT [123] could use back-propagation to highlight environment features (wind direction and speed, etc.) that can significantly influence the flight control decision (speed, direction, etc.). In barrier-escape tasks, obstacles identified by BSs from pixel-inputs of on-board cameras will be highlighted in the machine-human interface to explain the decision in real-time.

Train surrogate models: similar to twin-system, surrogate models ideally use a simple-structured model to fit the output of a complex model [63]. LIME, as shown in Fig. 1.2, is a model-agnostic algorithm, explains the complex prediction by approximation of the local case via an interpretable model (e.g. linear regression). It does so by generating a set of samples around instance to be explained; then use the set of samples as input in the black-box model; use the input-output set from black-box to train the weights in the explainable model, to reveal which features contribute significantly in black-box prediction. DNN-CBR using Case-Based Reasoning (CBR) to chose the local explanation of DNN by finding similar cases with the minimum distance from the database, while DNN provides outputs with high precision. [32] proposed a method to build a soft decision

tree created by DNN, to makes hierarchical decisions with the ability to provided better generalizations and robustness to unlabelled data.

1.1.2.3 Human Perception Mode

Using Examples/Comparison (Cases Based): Similar/opposite cases can guarantee users with confidence in the reason why AI models handle the recent problem in a specific way or not. The CBR method introduced above could select relevant similar cases from Case Base by selecting items with minimum distance to give out the reasons for model predictions.

Generate Explanations: Linguistic explanations could be one of the most powerful and intuitive explanations which should extract the interaction and logic from features, generate semantic sentences and even cooperate with visualisation methods to demonstrate intuitive explanations. Authors in [47] proposed a method to generate linguistic explanation by using the coupling of visual recognition and text definitions, which generate an understandable high-leveled explanation of the prediction.

1.2 Aims and Objectives

In this thesis, I aim to investigate the explainable artificial intelligence in (i) time series causal modelling and (ii) explainable neural inference with my published papers.

For (i), I aim to embed the priori domain knowledge in time series causal modelling and break it down into neuron-based dynamic system. Papers in chapter 2 take the climate and conflict as example, formulate its dynamic mechanism of time series via neural modelling with priori assumptions, and demonstrate the importance of domain expertise in explainable time series modelling.

For (ii), I aim to propose interpretation methodologies for neural model explainability in and distribution generalization and posteriori inference. Paper 3.4 is proposed to identify the “hard to learn” features by exploring the feature distribution generalization

in neurons, in order to give the further data collection guidance for incremental training; Paper 3.1 and 3.2 reveal the inner operation within neurons by symbolic approximation twins of neural networks; Paper 3.3 forms the posteriori inference into multimodal distributions, which represent the real data distribution, for further application enabling (e.g., reinforcement learning).

Besides, chapter 4 is proposed for an overall implementation and discussion of the methodologies in this thesis. The end-to-end framework is applied on a business case in semiconductor fabrication for manufacturing line forecasting, and shows the landing potential in dynamic engineering time series application.

Acceptance/publication of papers in chapter 2 & 3 are as follows:

Table 1.1: Papers in chapter 2 & 3.

Paper 2.1	Accepted	COMPLEXIS 2023
Paper 2.2	Published	2022 31st IJCAI
Paper 3.1	Published	2021 IEEE 93rd Vehicular Technology Conference
Paper 3.2	Published	2021 35th AAAI Conference (Student Paper)
Paper 3.3	Published	2022 IEEE 95rd Vehicular Technology Conference
Paper 3.4	Published	Neural Computing and Applications

1.3 Thesis Structure

The remainder of this “paper format” thesis is organised as follows. Paper cascade in chapter 2 focuses on time series neural modelling. Paper cascade in chapter 3 investigates the explainability in neurons. Chapter 4 discusses an overall implementation of the methodologies in this thesis over a real business case. Chapter 5 concludes this thesis and gives the future direction.

Chapter 2

Papers — Time Series Modelling

2.1 Climate-Conflict Causality Measures

2.1.0 Abstract

Climate change affects agriculture and human liveability, i.e. disrupting food availability, reducing access to food, and affecting food quality. The repercussions on social cohesion and conflict are less clear, and vary with geographic region, the time period (e.g., prevailing economy and social structure), and international trade (e.g., food prices). Several socio-economic mechanistic steps exist between local climate changes and conflict, and often the intermediate steps are hidden to us. Nonetheless, this chapter offer an exploratory data analysis in this chapter at a global scale, on the relationship between diverse climate indicators and conflict (both non-state terrorism and state conflict).

The results in this chapter can be used to inform policy and conflict prediction models such as Global Urban Analytics for Resilient Defence (GUARD) (e.g., how climate might be an external conflict excitation variable or induce migration. Here this chapter investigate the causal relationship process between climate change and conflict, including the causal direction, causal lag, and causal strength. Working with the MET Office, this chapter use historical climate and extreme environmental event data from the past 50 years across the world to identify geographic region-specific causal indicators.

The broad findings are: 1. The average monthly rainfall amount series is a universal causation factor to the terrorism series each region investigated; 2. The regions of Northern Africa, Southern Africa, Southern Asia and Southeastern Asia have the strongest causal link between the climate and the conflict; 3. There exists a common time lag of the causality between the climate and the conflict in the regions of Northern Africa, Southern Africa, Eastern Asia, Southern Asia, Southeastern Asia and Eastern Europe, which is worth further investigation.

2.1.1 Introduction

2.1.1.1 Literature Review

Among the most worrying of the mooted impacts of climate change is an increase in civil conflict as people compete for diminishing resources, such as arable land and water [49]. Research over the past decade has established that climate variability and climate change may influence the risk of violent conflict, including organized armed conflict [14, 53]. Authors in [52] examine 50 quantitative empirical studies and find a “remarkable convergence in findings” and “strong support for a causal association” between climatological changes and conflict at all scales and across all major regions of the world. However, under various hypothesis, different methodologies and datasets applied, the results are divergent or even contradictory [12, 91, 126]. Although the relationship between climate and conflict has been empirically tested in a wide variety of studies, the literature has yet to converge on a commonly accepted set of results [112].

2.1.1.2 Causality Methods Review

Granger Causality and Predictability Improvement Granger causality test is the first proposed and the best-known mathematical approach to detect causality based on statistical hypothesis test. Two time series X, Y given, under Granger’s theory, the causality (the alternative hypothesis in test) is defined as that time series $[..., X(t-1), X(t)]$ pro-

vide significant information about $[Y(t+1), Y(t+2), \dots]$ [36]. Although Granger causality achieve great success in econometrics, it is valid in stationary linear systems rather than dynamic systems. The predictability improvement (PI) causality test is proposed in response to this issue. Authors in [67, 68, 70] reconstruct the time series in multi-dimensional state spaces, which try to convert the non-linear dynamic system into linear in a manifold, where the Granger causality test can then work on.

Conditional Mutual Information Conditional mutual information (CMI) [48] is based on statistical information theory with key idea of transfer entropy. Discrete random variables (X, Y, Z) with support sets $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ given, the CMI $I(X, Y|Z)$ is defined as:

$$I(X, Y|Z) = \sum_{z \in \mathcal{Z}} p_Z(z) \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p_{X,Y|Z}(x, y|z) \log \frac{p_Z(z) p_{X,Y,Z}(x, y, z)}{p_{X,Z}(x, z) p_{Y,Z}(y, z)} \quad (2.1)$$

which estimates the directed information flow from a variable X to another variable Y under the condition of Z . Different from PI hypothesis test, CMI can gives the value of predictability improvement. CMI does not depend on any model in its formulation compared to other directed information flow measures like Granger causality, which makes this method capable of assessing both linear and non-linear interactions. In series causality detection application, with a causation lag τ , $I(X(n), Y(n+\tau)|Y(n))$ is normally used to evaluate the causality from $X(t)$ and $Y(t+\tau)$ with directionality index. CMI has been proven to work on time series causality [80, 111, 155].

Convergent Cross Map In [130, 139], convergent cross map (CCM) is proposed to detect the causal relationships in nonlinear dynamical systems. The key idea of CMM lays on the fundamental principle of Takens's theorem, which states that in a time series dynamical system with multi-variables, any single time series of one variable of the system recover the essential information of the system by high dimensional state space reconstruction. Two time series X, Y given in CCM, the causality is defined as the extent to which the time series $[X(t-E\tau), \dots, X(t-2\tau), X(t-\tau)]$ can recover the time series $Y(t)$.

The parameter τ here is the time step for the reconstruction while E is the dimension of the reconstruction. Authors in [130] also offer three one-minute animation videos on CCM operating details as the supplementary material [131]. CCM has been successfully applied in various field, such as the soil moisture-precipitation interaction in environment [149], the sensitivity of the carbon cycle to tropical temperature variations [147] and the relationship between temperature and green-house gases [142] in climate.

Long Short-Term Memory Long Short-Term Memory (LSTM) is an artificial recurrent neural network (RNN) architecture, which is capable of learning order dependence in series. The training process in LSTM can be considered as an adaptive non-linear regression from the input series to the output series. Owing to the non-linear expressive power of the neurons and the characteristics of RNN, LSTM has the following advantages [69]: (a) It is able to store information for an arbitrary duration; (b) It is resistant to noise (i.e. fluctuations of the inputs that are random or irrelevant for regression); (c) It is trainable in reasonable time. In this practice, two time series X, Y given, $X(t)$ and $Y(t)$ are set as the inputs while $Y(t+1)$ as the output. Then, this chapter can evaluate the causality of $X(t)$ by validating the regression error in LSTM [176].

Methodology Summary PI gives the non-linear Granger causality between two series through hypothesis testing at the most basic level of “Yes” or “No”; CMI models the causality strength using the mutual information between two series; CCM reconstructs one series into another and evaluates the causality using the correlation between reconstructed series and raw series; LSTM does an adaptive nonlinear regression from one series to another and assess the causality by prediction error.

2.1.2 Data and Toy Validation Model Descriptions

2.1.2.1 Data Descriptions

Temperature and Rainfall Data (MET Office) In this chapter, temperature and rainfall data are used to represent the climate conditions. I used the ERA5 dataset, which records the global temperature and rainfall grid in 0.5 resolution ($56\text{--}79\text{km}^2$ depending on latitude), and monthly in scope span 1970-present.

Extreme Climate Event Data In this chapter, natural disaster events (including tsunami, drought, tornado and etc.) are considered to be the extreme climate events and collected from Emergency Events Database (EM-DAT), which records these events around the world with the dates and locations, spans 1900-present.

Terrorism and State Conflict Data In this chapter, data from Global Terrorism Database (GTD) are selected as the indicator of terrorism. GTD is the most comprehensive database of terrorist attacks in the world, which contains over 200k terrorist attacks with the dates and locations, spans 1970-2018.

2.1.2.2 Assumptions and Testing via Validation Toy Model

Fig.2.1a shows the assumptions I used in this chapter. I assume the climate system (i.e., General Climate (x_1), Ambient Climate Temperature or Rainfall (x_2) and Natural Disaster (x_3)) is an autonomous system which has linear and nonlinear internal causality to each other, while there is a unidirectional linear causal link from climate system to the agriculture (y_1). The agriculture has a bidirectional linear causal link with the socio-economic (y_2) while also a unidirectional linear causal link from conflict (y_3) to the conflict itself. Then, my key assumption is that the agriculture has a unidirectional nonlinear causal link to the conflict.

Based on my assumptions, I establish a toy model using two unidirectional coupled Rossler systems (Fig.2.1b) with an adjustable coupling strength parameter C . The ini-

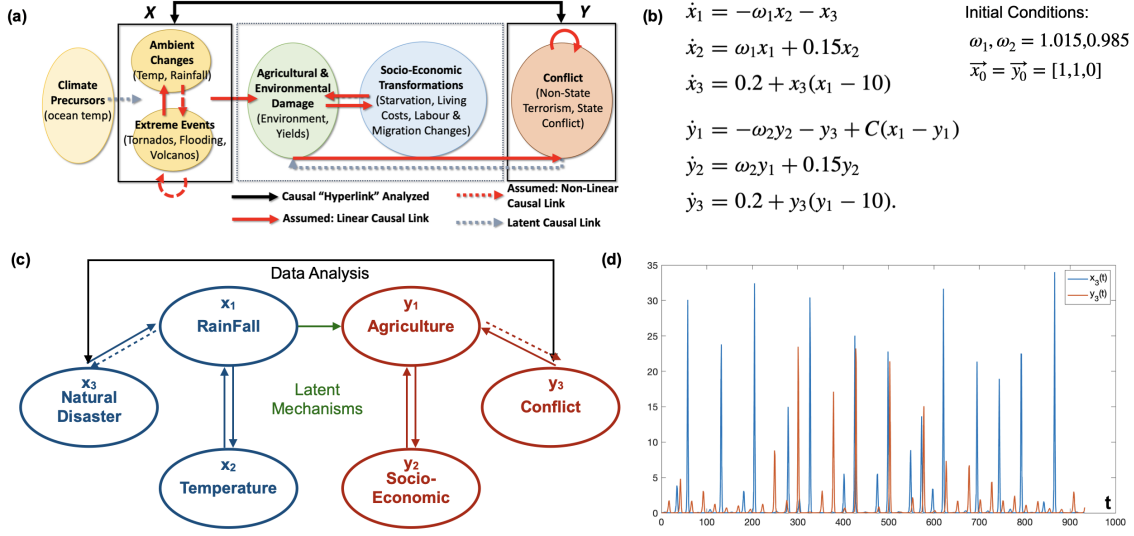


Figure 2.1: (a) Assumptions of the mechanism between climate and conflict. (b) & (c) The corresponding Rossler toy model, and (d) an example of time series $x_3(t)$ and $y_3(t)$.

tial conditions can be arbitrary, in experiments, I set the initial conditions of $\omega_1, \omega_2 = 1.015, 0.985$ and $\vec{x}_0 = \vec{y}_0 = [1, 1, 0]$. With my initial conditions, two systems would be synchronized when the coupling strength C is about 0.14, which means the system Y will be fully dependent on system X when coupling strength C is about 0.14.

2.1.3 Results and Validation

2.1.3.1 Experiment Setups

In my experiments, in order to keep data's attributes consistency in causality tests, the data are pre-processed as follows: (1) Data series spanning 1970-2018 are selected (2) Data series are unified into monthly resolution (3) For natural disasters and terrorist attacks, I quantify it in terms of its counting times per month (4) Data series for the following 12 regions are selected: (i) Northern Africa (AfricaN), (ii) Southern Africa (AfricaS), (iii) Central America (AmericaC), (iv) Northern America (AmericaN), (v) Southern America (AmericaS), (vi) Central Asia (AsiaC), (vii) Eastern Asia (AsiaE), (viii) Southern Asia (AsiaS), (ix) Southeastern Asia (AsiaSE), (x) Eastern Europe (EuropeE), (xi) Western Europe (EuropeW) and (xii) Oceania.

In addition, it is worth noting that I make an adjustment to the function in CMI to overcome an issue with directionality in CMI. Conventionally, two time series (X, Y) given, the condition mutual information $I(X(t), Y(t + \tau) | Y(t))$ would be considered as the causality of $X(t) \rightarrow Y(t + \tau)$ with directionality. However, this form of CMI causes instability in directionality with different time lags in my experiments. Thus, I use $I(X(t), Y(t + \tau) | X(t))$, which calculates the mutual information (MI) between $Y(t)$ and $Y(t + \tau)$ under the condition of $X(t)$. In this case, the directionality issue is solved. Then, by comparing the CMI and MI between $Y(t)$ and $Y(t + \tau)$ (i.e., $I(Y(t), Y(t + \tau))$), I can quantify the causality of the condition series X .

2.1.3.2 Validation using Theoretical Model

Firstly, I test my methods (i.e., PI, CCM and CMI) on the aforementioned theoretical toy model, in order to verify the validity of these methods in detecting my assumed causality mechanism between climate and conflict.

Fig.2.2 gives the causality results in my theoretical toy model with the coupling strength from 0 to 0.14 before the synchronization. The values in PI give the hypothesis test results on whether X_3 (Natural Disasters) causes the Y_3 (conflict); The values in CMI give the amount of mutual information given by X_3 to Y_3 ; The values in CCM give the correlation between the reconstructed Y_3 using X_3 and Y_3 itself; The values in LSTM give the regression error in percentage.

The result shows that all methods can detect the causality link before the synchronization of two systems. However, it can be observed that PI method has the most sensitivity to the causality among all my proposed methods, while CCM and LSTM have the longest lags in response to my proposed causality. Another notable finding is that LSTM is not consistent in performance. PI, CMI and CCM always give better results when the coupling strength goes up while LSTM does not. Due to this being unexplainable, the results from LSTM are less trustworthy than the others and I will analyze the LSTM result separately.

$X_3 \rightarrow Y_3$	PI	CMI	CCM	LSTM
C = 0	○	0.455	-0.025	45.7%
C = 0.01	○	0.479	0.050	50.8%
C = 0.02	○	0.502	0.183	54.5%
C = 0.03	○	0.564	0.123	48.3%
C = 0.04	○	0.671	0.071	38.4%
C = 0.05	√	0.714	-0.003	27.7%
C = 0.06	√	0.732	-0.010	17.6%
C = 0.07	√	0.778	0.065	25.1%
C = 0.08	√	0.790	0.147	16.4%
C = 0.09	√	0.803	0.420	12.6%
C = 0.10	√	0.805	0.517	9.80%
C = 0.11	√	0.819	0.626	6.41%
C = 0.12	√	0.820	0.759	2.96%
C = 0.13	√	0.826	0.849	7.42%
C = 0.14	√	0.837	0.907	2.3%

Figure 2.2: PI, CCM, CMI and LSTM results in the theoretical toy model. Red: no causality; Blue: causality may exist in between; Green: evident causality is detected.

2.1.3.3 Overall Causality Results by Region with PI, CCM and CMI

The raw quantitative overall causality results of PI, CCM and CMI by region are shown in Fig.2.3. The abbreviations in this figure: Var. = Variable; PI Test = Predictability Improvement Hypothesis Test; Acpt. = Acceptance; Rej. = Rejection; CCM = Convergent Cross Mapping; Corr = Correlation; RTS = Reconstruction Time Steps; C = Conflict/Terrorism; R = RainFall; T = Temperature; D = Nature Disaster; $R \rightarrow C$ = R cause C.

In each sub-figure, I show the CMI results for different cause factors (x_1, x_2, x_3) in different time lags with the coloured lines, and the PI and CCM results in the table embedded in each graph. For CCM, I have two values in the grid, the first value is the correlation value between the reconstructed conflict data and raw conflict data, while the second one is the corresponding reconstruction dimension. (e.g. In AfricaN, PI hypothesis test shows that rainfall and temperature has the causality to the conflict; CCM shows that the rainfall and temperature series can reconstructed into conflict series with the correlation of 0.76

and 0.67 respectively; CMI shows the mutual information over time lags, i.e. the condition of rainfall and temperature give the highest mutual information to the prediction from $C(t)$ to $C(t + \tau)$ over all the time while the condition of natural disaster series gives some mutual information just at the beginning, but ambiguous mutual information to the prediction after 18 months.)

In order to have more intuitive interpretation and analysis, I convert these results into qualitative representations with the following rules:

(1) For PI hypothesis test, I use “v” (coloured in green) to represent the accepted positive causality from the variables series towards terrorism series, “o” (coloured in blue) to represent no causality, and “x” (coloured in red) to represent reversed anti-nature causality from terrorism series towards climate series.

(2) For correlation degree in CCM, I define $\rho \geq 0.7$ as strong correlated (coloured in green), $0.4 \leq \rho < 0.7$ as moderate correlated (coloured in blue), $0.2 \leq \rho < 0.4$ as weak correlated (coloured in red), $\rho < 0.2$ as none correlated.

(3) For causality mechanism complexity degree in CCM, I define $E \leq 12$ as low complexity, $12 < E \leq 24$ as moderate complexity, $E > 24$ as high complexity.

(4) There are no absolute standards for CMI which makes the inter-comparison of CMI between different regions’ datasets meaningless, thus I use the CMI mean difference over MI of terrorism series:

$$mean_{\tau}(CMI - MI) = mean_{\tau}(I(C(t), C(t + \tau) | Var(t)) - I(C(t), C(t + \tau))) \quad (2.2)$$

to demonstrate the assistance of climate series causality in one region and internally compare this causality. The larger the difference between CMI and MI appears, the more causality effect the corresponding condition has. I colored the value larger than 0.5 in green, 0-0.5 in blue and negative value in red for clearer demonstrating.

Fig.2.4 shows the processed qualitative interpretation of the causality results both in region domain and methods domain according to my rules. I am going to analyse these



Figure 2.3: Overall causality results of PI, CCM and CMI by region.

(a)

AfricaN					AfricaS					AmericaC				
Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI
R	✓	strong	moderate	0.78	R	✓	strong	high	1.07	R	✓	weak	high	0.66
T	✓	moderate	high	0.64	T	✓	strong	high	0.56	T	×	moderate	moderate	-0.06
D	○	strong	high	-0.06	D	○	strong	moderate	-0.08	D	×	moderate	moderate	-0.24

AmericaN					AmericaS					AsiaC				
Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI
R	✓	weak	moderate	1.29	R	✓	None	low	0.81	R	✓	None	low	1.35
T	○	weak	low	1.31	T	✓	None	moderate	0.59	T	○	None	low	1.29
D	○	weak	low	0.48	D	○	None	low	0.47	D	✓	None	low	-0.81

AsiaE					AsiaS					AsiaSE				
Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI
R	✓	None	low	1.10	R	✓	strong	high	0.76	R	✓	strong	high	0.93
T	○	None	moderate	0.75	T	✓	strong	high	0.57	T	✓	strong	moderate	0.58
D	○	None	moderate	0.02	D	○	strong	moderate	-0.03	D	○	strong	moderate	0.38

EuropeE					EuropeW					Oceania				
Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI	Var.	PI Test	CCM Corr.	CCM E	CMI
R	✓	weak	high	0.93	R	✓	None	moderate	0.79	R	✓	None	moderate	1.20
T	✓	None	high	0.84	T	✓	None	low	0.85	T	○	None	high	1.32
D	○	None	low	-0.10	D	○	weak	moderate	0.44	D	○	None	moderate	0.14

(b.1)

PI Test												
Var. \ Region	AfricaN	AfricaS	AmericaC	AmericaN	AmericaS	AsiaC	AsiaE	AsiaS	AsiaSE	EuropeE	EuropeW	Oceania
R	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
T	✓	✓	×	○	✓	○	○	✓	✓	✓	✓	○
D	○	○	×	○	○	✓	○	○	○	○	○	○

(b.2)

Correlation between CCM Reconstruction Estimated Conflict Series and Original Conflict Series												
Var. \ Region	AfricaN	AfricaS	AmericaC	AmericaN	AmericaS	AsiaC	AsiaE	AsiaS	AsiaSE	EuropeE	EuropeW	Oceania
R	strong	strong	weak	weak	None	None	None	strong	strong	weak	None	None
T	moderate	strong	moderate	weak	None	None	None	strong	strong	None	None	None
D	strong	strong	moderate	weak	None	None	None	strong	strong	None	weak	None

(b.3)

$\text{mean}(CMI - MI) = \text{mean}((C(t), C(t+\tau) Var(t)) - (C(t), C(t+\tau)))$												
Var. \ Region	AfricaN	AfricaS	AmericaC	AmericaN	AmericaS	AsiaC	AsiaE	AsiaS	AsiaSE	EuropeE	EuropeW	Oceania
R	0.78	1.07	0.66	1.29	0.81	1.35	1.10	0.76	0.93	0.93	0.79	1.20
T	0.64	0.56	-0.06	1.31	0.59	1.29	0.75	0.57	0.58	0.84	0.85	1.32
D	-0.06	-0.08	-0.24	0.48	0.47	-0.81	0.02	-0.03	0.38	-0.10	0.44	0.14

Figure 2.4: (a) Qualitative interpretation of PI, CCM and CMI in each region (region domain). (b) Parallel comparison of qualitative interpretation of PI, CCM and CMI in each region (method domain).

results in both domains:

(1) Region Domain: It can be observed that the regions of AfricaN, AfricaS, AsiaS and AsiaSE give the strongest causality evidence of climate series affecting terrorism series from the results of all these three methods. However, although the results for other regions is ambiguous, it does not necessarily mean that the causality does not exist in these regions, it is also possible that my methods cannot detect the causality completely.

Another notable issue is the inconsistency in results from different methods. The representative examples are AmericaC and AsiaC, where causality of natural disaster series results given from PI, CCM and CMI are conflicting with each other. The reason for this would be that the real mechanism in these regions is beyond my assumptions.

(2) Method Domain: PI: With PI hypothesis test, terrorism series in each region shows the predictability improvement from the corresponding rainfall series. However, temperature series acts as a cause to terrorism series in just some specific regions, while natural disaster series affect only one region (i.e., AsiaC). Based on this PI method, the rainfall series can be concluded as a universal causation factor to the terrorism series.

CCM: The correlation offered by CCM is always contiguous with climate series in each region — there is no case where one variable series has strong CCM correlation while others' have it weak. This result indicates that the climate variables in my experiments can be considered as an autonomous system which the terrorism system is coupled on.

CMI: In the CMI method, the condition of rainfall and temperature series offers the positive mutual information between the terrorism series $C(t)$ and $C(t + \tau)$ in all of the regions except for AmericaC. However, the mutual information given by the condition of natural disaster series is ambiguous over these regions, which match the results in [126]—natural disasters are not the factors to be blamed for conflict.

2.1.3.4 LSTM Causality Results

I analyse the LSTM result independently as its results have inconsistencies with the other methods which raises doubts on its validity. In my experiments, I establish a LSTM network with two LSTM layers with 128 hidden nodes and three full connect layers with 64,128,16 hidden nodes respectively. I applied one-step prediction along the last ten years (1998-2018), which means for each month in each region, an independent LSTM is applied, while the average prediction percentage error is considered as the causality indicator.

Var. Error	LSTM											
	AfricaN	AfricaS	AmericaC	AmericaN	AmericaS	AsiaC	AsiaE	AsiaS	AsiaSE	EuropeE	EuropeW	Oceania
C	29.9%	26.1%	25.9%	34.8%	31.7%	16.7%	34.4%	19.1%	34.3%	29.3%	22.2%	38.2%
C & R	9.8%	16.6%	33.4%	7.5%	25.9%	18.6%	30.4%	5.6%	12.7%	13.2%	12.7%	68.1%
C & T	14.7%	19.7%	67.9%	19.5%	33.8%	4.3%	20.6%	7.9%	19.9%	15.8%	14.1%	35.6%
C & D	31.4%	22.8%	42.1%	14.4%	42.6%	32.1%	17.5%	11.6%	13.9%	44.5%	16.8%	87.5%
ALL	12.4%	14.2%	54.4%	11.3%	37.0%	5.3%	21.2%	6.3%	9.4%	18.4%	13.5%	52.1%

Figure 2.5: The LSTM prediction results — Average error in percentage is considered as the causality indicator. 20% error rate as the threshold of causality confirmation (in green).

Fig.2.5 shows the results from LSTM. In AmericaC, AmericaS, AsiaE and Oceania, the causality between climate and conflict is ambiguous from a LSTM regression sight, while others shows climate series may help in the conflict prediction which indicates the causality between climate and conflict. However, I can also observe the instability of the algorithm from the results, e.g., in AfricaN, AsiaC, EuropeE and Oceania, the prediction with inputs of C&D is worse than that with only C. This means LSTM has learned some odd mechanisms which actually do not exist and cause an overfitting. In this chapter, I will only take the LSTM results as a reference, not as a decisive factor.

2.1.3.5 Temporal Lag and Further Questions

Looking at Fig.2.3, I find an interesting phenomenon in $CMI(\tau)$ results. Obviously, when temporal lag $\tau = 0$, the mutual information between terrorism series $C(t)$ and $C(t\tau)$

reaches the maximum. Nevertheless, the rainfall and temperature series $CMI(\tau)$ reaches another peak when the temporal lag τ is around 10-14 months (1 year) and around 34-38 months (3 years) in the regions of AfricaN, AfricaS, AsiaE, AsiaS, AsiaSE and EuropeE. This indicates that in these regions, the climate series may have the same causation mechanism in climate $\rightarrow$ agriculture $\rightarrow$ social-economic $\rightarrow$ conflict/terrorism under my assumptions. A possible preliminary answer would be that, in low socioeconomic development and low capability regions, or regions dominated by agriculture, the climate has a distinct causation to the region conflict/terrorism with the time lag of one year and three years [91]. However, the mechanism behind this phenomenon is still an open question and becomes a further issue for us.

2.1.3.6 Weaknesses in Work

In this work, my assumptions are based on the common knowledge and lack of professional in agriculture and social science. Although I get the expected result that, in Africa, the causality is strong between the climate and the conflict from a data analysis perspective, this is still questionable from social science perspective [12, 91, 126]. My work so far cannot identify the spurious causality.

2.1.4 Conclusions and Next Steps

2.1.4.1 Conclusion

Climate change affects agriculture and human liveability, i.e. disrupting food availability, reducing access to food, and affecting food quality. The repercussions on social cohesion and conflict are less clear, and vary with geographic region, the time period (e.g., prevailing economy and social structure), and international trade (e.g., food prices) [12, 14, 49, 52, 53, 91, 112, 126]. Several socio-economic mechanistic steps exist between local climate changes and conflict, and often the intermediate steps are hidden to us. Nonetheless, I am offering an exploratory data analysis in this chapter at a global scale,

on the relationship between diverse climate indicators and conflict (both non-state terrorism and state conflict).

The results in this chapter can be used to inform policy and prediction models such as Global Urban Analytics for Resilient Defence (GUARD), e.g., how climate might be an external conflict excitation variable or induce migration. Here I investigate the causal process between climate change and conflict, including the causal direction, causal lag, and causal strength. Working with the MET Office, I use historical climate and extreme environmental event data from the past 50 years across the world to identify geographic region-specific causal indicators.

The broad findings are: 1. The rainfall series is a universal causation factor to the terrorism series in each region I investigated. 2. The regions of AfricaN, AfricaS, AsiaS and AsiaSE have the strongest causal link between the climate and the conflict. 3. There exists a common time lag of approximately 1 and 3 years of the causality between the climate and the conflict in the regions of AfricaN, AfricaS, AsiaE, AsiaS, AsiaSE and EuropeE, which is worth further investigation.

2.1.4.2 Next Step

In order to identify the spurious causality in my results from pure data analysis perspective, I propose to embed the knowledge from social science (e.g. regional politics and economy conditions) into my initial assumptions about the mechanisms within climate and conflict [12].

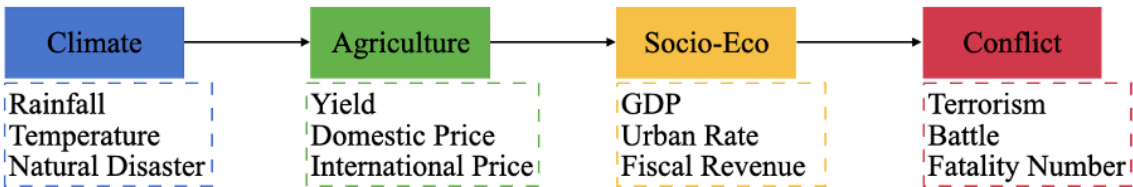


Figure 2.6: Proposed detailed causal link from climate to conflict.

Precisely, I assume a detailed causal link (Fig.2.6) and will investigate the causality strength one by one to discover the true causality mechanism from climate to conflict.

2.2 Revealing the Excitation Causality between Climate and Political Violence via a Neural Forward-Intensity Poisson Process

2.2.0 Abstract

The causal mechanism between climate and political violence is fraught with complex mechanisms. Current quantitative causal models rely on one or more assumptions: (1) the climate drivers persistently generate conflict, (2) the causal mechanisms have a linear relationship with the conflict generation parameter, and/or (3) there is sufficient data to inform the prior distribution. Yet, it is known that conflict drivers often excite a social transformation process which leads to violence (e.g., drought forces agricultural producers to join urban militia), but further climate effects do not necessarily contribute to further violence. Therefore, not only is this bifurcation relationship highly non-linear, there is also often a lack of data to support prior assumptions for high resolution modeling.

Here, this chapter aims to overcome the aforementioned causal modeling challenges by proposing a neural forward-intensity Poisson process (NFIPP) model. The NFIPP is designed to capture the potential non-linear causal mechanism in climate induced political violence, whilst being robust to sparse and timing-uncertain data. My results span 20 recent years and reveal an excitation-based causal link between extreme climate events and political violence across diverse countries. My climate-induced conflict model results are cross-validated against qualitative climate vulnerability indices. Furthermore, I label historical events that either improve or reduce my predictability gain, demonstrating the importance of domain expertise in informing interpretation.

2.2.1 Introduction

Armed violence is on the rise. Whilst the rate of major wars has decreased over the past few decades, the number of civil conflicts has doubled since the 1960s, and political

violence (e.g., terrorism, sectarian violence) have become more frequent in the past ten years. Governments and the international community often have little warning of impending crises. Current practices in conflict risk interpretation rely on diplomatic reports and extrapolating from statistical data. Machine learning is poised to boost the power of these approaches [43].

Among the most worrying of the mooted impacts of climate change is an increase in conflict as people compete for diminishing resources such as arable land and water. Research over the past decade has established that climate variability may influence the risk of conflict [8,13,51]. Despite broad expert agreement that there is a strong climate and conflict association [50], quantitative modeling results are divergent or even contradictory [13] [113]. Modelling the impact of climate on conflict and predicting out-of-sample events remains challenging, especially when only a few variables are considered [33]. As such, there remain serious challenges in the assumptions laid out in current causal inference models. Based on the general idea of a causality test [19], many papers and ourselves here propose to evaluate the causal link by testing whether additional extreme climate events would improve the performance of political violence prediction.

2.2.1.1 Background

Causal mechanisms that generate political violence often have an excitation mechanism [171]. The excitation effect that extreme climate events have on armed violence is often non-linear and bifurcates, e.g., extreme weather can cause agriculture community to shift to political violence and thereafter future climate changes are unlikely to have the same impact. Consequently, continuous functions of causal modeling that do not adjust for uncertain functional behaviour change is unlikely to be successful in complex societies [29]. These continuous causal models include [71]: Granger causality and Predictability Improvement (PI), Conditional Mutual Information (CMI), and Convergent Cross Map (CCM) methods.

For discrete excitation causal models, they are typically heavily parameterized espe-

cially on the time window, such that they are generally applicable only to a particular genre (e.g., IED attacks in a particular area [137]), and lack general understanding of the nature of climate vs. terrorism. Furthermore, terrorism data often suffer from sparsity, which is caused by a variety of factors ranging from under-reporting, lack of data in an emerging crisis area, to poor data quality. This can have an effect in non-linear likelihood estimators which is commonly used in current deep methods with linear loss function (e.g., LSTM).

2.2.1.2 Challenges of State-of-the-Art Models

In literature, neural network modeling of conflict events attempt to account for the non-linear relationships in society, but this suffers from two challenges. For most countries, political violence data is sparse. As such, the prior distribution assumption of output (e.g., conflict events) becomes critical [31]. In neural networks, the output value is considered to be a random variable conforming to a Gaussian distribution with corresponding mean and variance related to the inputs [74]. However, the number of conflict events have been shown to follow a Poisson point process [38]. As such, the divergence between Gaussian and Poisson distribution can not be ignored when they have small mean values in sparse data. Hence, in modeling, I propose that any model should set *Poisson processes as the prior* for the prediction model. Another challenge is predicting the precise time interval between attacks, where the state-of-the-art window size agnostic algorithm requires massive amounts of time series data [94], often unavailable to the conflict modeling genre. Therefore, in evaluation, I propose to set the *posterior likelihood* of probabilistic prediction as the indicator of the model performance.

2.2.1.3 Motivation for a Neural Forward-Intensity Model

My goal is to develop a general method that can self-configure to identify causal excitation signals and relate them to contextualised recent historical events. First of all, a specialized predictor is required for the excitation-based non-linear mechanism. Suppose

a deep learning model has the inputs of the historical terrorism and climate data, and the target output of the attack forecasting. Then, there would exist two main challenges in this task, which raise additional requirements to both modeling and its evaluation.

Poisson process has been recognized as a principled framework for modeling event data [129], where the intensity parameter is a function determined by relevant covariates. There are various derivatives of Poisson process, which focus on optimizing the linear form of intensity function with its covariates, e.g. Hawkes process [137, 171], self-correcting process [58], reinforced Poisson processes [119] and etc. In the most recent work [93, 162], the intensity function is modeled via a neural network [118], which enables more non-linear covariates being captured.

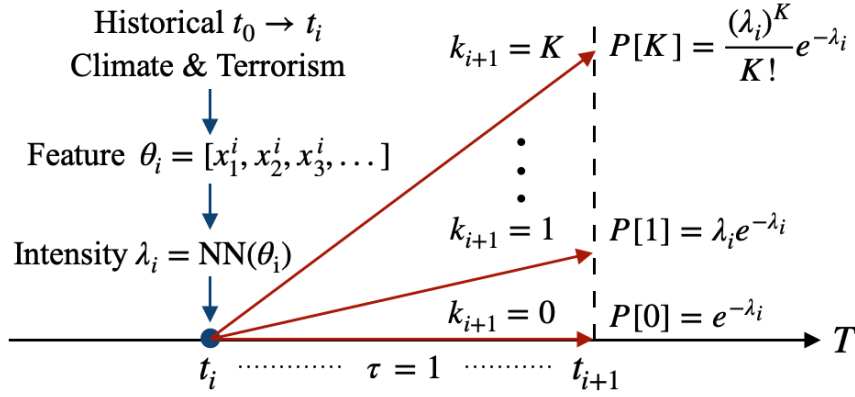


Figure 2.7: The idea of neural forward-intensity Poisson forecasting at each time step is that: (1) Blue arrow: Intensity inference via a neural network (NN) using the extracted the features from historical covariates; (2) Red arrow: One step probabilistic forecasting based on the Poisson process with estimated intensity given.

As shown in Fig.2.7, in my proposed model, I adopt the NN-based intensity function to fit the non-linear mechanism between climate and conflict generation process. I make a one-step forward probabilistic forecasting via the Poisson process with the intensity value in each day. My proposed neural forward-intensity Poisson process (NFIPP) model meets the requirements raised by the aforementioned challenges using the *prior of Poisson* and the *evaluation of likelihood*.

2.2.1.4 Contribution and Novelty

In this chapter, I evaluate the excitation causality between climate and terrorism using the proposed neural forward-intensity Poisson process (NFIPP). Two major contributions of this chapter are:

(i) In order to fit the non-linear mechanism in the sparse and timing-uncertain terrorist attack time series, I propose to embed a neural network based forward-intensity function in the Poisson process. This algorithm has been tested on a toy model and proven to have advantageous performance compared to conventional methods.

(ii) NFIPP is applied on the real terrorism and climate datasets for daily attack occurrence intensity inference. I reveal the excitation causal link from extreme climate events to terrorism via the posterior likelihood analysis. I find that the extreme climate events and other events (e.g., economy and politics) have the discrete excitation causal links to the terrorism rather than a continuous time causality, which require domain expertise in informing interpretation.

The remainder of this chapter is organized as follows. In Section 2, I formulate the end-to-end learning framework for NFIPP model and verify its application scope with my assumption. In Section 3, I apply NFIPP to the real data and analyze the results with contextualised recent historical events. Section 4 concludes this chapter and the ideas for future work.

2.2.2 Algorithm Formulation and Validation

My neural forward-intensity Poisson process (NFIPP) model is based on the assumption in function (2.3) that, the target time series $\mathbf{K} \in \mathbb{N}^{1 \times T}$ (i.e., daily count N of terrorist attacks) is the posterior observation $\{k_t\}_{t=1}^T$ from independent Poisson process with the corresponding forward-intensity $\lambda_t \in \mathbb{R}_+$ at each time slot $[t, t + \tau]$, while the intensity comes from a neural network (NN) which can be arbitrarily associated with the features

$(x_1^t, x_2^t, x_3^t \dots)$ generated from former covariate series (e.g., terrorism and climate).

$$P[N(t + \tau) - N(t) = k] = \frac{(\lambda_t \tau)^k}{k!} e^{-\lambda_t \tau} \quad (2.3)$$

$$\lambda_t = \text{NN}(x_1^t, x_2^t, x_3^t \dots)$$

However, in reality the accurate intensity values are unknowable in most cases, hence I create a toy scenario to verify the performance and scope of my model before the implementation on the real-world data.

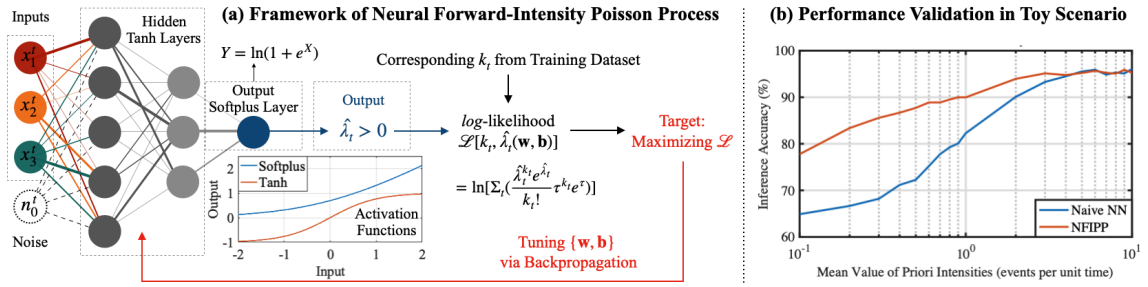


Figure 2.8: (a) The end-to-end learning framework of my proposed neural forward-intensity Poisson process (NFIPP) model. (b) The performance validation result of NFIPP in my supposed toy scenario, which shows its the advantage in low intensity range.

2.2.2.1 End-to-End Learning Framework

Fig.2.8(a) demonstrates the end-to-end NFIPP learning framework. Taking the D dimension feature vector at time t , $\mathbf{X}_t = [x_1^t; x_2^t; \dots; x_D^t] \in \mathbb{R}^{D \times 1}$, as the input, then NN is expected to has the output of the forward-intensity estimation $\hat{\lambda}_t$. Specifically, I introduce the “soft-plus” function $Y = \ln(1 + e^X)$ in the last layer to ensure $\hat{\lambda}_t \geq 0$ meeting the positive constraint of Poisson process, while avoid vanishing gradient problem if “relu” function is used [93].

With the outputs $\{\hat{\lambda}_t\}_{t=1}^T$ and the observation $\{k_t\}_{t=1}^T$ in the training data, the posterior log-likelihood $\sum \ln[P(k|\lambda)]$ can be derived as a function $\mathcal{L}(\mathbf{w}, \mathbf{b})$ with the weight and bias parameters in the NN using function (2.3), which enable the training of $\{\mathbf{w}, \mathbf{b}\}$ according

to maximum log-likelihood via gradient back-propagation with learning rate η as:

$$\begin{bmatrix} \mathbf{w}_{i+1} \\ \mathbf{b}_{i+1} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{w}_i \\ \mathbf{b}_i \end{bmatrix} + \eta \begin{bmatrix} \frac{\partial \mathcal{L}(\mathbf{w}_i, \mathbf{b}_i)}{\mathbf{w}_i} \\ \frac{\partial \mathcal{L}(\mathbf{w}_i, \mathbf{b}_i)}{\mathbf{b}_i} \end{bmatrix} \quad (2.4)$$

2.2.2.2 Model Scope Validation

In order to verify the performance and scope of NFIPP model, I create a toy scenario with transparent intensity functions. In the toy scenario, suppose (i) $\mathbf{K} \in \mathbb{N}^{1 \times T}$ are the posterior observations of corresponding Poisson process with the intensities $\Lambda \in \mathbb{R}_+^{1 \times T}$; (ii) Each element $\lambda_t \in \Lambda$ is the function of potential features $\mathbf{x}^t \in [0, 1]^D$, conditioned on $\lambda_t = \mathcal{F}(\mathbf{x}_t) \geq 0$. In the model validation, I set $T = 1e3$, $D = 4$, which are closed to the terrorism data training scale. For robustness test, I set x_4^t to be an independent variable of the function $\mathcal{F}$, which act as a noise input. $\mathcal{F}$ randomly takes its form from a bunch of composite non-linear functions, including power, exponential, trigonometric and etc. (e.g. $\lambda = e^{x_1} \sin(x_2) + x_3^{\frac{1}{4}}$). In the test, I validate the average accuracy performance of my NFIPP over scenarios with mean of intensities varying from 10^{-1} to 10^1 , compared to conventional naive NN with the same neuron topology. With massive Monte-Carlo, the validation results are demonstrated in Fig.2.8(b).

Both models are robust and without overfitting to the noise feature. my NFIPP shows distinct advantage in prediction when the mean event occurrence rate is lower than 1 per unit time. This advantage comes from my improvement to the model priori of Poisson distribution. With the occurrence rate increasing to 4, Poisson distribution approximates to Gaussian distribution, where the advantage gradually fades away while the model performance keeps well. In reality for each country in last two decades, rarely do mean daily terrorist attack rate rises above 3 even in most dangerous countries (see Section 3). This means my NFIPP model would lead to significant improvement in the terrorism prediction.

2.2.3 Implementation and Results

As discussed in Section 2.2.1.1, the excitation of climate in terrorism is non-linear and transient [29]. In this section, I aim to reveal this kind of causal link with a few representative countries with proposed NFIPP model, and analyze the potential underlying reasons backward from the results.

2.2.3.1 Evaluation Metrics

Based on the general idea of statistical causality test, I recognize a positive causality when additional historical climate data improve the performance of terrorism predictor comparing to historical terrorism only. In NFIPP, the prediction performance is evaluated by the posterior Poisson likelihood using the forward-intensity output given by predictor. Then the improvement can be calculated with the difference between the log-likelihood given by only historical terrorism data $\mathcal{L}_1$ and that the log-likelihood given by both historical terrorism and climate data $\mathcal{L}_2$. Since I only focus on the positive/negative effects and relative magnitude of improvement, for intuitive visualization, I define the likelihood gain with the difference normalized by a scalable tanh function:

$$\text{Likelihood Gain} = \tanh[\alpha \times (\mathcal{L}_2 - \mathcal{L}_1)], \quad (2.5)$$

where α is an adaptive scaling parameter.

Besides, in order to give a comprehensive analysis, I introduce the prediction rate for reference. Since the model output is the forward-intensity λ_t for time $[t, t + \tau]$ other than an exact value, the count of terrorist attacks prediction within a period $[t_1, t_2]$ can only be calculated in a statistical way based on Poisson's property, i.e., $\hat{K} = \sum_{t_1}^{t_2} \lambda_t$. I define the prediction rate of the model over a specific time as:

$$\text{Prediction Rate} = \frac{\min(\hat{K}, K)}{\max(\hat{K}, K)}, \quad (2.6)$$

which indicate the accuracy rate of matched prediction.

Furthermore, the overall impact of climate on the terrorism via the causal link is important for the cross-validation with results from political science [17]. Here I use the ratio of additional correct predicted counts via climate data to the terrorism data as the index. For climate gain, I only consider the years that have positive likelihood gain, then the climate gain ratio is defined as:

$$\text{Climate Gain Ratio} = \frac{\max(0, \text{TP Difference})}{\text{Total Terrorism Counts}}, \quad (2.7)$$

where the TP Difference is the true positive prediction difference between the counts from model with and without the climate data.

2.2.3.2 Data Description

Data Sources The terrorism data come from Global Terrorism Database¹ (GTD) which is one of the most comprehensive global datasets on domestic and international terrorist attacks around the world. For extreme climate events, I consider two main aspects — extreme rainfall and temperature events (e.g., floods and droughts). For flood and drought, the data come from EM-DAT², which contains global essential core data on the occurrence of floods and droughts in countries. For extreme temperature, I consider the absolute anomaly value in the mean, lowest and highest temperature value, compared to standard baseline of local temperature, as the extreme temperature events. The corresponding data come from Berkeley Earth³, which supplies timely, impartial, and verified temperature data for countries. For cross validation, I use the *country climate impact sensitivity index* introduced by ND-GAIN⁴ [17].

¹<https://www.start.umd.edu/gtd/>

²<https://www.emdat.be>

³<http://berkeleyearth.org>

⁴<https://gain.nd.edu/our-work/country-index/rankings/>

Feature Pre-Processing In my implementation, each country are studied independently and the minimum time unit is one day. The target output for training is the count of terrorist attacks within country by day. The input feature attributes are formulated as follow: (i) Historical terrorism feature. I use the mean daily occurrence of attacks in past 15, 90 and 365 days to represent the trend of attack risks. (ii) Historical extreme climate event feature. I consider the climate influence in both short term (6 months) and long term (36 months), while 6 months is the average cycle period of crops, 36 months is the average period of El Niño and La Niña. For rainfall feature, I use the count of floods and droughts in past 6 and 36 months. For temperature feature, I take the sum of the absolute value of anomaly in past 6 and 36 months.

Training and Inference In the experimental analysis, I perform a one-year step forward training and inference. In each step, six years' data are applied for training and thereafter the model gives the intensity prediction day by day in the next year. I mainly focus on the results from year 2000 to year 2019 whilst 6 years' more data from year 1994 are used for the initial training.

2.2.3.3 Results and Discussion

I select a diverse range of 8 countries based on a combination of factors: total volume of political violence (1993-2019), geographic and ethnolinguistic diversity from each other, and genre of political violence. Fig.2.10 demonstrates the results for Afghanistan (AFG), Thailand (THA), Nigeria (NGA), India (IND), Philippines (PHL), Algeria (DZA), United Kingdom (GBR), Colombia (COL). The raw output from the NFIPP is daily based, but for simplicity of overall understanding, I take the period of one year in evaluation (e.g., likelihood gain).

Cross-Validation Against Climate Resilience Index The climate gain ratio indicate that to what extend can climate non-linearly excite the additional terrorism. In order to cross-validate the results, I use the ND-GAIN country climate impact sensitivity index

[17], which indicates the extent to a country depending upon a sector negatively affected by climate hazard and the proportion of the population particularly susceptible. This includes multiple sectors such as food, water, healthcare, ecosystem, human habitat, and infrastructure. The comparison is demonstrated in Fig.3.3. The results from NFIPP and ND-GAIN show a strong similarity in trend. This allows us to verify that a country has a high causal link between extreme climate events (climate gain ratio) in my model also has a corresponding high sensitivity evaluated using more qualitative methods (e.g. ND-GAIN).

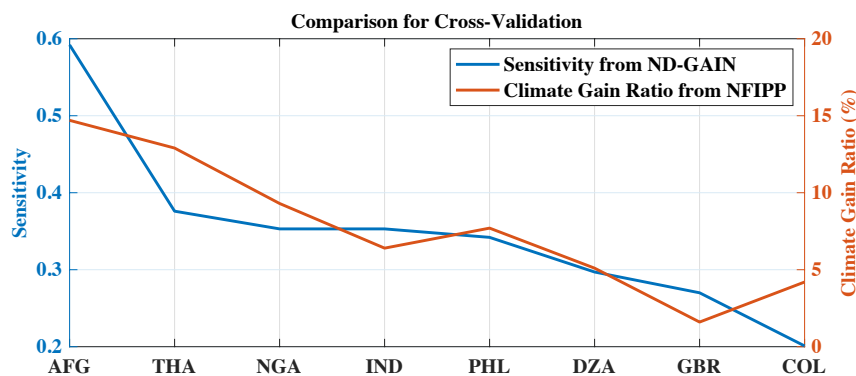


Figure 2.9: Cross-validation results across diverse countries and conflict situations.

General Analysis The positive likelihood gain represents the relative strength of the causal link from climate to terrorism, while the negative likelihood gain indicates that the attacks' mechanism is different from the climate induced pattern and have other causes.

Firstly, I focus on the years with negative likelihood gain in Fig. 2.10a. From an economic and political point of view, most negative values can correspond to explanation in economic and political science. e.g., the 2008 financial crisis deeply affect the pattern in India, UK, and Philippines; the new allegiance of Boko Haram to the Islamic State of Iraq in Nigeria 2015 and the Royal Thai Army coup in Thailand 2014. These events acted as additional causal excitation to the terrorism mechanism and overwrote the causal link with climate. Secondly, if I trace backwards the climate data that have a spike positive likelihood gain via climate, the corresponding reasons can be found, e.g., Philippines reported over 60 floods and droughts during 2009-2011 which is much higher than usual

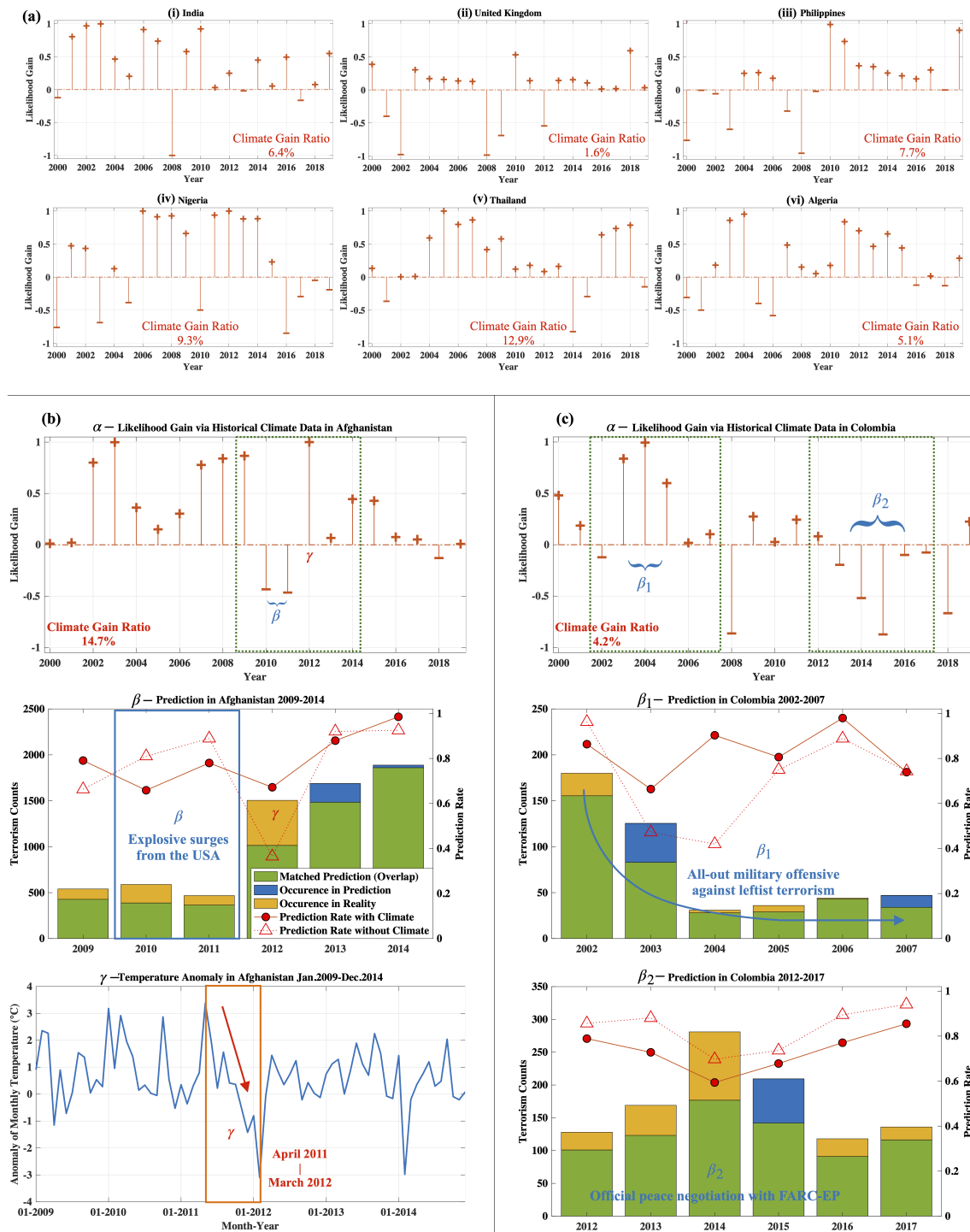


Figure 2.10: Experiment results. (a) General pattern analysis in representative countries; (b) Detailed analysis in Afghanistan; (c) Detailed analysis in Colombia.

years; Nigeria continuously suffered from anomaly high temperatures from 2002 to 2019.

Table 2.1: Example events that influence political violence.

Category	Events Example	Affected Countries
Climate	Typhoon, Drought	ALL
Economy	Financial Crisis	IND, GBR, PHL, COL
Politics	Coup, Surge	NGA, THA, AFG, COL

In Tbl.2.1, I summarize the key recent historical events that I think have influenced the development of political violence in different countries. These conclusions are reinforced by various political and social science literature, but I acknowledge that further cross-domain expertise integration is the key to informing interpretation. In the next sub-section, I take two countries which have different geographic, ethnolinguistic, and political background, i.e. Afghanistan and Colombia, as the examples for a more detailed analysis.

Detailed Analysis per Country (i) Afghanistan (Fig.2.10b). Despite Afghanistan is involved in a protracted War on Terror (2001-2019), the causal link from climate to terrorism existed in most of these years and offer a significant 14.7% gain ratio. Of interest is the likelihood gain fluctuation (Fig.2.10b- α) between 2009 and 2014. From Fig. 2.10b- β , I know that the total number of attacks in 2009-2011 are approximately the same, whilst the causal link break up in last two years. One possible reason was that the explosive surges from the USA (i.e. the troop level is doubled by the new president Barack Obama in those two years [109, 110]) just before the 2012 troop withdrawal, which resulted in the modification of terrorism pattern in 2010-2011. Besides, I notice that in 2012, the causal link reverse into a strong link and boost the prediction rate in that year. By looking back at the climate data, I found that, during the period of April 2011 to March 2012, the extreme temperature in Afghanistan experienced a cliff change from over-heating to under-cooling (Fig.(2.10)b- γ), while the events of drought and flood tripled compared to the past three years. By comprehensive analysis with Fig.(2.10)b, I believe that the there exist dominated excitation of climate in terrorism in Afghanistan.

(ii) Colombia (Fig. 2.10c). Climate events do not have great excitation on the terrorism in Colombia with the climate gain ratio of 4.2%. However, potential causes can still be found if I take a closer look at the period of 2002-2007 (β_1) and 2012-2017 (β_2). Fig.(2.10)c- β_1 shows a sharp fall in the attack number from 2002, where the likelihood gain via climate become strong positive. One of the major reasons is that Álvaro Uribe was elected as the president of Colombia. He then led an all-out military offensive against leftist terrorism just following his 2002 election, which eliminated the political-based terrorist attacks. The remaining follow the unorganized attack pattern which cause by extreme climate events and drive a raise in the likelihood gain. Another notable period is 2012-2017 (β_2) where the likelihood gain gradually goes down and remain negative. Actually, from 2012 to 2016, the government of Colombia was on an official peace negotiation with the largest terrorism group FARC-RP. However, Fig.(2.10)c- β_2 shows an increase in attack number instead of decrease during the peace negotiation. By comprehensive analysis with Fig.(2.10)c, I believe that the politics and domestic terrorist groups are more dominated on the causal excitation terrorism in Colombia.

The terrorism patterns are various from countries and time to time. In my analysis, the period of one year is taken for analysis. With further domain expertise, my results are day-by-day and adaptive for any time range analysis.

2.2.4 Conclusion and Future Work

The causal mechanism between climate and political violence is fraught with complex mechanisms. Current quantitative causal models largely rely on assumptions unsuitable for conflict mechanisms, namely: continuous and linear drivers, and sufficient data to model priors. Given that there is sufficient literature to show that drivers are often excitation based (e.g., a single event triggers a transformation but not thereafter), I overcome the aforementioned causal modeling challenges by proposing a neural forward-intensity Poisson process model. The model is designed to capture the potential non-linear excitation causal mechanisms in climate induced political violence, whilst being robust to

sparse and timing uncertain data. Furthermore, the raw predictive output is a day-by-day Poisson process, improving temporal resolution over aggregated monthly/annual predictions. My results span 20 recent years and reveal an excitation-based causal link between extreme climate events and political violence across diverse countries. Furthermore, I label historical events that either improve or reduce my predictability gain, demonstrating the importance of domain expertise in informing interpretation.

My future work will try to take account of neighbourhood relations between countries to understand larger scale climate effects, political relations, and the diffusion of conflict, via a spatio-temporal point process [171] and a combined neural training of the diffusion intensity process, with an explainable neural network [135] embedded in. I hope the techniques developed help to forecast and limit future conflicts [43].

Acknowledgements

I acknowledge Alan Turing Institute's Defence & Securities Program for funding this under GUARD (CONTINENT extension) project. I thank the MET Office's James Bacon and Edward Pope in assisting us with interpreting extreme climate data.

Chapter 3

Papers — Explainable Neural Inference

3.1 Approximate Symbolic Explanation for Neural Network Enabled Water-Filling Power Allocation

3.1.0 Abstract

Water-filling (WF) is a well-established iterative solution to optimal power allocation in parallel fading channels. Slow iterative search can be impractical for allocating power to a large number of OFDM sub-channels. Neural networks (NN) can transform the iterative WF threshold search process into a direct high-dimensional mapping from channel gain to transmit power solution. My results show that the NN can perform very well (error 0.05%) and can be shown to be indeed performing approximate WF power allocation. However, there is no guarantee on the NN is mapping between channel states and power output. Here, I attempt to explain the NN power allocation solution via the Meijer G-function as a general explainable symbolic mapping. My early results indicate that whilst the Meijer G-function has universal representation potential, its large search space means finding the best symbolic representation is challenging.

3.1.1 Introduction

3.1.1.1 Multi-Channel Power Allocation

Deep learning is capable of achieving effective high-dimensional non-linear mapping that can improve the efficiency of traditional low-dimensional linear or non-linear iterative algorithms [27, 78, 92]. The classic MIMO Water-filling (WF) power allocation is selected as a universally understood classic example to demonstrate explainability. It's a well established mechanism of Lagrangian optimisation, which produces a WF threshold that has to be iteratively searched to achieve optimal power allocation under a fixed power budget. WF is an important algorithm to analyse, because it is widely used in multiple channel systems (IEEE 802.xx series, OFDM systems), and innovations have been made to accelerate the search algorithm [5, 99], adapt it to discrete modulation schemes (e.g. Mercury-filling [84]), and use neural networks (NN) to accelerate the mapping [82, 177].

The essential functional role of that optimisation plays can be universally represented as $\mathbf{y} = f(\mathbf{x}; \lambda)$, where inputs $\mathbf{x} \in \mathbb{R}^{n \times 1}$ (e.g. channel gains) map to an output $\mathbf{y} \in \mathbb{R}^{n \times 1}$ (e.g. power allocation) via a model $f(\cdot)$. In classic WF, the Lagrangian optimisation produces an iterative solution (see Fig. 3.1a); and in NNs, an approximate non-linear mapping of $\hat{\mathbf{y}} = f^*(\mathbf{x})$ can achieve effective power allocation $\hat{\mathbf{y}}$ without iterative search for λ the WF level (see Fig. 3.1b with illustrative NN structure). This is useful for real-time power allocation [134] over a large number of parallel channels, e.g. OFDMA.

NNs lack the performance guarantee and explainable power of the classic WF solution. An open challenge with Deep Learning (DL) is the lack of transparency and trust compared to traditional model-based optimisation [26]. Black-box NNs cannot explain the essential features that influence actions (from PHY layer signal detection to MAC layer reinforcement learning optimisation [39]).

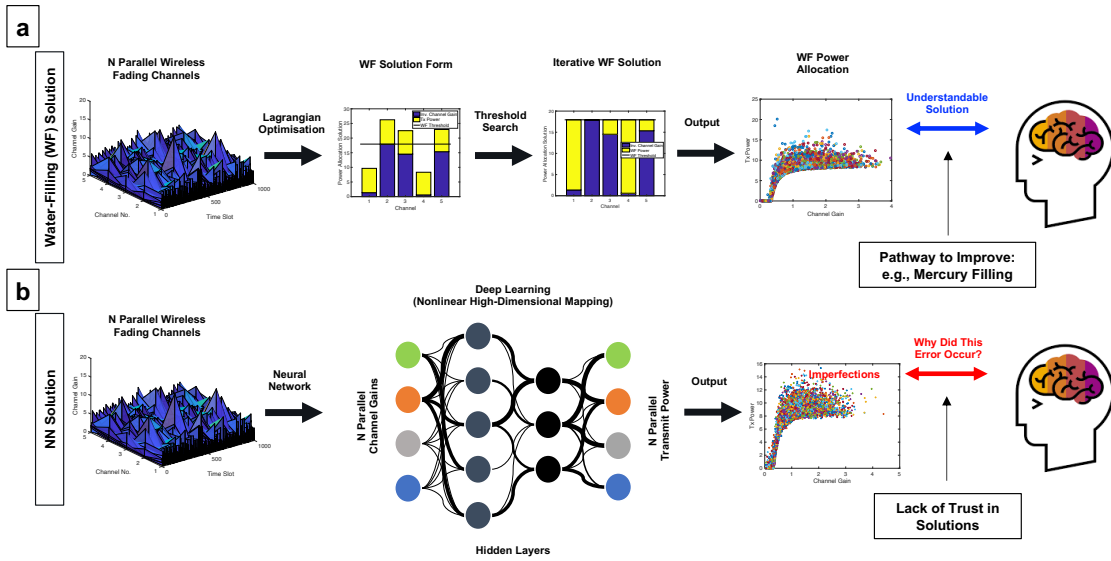


Figure 3.1: Parallel Channel Power Allocation: a) Water-filling (WF) solution via Lagrangian optimisation and iterative search with clear trust and understanding, and b) deep learning non-linear solution with no transparency and trust.

3.1.1.2 Legal and Safety Motivations to Explainable AI (XAI)

At the heart of my need to add explainability / interpretability to deep learning (DL) is the need to build trust in a quantifiable way. Traditional model-based algorithms have reasonably high clarity in how an *assumed model* and the *input data* leads to *output decisions*. However, deep learning (DL) is opaque and doesn't tell us:

- What data features are contributing to decisions
- How to improve the algorithm design
- Uncover hidden bias in the input data and algorithm
- Reverse teach human experts to uncover new insights

The *legal framework* for AI is still in its infancy, and there are several explicit cases for XAI in different regions:

- EU: 2018 General Data Protection Regulation (GDPR) in EU requires machine learning algorithms to be able to explain their decisions (see Recital 71).

- USA: 2017 Equal Credit Opportunity Act update (Reg B, art 1002.9) requires agencies to provide an official set of reasons on the main factors that affect the credit score.
- National: 2016 French Digital Republic Act requires the degree and mode of algorithms that contribute to decisions, the data used and its provenance, the weight of different data features, and the resulting actions.

As such, there is the need to develop a range of **explainable AI (XAI)** solutions that attempt to de-mystify NN mappings - building trust between machine intelligence and human expertise [39]. These XAI techniques range from visualising key hidden layer features to localised linear models (LIME) [26, 108]. Visualising feature propagation can reveal useful component level insights [42] but cannot construct a general symbolic mapping of $h \rightarrow p$. LIME models offer local linear or simple symbolic models, but lack the general insight. Recent work [3] have shown that Meijer G-functions are general input output symbolic meta-representations and have the potential explain neural network mappings.

3.1.1.3 Contribution

In this initial results on XAI for wireless power control, I outline the a symbolic method to partially explain NN representation of WF. The method can be used much more widely to other areas of optimisation. My novelties in this chapter is to derive a low-dimensional symbolic representation of the NN's high-dimensional non-linear mapping via the Meijer G-function. This chapter presents a useful framework for partial XAI in wireless systems and can be expanded to more complex multi-user/-cell scenarios.

3.1.2 System Setup

3.1.2.1 Water-Filling Power Allocation

I consider a classical wireless parallel channel power allocation problem comprised of N channels with independent fading characteristics. WF power allocation is well-established and I will not detail it here. Suffice to say, under the Shannon capacity assumption and a total power budget, the solution form for power in channel $n \in N$ is of:

$$p_n = f[h_n; \lambda] = \left(\frac{1}{\lambda} - \frac{N_0}{|h_n|^2} \right)^+, \quad (3.1)$$

where N_0 is the AWGN power, h_n is the fading gain, and the parameter λ is the Lagrangian multiplier (WF level) and a total power budget of 40W is applied. The iterative search for the WF threshold $\frac{1}{\lambda}$ enables each channel to be allocated the optimal level of power. Channels with a gain that is too low will be allocated no power (i.e. $(\cdot)^+$ operator). I implement this WF algorithm for N parallel Rayleigh fading channels with an iterative search solution for parameter λ , so that to achieve power allocation p_n by $f(\mathbf{h}; \lambda)$. WF optimisation is explainable to engineers.

3.1.2.2 Neural Network Power Allocation

I implement a 2 hidden-layer NN with 10 sigmoid neurons per hidden layer, using random data division for validation and testing. The input are the channel states and the output is the Lagrangian multiplier λ . The Lagrangian parameter in turn gives the power allocation output ($\lambda \rightarrow p_n$).

The implementation parameters are given in Tbl.3.1. The training data is based on the conventional iterative WF solution. The resulting performance of the NN is as follows: training $R^2 = 0.996$, and out-of-sample test $R^2 = 0.993$, achieved convergence at 90-200 out of 1000 epochs. Details of the training, validation, and out-of-sample testing statistics can be found in Fig. 3.2.

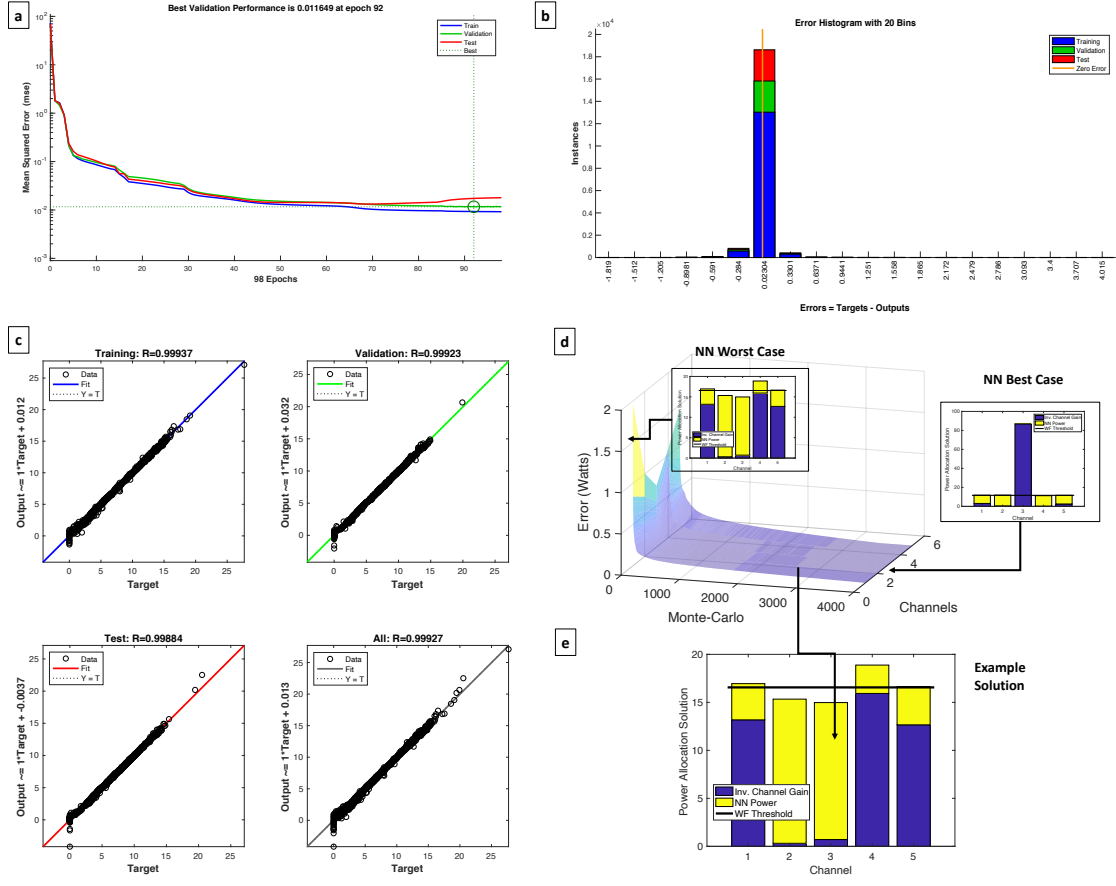


Figure 3.2: NN training & testing statistics: a) MSE convergence vs. training epochs, b) error distribution, c) regression statistics against supervision data for testing, validation, testing, and combined samples, d) ranked error distribution over Monte-Carlo runs with example of worst and best case allocation, and e) example of worst case NN power allocation.

3.1.3 Symbolic Explainability via Meijer G-function

Meijer G-function is a general function intended to include most known special functions.

A general definition is given by a line integral in the complex plane:

$$G_{p,q}^{m,n} \left(\begin{matrix} a_1, \dots, a_p \\ b_1, \dots, b_q \end{matrix} \middle| z \right) = \frac{1}{2\pi i} \int_L \frac{\prod_{j=1}^m \Gamma(b_j - s) \prod_{j=1}^n \Gamma(1 - a_j + s)}{\prod_{j=m+1}^q \Gamma(1 - b_j + s) \prod_{j=n+1}^p \Gamma(a_j - s)} z^s ds, \quad (3.2)$$

where $\Gamma(\cdot)$ denotes the gamma function. The Meijer G-function is essentially a contour integral. If solving it using the residue theorem, it becomes evident that it defines the

Table 3.1: System Setup

Parameter	Value
Channel	2.4GHz 3GPP Micro Rayleigh Fading
Wireless System	5 channels with 40W budget
Iterative WF Monte-Carlo Loops	5e4
Hidden Layers	2 with 10 Sigmoid neurons/layer
Training	Random division, ≤ 1000 epochs
Explainability	Meijer G-function

shape of the function using zeros and poles similar to filters, thereby achieving full adaptability.

As such, the Meijer G-function provides a flexible framework to discover the mapping between input variables (e.g. channel gain z) and output solution (e.g. power allocation p) through tuning integral residue by zeros and poles. Indeed, this was the motivation behind recent work in the biomedical domain [3].

Fitting the Meijer G-function to any set of data is almost impossible due to the large search space of the discrete hyper-parameters m, n, p, q . Inspired by empirical work on mapping empirical information rate contours in discrete modulation and coding schemes (see [45]), I have fitted the observed NN output of the Lagrangian multiplier λ (WF level) to an inverse tangent function:

$$\begin{aligned}
\lambda &= a \times \sum_{n=1}^5 \tan^{-1}((|h_n|^2 + b_n)^{-2} + c_n) + d \\
&= a \times \sum_{n=1}^5 \frac{1}{2|h_n|^2} G_{2,2}^{1,2} \left(\frac{1, \frac{3}{2}}{1, \frac{1}{2}} \middle| z_1 + c_n \right) + d
\end{aligned} \tag{3.3}$$

where

$$z_1 = G_{0,1}^{1,0} \left(\frac{-2}{-} \middle| |h_n|^2 + b_n \right) + G_{1,2}^{1,1} \left(\frac{-1}{-1, -2} \middle| |h_n|^2 + b_n \right) \tag{3.4}$$

Given λ , I can find the corresponding power allocation values p_n . This function was fitted to achieve the matched result showing excellent agreement with coefficient of determination $R^2 = 0.989$. The explainable symbolic result show one of many plausible

solutions and whilst the fitting is strong, this has the opportunity to expand to a wider range of symbolic representations.

3.1.4 Conclusions and Future Work

Water-filling (WF) is a well-established iterative solution to optimal power allocation in parallel fading channels. Slow iterative search can be impractical for allocating power to a large number of OFDM sub-channels. Neural networks (NN) can transform the iterative WF threshold search process into a direct high-dimensional mapping from channel gain to transmit power solution. My results show that the NN can perform very well (error 0.05%) and can be shown to be indeed performing approximate WF power allocation. However, there is no guarantee on the NN is mapping between channel states and power output.

Here, I applied a general Meijer G-function representation to neural network mapping in iterative WF optimisation to reveal the symbolic insight. However, this work is limited by the large search space of the Meijer G-function and I have only found one of many plausible solutions. I used an observational target function to alleviate this challenge here, but this cannot guarantee that it is the most representative function for the neural network.

I will advance this work by identifying low complexity search methods to find the general optimal representation to neural networks, as well as extend this explainability framework to predictive AI solutions (e.g. data-driven proactive optimisation [89] and distributed learning [174]).

3.2 Scalable Partial Explainability in Neural Networks via Flexible Activation Functions

3.2.0 Abstract

Achieving transparency in black-box deep learning algorithms is still an open challenge. High dimensional features and decisions given by deep neural networks (NN) require new algorithms and methods to expose its mechanisms. Current state-of-the-art NN interpretation methods (e.g. Saliency maps, DeepLIFT, LIME, etc.) focus more on the direct relationship between NN outputs and inputs rather than the NN structure and operations itself. In current deep NN operations, there is uncertainty over the exact role played by neurons with fixed activation functions. In this chapter, I achieve partially explainable learning model by symbolically explaining the role of activation functions (AF) under a scalable topology. This is carried out by modelling the AFs as adaptive Gaussian Processes (GP), which sit within a novel scalable NN topology, based on the Kolmogorov–Arnold Superposition Theorem (KST). In this scalable NN architecture, the AFs are generated by GP interpolation between control points and can thus be tuned during the back-propagation procedure via gradient descent. The control points act as the core enabler to both local and global adjustability of AF, where the GP interpolation constrains the intrinsic auto-correlation to avoid over-fitting. I show that there exists a trade-off between the NN’s expressive power and interpretation complexity, under linear KST topology scaling. To demonstrate this, I perform a case study on a binary classification dataset of banknote authentication. My model converge at better precision rate than state-of-the-art SVM algorithms which indicates that I do not make performance sacrifices in my approach. Meanwhile, by quantitatively and qualitatively investigating the mapping relationship between inputs and output, my explainable model can provide interpretation over each of the one-dimensional attributes. These early results suggest that my model has the potential to act as the final interpretation layer for deep neural networks.

3.2.1 Introduction

Improving the trust in deep neural networks (DNN) can be achieved by developing the statistical and explainable foundations. Trust in deep learning is important, because of the increasing widespread use of commercial DNN based artificial intelligence (AI), especially in the areas that engage with human life such as autonomous vehicles and bank transactions [1]. There are a multitude of potential risks (e.g. discrimination, adversarial attacks, over-fitting) [169] and explainable AI [4] has the potential to both offer insight during the operations and in a post-hoc manner.

3.2.1.1 Motivation and Related Work

At the heart of my need to add explainability / interpretability to DNNs is the need to build trust in a quantifiable way. Traditional mathematical model-based algorithms have reasonably high clarity in how a model and the input data leads to output decisions. Bayesian methods can quantify the uncertainty both in the forward and inverse problem. Whilst DNNs have been shown to be able to accelerate the solution discovery of many iterative optimisation problems [40], they remain opaque and doesn't tell us the impact of input data and bias on decisions, the reasoning for decisions, and how the DNN logic can reverse teach human experts.

Beyond these technical requirements, the legal framework for AI is still in its infancy, and there are several explicit requirements for XAI in different regions, such as EU GDPR (see Recital 71) requires machine learning algorithms to be able to explain their decisions. The key is that rightly or wrongly, humans can attempt to explain if prompted to, and I need DNNs to have that equal capability in order to ensure trust and a legal pathway towards improving safety and reliability.

The most common way to achieving explainability for NN is to evaluate the impact of each input on the output, e.g. Saliency maps [125], DeepLIFT [124], LIME [106] can obtain the approximate solution to provide aforementioned type of explanation by reverse analysis for instances. However, these methods are more focusing on the direct relation

between inputs and output rather than the NN structure and inner operations. Authors of [4] propose to demystify black-box models with symbolic meta-models which leads a pathway to split and explicit the inner operations of NN and inspired us to improve transparency in NNs from an activation function and topology perspective.

Flexible Activation Function Conventional NN typically have a fixed, bounded continuous non-linear activation function (AF) at each neuron, which is the key to the overall nonlinear behavior of NN. However, with the fixed AF, the expressive power of each layer is capped [86, 95, 156], thus the neural network can only become deeper in order to fit complex high dimensional nonlinear data. Therefore, there is a strand of works [2, 98, 100, 114, 115, 120, 121, 145, 175] tried to train the NN by tuning AF for purpose of enhancing the expressive power of nodes so that to reduce the topological complexity. The core idea of flexible AF is using control parameters for the curve shaping, which can be considered as interpolation between control points, while these parameters being optimized during the back-propagation process.

Interpolation methods have applied different AFs: (i) Piecewise linear interpolation [2] has the best flexibility, but easily leads to overfitting. Thus, the penalty function is crucial and sensitive to the model; (ii) Polynomial interpolation [98] avoids the overfitting to some extent due to the constraint of the function, but the local flexibility is sacrificed; (iii) Spline interpolation [23, 114, 145, 175] has both locally and globally flexibility while avoids the overfitting, but the splines are hard to express in terms of symbolic functions, which is the gateway to all explanations [4]; (iv) Gaussian Processes (GP) interpolation [115, 140] is a non-parametric model that can give the symbolic expression of AFs, while offer an expressive power vs. explainability trade-off in training set from the kernel space. Due to the kernel function constrains, GP AFs can effectively prevent overfitting as well as give a symbolic function that conforms to intrinsic autocorrelation which is better than others for explanation purpose. Therefore, in my proposed NN, I make each AF be generated by GP interpolation between control points within it, which can thus

be tuned during the back-propagation procedure via gradient descent. By visualizing the activation, I can decrease difficulty in explaining the role of each node and layer in NN.

NN Topology In common NN, topology is chosen from a set of known models (AlexNet, VGGNet, GoogleNet, etc) or customized with few limitations [9,64,168]. Moreover, from the field of evolutionary computing, search algorithms for neural network topologies are proposed in [6,37,56,167,172]. However, dynamic and complex topologies are not suitable for my main purpose – Explainability. In this chapter, I propose lay my explainability scheme on a fixed topology mode. [11,55,66] shows that Kolmogorov–Arnold Superposition Theorem (KST) can offer an approximation to any continuous function in high dimensional space using a finite composition of (a) univariate continuous functions and (b) addition operation. Therefore, based on KST, I establish a scalable NN topology as the foundation of my explainability since both (a) and (b) are the basic elements of all operations which are easier for understanding.

3.2.1.2 Novelty and Contribution

In this chapter, I approach the problem of NN explainability framework by introducing flexible activation functions into Kolmogorov–Arnold Superposition Theorem (KST) based topology NN for interpreting its inner workings in terms of symbolic and visualized functions for each neuron. Under my proposed model, I achieve global transparency to the NN and show the trade-off between NN expressive power vs. explainability. The remainder of this chapter is organised as follows. In Section 2, I build a system model step by step. In Section 3, I apply the model to the a binary classification dataset of banknote authentication for case study and conduct the explainability analysis of it. Section 4 concludes this chapter and proposes the ideas for future work.

3.2.2 System Model

3.2.2.1 Neural Network Topology

Kolmogorov–Arnold Superposition Theorem Hilbert’s 13th problem solved by Kolmogorov–Arnold superposition theorem (KST) [66], neural networks (NN) can prove to be universal approximators for every continuous function mapping [20, 76]. There are many generalizations and refinements of KST. I state one of these, which uses the primary work in [66]. To be specific, for any $D \in \mathbb{N}$, there exist $R \leq 2D$ and continuous functions $\phi_{rd}(\lambda_d) : \mathbb{I} \rightarrow \mathbb{R}$ for $d = 1, 2, \dots, D$ and $r = 0, 1, \dots, R$, such that: for every arbitrary multivariate continuous function $f(\boldsymbol{\lambda}) : \mathbb{I}^D \rightarrow \mathbb{R}$, where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_d, \dots, \lambda_D]^T$ there exist continuous functions $\Phi_r : \mathbb{R} \rightarrow \mathbb{R}$ for $r = 0, 1, \dots, R$, such that I may define:

$$F(\boldsymbol{\lambda}) = \sum_{r=0}^R \Phi_r \left(\sum_{d=1}^D \phi_{r,d}(\lambda_d) \right) \quad (3.5)$$

as an approximate realization of function $f(\boldsymbol{\lambda})$; that is, given any $\varepsilon > 0$, $|F(\boldsymbol{\lambda}) - f(\boldsymbol{\lambda})| < \varepsilon$ for each $\boldsymbol{\lambda} \in \mathbb{I}^D$, which means functions of the form $F(\boldsymbol{\lambda})$ are dense in $C(\mathbb{I}^D)$. Kolmogorov also showed that the inner functions ϕ_{rd} are independent with the outer functions Φ_r may have latent dependence of the target function f .

KST-Based NN Topology I construct my initial NN topology model according to the state of KST. Each neural node contains an adjustable activation function which maps the sum of inputs into its output. I set the $R = 0, 1, 2, \dots$ in (3.5), the *repetition level*, as the only parameter of my proposed topology, which control scaling as well as the trade-off between potential approximation ability and width of the NNs model while the depth of NN is fixed. Meanwhile, I define each repetition in topology as a *unit*, and the final estimated output $\hat{y}$ is the sum of $R + 1$ units. This enable us to alleviate the complexity of explanation into a fixed mode. Schematic diagrams in Fig.3.3 demonstrate two topology examples and indicate that the KST topology is the key to the trade-off between model expressive power and explainability while flexible activation function is

applied to enhance the expressive power of each node – as I will show in Section 2.2.

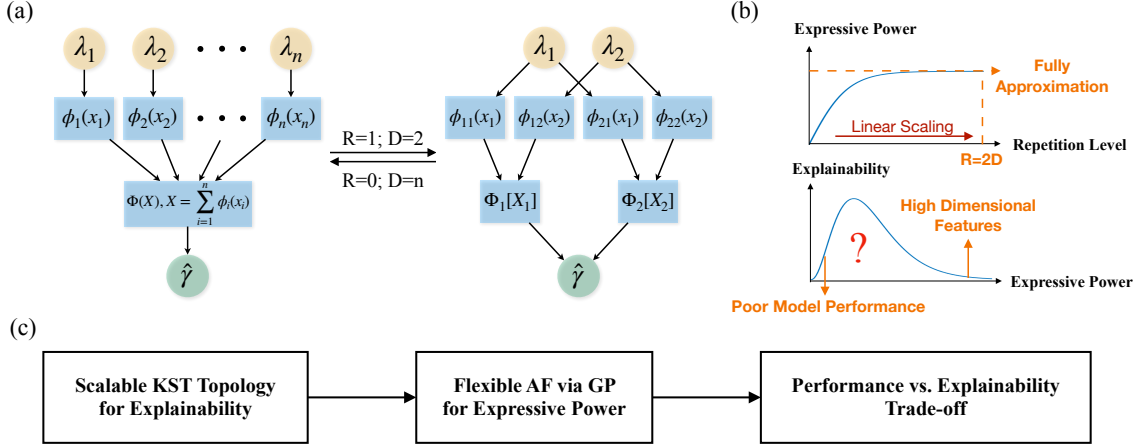


Figure 3.3: (a) The KST based neural network topology. Φ and ϕ are the flexible GP AFs. Left gives an example for one unit with n dimensions inputs while right is two units topology for two dimensions inputs; (b) Qualitative relationship within expressive power, scaling and explainability in my model; (c) My working flow for scalable partial explainability in neural networks via flexible AFs.

3.2.2.2 Activation Function

Different from the conventional NN using fixed AF while training the additional weight and bias of each input for the neural node, I use unmodified input for each proposed neural node where an adjustable AF is within. In other words, I embed the weight and bias of each input into the AF it is designated, and at the same time, tuning the shape of AF.

In order to tackle the aforementioned problem in 3.2.1.1 (i.e. function discontinuity, local tuning difficulty, parameter dependency and model over-fitting), I proposed to apply the noise-contained Gaussian Processes (GP) to fit the AF with control points. Specifically, each AF is generated by a GP regression of the control points while I tolerate the existence of noise on control points.

In this case, I can achieve the following objectives for the AF: (1) Ensure of the intrinsic autocorrelation within the function for smoothness and explainability; (2) Gain both local and global function adjustability; (3) Avoid over-fitting.

Priori Gaussian Processes Consider each activation function is assumed to follow a latent GP plus noise which can be expressed as:

$$\phi(x) = \text{GP}(x) + \varepsilon \quad (3.6)$$

where $\text{GP}(x)$ is the random variable (RV) which follows a distribution given by GP, and ε is the additive Gaussian noise with zero mean and variance σ^2 . From the continuous AF domain, finite number of control points are taken as $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, with $\mathbf{y} = \phi(\mathbf{x}) = [y_1, y_2, \dots, y_n]^T$, can be assumed to follow the multivariate Gaussian as

$$\text{GP}(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \mathbf{K}(\mathbf{x}, \mathbf{x})) \quad (3.7)$$

where $\boldsymbol{\mu}(\mathbf{x})$ is the mean function and $\mathbf{K}(\mathbf{x}, \mathbf{x})$ is the covariance matrix given by:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \quad (3.8)$$

where $k(x_i, x_j)$ is the covariance between RVs $\text{GP}(x_i)$ and $\text{GP}(x_j)$ represented by the kernel function. Thus, according to 3.6 and 3.7, the priori GP probability model can be expressed as:

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \mathbf{C}(\mathbf{x}, \mathbf{x})). \quad (3.9)$$

where $\mathbf{C}(\mathbf{x}, \mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}_n$ and $\mathbf{I}_n$ is the diagonal identity matrix.

Kernel Function In GP, the covariance between every two RVs is quantified by the kernel function which interprets the potential correlation between RVs. Several appropriate kernels can be selected for different priori knowledge, e.g. smooth curve is obtained by fitting with radial-basis function (RBF) kernel and exp-sine-squared kernel is designed for

periodic patterns. In my proposed activation function, I aim for two goals, i.e. (1) Global autocorrelation with local adjustability; (2) Continuity and smoothness, so that rational quadratic (RQ) kernel would be a default choice in my experiments:

$$k(x_i, x_j) = \sigma^2 \left(1 + \frac{(x_i - x_j)^2}{2\alpha l^2}\right)^{-\alpha} \quad i, j = 1, 2, \dots, n \quad (3.10)$$

where σ^2 determines the variance magnitude, l is length-scale parameter and α is scale mixture parameter. RQ kernel can be considered as a combination of infinite sum of RBF kernels with various length-scales hence to vary smoothly across multiple length-scales which makes it more flexible than the RBF kernel to fit local adjustment of control points and to alleviate gradient vanishing problem while maintain the noise resistance property of GP [157]. Further more, various common and customized kernels alternatives may also be applicable in some specific scenario with priori knowledge embedded [117, 128, 132].

Posterior Gaussian Processes Kernel function selected, the hyper-parameters $\boldsymbol{\theta}$ of the kernel can be tuned by maximizing the corresponding log marginal likelihood function which is equivalent to minimizing the cost function:

$$\arg \min_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \mathbf{y}^T \mathbf{C}^{-1} \mathbf{y} + \log |\mathbf{C}| \quad (3.11)$$

The conventional quasi-Newton and gradient descent methods can be used in this optimization problem while [164] also offers an algorithm for reducing the computational complexity of hyper-parameter learning from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ without performance loss.

In the kernel hyper-parameters tuning process, due to the allowance of noise ε in $\phi(\mathbf{x})$, the AFs effectively avoid over-fitting, since the GP will follow the covariance constraints given by kernel and result in a curve which is not guaranteed to pass through all the control points. Therefore, the GP can give penalty to a potential over-fitted control point by considering it as noise, so that to keep itself still being an autocorrelation function.

Kernel hyper-parameters tuned and optimized, GP can give the posterior Gaussian

distribution for every RV $y_* = \phi(x_*)$ within the AF domain with mean and variance as [104]:

$$\overline{\phi(x_*)} = \boldsymbol{\mu} + \mathbf{k}^T(\mathbf{x}, x_*) \mathbf{C}^{-1}(\mathbf{y} - \boldsymbol{\mu}) \quad \text{with} \quad V[\phi(x_*)] = k(x_*, x_*) - \mathbf{k}^T(\mathbf{x}, x_*) \mathbf{C}^{-1} \mathbf{k}(\mathbf{x}, x_*) \quad (3.12)$$

and here I can assume that $\boldsymbol{\mu} = \mathbf{0}$ [158]. Same as Bayesian deep learning, the GP AFs give not only the mean values but also the variance which lead the pathway to quantify the epistemic and aleatoric uncertainties of the model [62]. In this chapter, I only use the mean for activation value and discuss the explainability.

3.2.2.3 Back-Propagation for Control Points Tuning

Feed-forward NN structure established, I can use back-propagation algorithm for control points' coordinate tuning so that to adjust the AF. Let $L(\Theta)$ denote the loss for a batch of instances, where $\Theta = \{x_1, x_2, \dots, y_1, y_2, \dots\}$. In each epoch for a batch of training instances, I can perform an update on control points' coordinate with a learning rate η as:

$$\begin{bmatrix} x_i^{t+1} \\ y_i^{t+1} \end{bmatrix} \leftarrow \begin{bmatrix} x_i^t \\ y_i^t \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial L(\Theta)}{\partial x_i} \\ \frac{\partial L(\Theta)}{\partial y_i} \end{bmatrix} \quad (3.13)$$

Similar to conventional NN, the chain rule still works for loss gradient descent whatever loss function (e.g. cross entropy, MSE loss, etc.) is chosen while the key is to obtain the function slope $\kappa(x_*) = \frac{\partial \overline{\phi_*}}{\partial x_*}$ at each point x_* and the gradient $\Delta_i = [\frac{\partial \overline{\phi_*}}{\partial x_i}, \frac{\partial \overline{\phi_*}}{\partial y_i}]$ for each control point, where $i = 1, 2, \dots, n$. Due to the complex optimization processes in GP regression, for calculation reduction, I use finite difference method in (3.12) for $\kappa(x_*)$ and Δ_i approximate solutions. I also perform some experiments in which $\kappa(x_*)$ and Δ_i are given precisely, however the accuracy improvement is ambiguous.

3.2.2.4 Symbolic Model and Visualized Explainability

Using all trained AFs, the symbolic representation of the model (3.14) can be directly derived from (3.5) and (3.12) which gives the transparency to the full mapping relationship. Furthermore, with figures of the AFs (e.g. Fig.3.4a), I can backtrack how each instance inputs map to the output visually. i.e. At the last layer, contribution from each unit can be evaluated, so that the reason why model perform well/bad in each instance can be attributed to some specific units. In the same way, I can discover which input features are decisive at the first layer. Moreover, I can easily reverse this model in order to explore the data group which lead to my concerned output. Therefore, the noise attributes and potential feature discrimination might be dug out. In the next section, I present a case study to illustrate how my proposed AFs offer the transparency and partial explainability for the neural network.

3.2.3 Case Study and Result Discussion

In this section, I perform a case study on a 1372 instances, four-attributes binary classification dataset for banknote authentication¹ with attribute information: (λ_1) variance of wavelet transformed image; (λ_2) skewness of wavelet transformed image; (λ_3) kurtosis of wavelet transformed image; (λ_4) entropy of image. Each attribute is standardized into $[-1, 1]$ interval for clearer illustrating. Furthermore, I add an attribute with artificial noise (λ_5), which uniformly distribute over $[-1, 1]$, in order to evaluate the model's ability to deal with noise. I set two criteria to evaluate the model performance, namely the classification accuracy and the numerical loss. I use one for genuine and zero for forged to represent the authenticity of the banknote. I use the nearest neighbor principle for classification while calculate the numerical loss for back-propagation. The model and system setup of my experiments implementation are given below in Tbl.3.2.

¹Data Source: <https://archive.ics.uci.edu/ml/datasets/banknote+authentication>

Table 3.2: System Setup

Parameter	Value
Number of Units	2
Number of Control Points	6 for Each Neuron
Control Points Initialization	Random
GP Interpolation Kernel	Rational Quadratic
Learning Rate	Layer 1: 1e-1; Layer 2: 1e-3
Division	Random Division; Training: Validation (Test)=7:3
Back-propagation Epochs	1000
Processor	2.3 GHz 8-Core Intel Core i9
Memory	16 GB 2400 MHz DDR4

3.2.3.1 Results Overview

Symbolic Explainability - After 428s running time, I obtain the symbolic trained model in the form of:

$$F(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = \sum_{r=0}^1 AF_r^{Out} \left(\sum_{d=1}^5 AF_{r,d}^{In}(\lambda_d) \right) \quad (3.14)$$

Fig.3.4(a) visualizes the trained NN networks with every AF while I backtrack two typical representative data from two classes in the AFs with green points for the genuine banknote and red crosses for the forged one. Fig.3.4(b) shows the classification accuracy achieve 100% for both training and validation set after around 120 epochs, which is higher than the classical SVM (99.2%) on this dataset while the numerical loss is converged within $[10^{-3}, 10^{-2}]$ for each instance and no overfitting has occurred. Fig.3.4(c) focuses on three noteworthy AFs.

3.2.3.2 Model Interpretation

In my model, the output result is the sum of the outputs from the two units. Consider that each unit represents a feature of the data which act as an additive to the model result, thus I separately analyze how these units effect the model result. At layer 2, the AF1 and AF2 (Fig.3.4a) give different value ranges with $\Delta_1 \approx 0.8 > \Delta_2 \approx 0.25$ which indicates that the unit 1 has a more decisive impact than unit 2 on the model result, meanwhile it

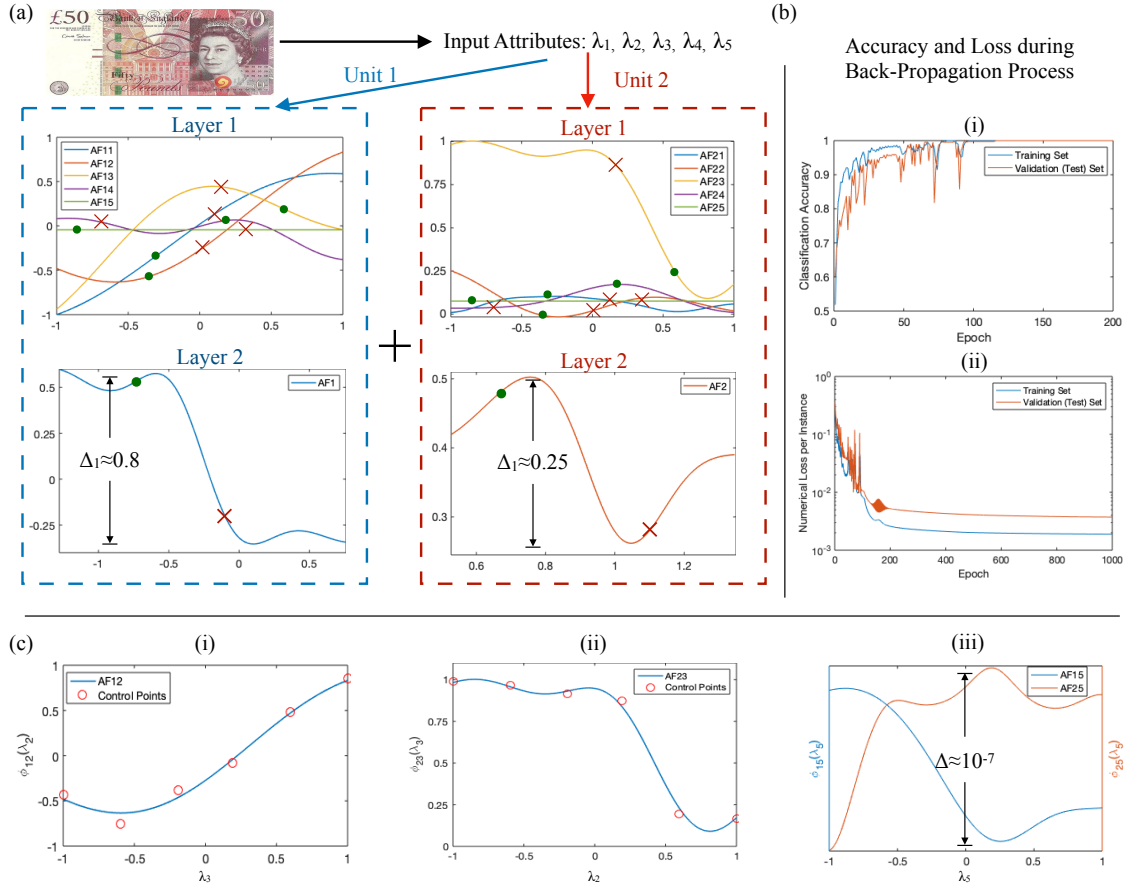


Figure 3.4: Visual Explanation: (a) The AFs in each neuron after training process are demonstrated. The green points and red crosses marked on the AFs show how two typical representative inputs from two classes data map to the output; (b) The model performance are evaluated with both the classification accuracy and the numerical loss during the back-propagation process; (c) Three noteworthy AFs are displayed separately for analysis. — (i) Approximately linear AF12 regression; (ii) Cliff nonlinear AF23 for classification; (iii) Uninfluential AF15&AF25 for noise filtering.

is worth mentioning that AF1 demonstrate a distinct trend towards binary classification, which such is the case in the previous description of KST – the outer functions may have latent dependence of the target function. Firstly, focus on unit 1, it can be qualitatively observed that the λ_1, λ_2 and λ_3 are the main attributes which effect most on the unit 1 output. Fig.3.4c(i) gives an example for AF12 with its control points which start with a small decrease followed by an increase. Analyze it together with the layer 2, I can conclude that the lower λ_2 make the banknote more easily to be classified as genuine. Turn to unit 2, the λ_3 almost dominate on the output of unit 2. With a rapid change

over $[0, 0.5]$ interval in AF23 (Fig.3.4c(ii)), λ_3 attribute has the ability to distinguish the two classes of data in unit 2. In both units, the impact of attribute λ_4 is not notably on the classification result of my model due to relatively narrow range of AF14 and AF24. Meanwhile, Fig.3.4c(iii) shows that my model has the robustness to noise attribute λ_5 – AF15 and AF25 give extremely little contribution to the model result other than over-fit the data.

In my model, the symbolic expressions can help data scientists with better understanding of the global mapping relationship within the NN. Furthermore, by analyzing the trained model reversely, scientists can anticipate potential risks in advance about how banknote would be forged to pass the detector under this case. Meanwhile, for AI users, the one-dimensional visual functions flow offer the transparency and partial explainability on how model result come from each attribute input, which can enhance the trust to the model.

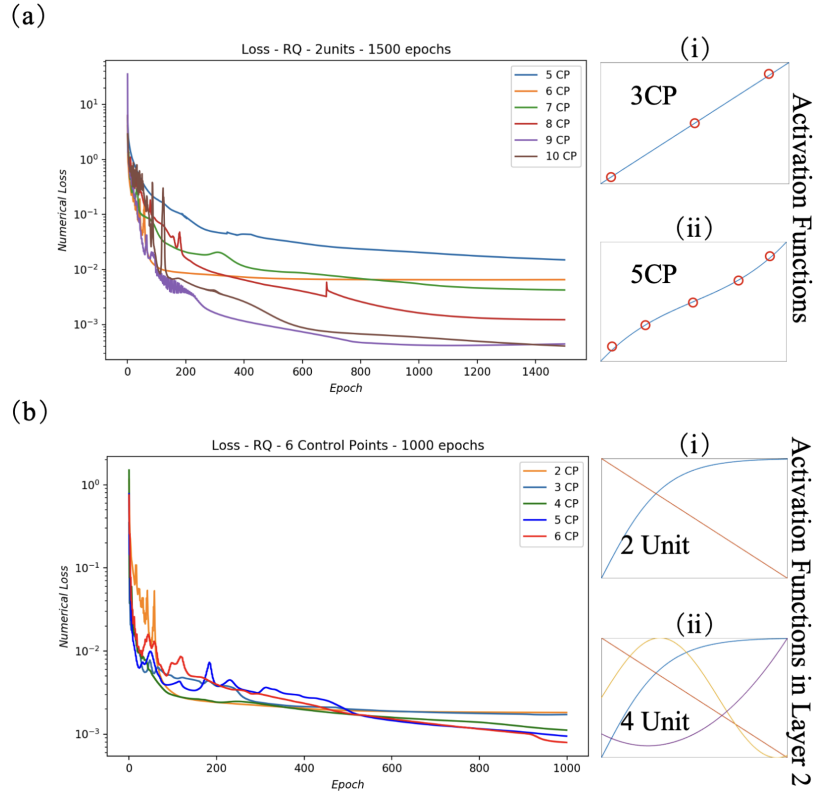


Figure 3.5: More control points in each neural (a) and more units in topology (b) can lead to better model performance, however as such, the difficulty of model explaining for AI users will increase at the meantime.

3.2.3.3 Scalability, Expressive Power and Explainability

My model is linear scalable with the repetition level in topology and the number of control points in each neuron. The number of control points determines how much details can the function has while the units number determines the maximum number of potential features that can be extracted. Fig.3.5 shows that more units and more control points lead the model converging at lower numerical loss which means stronger model expressive power. However, the difficulty of model explaining for AI users is increasing at the meantime – The AI users are not sensitive to small variations in the function (Fig.3.5a) while confuse the role of multiple units (Fig.3.5b). The ascent of model accuracy and model explaining difficulty are subject to different scales – $\log$ scale and linear (or even exponential) scale. Thus, I need to find a balance between these two or establish a trade-off based on different scenario requirements.

3.2.4 Conclusion and Future Work

In this chapter, I found that by modeling neural networks as a combination of linear scalable architectures via the Kolmogorov–Arnold Superposition Theorem (KST) and generalised Gaussian Process activation functions, I are able to demonstrate a new degree of interpretability. In particular, I have demonstrated a trade-off between expressive power and explainability of the NN through controlling the KST repetition level. This work expands on current state-of-the-art interpretation methods (e.g. Saliency maps, DeepLIFT, LIME, etc.) by focusing more on the role played by activation functions and the NN structure itself.

To demonstrate applicability, I perform a case study on a binary classification dataset of banknote authentication. My model converge at better precision rate than state-of-the-art SVM algorithms which indicates that I do not make performance sacrifices in my approach. Meanwhile, by quantitatively and qualitatively investigating the mapping relationship between inputs and output, my explainable model can provide interpretation

over each of the one-dimensional attributes. These early results suggest that my model has the potential to act as the final interpretation layer for deep neural networks.

My future work will focus on (i) fitting orthogonal features to different unit in order to compress my network by reducing the repetition level while keeping approximately the same model accuracy [59,97]; (2) establishing a pattern library of common features using activation functions for easier identifying the feature represented by a unit.

3.3 Forecasting Wireless Demand with Extreme Values using Feature Embedding in Gaussian Processes

3.3.0 Abstract

Wireless traffic prediction is a fundamental enabler to proactive network optimisation in 5G and beyond. Forecasting extreme demand spikes and troughs is essential to avoiding outages and improving energy efficiency. However, current forecasting methods predominantly focus on overall forecast performance and/or do not offer probabilistic uncertainty quantification. Here in this chapter I design a feature embedding (FE) kernel for a Gaussian Process (GP) model to forecast traffic demand. The FE kernel enables us to trade-off overall forecast accuracy against peak-trough accuracy. Using real 4G base station data, I compare its performance against both conventional GPs, ARIMA models, as well as demonstrate the uncertainty quantification output. The advantage over neural network (e.g. CNN, LSTM) models is that the probabilistic forecast uncertainty can directly feed into decision processes in optimisation modules.

3.3.1 Introduction

Wireless traffic prediction is a key enabler for proactive resource optimisation in 5G and beyond [24, 90, 127]. Proactive optimisation can create user-centric quality-of-service (QoS) and -experience (QoE) improvements across 5G network slices [73, 116]. Direct prediction from historical data [16, 165, 173] and inference from proxy social media data [166] are important inputs to proactive optimisation modules [116] being considered for 5G and beyond applications [41], such as interference management, load balancing, localization [152], and multi-RAT offloading; with implementation on the edge or in CRANs. This is more pertinent for hyper-dense mass autonomy applications where spikes in demand across different quality of service/experience/trust demands require forward prediction [77].

I begin with a review of time-series forecasting algorithms used in wireless traffic prediction and identify a lack of research in both high and low extreme value predictions, which is of critical importance to avoiding network congestion and inefficiencies.

3.3.1.1 State-of-the-Art

Time-series prediction methods can be classified into several types, with training data in high demand.

Statistical Models Statistical time-series modelling using a variety of signal processing and machine learning approaches have been widely applied to predict the wireless traffic. Moving average models with smoothing weights and seasonality works well for univariate forecasting. For example, in [163], seasonal auto-regressive integrated moving average (ARIMA) models were fitted to wireless cellular traffic with two periodicities for prediction. However, this model is insensitive to anomalous values, such as event-driven spike demand. Indeed, predicting and avoiding spike demand is critical to avoiding network outages and improving the consumer experience. Other methods rely on statistical generative functions assuming a quasi-static behaviour, such as the exponential or α -stable model [79], but these do not offer the adaptivity of machine learning techniques below.

Machine Learning Models In terms of machine learning approaches, artificial neural networks (ANNs) has been used to predict the self-similar traffic with burstiness in [161]. Although ANNs and deep learning approaches (CNN, LSTM, wavelet, Elman) neural networks [28, 96, 148, 173] performed well in cumulative learning and prediction accuracy, it cannot give a *quantitative uncertainty* due to its intermediate black-box process. Alternatively, Gaussian Processes (GPs) have been used [165] and showed a strong adaptivity to the wireless traffic data. Nevertheless, the usage of traditional kernels are unable to capture long-range period-varying dependent characteristics which limits the efficiency of existing training data.

Gaussian Process (GP) Gaussian process (GP) is widely used because of its adaptability to manifold data. As a non-parametric machine learning method, the prior GP model is firstly established with compound kernel functions based on the background of the data. One optimizes the hyper-parameters using the training data to extract its posterior distribution for the predicted outcome. The prediction results given by GPs quantify the statistical significance, which is an important advantage over other black-box machine learning. As such, whilst GPs may not achieve the performance level of ANNs, they are able to quantify risk and that risk can be interpreted back to the features of the data [159].

Feature Extraction and Wireless Context The features of traffic patterns may be correlated if the patterns are driven by the same specified events, i.e. the rush hours, concerts, etc. In these cases, the key point is to find the implied events information from the current flow trend by identifying where its features are close to those in historical data, hence, to predict how will the traffic demand change according to it in the past. [15, 103, 151, 153, 160] addressed the problem of feature selection, in order to determine the most discriminative and relevant features of the classified data.

In the context of wireless traffic forecasting, current literature employ classic kernel functions [165], which cannot memorize the non-periodic data pattern for extended periods. This means the GP model do not make full use of the training data. Furthermore, wireless traffic forecasting is often interested in predicting extreme events as opposed to the overall pattern of the traffic variation. Extreme demand values are useful in driving proactive network actions (e.g. extreme high demand requires spectrum aggregation and cognitive access, whereas extreme low demand can lead to proactive sleep mode and coverage compensation [44]). Therefore, what is needed is an adaptive kernel in GP models to trade-off prediction accuracy between overall traffic variations and extreme values.

3.3.1.2 Novelty and Contribution

In this chapter, I propose to embed the relevant data features in a flexible kernel functions, which enable the GP model to achieve this trade-off. I make three major contributions:

1) A novel feature embedding (FE) kernel GP model is proposed for forecasting wireless traffic. Specifically, fewer hyper-parameters are required in this model, which reduce the computation burden compared with that uses classic hybrid kernels. Meanwhile, the learning rate is improved significantly for irregular training data;

2) The predicted results are quantified into probability density function (PDF), which are more useful to plug into optimisation modules than the mean prediction value. Precisely, the predicted traffic is described to follow a weighted superposition distribution of mixed Gaussian distributions instead of the sum of those in traditional GP;

3) Demonstrating my forecast model on real wireless traffic data, the cumulative error curve of my model is compared against state-of-the-art algorithms used in literature (seasonal ARIMA [163] and traditional GP model [165]). My model shows the best adaptivity and prediction accuracy trade-off between overall accuracy and extreme value accuracy.

The remainder of this chapter is organised as follows. In Section 2, I build a system model step by step from pre-processing to prediction. In Section 3, I apply the model to the real wireless traffic data and evaluate the performance of it. Section 4 concludes this chapter and proposes the ideas for future work.

3.3.2 System Model

In this chapter, I use a sliding window of historical traffic data to predict future traffic demand. In this chapter, I focus on wireless downlink (DL) traffic data demanded by end-users at 15m intervals over a two week period - see Fig. 3.6.

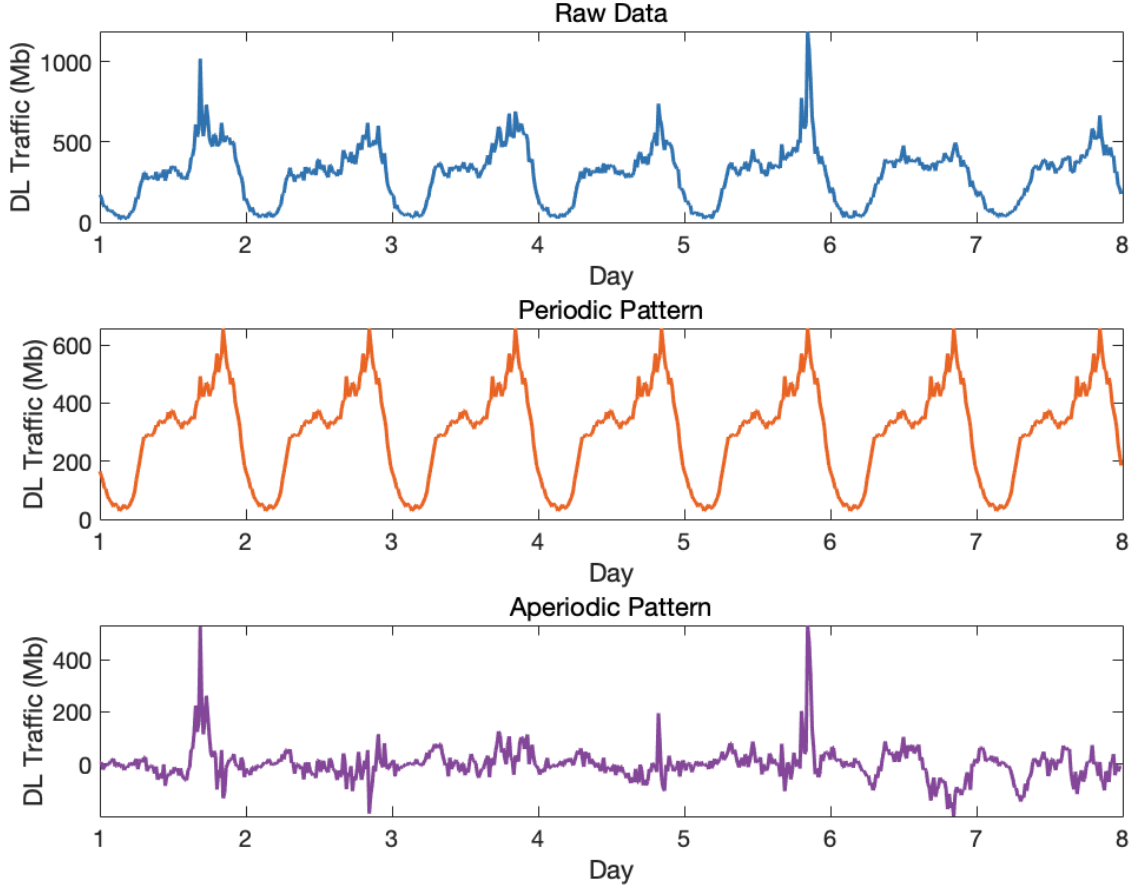


Figure 3.6: The traffic demand data is decomposed into daily periodic and aperiodic components.

3.3.2.1 Data Decomposition

The raw data is considered to be composed of a daily periodic and an aperiodic pattern from my observation and existing literature. By using a band-pass filter, the raw data can be decomposed into the aforementioned two components, as shown in Fig.3.6. In order to set the model free from the domination of large-scale periodic patterns, I fix the daily periodic pattern which is derived from the historical data as the established baseline² and only make prediction on the rest aperiodic pattern.

I assume that the aperiodic traffic consists of a noise flow and a event-driven flow which has an implicit intrinsic correlation. The latter is predictable if I can identify the

²I acknowledge that there are baseline variations between each day of the week, but I focus on the aperiodic prediction, which is the main challenge.

features relevance in this kind of flow from the noise.

3.3.2.2 Priori Gaussian Process Model

The DL traffic value at time point t is assumed to be a latent GP plus noise as

$$y(t) = f(t) + \varepsilon(t), \quad (3.15)$$

where $f(t)$ is the random variable (RV) which follows a distribution given by GP, and ε is the additive Gaussian noise with zero mean and variance σ_n^2 . From the continuous time domain, finite number of time points taken as $\mathbf{t} = [t_1, t_2, \dots, t_n]^T$, the RVs, $\mathbf{f}(\mathbf{t}) = [f(t_1), f(t_2), \dots, f(t_n)]^T$, can be assumed to follow the multivariate Gaussian as [159]

$$\mathbf{f}(\mathbf{t}) \sim \mathcal{N}(\mathbf{M}(\mathbf{t}), \mathbf{K}(\mathbf{t}, \mathbf{t})) \quad (3.16)$$

where $\mathbf{M}(\mathbf{t})$ is the mean function and $\mathbf{K}(\mathbf{t}, \mathbf{t})$ is the covariance matrix given by

$$\mathbf{K}(\mathbf{t}, \mathbf{t}) = \begin{bmatrix} k(t_1, t_1) & k(t_1, t_2) & \cdots & k(t_1, t_n) \\ k(t_2, t_1) & k(t_2, t_2) & \cdots & k(t_2, t_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(t_n, t_1) & k(t_n, t_2) & \cdots & k(t_n, t_n) \end{bmatrix} \quad (3.17)$$

where $k(t_i, t_j)$ is the covariance between $f(t_i)$ and $f(t_j)$ represented by the kernel function.

According to (3.15) and (3.16), the priori GP probability model of DL traffic can be expressed as

$$\mathbf{y}(t) \sim \mathcal{N}(\mathbf{M}(\mathbf{t}), \mathbf{K}(\mathbf{t}, \mathbf{t}) + \sigma_n^2 \mathbf{I}_n). \quad (3.18)$$

where $\mathbf{I}_n$ is the diagonal identity matrix.

3.3.2.3 Feature Embedding Kernel Function

In GP, the covariance between every two RVs is quantified by the kernel function which interprets the potential correlation of RVs in a high dimensional space. Here I use the Gaussian radial basis function (RBF) kernel with a feature embedding (FE) norm:

$$k_G(t_i, t_j) = \sigma^2 \exp\left(-\frac{\|\mathbf{\Lambda}_i^\Theta - \mathbf{\Lambda}_j^\Theta\|_2^2}{2\beta^2}\right) \quad (3.19)$$

$$\mathbb{R} \times \mathbb{R} \xrightarrow{\text{FE}} \mathbb{R}^\Theta \times \mathbb{R}^\Theta \xrightarrow{k_G} \mathbb{R},$$

where $\mathbf{\Lambda}_l^\Theta$ is defined as the Θ dimensions weighted feature matrix of the RV at time point t_l :

$$\mathbf{\Lambda}_l^\Theta = [w_1 \lambda_l^1, w_2 \lambda_l^2, \dots, w_\Theta \lambda_l^\Theta]^T, \quad (3.20)$$

where the θ^{th} feature of RV $f(t_l)$ in the matrix is from a feature generator function $h_\theta(\cdot)$ which can either be homogeneous or non-homogeneous of former values ($L < l$):

$$\lambda_l^\theta = h_\theta[y(t_{l-1}), y(t_{l-2}), \dots, y(t_{l-L})]. \quad (3.21)$$

Due to the symmetry, it can be easily proved that my new kernel function still meets the conditions of Mercer's theorem.

The feature generator $h_\theta(\cdot)$ is the key to capture the events. Here I describe an event from three perspectives. Firstly, the baseline of the events, i.e. $\sum_1^i y(t_{l-i})$ with $(i < l)$. This feature helps to measure the base size of traffic and works for events traffic that are proportional to the former values. Secondly, the differences in time series values, i.e. $y(t_{l-i}) - y(t_{l-j})$ with $(i, j < l)$. This allows the absolute change, which is associated with an event, being captured. Thirdly, the fluctuation degree, i.e. $\frac{y(t_{l-a}) - y(t_{l-b})}{y(t_{l-c}) - y(t_{l-d})}$ with $(a, b, c, d < l)$ or the standard deviation of former values. This feature enables the prediction giving a better result with the evaluation of the current traffic volatility.

In BS (coordinated) control systems (e.g. radio resource management or beamforming), understanding sharp changes in traffic demand (especially when above the cell ca-

capacity or significantly below economic profitability thresholds) is more important than average demand trends. As such, the proposed feature weighting process in this chapter focus on building a trade-off between general prediction accuracy and the aforementioned *extreme demand values*. Another advantage of feature embedding in this way is that all the features are explainable (see Section III-B), which is critical from an explainable AI perspective [41].

To achieve this, I set a threshold ξ of traffic varying value Δy at each sample time point based on historical data as shown in Fig.3.7. If Δy_p at t_p is outside the $\xi \times 100\%$ confidence interval in the distribution of Δy , the associated feature $\mathbf{\Lambda}_p^\Theta$ will be tagged as an outlier and assigned to category $\mathcal{A}$; otherwise it is assigned to category $\mathcal{B}$. The *Relief* idea in [136] is utilized, whereby the feature weights are optimized by maximizing the sum of margin from each $\mathbf{\Lambda}_{\mathcal{A}n}^\Theta$ to the nearest point with a different category $N_{\mathcal{A}n}(\mathbf{\Lambda}_{\mathcal{B}}^\Theta)$. This process is expressed as:

$$\max_w \sum_n^{|B|} (M_w(\mathbf{\Lambda}_{\mathcal{A}n}^\Theta, N_{\mathcal{A}n}(\mathbf{\Lambda}_{\mathcal{B}}^\Theta))) \text{ s.t. } \|w\|_2^2 = 1, w \geq 0 \quad (3.22)$$

where $M_w(\mathbf{\Lambda}_p^\Theta, \mathbf{\Lambda}_q^\Theta) = \sum_{\theta=1}^\Theta w_\theta \left| \mathbf{\Lambda}_p^\theta - \mathbf{\Lambda}_q^\theta \right|$, which projects the high dimensional feature vectors' norm onto one dimension, and w_θ is the weight of the θ^{th} feature.

In the Gaussian RBF kernel (3.19), the feature space can be mapped to an infinite dimensional kernel space ($e^x = \sum_{n=0}^\infty \frac{x^n}{n!}$). The hyper-parameter β controls the higher dimensional attenuation rate and has amplitude σ . Hyper-parameters are tuned by maximizing the corresponding log marginal likelihood function which is equivalent to minimizing the cost function $l(\beta, \sigma)$ [165]:

$$\arg \min_{\beta, \sigma} l(\beta, \sigma) = \mathbf{y}^T \mathbf{C}^{-1} \mathbf{y} + \log |\mathbf{C}|, \quad (3.23)$$

where $\mathbf{C} = \mathbf{K}(\mathbf{t}, \mathbf{t}) + \sigma_n^2 \mathbf{I}_n$ and $\mathbf{y}$ is the matrix of known values $[y(t_1), y(t_2), \dots, y(t_n)]^T$. The quasi-Newton and gradient descent methods can be used in this optimization problem.

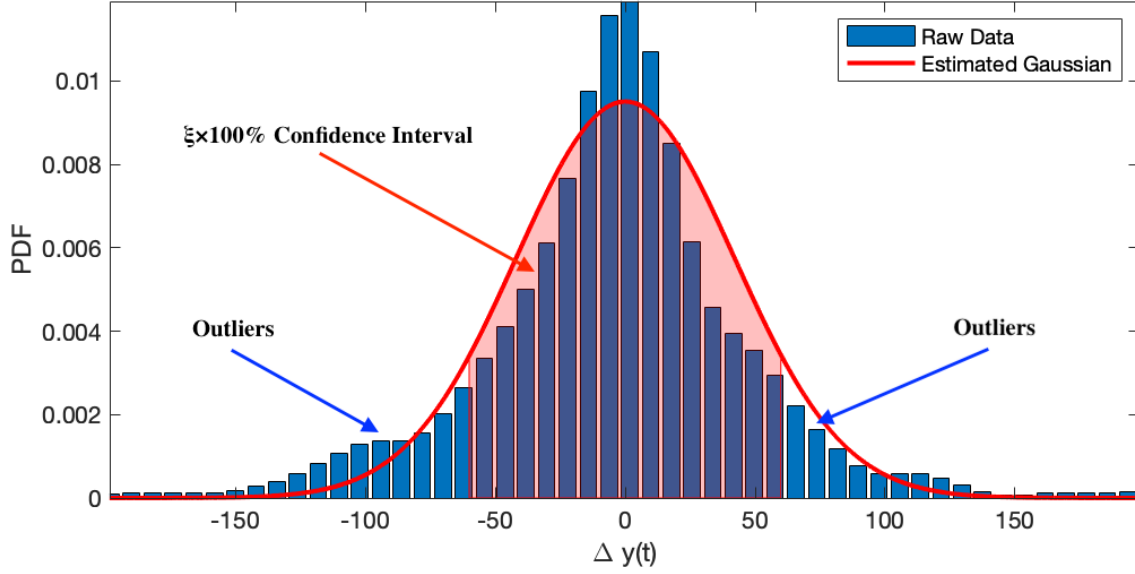


Figure 3.7: Historical time points are collected into two categories according to their estimated Gaussian distribution.

3.3.2.4 Posteriori Prediction

After the hyper-parameters are determined, the covariance of every two RVs in the training set can be quantified by $\mathbf{C}(\mathbf{t}, \mathbf{t} | \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}})$, where $\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}}$ are the optimized parameters. Let us assume that at a future time point t_f , the RV $y(t_f)$ follows the same model as the $y(t_1) \sim y(t_n)$ training set. Therefore, $\mathbf{K}(t_f, \mathbf{t} | \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\sigma}})$ yields the covariance of $y(t_f)$ with historical RVs. The multivariate distribution for any $t_i (i \in \{1, 2, \dots, n\})$ and t_f is

$$\begin{bmatrix} y(t_i) \\ y(t_f) \end{bmatrix} \sim \mathcal{N}(\mathbf{M}_{i,f}, \boldsymbol{\Sigma}_{i,f}^2) \quad (3.24)$$

with mean $\mathbf{M}_{i,f} = \begin{bmatrix} M(t_i) \\ M(t_f) \end{bmatrix}$, and covariance matrix $\boldsymbol{\Sigma}_{i,f}^2 = \begin{bmatrix} k(t_i, t_i) + \sigma_n^2 & k(t_i, t_f) \\ k(t_f, t_i) & k(t_f, t_f) + \sigma_n^2 \end{bmatrix}$.

So $\mathcal{Y}_i = [y(t_i), M(t_i), \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\beta}}]$ given, the posterior distribution of $y(t_f)$ can be derived as

$$y_i(t_f) | \mathcal{Y}_i \sim \mathcal{N}(\hat{\mu}_{i,f}, \hat{\sigma}_{i,f}^2) \quad (3.25)$$

with

$$\begin{aligned}\hat{\mu}_{i,f} &= M(t_f) + \frac{k(t_f, t_i)}{k(t_i, t_i) + \sigma_n^2} (y(t_i) - M(t_i)) \\ \hat{\sigma}_{i,f}^2 &= \sigma_n^2 + k(t_f, t_f) - \frac{k(t_f, t_i)k(t_i, t_f)}{k(t_i, t_i) + \sigma_n^2}.\end{aligned}\tag{3.26}$$

For each previous time point t_i in this model, a posterior distribution component of $y_i(t_f|\mathcal{Y}_i)$ can be generated. In naive GP, the predicted distribution $y(t_f)$ is also a Gaussian distribution which sums the influence of each previous point on its mean and variance [159]. In my proposed FE-GP forecasting model, the predicted distribution uses a Gaussian mixed model (GMM). Consider the GMM resultant PDF of $y(t_f)$ is the superposition of every individual distribution components from each $y(t_i)$ and $y(t_f)$ with a normalization coefficient as

$$P(y(t_f)|\mathcal{Y}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{2\pi\hat{\sigma}_{i,f}^2}} \exp\left(-\frac{(y_f - \hat{\mu}_{i,f})^2}{2\hat{\sigma}_{i,f}^2}\right)\tag{3.27}$$

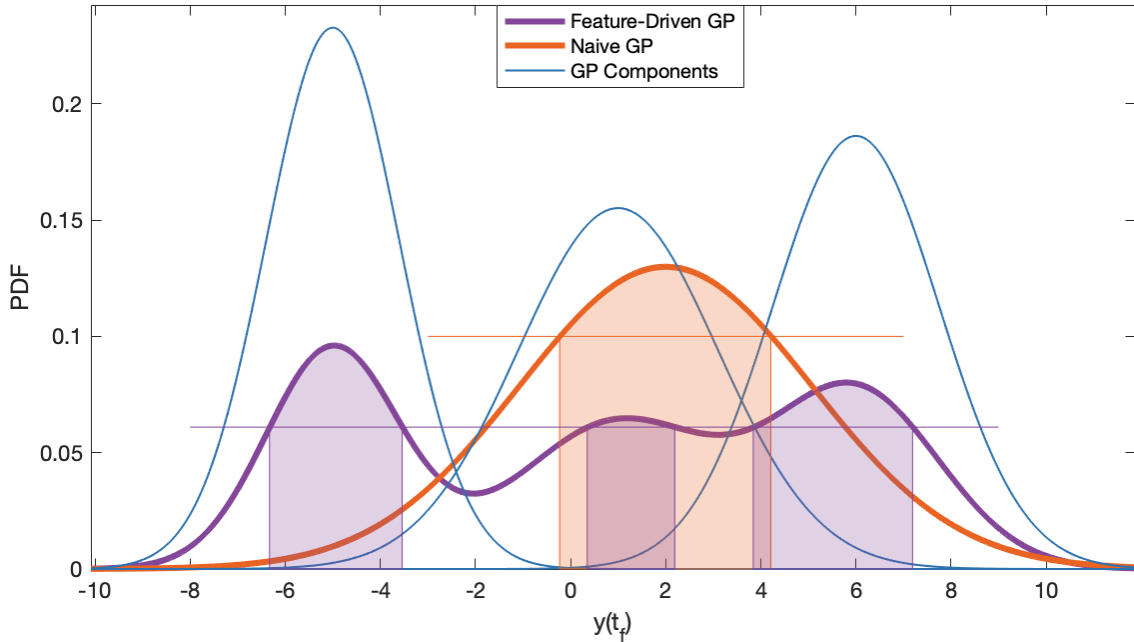


Figure 3.8: The purple and orange shadows have the same area representing the same probability.

An example is shown in Fig.3.8. Blue lines are distribution components, derived by the covariance matrix of three previous points with the future point. Naive GP gives the average prediction result of this future point, i.e. also a Gaussian distribution, under integrated impacts from all components. While in FE-GP, the GMM prediction result of this future point is assumed to have an equal probability to follow one of these three distribution components. The purple line gives the overall PDF of FE-GP.

3.3.3 Experimental Results and Discussion

3.3.3.1 Data Source

The data I use for training comes from base stations (BSs) in a 4G metropolitan area. The anonymous data is given by my industrial collaborator. It consists of aggregated downlink (DL) and uplink (UL) traffic demand volume per 15 minute interval over several weeks. I have selected a few example BSs at random to demonstrate my forecasting algorithm's performance.

3.3.3.2 Explainable Feature Matrix for FE-GP

When applying FE-GP to wireless DL traffic forecasting, the first to be considered is what does the feature matrix consist of. In my experiment, the features are set to be:

$$\begin{aligned}
 \lambda_l^1 &= y(t_{l-1}); \quad \lambda_l^2 = y(t_{l-2}); \\
 \lambda_l^3 &= y(t_{l-3}); \quad \lambda_l^4 = y(t_{l-4}); \\
 \lambda_l^5 &= y(t_{l-2}) - y(t_{l-5}); \quad \lambda_l^6 = y(t_{l-3}) + y(t_{l-4}); \\
 \lambda_l^7 &= \frac{y(t_{l-2}) - y(t_{l-1})}{y(t_{l-3}) - y(t_{l-2})}; \quad \lambda_l^8 = \frac{y(t_{l-2}) - y(t_{l-3})}{y(t_{l-3}) - y(t_{l-4})}; \\
 \lambda_l^9 &= std.[y(t_{l-1}), y(t_{l-2}), y(t_{l-3}), y(t_{l-4}), y(t_{l-5})];
 \end{aligned} \tag{3.28}$$

where $std.(y)$ is the standard deviation of elements of y .

3.3.3.3 Performance Metrics

I use the absolute cumulative error (ACE) as the performance metric:

$$\text{ACE} = \sum_{n=i}^j |\hat{y}(n) - y(n)|, \quad (3.29)$$

where $\hat{y}$ is the predicted DL traffic and $y(n)$ is the real data. For a fixed value forecast (one-step-ahead forecasting of the DL traffic), I assign $\hat{y}$ to be the value that has the maximum posterior probability.

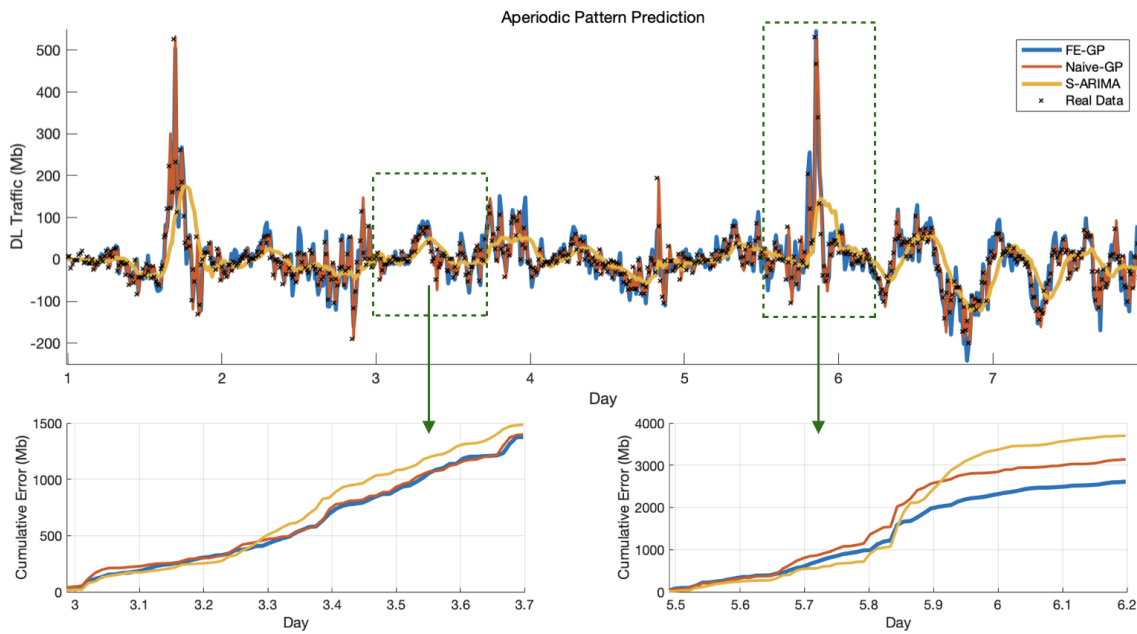


Figure 3.9: A comparison of aforementioned forecasting algorithms over a week (672 points). The cumulative error for 2 representative parts: (left) Average demand shows similar performance between FE-GP and Naive-GP; (right) Extreme spike demand shows superior performance by FE-GP against both Naive-GP and S-ARIMA.

3.3.3.4 Results Analysis

Fig.3.9 shows a comparison of forecasting algorithms over a week (672 points): (1) proposed FE-GP, (2) classical Naive-GP, (3) Seasonal ARIMA, against real 4G DL data. The cumulative error is shown for 2 different representative parts of the data: (left) average demand shows similar performance between FE-GP and Naive-GP; and (right) ex-

treme spike demand shows superior performance by FE-GP against both Naive-GP and S-ARIMA. From the GP models perspective, in the average part (left), both FE-GP and Naive-GP consider most of the traffic demands as noise flow, i.e. $\varepsilon(t)$ in the initial model, thus they perform similarly; In the extreme spike part (right), FE-GP can correctly recognize a potential event-driven flow, which has happened before, using features from the last few points, yet Naive-GP cannot, hence FE-GP gives a better prediction. In Fig. 3.11, I demonstrate that the proposed FE-GP perform the best overall due to its adaptive to the extreme values, even though it might be a little worse on average demand in few specific time stamps.

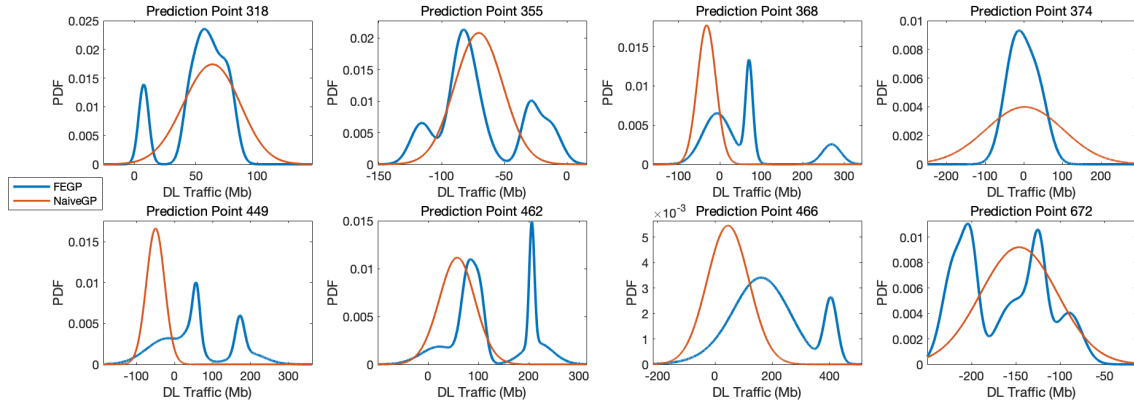


Figure 3.10: Posterior distribution of forecast produced by naive-GP (single peak) and FE-GP (multiple peaks) quantifies uncertainty.

3.3.3.5 Uncertainty Quantification

Posterior distribution of both models at a few representative points are given in Fig.3.10. Different from single peak Gaussian distribution predicted by Naive-GP, the GMM in FE-GP gives more general distributions for prediction. In the absence of a known periodicity, Naive-GP sums the effect of the last few time points, while FE-GP consider the effect from all time points according to their similarity in features with the predicted point. Consequently, there may be several peaks scattered over the forecast, which will inform proactive optimisation modules.

In data-driven wireless resource proactive optimization system [24], I ought to focus on not only the benefits brought by the system decision, but also the potential risks that drive regret functions, i.e., the occurrence of extreme demands. In my proposed FE-GP prediction model, the risks can be quantified from posterior distribution. For example, in Fig.3.10:

(1) **Low-traffic triggers proactive sleep mode and coverage compensation:** My FE-GP prediction points 318 and 672 demonstrates clear non-negligible probability of low traffic whilst the mean prediction is similar to that of the naive-GP. That is to say, I may need to proactive sleep selected cells to achieve more energy efficient operations, whilst using other neighbouring cells across RATs to compensate [44]. The risk of doing so is quantified by the posterior distribution (e.g., there is a small risk that there is actually high demand and compensated coverage is not enough).

(1) **Spike-traffic triggers proactive spectrum aggregation and offloading:** prediction point 368, there is a non-negligible high probability density area appearing at extreme value, which is far away from the predicted mean value. This can be used to inform proactive spectrum aggregation and off-loading of non-vital traffic to delay-tolerant RATs. The risk of doing so is quantified by the posterior distribution (e.g., there is a small risk that there is actually no demand for high capacity).

3.3.3.6 Training Process

As the training set increases over time, the FE-GP model becomes more sensitive to spikes due to its adaptivity to features. Nevertheless, the cost of computing goes up with the size of training set as well, thus I have to set a size threshold to the training set. In Naive-GP, I can discard data in reverse chronological order without affecting the performance of the model. However, in FE-GP, a trade-off must be made between the sensitivity of spikes and overall prediction accuracy, i.e., keeping more extreme value time points means the model is more sensitive to spikes prediction but may reduced overall performance. This need to be done case by case with each pre-existing resource proactive optimization system.

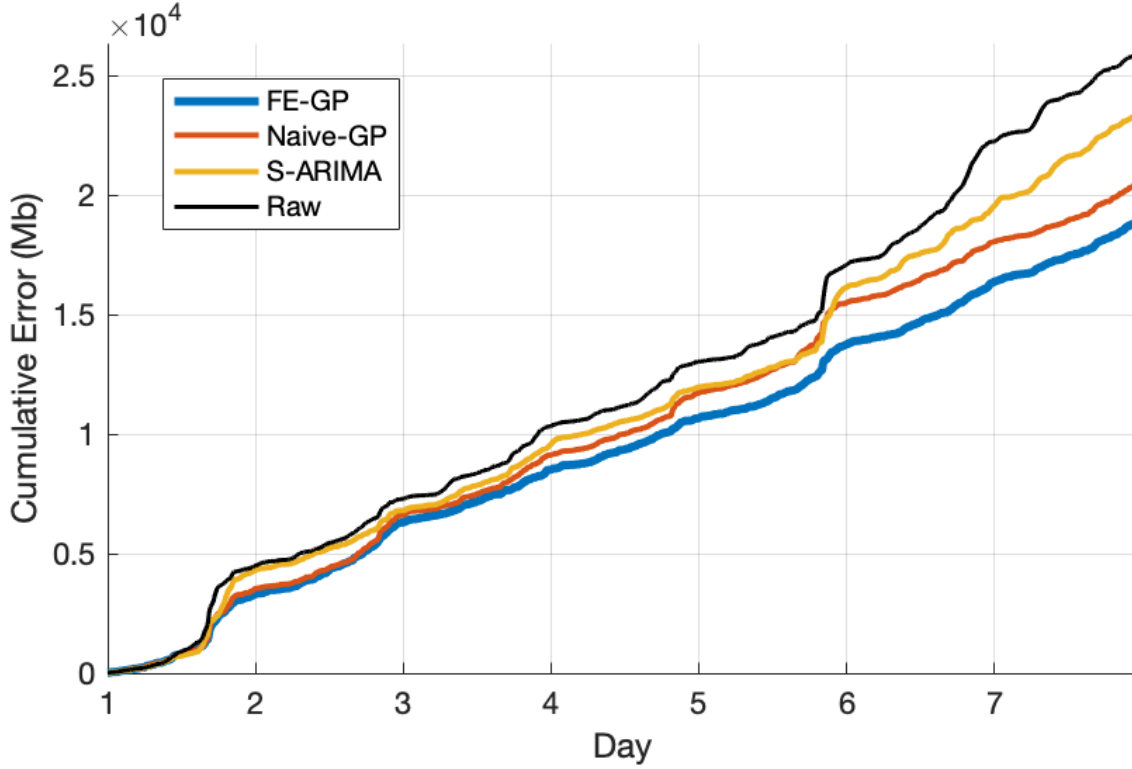


Figure 3.11: Cumulative error comparison between forecasting algorithms.

3.3.4 Conclusion and Future Work

Forecasting extreme demand spikes and troughs is essential to avoiding outages and improving energy efficiency. Whilst significant research into traffic forecasting using ARIMA, GPs, and ANNs have been conducted, current methods predominantly focus on overall performance and/or do not offer probabilistic uncertainty quantification. Here, I designed a feature embedding (FE) kernel for a GP model to forecast traffic demand. The FE kernel enabled us to trade-off overall forecast accuracy against peak-trough accuracy. I compared its performance against both conventional GPs, ARIMA models, as well as demonstrate the uncertainty quantification output. The advantage over neural network (e.g. CNN, LSTM) models is that the probabilistic forecast uncertainty can directly feed into decision processes in self-organizing-network (SON) modules in the form of both predicted average KPI benefit and regret functions using methods such as probabilistic numerics. My future work will focus on expanding to spatial-temporal dimension

via Gaussian random fields integration, consider multi-variate forecasting across different service slices, as well as employing Bayesian training in Deep Gaussian Process (DGP) models [75] to avoid catastrophic forgetting and to combat the dynamiticity of the traffic process.

3.4 Scarce Data Driven Deep Learning of Drones via Generalized Data Distribution Space

3.4.0 Abstract

Increased drone proliferation in civilian and professional settings has created new threat vectors for airports and national infrastructures. The economic damage for a single major airport from drone incursions is estimated to be millions per day. Due to the lack of diverse drone training data, accurate training of deep learning detection algorithms under scarce data is an open challenge. Existing methods largely rely on collecting diverse and comprehensive experimental drone footage data, artificially induced data augmentation, transfer and meta-learning, as well as physics-informed learning. However, these methods cannot guarantee capturing diverse drone designs and fully understanding the deep feature space of drones. Here, this chapter show how understanding the general distribution of the drone data via a Generative Adversarial Network (GAN) and explaining the missing features using Topological Data Analysis (TDA) - can allow us to acquire missing data to achieve rapid and more accurate learning.

This chapter demonstrate my results on a drone image dataset, which contains both real drone images as well as simulated images from computer-aided design. When compared to random data collection (usual practice - discriminator accuracy of 94.67% after 200 epochs), my proposed GAN-TDA informed data collection method offers a significant 4% improvement (99.42% after 200 epochs). I believe that this approach of exploiting general data distribution knowledge from neural networks can be applied to a wide range of scarce data open challenges.

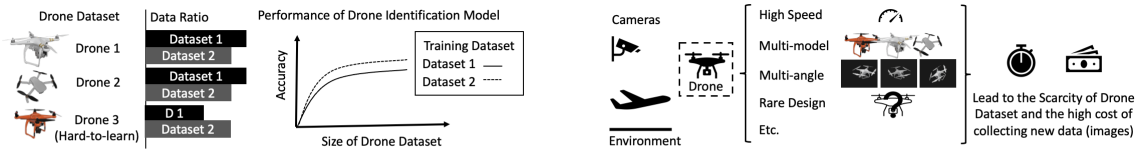
3.4.1 Introduction

Increased proliferation of drones and autonomous air vehicles can disrupt critical national services (e.g., Gatwick Airport 2018). The economic damage for air transport is estimated

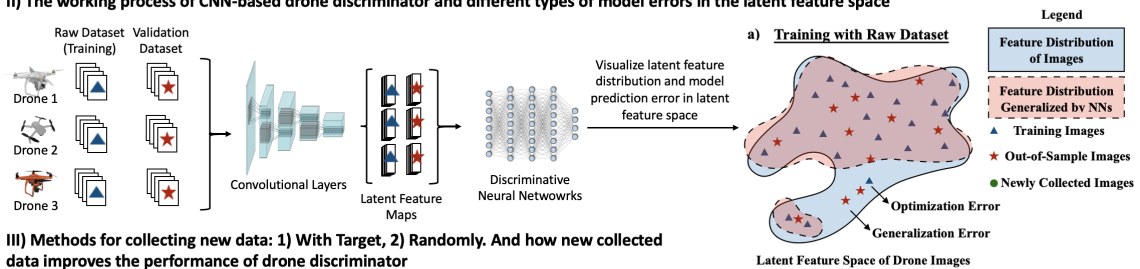
to be millions per day for airports and airlines. Furthermore, the drone industry makes up 1.9% of UK GDP and supports over 600,000 jobs in the drone's economy. Whilst many are amateur drones that pose no malicious intention, some may carry deadly capability and cause severe economic damage to critical infrastructure. Protection against drones is critical to ensuring smooth operation of services, whilst safeguarding it against the most severe threats. High resolution cameras can classify drones using deep learning, but accurate identification is critical for not disrupting normal day-to-day operations and maintaining an efficient economy [138]. Whilst the upper limit of accuracy for image classification has been increased by more complex deep learning (DL) architectures, the upper accuracy of DL model also limited by its logarithmic growth to the size of the training dataset (see - Fig.3.12.I) [133]. Simultaneously, lack of hard-to-learn (by DL model) drone data will affect the overall drone detection accuracy. It is a key open challenge to achieve extremely high accuracy by sourcing sufficient, relevant but rare training data sets (e.g., rare drone design) [178]. This often means a large amount of resources and time is dedicated to broad data collection and (re)training the neural network. Especially, high speed and tiny-size of drone challenge the image capture, at the same time, multi-model and numbers of shooting angle increase the data scarcity of drone dataset - see Fig.3.12.I. Therefore, the high cost of collecting new drone images adds to the cost of improving the accuracy of DL-based drone classification. However, a target of data collection can reduce the amount of new data to be collected compared to collecting data on all drones, saving overall DL performance improvement costs.

In this chapter, I show how understanding the general distribution of the drone data via a Generative Adversarial Network (GAN) and explaining the missing features using Topological Data Analysis (TDA) - can allow us to acquire missing data to achieve rapid and more accurate learning. The aim is to demonstrate how the improved accuracy can benefit air transport protection.

I) DL drone protection: The relationship between training data size and DL model accuracy, and the reason for the scarcity of drone data



II) The working process of CNN-based drone discriminator and different types of model errors in the latent feature space



III) Methods for collecting new data: 1) With Target, 2) Randomly. And how new collected data improves the performance of drone discriminator

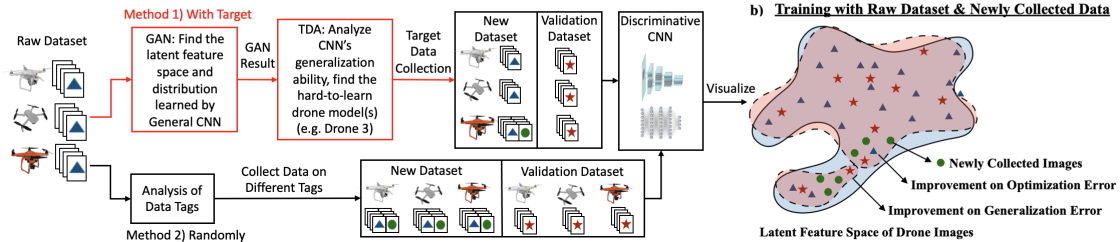


Figure 3.12: (I) The upper limit of DL drone classification accuracy is affected by the training data size, but various factors of drones lead to scarcity of drone data and high acquisition costs. (II) the nature of discriminative NNs' inference is to map an high-dimensional input features into a label class using the feature distribution (FD) generalized from the training data (convolutional layers extract latent features maps of input images for further discriminative work in NN). While lack of training data will result in the generalization error on out-of-sample data. (III) my aim is to detect the data that important for model training (e.g. hard-to-learn drone), reduce the amount of new data required for model improvement, and outperform randomly data collection method in both learning speed and predication accuracy of trained model.

3.4.1.1 Related work

One of the universal challenges in deep neural network training is when there is a lack of data, the out-of-sample performance can not be guaranteed [150]. As shown in Fig.3.12.II, the nature of discriminative NNs' inference is to map an high-dimensional input features into a label class using the feature distribution (FD) generalized from the training data [72] (convolutional layers extract latent features maps of input images for further discriminative work in NN). While lack of training data will result in the generalization error on out-of-sample data [60,83,170], data scarcity can also lead to the optimization error on known training data (see - Fig.3.12.a). However, collect extra training data could address these error as shown in Fig.3.12.b. In general discriminative NNs, the performance increases logarithmically based on volume of training data size [133], which means the marginal cost of training data for model performance improvement boosts exponentially. Due to the diminishing returns, exhaustive or randomly searching for data is not applicable to the scenarios where data collection is expensive (e.g., aerospace, military). Therefore, there is a need to create a method to identify which specific new data would be important for discriminative NNs' performance improvement based on an existing dataset, as the guidance to the new data collection work, so that to reduce data collection cost. This is the motivation for this chapter. As shown in Fig.3.12.III, my aim is to detect the data that important for model training (e.g. hard-to-learn drone), reduce the amount of new data required for model improvement, and outperform randomly data collection method in both learning speed and predication accuracy of trained model. There are related papers addressing the aforementioned challenges in the discriminative NNs training with limited data. I summarize several research achievements below and compare them in table 3.3.

Data augmentation improve the training set by adding slightly modified (e.g., translations, rotations and flips) copies of existing data to strengthen the invariance of NNs to the aforementioned modified data [122]. However, data augmentation is only based on raw training data, hence cannot offer additional generalization on the variation of the object itself other than its position to NNs. Transfer learning reduce the training cost of

new DL model by reusing the convolutional kernels from other related well-trained DL models [154]. By doing so, NNs can partially transfer the generalization ability got in the former relevant training into the latter target inference. Similarly, meta-learning tries to abstract more universal generalization ability from multiple training domains and “learn to learn” fast [54], which lays the foundation for the few-shot learning [105]. However, the performance gain via these two methods are not guaranteed, since the transferred generalization ability is context-agnostic (i.e., does not focus on the properties of the new data), which may not match the need to specific requirements. Physical informed learning is designed to embed given laws of physics (e.g., general nonlinear partial differential equations) into NNs to inform its generalization [85, 102]. Hence, there is less need for the diversity of the training data and more training data can be generated numerically from the given laws of physics. However, in most discriminative NNs applications, physical law is unknown.

	Data Augmentation [122]	Transfer Learning [154]	Meta Learning [54]	Physical Informed Learning [102]	Proposed GAN-TDA Method
Methodology	Add modified data into dataset	Reuse Layers from other well-trained DL models	Reuse layers trained in previous tasks	Apply physical properties in model training	Purposefully collect new data
Generalization Ability	Modified dataset	Other related data	Other related data	Data physical properties	Newly collected data
Advantage	No additional data required	Wide applicability; fast deploy speed	Learn to learn; learn fast for new tasks	Controllable generalization	Explainability; less data to be collected;
Disadvantage	Limited improvement; lack of explainability	Context-agnostic, lack of explainability	Context-agnostic, lack of explainability	Difficulties in processing physical properties	Need additional cost for collecting targeted new data

Table 3.3: A comparison of methods for training deep learning models with scarce data

Although data augmentation enhances the training set by adding augmented data

based on observed data (auxiliary variables method), there is no effect on un-observed data (e.g. Drone images by un-observed angle). However, newly collected training set could address this problem. Transfer learning and meta-learning can save the training time for new tasks, but can not enhance a trained model and are lack of explainability. Physical informed learning needs expert experience which is abstract and uncontrollable. Thus, I propose a method to reveal the relationship between NN's generalization ability and the composition of training data, so that to provide a feature-based target for new data collection to save the cost of collecting new data.

3.4.1.2 Innovation: GAN-TDA framework

In conventional work, methods focus on the generalization ability of the model itself, which are general methods with versatility for various applications. By contrast, the generalization ability brought by the training data attracts less attention. In this paper, I aim to identify the required data for discriminative NNs' generalization error reduction, by analyzing the existing training data through its potential feature distribution (FD) and that generalized by NNs (see - Fig.3.12.b).

Generative adversarial network Generative models are designed to generate data with the same FD as that learned by NNs. In principle, most common generative models, including variational auto-encoder (VAE) and variations of generative adversarial network (GAN), are trained to convert the initial distribution of latent variables into that learned by NNs with training data [7, 35, 61, 65, 141]. For scarce data problems such as my drone detection application, one might be interested in using GAN with the following reasons: (1) GAN is proven to be asymptotically consistent in FD approximation while VAE may have bias due to the variational lower bound, thus the generated data from GAN would be more precise in representing the FD learned by the NNs [34]; (2) With the same training data, the discriminator in GAN gives similar but more smooth convolution kernels as that in CNN [146]. More specifically, kernels in GAN have the same generalization

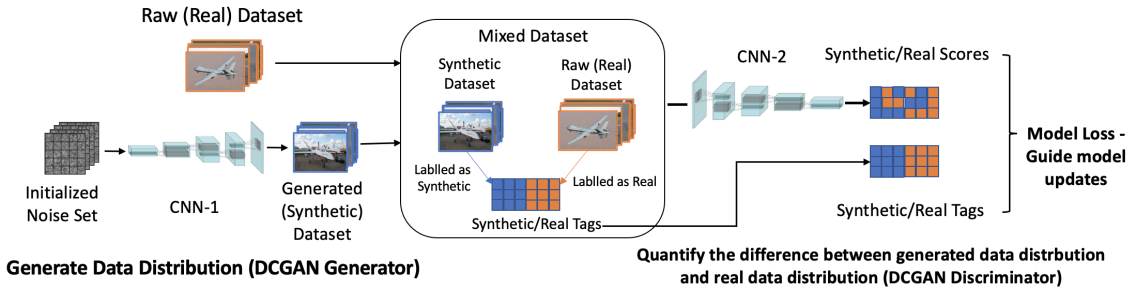
way but weaker ability as CNN. Accordingly, the analysis in GAN can be considered as the representation of general CNN. (3) The discriminator in GAN can be considered as the pre-trained target discriminative NN for methodology validation so as to eliminate generalization ability bias caused by another arbitrary NNs [81].

In my experiment, color information in each pixel of drone images are use as inputs, the viewing of raw feature space built by pixel information will be over-dimensioned and lacks practical interpretability. Thus, to build a more informative feature space for image data, latent feature learned and processed by convolutional layers in deep learning models helps. Here, each latent feature is represented as the degree of response to a certain kernel feature in different receptive fields of the image. However, GAN does not give explicit expressions of the distribution, hence Monte Carlo synthetic data from the generator would be taken for the further FD analysis in the next step.

Topological data analysis For high-dimensional data analysis, dimension reduction approaches, such as PCA, MDS, t-SNE and etc., are commonly applied [25, 30]. However, due to existing of generalization in NNs, the generated data may have more complicated topology than raw dataset in the high-dimensional feature space, while conventional methods fail to capture any structure from the data, which cause catastrophic lose in high-dimensional distance information for my analysis after dimension reduction. In my framework, topological data analysis (TDA) mapper is proposed to address this issue. With the key idea of multidimensional persistence, TDA can capture data structure and then preserve the high-dimensional distance information with simplicial complex [10, 87, 143].

In my experiment, TDA is applied to tell the difference between the potential FD of data from raw dataset and that of synthetic data from the generator in GAN. With the high-dimensional clustering in TDA, discrete nodes are used to represent the original continuous feature distribution, while the connection between nodes indicates the distance in feature space. Before I do TDA, I mix these two dataset into one with data labels attached (i.e., real, synthetic), so that to maintain the consistency of the topological space

Step 1) Find High-dimensional Distribution of Raw Data Through the Distribution of DCGAN Generated Data



Step 2) Visualize High-dimensional Data Distribution by Dimensionality Reduction Algorithm

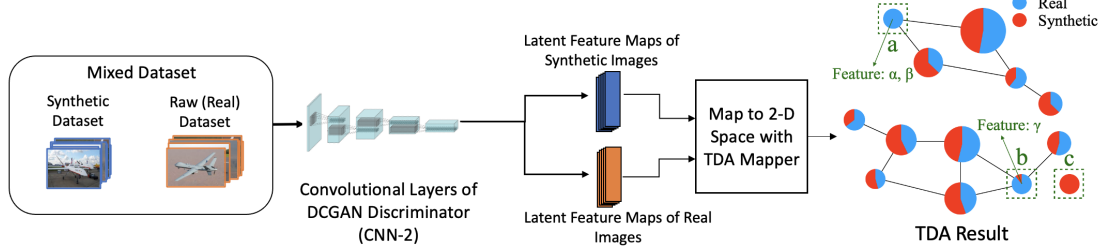


Figure 3.13: Demonstration of Method: Step 1) Find high-dimensional (latent features) distribution of raw dataset by DCGAN which indirectly expressed by the distribution of DCGAN generated synthetic data; Step 2) Use the outputs from the convolutional layers in DCGAN discriminator on both raw dataset and synthetic dataset to represent their latent feature maps respectively, and use TDA on both the latent feature maps to visualize the difference in their distributions. In Step 2): The TDA Mapper generates a network representation of the feature space, in which each node corresponds to a set of data with similar features. The size of the node indicates the quantity of the data assigned in it while the color is used to distinguish whether the data is real or synthetic in the node. In ideal conditions, the synthetic data would have roughly equivalent proportion in each node. Accordingly, the categories of nodes are listed as follows: (a) No synthetic data are generated in this node; (b) both synthetic data and real data are placed in this node; (c) Synthetic data are generated anomalously outside of the real data feature distribution. (a) indicates the GAN (representation of DNN) is failing to generalize the data with the feature α, β , (c) indicates that the GAN is still incomplete convergence, hence I will train the GAN further while (a) or (c) nodes occur. Part of (b) nodes containing few generated synthetic data with feature γ , thus I propose to emphasize my data collection procedure on data with γ .

in TDA. Then, by analyzing the proportion of real/synthetic data in each node, one can discover the weak nodes, which lack the synthetic data, and then identify the required data by the real data tags in these nodes to guide the new data collection.

It is worth noting that, in my proposed methodology, the description for required training data cannot exceed human knowledge about the data. Although GAN-TDA approach works on the feature space in NNs, the result is still interpreted using the human feature space (tags), which may not match the need of neurons. What I can do here is to tag data with my best knowledge, so that to make the data collection guidance more targeted.

Contribution and novelty In this paper, I propose GAN-TDA to identify which specific new data would be important for discriminative NNs' performance improvement based on an existing drone dataset, as the guidance to the new data collection work. To my best knowledge, this chapter is the first to reveal the relationship between NN's generalization ability and the composition of training data, so that to improve the model performance via newly collected data.

I make three major contributions:

(i) GAN-TDA framework is proposed to guide the new data collection. Specifically, I use GAN to capture the feature distribution in inference generalized by discriminative NNs from the training data, and TDA to identify the generalization weakness on the training data.

(ii) A drone image dataset using both real drone images as well as simulated images in CAD is established for my experiment. Each image is tagged with the drone's features (e.g., model, color, frame shape, camera position...) as many as I can.

(iii) I demonstrate my results on a drone image dataset, which contains both real drone images as well as simulated images from computer-aided design. When compared to random data collection (usual practice - discriminator accuracy of 94.67% after 200 epochs), my proposed GAN-TDA informed data collection method offers an incremental 4% improvement (99.42% after 200 epochs).

The remainder of this chapter is organised as follows. In Section 2, I demonstrate the working flow of my proposed GAN-TDA framework. In Section 3, I apply my model on drone picture data for evaluation and validation. Section 4 concludes this chapter and proposes the ideas for future work.

3.4.2 Method

Given a dataset with clear properties labelled in tags that are detailed enough to guide the direction of new data collectivity (e.g., images in a vehicle dataset labelled with the vehicle's maker, model, type, color, number of wheel, etc.), my methodology is to identify which kind of data the discriminative model has weak generalization on by viewing the data distribution in the dataset and distribution learned by deep learning (DL) models.

3.4.2.1 Step 1) - Learn data distribution by GAN

The first step is to find the high-dimensional distribution of the raw dataset (drone images). According to the demonstration of *Step 1*) in Fig. 3.13, two networks named Generator G and Discriminator D with different network structures will be established and initialized with individual aim to generate synthetic images and to recognise the input image is real or synthetic. Before the training of GAN, all the image from the raw dataset will firstly be pre-processed into the same size (e.g., 64*64 pixels) and secondly be normalized into the same scale. These steps are to ensure each input parameter (pixel intensities in each color channel) has a similar data distribution to guarantee the convergence of DL models [57].

Processed images from the raw dataset (real image dataset) will be divided into batches $B = \{B_1, B_2 \dots B_i\}$ with a fixed size (e.g., 64 pictures per batch). During the training, a batch of initialized noise set n that follows a certain distribution (e.g., Gaussian Noise) will be generated whose batch size is the same as data batch size (e.g., 64*[100 samples from Gaussian Noise]). The initialized noise set will be re-produced at the beginning of each training iteration and then be processed into a set of synthetic images $G(n)$ by fractionally-strided convolutions [101] in G . The optimization of G is expressed by mini-

mizing the generative loss, which could be quantified by the discriminative result from D . And the optimization of D is to minimize the discriminative loss on both synthetic images and raw images.

During the quantification of model loss, raw images B_i will be labeled as *True* and generated synthetic data $G(n)$ will be labeled as *False*, these information are stored into the real and synthetic tags matrix $T_{tag}(G(n), B_i)$ (mixed dataset). D will scoring the real and synthetic rate on both B_i and $G(n)$, the generated scores are stored in the real or synthetic score matrix as $T_{prediction}(G(n), B_i) = [D(G(n)), D(B_i)]$. The generative and discriminative loss could be further expressed by the divergence between $T_{tag}(G(n))$ and $T_{prediction}(G(n))$ and that between $T_{tag}(G(n), B_i)$ and $T_{prediction}(G(n), B_i)$. The divergence quantification uses Wasserstein distance to guarantee the stability of model training and avoid the collapse mode issue [7].

The processes of training a DCGAN with Wasserstein distance is shown as Algo. 1. During the model training, optimizing model parameters by backpropagation in both G and D will let the distribution of generated synthetic data $G(n)$ gradually approaching that of the real dataset B . After the model converges, the distribution of the raw dataset is considered to have been learned and captured by GAN that the distributions of the GAN generated synthetic image set and the raw dataset is in a high similarity. Simultaneously, the ability to convert the given certain distribution noise set into real-enough synthetic data which follows the distribution of the raw dataset will be hiddenly stored with the form of model parameters in the GAN generator.

3.4.2.2 Step 2) - Latent feature maps extraction and TDA

In CNN, convolutional kernels are designed with weight sharing property, and each multidimensional kernel representing a unique hidden feature. So, in the convolution process, the output of each convolutional kernel means the degree of activation of an image property in a series of adjacent receptive fields controlled by stride.

The latent feature map of an image contains a set of activation degrees on different

Algorithm 1 Training of DCGAN with Wasserstein distance**Require:** Training set $B = \{B_1, B_2 \dots B_i\}$ **Require:** Generator G , Discriminator D , Gaussian Noise n , Wasserstein distance calculator W , Optimizer Opt **Require:** Number of total epochs j Initialize G, D **for** epoch $\leq j$ **do** **for** $B_i \in B$ **do** Initialize Gaussian Noise n $T_{tag}(G(n), B_i) =$
 $[False * G(n), True * B_i]$ $T_{tag}(G(n)) = [False * G(n)]$ $T_{prediction}(G(n), B_i) =$
 $[D(G(n)), D(B_i)]$ $T_{prediction}(G(n)) = [D(G(n))]$ $loss_D =$
 $W(T_{tag}(G(n), B_i), T_{prediction}(G(n), B_i))$ $loss_G =$
 $W(T_{tag}(G(n)), T_{prediction}(G(n)))$ $D \leftarrow Opt(loss_D, D)$ $G \leftarrow Opt(loss_G, G)$ **end for**

epoch = epoch + 1

end for **return** D, G

high-dimensional features and each value in the latent feature map can be seen as the activation intensity of different convolution kernels. It can also be regarded as the expression of high-dimensional image features to a low-dimensional space like the encoded latent features by generative models like VAE. The latent feature map can express the characteristics of the image to a certain extent, and be used to reduce the image dimension to facilitate the analysis of the image.

Certain data distribution will be expressed differently by latent feature spaces formed from different convolutional layers. To appropriately express the feature space for the raw dataset distribution, a proper multi-dimensional scale needs to be confirmed with a set of latent features learned by convolutional kernels. According to the fact that front convolutional layers tend to learn lower-dimensional visual features (e.g., line, Polyline, arc), later convolutional layers trying to extract high-dimensional latent features which contain complex information about positioning and relationship based on features extracted by

the front layers. The feature space formed by kernels from the last convolutional layer is chosen to keep the deeply high-dimensional global information used for feedforward layers inferences. And this space could clearly express the distance between distributions of the synthetic dataset and raw dataset.

As shown in Algo. 2, the latent feature map of a given image is defined as the output of GAN-discriminator's last convolutional layer after feeding the image into D (Step 2 of Fig.3.13). Suppose the convolutional layers in D is defined as a layer set $C = \{c_1, c_2 \dots c_k\}$ and the generated synthetic image from G as $S = \{G(n_1), G(n_2) \dots G(n_i)\}$ (i is the batch number in B : to guarantee the generated synthetic images have the same amount of data as raw dataset). The tags of all synthetic image is set to *False* as $T_{tags}(S) = [S * False]$, and *True* for all images from real dataset B as $T_{tags}(B) = [B * True]$. As shown in Algo. 2, the algorithm will return a dataset that contains both the latent feature maps for synthetic images L_S and real images L_B with their tags $T_{tags}(S)$ and $T_{tags}(B)$ for further TDA use.

Algorithm 2 Extraction of latent feature maps

Require: Real image dataset $B = \{B_1, B_2 \dots B_i\}$, Synthetic image dataset $S = \{G(n_1), G(n_2) \dots G(n_i)\}$

Require: Convolutional layers of DCGAN's discriminator $C = \{c_1, c_2 \dots c_k\}$

Require: Tags for synthetic images $T_{tags}(S)$, and for real images $T_{tags}(B)$

Define $L_S, L_B = \{\}, \{\}$

for $i \leq \text{len}(B)$ **do**

$l_B = B_i$

$l_S = G(n_i)$

for $j \leq \text{len}(C)$ **do**

$l_B = c_j(l_B)$

$l_S = c_j(l_S)$

$j = j + 1$

end for

$L_S.append(l_S)$

$L_B.append(l_B)$

$i = i + 1$

end for

return $\{L_S, L_B, T_{tags}(S), T_{tags}(B)\}$

The viewing of data distribution in the chosen feature space is barricaded by its high dimensionality, where TDA makes reasonable dimensionality reduction representation to help the discovery of distance between two high-dimensional data distributions. I use

Kepler Mapper for TDA visualization [143].

3.4.2.3 TDA interpretation

In perfect training, the percentage of synthetic data in each node should be close, which means discriminative NNs give roughly equivalent generalization to every models. In practical, there always has bias on the generalization to different models, which can be observed from uneven percentage in each node. Low percentage indicates the lack of generalization and vice versa.

3.4.2.4 Experimental setup

Recent deep learning drone detection models make inferences mainly based on images and video information captured by surrounding cameras. These captured visual data will be processed to complex high-dimensional latent features by convolutional layers based on image properties of different receptive fields for further inference steps completed by feed-forward layers. Therefore, various appearance and shape factors of drones designed according to different working environments and purposes challenge CNN based drone identification models in recognition precision, generalization and robustness a lot.

To investigate which design property of drone is hard for general CNN-based classifiers to learn and discriminate, an experiment is established to apply the GAN-TDA method into a collected drone image dataset (raw image dataset). This experiment aims to prove that the drone discriminator trained using additional images collected with the GAN-TDA method performs better than the model using new images collected with the randomly method. During the experiment, the GAN-TDA model will analyze the raw dataset and generate the guidance on which kind of data should be collected additionally. Two models with the same net settings will be trained using different newly collected datasets (see *Method 1*) and *Method 2*) in Fig.3.12.III) in a control experiment to evaluate the feasibility of GAN-TDA guidance. The datasets for two groups are:

Group GAN-TDA - data from the raw dataset and additional data collected under

guidance from GAN-TDA (additional data for several drone models);

Group Random - data from the raw dataset and additional data collected averagely for all data categories (additional data for all drone models).

During the validation, two models trained by Group GAN-TDA dataset and Group Random dataset respectively will be tested on the same validation dataset which contains images for all drone models distinct from the images in training sets (Group GAN-TDA and Group Random dataset). According to the performance comparison between the model trained with Group GAN-TDA dataset and with Group Random dataset, the superiority of GAN-TDA guided data in improving general target CNN-based model generalization could be viewed. Details are listed as follows.

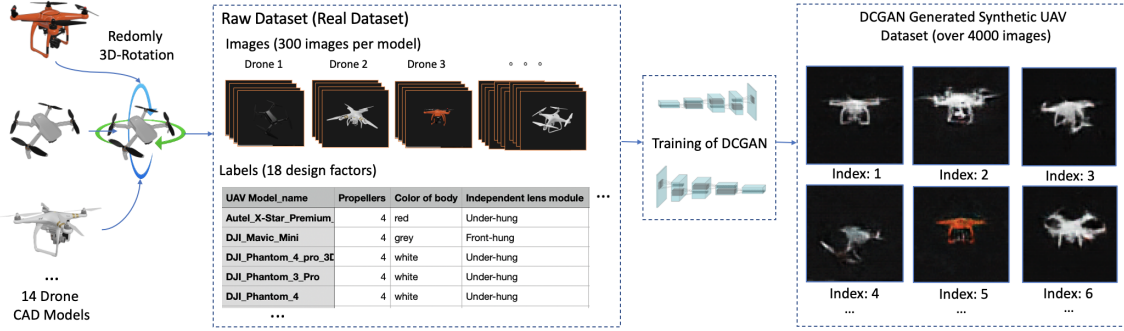


Figure 3.14: The Generating of the Raw Dataset and Synthetic Dataset

3.4.2.5 The dataset and hardware

The raw dataset contains over 4000 pictures averagely collected from 14 popular commercial drones' 3D models³ (e.g., DJI Phantom 3, Phantom 4 and Phantom 4 pro). As shown in Fig.3.14, to simulate the real pictures of flying drones caught by monitors and cameras with different angles, I randomly rotate these 3D models on x, y, z axis and take screenshots with drone centre-placed and 1800*1500 pixels resolution. During the data collection, the background of each drone model is set into black without ambient light effect to remove the high-frequency information from the background. These images are

³The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

store in different folders named by their drone model's name, with an additional label file that contains a unique image ID for each collected drone image with 18 different hardware and appearance characteristics of these drones (e.g., shape of propellers, number of propellers, position of floor stand).

The experiment environment is split into two part: the training of DCGAN is transferred into Cloud served by Google Cloud Platform with a VM (Ubuntu 16.04) established and 1 Tesla V100 GPU (NVIDIA-SMI 450.102.04, CUDA 11.0, 16G Memory) embedded; the evaluation experiment is processed by 8-Core Intel Core i9 (16G Memory).

3.4.2.6 DCGAN settings

To accelerate the training speed of GAN and avoid the missing of meaningful details in the appearance, each image in the real dataset is resized into a resolution of $3*64*64$ (R, G, B channels, $64*64$ pixels) and processed with pixel value normalization in each channel (mean = 0.5, standard deviation = 0.5). As shown in Table 3.4, the DCGAN-generator is designed with five fractionally-strided convolution layers to generate synthetic images with the same resolution as the pre-processed real image ($3*64*64$). The discriminator is designed with five convolutional layers which accept $3*64*64$ images as input, and the activation function of the last layer (Sigmoid in original DCGAN [101]) is removed to meet the requirements of use Wasserstein Loss [7]. During the training of DCGAN on the raw dataset, images from the dataset will be split into batches with 64 images each and shuffled before each training epoch. Gaussian noise is chosen to provide GAN-generator with original input, and in each batch training, a randomly initialized $64*100$ noise set will be processed in GAN-generator into 64 synthetic images. According to the indications from [18], discriminator should be trained before generator and with more epochs than generator. The training rate is set to five that generator will be trained once after five times training of discriminator [7] with the same training rate $3e^{-4}$ using Adam Optimizer. The generating quality of generator will be checked every 400 epochs training, and part of the synthetic images are sampled and shown in Fig.3.14 to demonstrate the generating quality

of generator after 5000 epochs training.

Table 3.4: DCGAN network training/validation settings

Generator			Discriminator		
Layer	Type	Size	Layer	Type	Size
Input	Gussian Noise	100	Input	Image	3*64*64
ConvTranspose 1	4*4 Fractionally-strided Convolutions	512	Convolution 1	4*4 Convolutions	32
-	Batch Normalization	512	-	Leaky ReLU	32*32*32
-	ReLU	512*4*4	-	-	-
ConvTranspose 2	4*4 Fractionally-strided Convolutions	256	Convolution 2	4*4 Convolutions	64
-	Batch Normalization	256	-	Batch Normalization	64
-	ReLU	256*8*8	-	Leaky ReLU	64*16*16
ConvTranspose 3	4*4 Fractionally-strided Convolutions	128	Convolution 3	4*4 Convolutions	128
-	Batch Normalization	128	-	Batch Normalization	128
-	ReLU	128*16*16	-	Leaky ReLU	128*8*8
ConvTranspose 4	4*4 Fractionally-strided Convolutions	64	Convolution 4	4*4 Convolutions	256
-	Batch Normalization	64	-	Batch Normalization	256
-	ReLU	64*32*32	-	Leaky ReLU	256*4*4
ConvTranspose 5	4*4 Fractionally-strided Convolutions	3	Convolution 5	4*4 Convolutions	1
Output	Tanh	3*64*64	Output	None Activation Function/Sigmoid (validation)	1

3.4.2.7 Validation settings

The evaluation of the method is to show the performances of models trained with different additional datasets on distinguishing the synthetic and real drone data.

To control the experiment variable (avoid influences brought by different initialization ways), the network in DCGAN-discriminator is chosen to be the identical network initialization whose convolutional layers are pre-trained on the raw dataset. During the training of DCGAN, the increasing of discriminative ability in discriminator is suppressed by the gradually increasing adversarial power from DCGAN-generator. Once the generator is fixed, the training of the discriminator will no longer be limited and be seen as the training of a general discriminative DNN model.

Based on the result from GAN-TDA (details are listed in the Section: TDA Result), I collect two additional datasets for Group GAN-TDA and Group Random respectively. The additional data for Group GAN-TDA is evenly collected from the 3 detected hard-to-learn drone models (DJI Phantom 3 Pro, DJI Phantom 4 and DJI Phantom 4 pro) and Group Random additional data is evenly collected from all drone models. In the practice

environment, data accessibility for different models varies. Therefore, the evenly collected method is used in this experiment to represent the general randomly collection of new data. To control variables, I collect 140 additional images per hard-to-learned drone model for Group GAN-TDA and 30 additional images among all 14 drone models for Group Random (the newly collected data accounts for 10% of the raw dataset). With the use of discriminator as the initialization, the network structures of both models trained by Group GAN-TDA dataset and Group Random dataset are the same which shown in Table 3.4, but Sigmoid activation function is added to the last feed-forward layer for the binary classification task.

The validation set is collected on all drone models evenly (50 images per drone model) and independent from the raw dataset and any additional dataset. Each image in the validation set will be pre-processed as the training sets before model inference. During training, model performance on the validation set will be supervised in each epoch.

3.4.3 Results

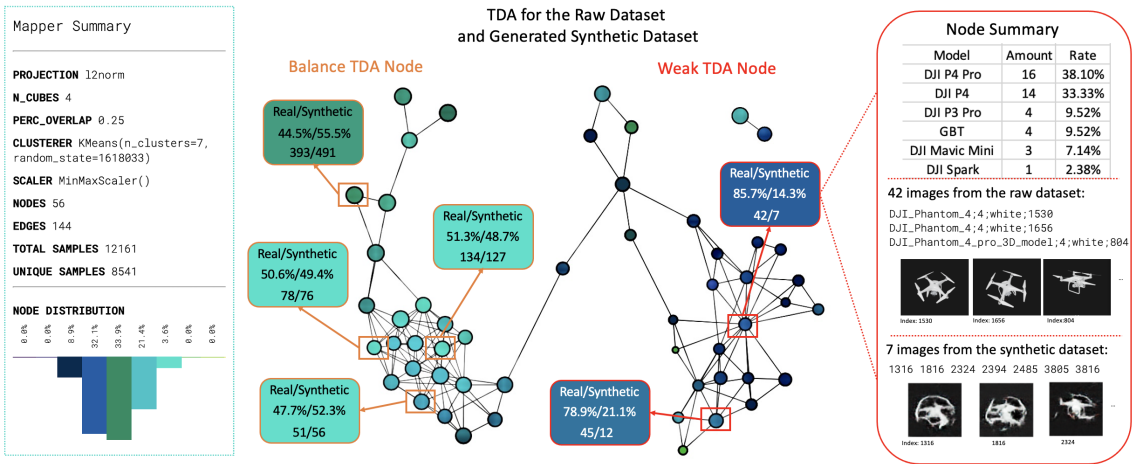


Figure 3.15: TDA output and analysis

3.4.3.1 TDA result

The outcome by TDA shows a topological analysis result to the distribution of both synthetic data and real data. By analyzing the TDA diagram shown in Fig.3.15, I found three drone models out of 14 are more difficult for deep learning models to learn.

According to the *mapper summary*, I can see none (a) or (c) type TDA node (clarified in Fig.3.13) occurs that the trained GAN do capture the data distribution of the raw dataset. There are two kinds of type (b) TDA nodes: Balance TDA node (the amount of real data and synthetic data is balance) and Weak TDA node (few synthetic data in this node compared with real data). The color of nodes reflects the internal balance of data that dark color refers to weak TDA nodes with low internal data balance. The rate of real data and synthetic data in some selected TDA nodes are listed in Fig.3.15 for demonstration.

According to the node distribution from the mapper summary, 8.9% TDA nodes are in dark blue color so I will focus more on these weak nodes. Tracing the origin data whose latent feature map is placed in weak TDA points (which drone model are these data from) could provide clear guidance for new data collecting. I also list the details of an example weak TDA node as shown in Fig.3.15 - *Node Summary*. From here, the proportion of real data from different drone models will be summarized (e.g., 16 out of 42 real data are from drone model DJI Phantom 4 Pro, and count 38.1% of real data in this node).

By analyzing the data in all weak TDA nodes (with dark blue color), I found that the majority of real data placed in these nodes come from 3 drone models - DJI Phantom 3 Pro (22.7%), DJI Phantom 4 (27.6%) and DJI Phantom 4 (33.5%). The result shows the DCGAN generalization ability on these drone models is weak. The TDA result further forms the GAN-TDA guidance that new data collection should focus on these models.

3.4.3.2 Discriminator result

The discriminator result shows that, on the designed validation dataset, the drone detection ability of the model trained with GAN-TDA guided additional data is better than the model trained by randomly collected additional data.

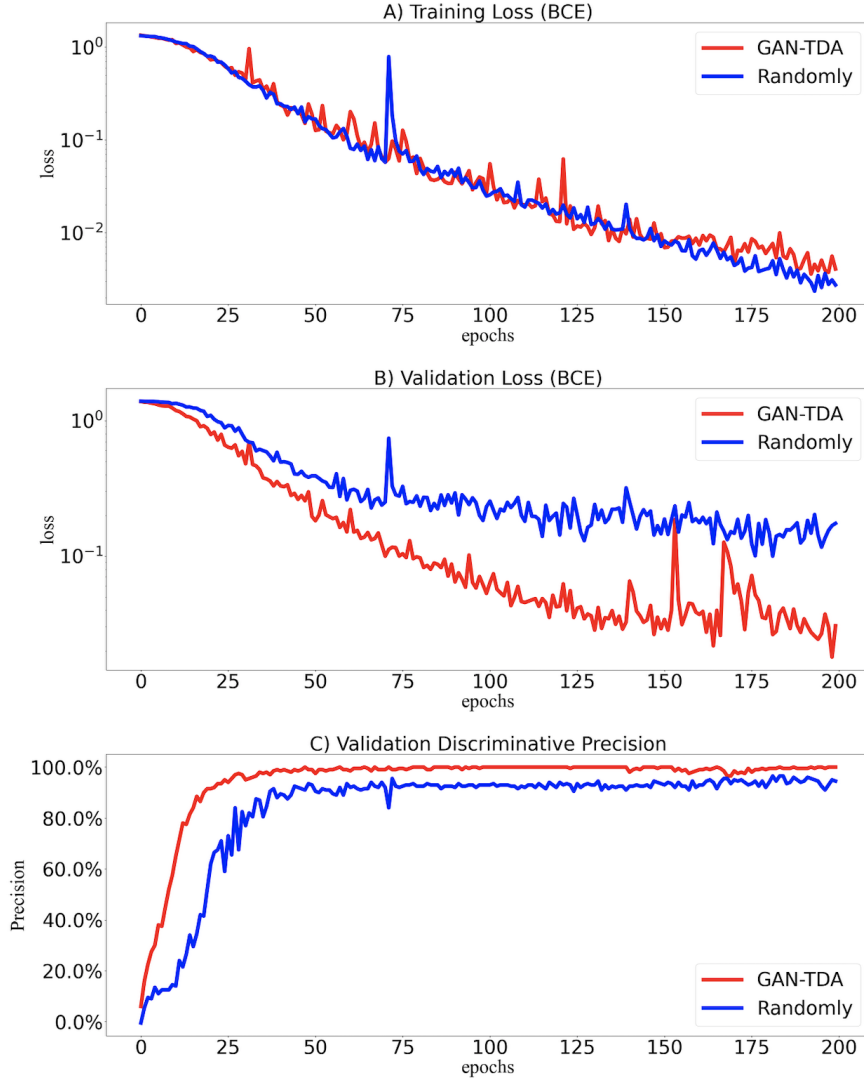


Figure 3.16: The performance comparison of targeted data collection method and random data collection method

The demonstration of the discriminator result is made in three parts: *A) Training loss in binary cross-entropy (BCE); B) Validation Loss in BCE; C) Discriminative precision on the validation dataset.*

According to A) in Fig.3.16, the additional data collected by different collection methods do not generate obviously influences on model training loss. This means, once the network settings and additional data amount are fixed, the learning speed of the deep learning model on instances from the training dataset will not be influenced.

But, as the models' BCE losses on the validation dataset are shown in B), the performance of model trained by Group GAN-TDA dataset shows a significant advantage than

that of model trained by Group Random dataset in BCE loss (Final loss: 0.0305 - 0.1740). This means the models' generalization ability on unseen new data could be affected by the quality of additional data for training. In other words, although both collection methods guarantee the learning of instances from the raw dataset, GAN-TDA guided additional data performs better in boosting the model's learning of data distribution, which leads to the reinforcing of model generalization ability.

The result in *C*) demonstrates the models' discriminative precision on validation dataset. The model trained with GAN-TDA guided additional data shows a quicker rising in inference precision of validation dataset compared with randomly collected data. And by the end of my training, GAN-TDA guided model's final precision on the validation dataset achieve a 4.75% increase on that of model trained by random additional data (99.42% - 94.67% on 350 images). Simultaneously, GAN-TDA guided additional data could make the deep learning model achieve high accuracy faster than randomly collected additional data.

3.4.4 Conclusion

Increased proliferation of drones and autonomous air vehicles can disrupt critical national services (e.g., Gatwick Airport 2018). The economic damage for air transport is estimated to be millions per day for airports and airlines. Protection against drones is critical to ensuring smooth operation of services, whilst safeguarding it against the most severe threats. High resolution cameras using deep learning is challenged by the lack of training data sets. This often means a large amount of resources and time is dedicated to broad data collection and (re)training the neural network - without a guaranteed convergence in improving accuracy. This chapter has used explainable deep learning to identify the missing data and guide data collection and generation.

For high-dimensional image data, the GAN-TDA provide a solution to extract the latent features of each data instance as feature maps and generate a demonstration of the generalization ability of the convolution kernels on different latent features. With the

mapping relationship among images, latent features and labels, the generalization ability of kernels on latent features could indicate that on different image properties (according to data labels). During model training, the training of hard-to-learned kernels with slow improvement in generalization abilities needs more training epochs and additional data feedings, which means an image instance with properties of these hard-to-learned kernels are more difficult for DL models to learn (in my experiment: images with these properties are hard for GAN to generate). Afterwards, analysing these learning-hardly images with their tagged properties can indicate the direction of new data collection. By applying GAN-TDA proposed in my chapter, I achieve a 4.75% precision boosting (99.42%) on drone discriminative NN compared with a control Model which uses randomly collection method (94.67%). Simultaneously, GAN-TDA guided data make the discriminative NN achieve the same inference performance with less training time.

Chapter 4

Overall Discussion with Implementation

4.1 XAI4TS Framework

In paper chapters of this thesis, I firstly embed the priori knowledge in time series and break it down into neuron-based dynamic modelling. Then secondly, I propose interpretation methodologies for neural model explainability in distribution generalization and inference aspects.

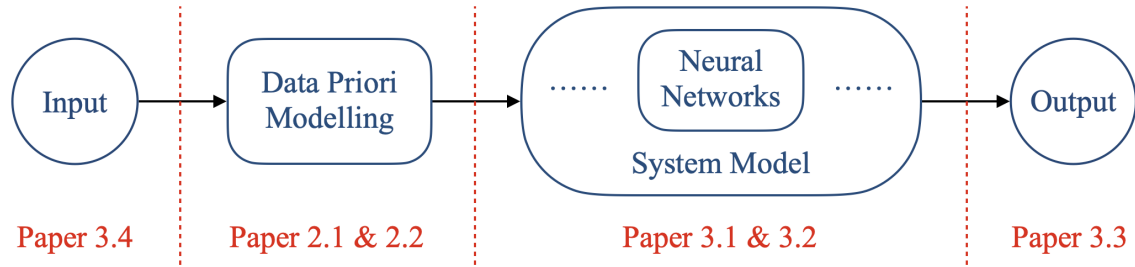


Figure 4.1: The end-to-end framework.

For an overall implementation, the end-to-end framework of explainable AI for time series modelling and causal inference is generated in Fig.4.1 via the papers in this thesis. In this chapter, I briefly apply the framework in a business case of semiconductor manufacturing for overall discussion.

4.2 Implementation in Smart Manufacturing

4.2.1 Introduction

4.2.1.1 Domain Knowledge

In semiconductor semicon fab ¹, wafers go through a series of operations (incl. etch, photo, diffusion, etc.) repeatedly and then become multi-layer chips called FabOut products. Each wafer has its own route prescription on the production line, called traveller, depend on different design IDs (DIDs).

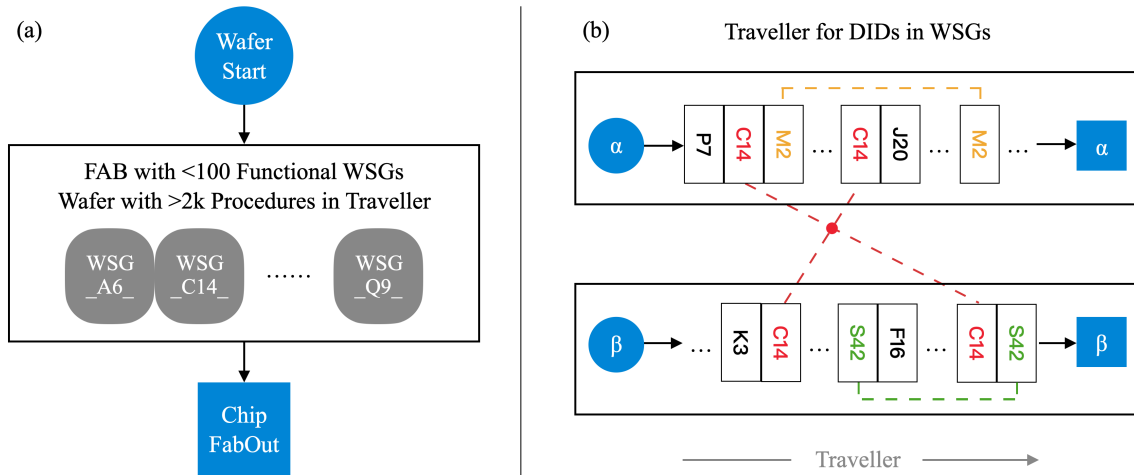


Figure 4.2: (a) Each wafer has over 2000 procedures in production traveller but can be done within 100 work station groups (WSGs); (b) WSGs always recur in (orange and green) and across (red) travellers for α and β DIDs.

In the implementation, a fab of dynamic random-access memory (DRAM) in Taiwan is investigated. As shown in Fig.4.2a, there are over 2000 procedures on traveller before a wafer becoming a DRAM chip in this fab within no more than 100 work station groups (WSGs). Due to the three-dimension (3D) stacking technique in integrated circuit (IC), most WSGs (e.g., etch, photo and diffusion WSGs) are recurring in multi-layer chips fabrication. Thus, in Fig.4.2b, each WSG may undertake the same task repetitively in one DID traveller (orange and green) and/or across several travellers for different DIDs (red).

¹An overview of the fabrication (5 mins): <https://www.youtube.com/watch?v=qILBM1NP3U4>

In addition to the order of WSGs, there are also other constraints given by the traveller. For example, the bundle prescription in traveller requires several procedures to be done within a specific time, otherwise wafers will be unqualified. This means, the relevant WSGs have to assign the corresponding priority to DID with such traveller. Nevertheless, some WSGs need long runtime but can process wafers with same DIDs in batch. These WSGs have to be idle until all the bundle prescription can be allocated properly. At the same time, 3-4 million wafers in progress (WIP) are distributed and queuing up at different WSGs in the same fab according to the traveller. As a result, fabrication is a nonlinear and time-varying dynamic system, and can be considered as a black-box to the planners.

4.2.1.2 Background and Motivation

The time it takes for a wafer finishing the traveller until a chip FabOut, which is called cycle time (CT), lasts around 9-14 weeks for DRAM in average. Therefore, it is crucial to plan the wafer start volume in advance for meeting the contractual orders. In addition, under instant varying market, fab also aims to optimize the strategy of wafer start (SWS) for other temporary requirements (e.g., max-out for specific DID product), while ensure the baseline of meeting the contractual demand. However, based on aforementioned WSG usage scheme, there exists implicit dynamic among the FabOut, CT and WIP (detailed discussion in 4.2.2.1), which brings the difficulties to SWS.

Since there is no explicit expressions for the fabrication, an event-based simulator has been developed as a model-driven digital twins, which plays a vital role for the virtual interaction between data scientists (DS) and real fab in SWS. However, this simulator freezes most advanced dynamic optimization/planning models, due to its time-consuming inner calculations in each WSG event (about 90 mins runtime in simulator for per avg. CT in fab). For example in reinforcement learning, the interactions with the environment must be inside the training loop, hence the event-based simulator will make the optimization/planning last forever.

Consequently, I develop a data-driven model as the twin of simulator, the SimTwin, with rapid inference response. The framework is shown in Fig.4.3 — the SimTwin is proposed to generalize the simulator function via historical data, so that DS model can have efficient interaction with the SimTwin.

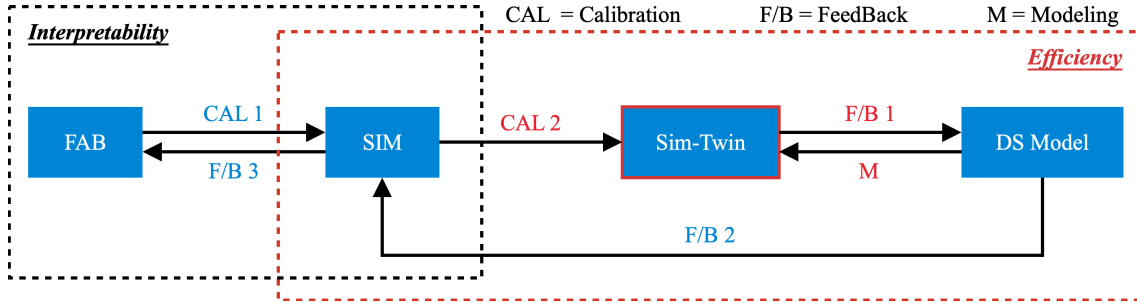


Figure 4.3: Motivation for Sim-Twin.

4.2.1.3 Contribution and Novelty

In this implementation, I model the fab manufacturing via a data-driven & neuron-based SimTwin trained by the data from model-driven & event-based simulator. Two major contributions are:

- (i) The SimTwin achieve the equivalent accuracy performance for SWS-FabOut inference and alleviate the runtime by $100\times$ less compared to the simulator;
- (ii) Symbolic expressions of the SimTwin is approximated base on domain boundary conditions for explainability;

4.2.2 Time Series Modelling

4.2.2.0 Notation

In the modeling, the notations in Tbl.4.1 are taken.

Table 4.1: Notations defined in subsequent derivations.

Notation	Meaning	Notation	Meaning
Λ	Traveller Length (steps)	λ	Travel Distance (steps)
ω	Wafer in Progress (lots)	v	Velocity (steps/unit time)
i	Wafer DID $\in \{\alpha, \beta, \gamma, \dots\}$	k	All DIDs $k = \{\alpha, \beta, \gamma, \dots\}$
θ	F/O = FabOut	ρ	FabOut Upper Bound
T	Cycle Time	τ	Instantaneous Time

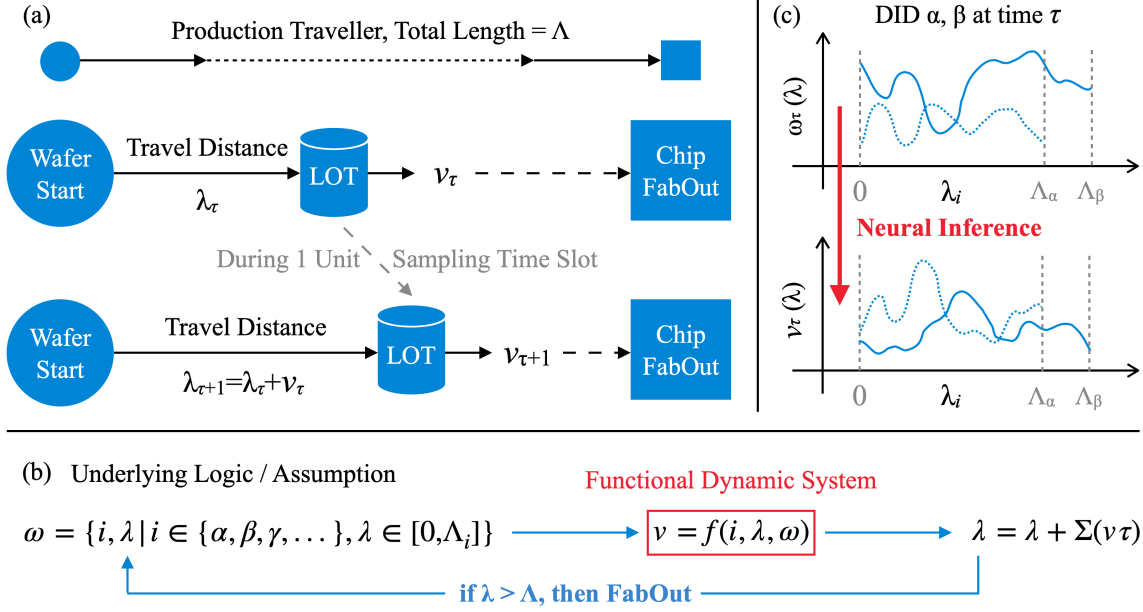


Figure 4.4: (a) Sketch of travel velocity model; (b) The underlying logic/assumptions show that the dynamic function of velocity is the key to the model; (c) The expected input/output inference function for the velocity model.

4.2.2.1 Fabrication Dynamic

With the idea in paper 2.1 and 2.2, the original time series is considered as a dynamic model via its underlying physical domain knowledge. The dynamic modeling focuses on the wafer move velocity v within WSGs (steps per unit time) as the core variable over the time, while is assumed to be constant across the unit sampling period (Fig.4.4a). The model dynamic takes the DID (i), travel distance (λ) and WIP (ω) into account with the underlying logic shown in Fig.4.4b, and can hence be expressed by functional complex system of:

$$\left. \frac{d\lambda_i}{dt} \right|_{t=\tau} = v_i^\tau(\lambda) = \mathbf{f}_i \left[\omega_\alpha^\tau(\lambda), \omega_\beta^\tau(\lambda), \omega_\gamma^\tau(\lambda), \dots \right], i = \alpha, \beta, \gamma, \dots \quad (4.1)$$

where $\mathbf{f}_i(\cdot)$ is the functions of $\omega_k^\tau(\lambda)$ with the operation demonstrated in Fig.4.4c. To the best knowledge, there is no empirical prior forms of this functional, thus, neuron-based model is proposed to fit this functional operation as a black-box.

4.2.2.2 Neural Modeling

Base on the domain knowledge of fabrication, the travellers can be divided into blocks according to fab operation suggestions. Then, the original model is converted as a nonlinear sequence to sequence (Seq2Seq) inference problem as shown in Fig4.5.

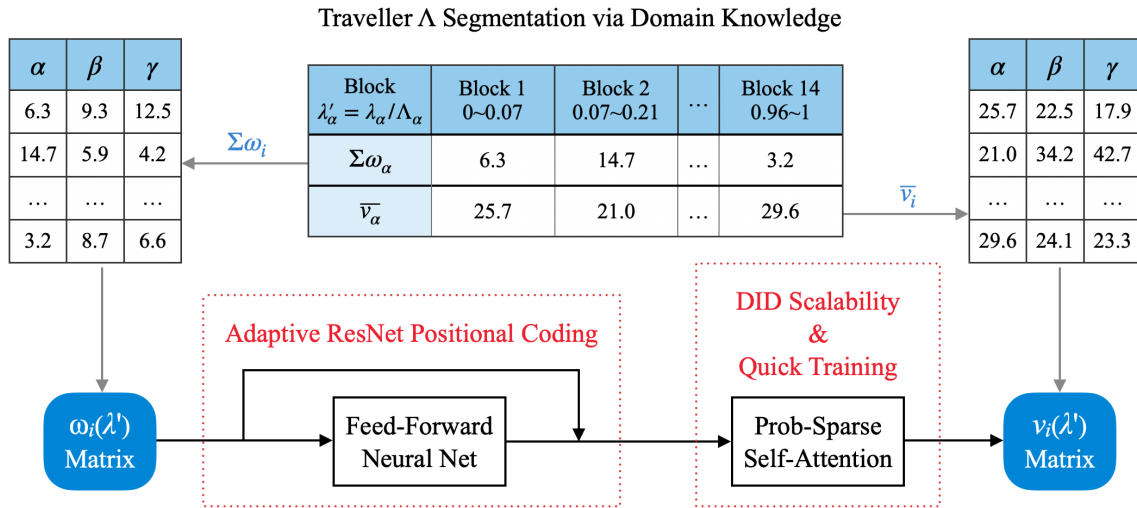


Figure 4.5: Neuron-based data-driven model.

Here I use the attention-based neural network to model the Seq2Seq inference [144]. In order to have scalability for the DID number and quick training/inference, probability-sparse self-attention layer is selected as the main part of the model. Since the position of data in my problem is not as important as that in natural language processing (NLP) but still need encoding, I apply the idea from ResNet [46] for an adaptive positional weak embedding.

4.2.2.3 Symbolic Boundary Condition

In the fabrication, base on the domain knowledge, the following assumption can be made for boundary condition:

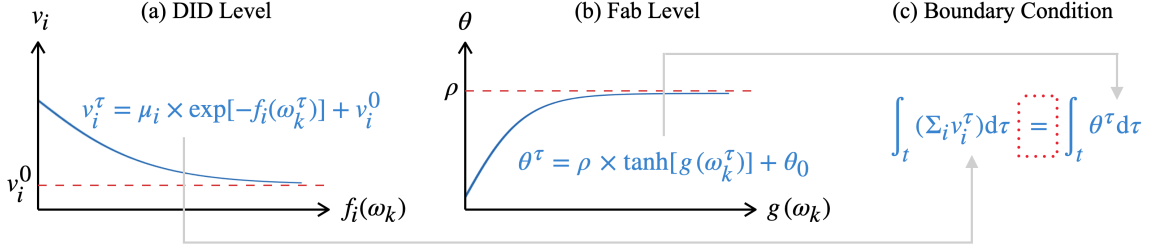


Figure 4.6: (a) DID level constrain; (b) Fab level constrain; (3) Boundary condition constrain.

(a) DID level: For each DID i at time τ , as shown in Fig.4.6a, the travel velocity:

$$v_i^\tau(\lambda) = \mu_i \times \exp[-f_i(\omega_k^\tau(\lambda))] + v_i^0 \quad (4.2)$$

For each DID, the travel velocity is always positive but depends on how many wafers are queuing up on its traveller, and achieves the maximum velocity when no wafer is waiting.

(b) Fab level: For the Fab at time τ , as shown in Fig.4.6b, the FabOut:

$$\theta\tau = \rho \times \tanh[g(\omega_k^\tau)] + \theta_0 \quad (4.3)$$

For the whole fab, there is a "Max-Out" limit. No matter how SWS be optimized / WIP be changed, a bottom neck will always determine this limitation.

Note that, in this implementation, I take " $\exp(-x)$ " and " $\tanh(x)$ " expressions here, but they can be arbitrary bounded monotone asymptotic functions without altering the underlying assumption. Then theoretically:

(c) The summation of DID level FabOut via DID level assumption ought to be always equal to that from Fab level as shown in Fig.4.6c. The inner functions $f(*)$ and $g(*)$ are unknown in reality but may have possible answers under this constrain.

4.2.3 Explainable Neuron

4.2.3.1 Targeted Data Collection

Two types data are collected for the training — (i) historical data from fab and (ii) generated data from model-driven event-based simulator. Although simulator can generate data much faster and more flexible than the real fab, it is still too time-consuming (i.e., 90 mins runtime for 12 weeks' simulation) to exhaust all possible conditions.

Thus, the GAN-TDA methodology in chapter 3.4 is applied for a targeted data collection guidance from simulator. In reality, this fab have totally 14 overall manufacturing mode setting as shown in the Fig. 4.7 (left). Accordingly, the sub-labels for input matrix can be tagged with former and current condition of each DID production line, as shown in Fig. 4.7 (right).

Condition	DID α_1	DID α_2	DID α_3	DID α_4	DID β_1	DID γ_1	DID γ_2		Sub Label	Former Condition	Current Condition
1	MaxOut	Baseline	Lean	Lean	Baseline	Baseline	Baseline		a	4	→ 2
2	MaxOut	Baseline	Lean	Baseline	MaxOut	Baseline	Lean		b	7	→ 7
...	...	...	...	...	...	...	...		c	6	→ 14
14	Baseline	Baseline	Baseline	MaxOut	MaxOut	Baseline	Lean		...	...	...

Figure 4.7: The production line condition can vary with the change of wafer start strategy. In most cases, the strategy will be executed for less than a complete cycle time, so one-step former condition is taken with the current condition as the sub-label of the WIP matrix.

4.2.3.2 Explainable Neuron Twins

With boundary conditions stated in 4.2.2.3, the explainability can be achieved by revealing the explicit expression of $f(*)$ and $g(*)$. In order to apply the symbolic approximation stated in chapter 3.1 and 3.2, the number of variables has to be limited. For the purpose of approximation explainability only, I assume the WIP are evenly distributed on the traveller and moving with the same velocity. Then, Meijer G-Function can be optimized as the symbolic twin for the neural networks via gradient descent [4].

4.2.3.3 Probabilistic Inference

Due to the unscheduled down of the tools in WSGs, there exists the stochasticity in the dynamic, which may lead to bifurcation in the forward inference time series. Remember that the forecasting of the Sim-Twin will be used for further optimization on SWS series planning, the distributions of bifurcation are more important than the mean predicted values. Hence I adopt the idea of chapter 3.3, using Gaussian mixture model as the output of the neural network, then mixture density network (MDN) is applied at the last layer to generate the multimodal distribution for the probabilistic inference.

4.2.4 Results

4.2.4.1 GAN-TDA

I merge the existing real data and generated data into the same TDA for analysis as shown in Fig.4.8.

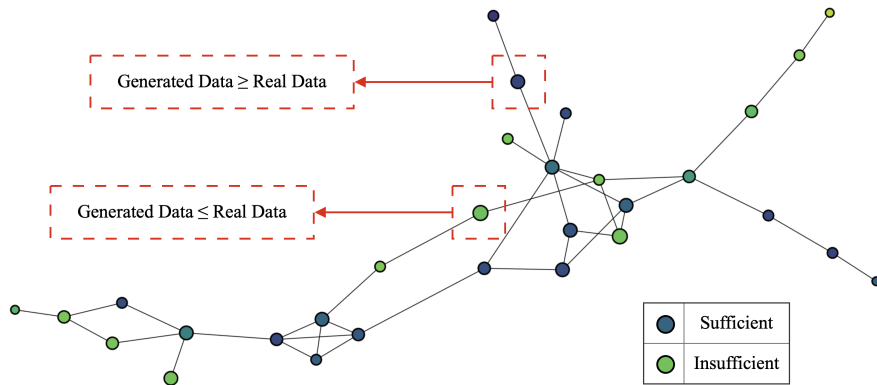


Figure 4.8: TDA informed further data collection guidance.

By investigating the components of each insufficient node, I found that data in the following conditions are more hard to be learnt by neurons:

- (i) In the same series, more than one DIDs are set into “Lean” mode (e.g. DID α_3 and α_4 in Condition 1);
- (ii) The manufacturing mode for DID β transfers from “Baseline” into “MaxOut” or “Lean” (e.g., from Condition 1 to Condition 2);

(iii) From none “MaxOut” to more than 2 DIDs’ “MaxOut” at the same time.

According to the possible modes in real fab (Fig.4.7) and these conditions, I can then generate more data for training from the model-driven event-based simulator.

4.2.4.2 Symbolic Approximation

For trustworthy usage of end-users, symbolic approximation is applied to explore the neuron inner operations. For simplicity, in the approximation, I assume the wafer moving velocity keep the same across the traveller as one block (Fig.4.5). Then I get an example explicit expression at time τ for DID γ as:

$$v_{\gamma}^{\tau} = 19.022 \times \exp \left\{ - (0.163\omega_{\gamma} + 0.042\omega_{\beta}) \right. \\ \left. \times [1 + 0.585 \sin (7.302\omega_{\beta} - 3.792\omega_{\alpha})] \right\} + 6.147 \quad (4.4)$$

which matches the common sense — the velocity is generally decreasing when the WIP is increasing. The inner $\sin(*)$ function implies the potential relationship affected by the batch and cascade process blocks.

4.2.4.3 Multimodal Distribution Inference

Due to the non-normal stochasticity in the fabrication, the distribution of the velocity is more important than the mean value in the inference for further planning and optimization actions on the SWS and etc. With mixture Gaussian multimodal (GMM), three typical patterns as examples in different processing blocks are shown Fig.4.10.

(i) For general blocks, the velocity is close to the normal distribution. The uncertainty in this kind of blocks is given by the tool/equipment unscheduled downtime/failure;

(ii) For batch process blocks, the velocity can be considered as uniform distributed over a certain range and also with the uncertainty as it in (i). The flat difference in velocity is caused by the queuing in the batching procedure, while tool failure always exists;

(iii) For cascade process blocks, the velocity shows a bimodal distribution. Both cascade WSGs clearing and queuing vary the waiting time for each wafer and make this

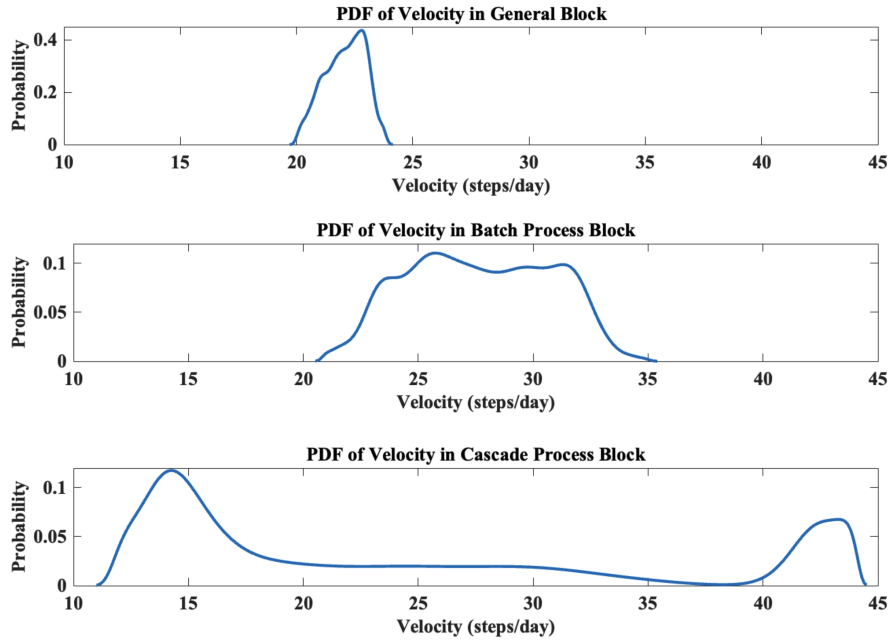


Figure 4.9: Velocity distribution patterns in three typical kinds of processing blocks.

pattern in velocity. The formulation of this kind of distribution is crucial for further planning and optimization AI agent.

4.2.4.4 Time Series Forecasting

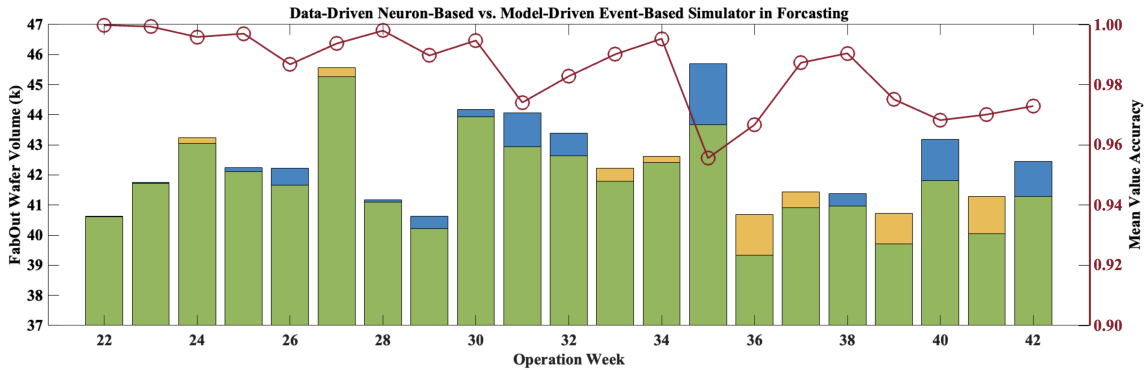


Figure 4.10: Blue: the ground-true FabOut given by model-driven event-based simulator; Yellow: the forecasting FabOut given by data-driven neuron-based SimTwin; Green: the matched/hit part of two methods; Red: the hit rate of the SimTwin.

If I set the result from model-driven event-based simulator as the ground-true SWS-FabOut forecasting, then the accuracy of the data-driven neuron-based SimTwin can be evaluated by mean value FabOut hit rate. The hit rate is consistently maintained over 95%

across more than one complete cycle time. However, due to the cumulative uncertainty, the accuracy increasingly fluctuates with the length of forecasting time.

4.3 Overall Discussion

The proposed XAI4TS framework is implemented on the basis of the papers in this thesis — Firstly embed the priori knowledge in time series and break it down into neuron-based dynamic modelling; Then secondly, propose interpretation methodologies for neural model explainability in distribution generalization and inference aspects.

In this implementation, I model the time series of SWS-FabOut as a dynamic system by introducing WIP moving velocity on traveller. Then neuron-based methods are applied for training series generation, system symbolic approximation and probabilistic bifurcation inference. Different types of explainability are explored in the framework—

(i) For data scientist: In data collection procedure, GAN-TDA helps to understand the crucial features of data for the neural model learning and leads to the guidance of targeted data collection for more rapid and generalized training.

(ii) For domain users: In data modeling procedure, time series are broken down into dynamic system via the priori domain knowledge. Then in the neuron training procedure, users can explore the symbolic/qualitative approximations of function of neural networks base on the boundary conditions for trustworthy usage.

(iii) For AI expert: In causal inference procedure, the forecasting result is formed in data-driven multimodal distribution which gives the bifurcation probabilistic inference without any priori distribution assumption. The posteriori PDFs can hence drive further planning and optimization AI agent (e.g., reinforcement learning) in next stages.

This implementation shows the landing potential of my XAI4TS framework in dynamic engineering time series application.

Chapter 5

Conclusions and Future Work

Due to the concern over the trust issue of black-box artificial intelligence (AI) in daily human life, “right to an explanation” requirement is hence proposed for any algorithm in Europe general data protection regulation (GDPR). However, the framework of explainable AI (XAI) is still in its infancy. In this thesis, I generate and implement the framework of explainable AI for time series (XAI4TS) modelling and causal inference.

My key findings can be elaborated as: (i) I embed the priori domain knowledge in time series causal modelling and break it down into neuron-based dynamic system. In the thesis, I take the climate and conflict as example, formulate its dynamic mechanism of time series via neural modelling with priori assumptions, and demonstrate the importance of domain expertise in explainable time series modelling. (ii) I propose interpretation methodologies for neural model explainability in and distribution generalization and posteriori inference. GAN-TDA workflow is proposed to identify the “hard to learn” features by exploring the feature distribution generalization in neurons, in order to give the further data collection guidance for incremental training; I formulate the flexible activation function to reveal the inner operation within neurons by symbolic approximation twins of neural networks; I use the feature embedding kernel in Gaussian Processes form the posteriori inference into multimodal distributions, which represent the real data distribution, for further application enabling (e.g., reinforcement learning).

Through this framework in my thesis, diverse explainability can be achieved for various trustworthy usage of different users. Specifically, (i) Domain end-users can trust the model base on the dynamic via priori knowledge with their expertise; (ii) Data scientists can trust the data and neurons base on the comparison of generalized feature distribution via GAN-TDA workflow with graphic / tabular visualization; (iii) AI expert can trust the posteriori inference base on multimodal distribution via data-driven probabilistic bifurcation with no assumptions. Meanwhile, I acknowledge the importance of domain expertise in my XAI4TS framework for human comprehensible interpretation, whereas, that implies the principle of epistemology in philosophy — explanation cannot be revealed without methodological ontology and logic.

Time series forecasting is one the key enablers for the static decision making and/or dynamic actions optimization in deep reinforcement learning. Whereupon my future work plans to apply XAI4TS framework as the explainable stochastic emulator environment for the AI agent, and investigate further explainability in neuron-based reinforcement learning algorithms.

References

- [1] On artificial intelligence - a european approach to excellence and trust. European Commission White Paper, 2020.
- [2] AGOSTINELLI, F., HOFFMAN, M., SADOWSKI, P., AND BALDI, P. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830* (2014).
- [3] ALAA, A., AND SCHAAR, M. Demystifying black-box models with symbolic metamodels. In *NIPS* (2019).
- [4] ALAA, A. M., AND VAN DER SCHAAR, M. Demystifying black-box models with symbolic metamodels. In *Advances in Neural Information Processing Systems* (2019), pp. 11301–11311.
- [5] ALTMAN, E., AVRACHENKOV, K., AND GARNAEV, A. Closed form solutions for symmetric water filling games. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications* (2008), IEEE, pp. 673–681.
- [6] ANGELINE, P. J., SAUNDERS, G. M., AND POLLACK, J. B. An evolutionary algorithm that constructs recurrent neural networks. *IEEE transactions on Neural Networks* 5, 1 (1994), 54–65.
- [7] ARJOVSKY, M., CHINTALA, S., AND BOTTOU, L. Wasserstein generative adversarial networks. In *International conference on machine learning* (2017), PMLR, pp. 214–223.

- [8] ASAKA, J. O. Climate change-terrorism nexus? a preliminary review/analysis of the literature. *Perspectives on Terrorism* 15, 1 (2021), 81–92.
- [9] BALLESTER, P., AND ARAUJO, R. M. On the performance of googlenet and alexnet applied to sketches. In *Thirtieth AAAI Conference on Artificial Intelligence* (2016).
- [10] BERGOMI, M. G., FROSINI, P., GIORGI, D., AND QUERCIOLO, N. Towards a topological–geometrical theory of group equivariant non-expansive operators for data analysis and machine learning. *Nature Machine Intelligence* 1, 9 (2019), 423–433.
- [11] BRAUN, J., AND GRIEBEL, M. On a constructive proof of kolmogorov’s superposition theorem. *Constructive approximation* 30, 3 (2009), 653.
- [12] BUHAUG, H. Climate not to blame for african civil wars. *Proceedings of the National Academy of Sciences* 107, 38 (2010), 16477–16482.
- [13] BUHAUG, H., BURKE, M., FEARON, J., FIELD, C., MAYSTADT, J., AND LOUGHLIN, J. Climate as a risk factor for armed conflict. *Nature* (2019).
- [14] BURKE, M. B., MIGUEL, E., SATYANATH, S., DYKEMA, J. A., AND LOBELL, D. B. Warming increases the risk of civil war in africa. *Proceedings of the national Academy of sciences* 106, 49 (2009), 20670–20674.
- [15] CAO, B., SHEN, D., SUN, J.-T., YANG, Q., AND CHEN, Z. Feature selection in a kernel space. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 121–128.
- [16] CAO, X., ZHONG, Y., ZHOU, Y., WANG, J., ZHU, C., AND ZHANG, W. Interactive temporal recurrent convolution network for traffic prediction in data centers. *IEEE Access* 6 (2018), 5276–5289.

- [17] CHEN, C., NOBLE, I., HELLMANN, J., COFFEE, J., MURILLO, M., AND CHAWLA, N. University of notre dame global adaptation index country index technical report.
- [18] CHINTALA, S., DENTON, E., ARJOVSKY, M., AND MATHIEU, M. How to train a gan? tips and tricks to make gans work. *Github. com* (2016).
- [19] CHVOSTEKOVÁ, M., JAKUBÍK, J., AND KRAKOVSKÁ, A. Granger causality on forward and reversed time series. *Entropy* 23, 4 (2021).
- [20] CYBENKO, G. Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2 (1989), 183–192.
- [21] DOSHI-VELEZ, F., AND KIM, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).
- [22] DOŠILOVIĆ, F. K., BRČIĆ, M., AND HLUPIĆ, N. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)* (2018), IEEE, pp. 0210–0215.
- [23] DOUZETTE, A. S. B-splines in machine learning. Master’s thesis, 2017.
- [24] DU, Z., SUN, Y., GUO, W., XU, Y., WU, Q., AND ZHANG, J. Data-driven deployment and cooperative self-organization in ultra-dense small cell networks. *IEEE Access* 6 (2018), 22839–22848.
- [25] ENGEL, D., HÜTTENBERGER, L., AND HAMANN, B. A survey of dimension reduction methods for high-dimensional data analysis and visualization. In *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011* (2012), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

- [26] ET AL., S. C. Interpretability of deep learning models: A survey of results. In *IEEE SmartWorld* (Aug 2017), pp. 1–6.
- [27] FARSAF, N., AND GOLDSMITH, A. Neural network detection of data sequences in communication systems. *IEEE Transactions on Signal Processing* 66, 21 (Nov 2018), 5663–5678.
- [28] FENG, J., CHEN, X., GAO, R., ZENG, M., AND LI, Y. Deeptp: An end-to-end neural network for mobile cellular traffic prediction. *IEEE Network* 32, 6 (November 2018), 108–115.
- [29] FIELD, C., BARROS, V., STOCKER, T., AND QIN, D. Managing the risks of extreme events and disasters to advance climate change adaptation. *Intergovernmental Panel on Climate Change (IPCC)* (2012).
- [30] FODOR, I. K. A survey of dimension reduction techniques. Tech. rep., Citeseer, 2002.
- [31] FROGNER, C., ZHANG, C., MOBAHI, H., ARAYA, M., AND POGGIO, T. A. Learning with a wasserstein loss. *Advances in Neural Information Processing Systems* 28 (2015), 2053–2061.
- [32] FROSST, N., AND HINTON, G. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784* (2017).
- [33] GLEDITSCH, K., AND WARD, M. Forecasting is difficult, especially about the future: Using contentious issues to forecast interstate disputes. *Journal of Peace Research* (2013).
- [34] GOODFELLOW, I. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* (2016).

- [35] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAI, S., COURVILLE, A., AND BENGIO, Y. Generative adversarial nets. In *Advances in neural information processing systems* (2014), pp. 2672–2680.
- [36] GRANGER, C. W. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society* (1969), 424–438.
- [37] GRUAU, F., WHITLEY, D., AND PYEATT, L. A comparison between cellular encoding and direct encoding for genetic neural networks. In *Proceedings of the 1st annual conference on genetic programming* (1996), pp. 81–89.
- [38] GUO, W. Common statistical patterns in urban terrorism. *Royal Society open science* 6, 9 (2019).
- [39] GUO, W. Explainable Artificial Intelligence (XAI) for 6G: Improving Trust between Human and Machines. *arXiv:1911.04542* (Nov 2019).
- [40] GUO, W. Explainable Artificial Intelligence (XAI) for 6G: Improving Trust between Human and Machine. *IEEE Communications Magazine* (2020).
- [41] GUO, W. Explainable Artificial Intelligence (XAI) for 6G: Improving Trust between Human and Machine. *IEEE Communications Magazine* 58, 6 (2020).
- [42] GUO, W. Partially Explainable Big Data Driven Deep Reinforcement Learning for Green 5G UAV. In *IEEE International Conference on Communications (ICC)* (2020), IEEE.
- [43] GUO, W., GLEDITSCH, K., AND WILSON, A. Retool ai to forecast and limit wars. *Nature* (2018).
- [44] GUO, W., AND O’FARRELL, T. Dynamic cell expansion with self-organizing cooperation. *IEEE Journal on Selected Areas in Communications* 31, 5 (May 2013), 851–860.

- [45] GUO, W., WANG, S., AND CHU, X. Capacity expression and power allocation for arbitrary modulation and coding rates. In *IEEE Wireless Communications and Networking Conference (WCNC)* (April 2013), pp. 3294–3299.
- [46] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.
- [47] HENDRICKS, L. A., AKATA, Z., ROHRBACH, M., DONAHUE, J., SCHIELE, B., AND DARRELL, T. Generating visual explanations. In *European Conference on Computer Vision* (2016), Springer, pp. 3–19.
- [48] HLAVÁČKOVÁ-SCHINDLER, K., PALUŠ, M., VEJMEĽKA, M., AND BHATTACHARYA, J. Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports* 441, 1 (2007), 1–46.
- [49] HOMER-DIXON, T. F. Environment, scarcity, and violence. In *Environment, scarcity, and violence*. Princeton University Press, 2010.
- [50] HSIANG, S., AND BURKE, M. Climate, conflict, and social stability: what does the evidence say? *Climate Change* (2014).
- [51] HSIANG, S., MENG, K., AND CANE, M. Civil conflicts are associated with the global climate. *Nature* (2011).
- [52] HSIANG, S. M., AND BURKE, M. Climate, conflict, and social stability: what does the evidence say? *Climatic change* 123, 1 (2014), 39–55.
- [53] HSIANG, S. M., MENG, K. C., AND CANE, M. A. Civil conflicts are associated with the global climate. *Nature* 476, 7361 (2011), 438–441.
- [54] HUISMAN, M., VAN RIJN, J. N., AND PLAAT, A. A survey of deep meta-learning. *Artificial Intelligence Review* (2021), 1–59.

- [55] IGELNIK, B., AND PARIKH, N. Kolmogorov's spline network. *IEEE transactions on neural networks* 14, 4 (2003), 725–733.
- [56] ILONEN, J., KAMARAINEN, J.-K., AND LAMPINEN, J. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters* 17, 1 (2003), 93–105.
- [57] IOFFE, S., AND SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (2015), PMLR, pp. 448–456.
- [58] ISHAM, V., AND WESTCOTT, M. A self-correcting point process. *Stochastic processes and their applications* 8, 3 (1979), 335–347.
- [59] JIANG, H., CHING, W.-K., AND HOU, W. On orthogonal feature extraction model with applications in medical prognosis. *Applied Mathematical Modelling* 40, 19-20 (2016), 8766–8776.
- [60] JIN, P., LU, L., TANG, Y., AND KARNIADAKIS, G. E. Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness. *Neural Networks* 130 (2020), 85–99.
- [61] KARRAS, T., LAINE, S., AND AILA, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4401–4410.
- [62] KENDALL, A., AND GAL, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems* (2017), pp. 5574–5584.
- [63] KENNY, E. M., AND KEANE, M. T. Twin-systems to explain artificial neural networks using case-based reasoning: comparative tests of feature-weighting methods

- in ann-cbr twins for xai. In *Twenty-Eighth International Joint Conferences on Artificial Intelligence (IJCAI), Macao, 10-16 August 2019* (2019), pp. 2708–2715.
- [64] KHAN, R. U., ZHANG, X., AND KUMAR, R. Analysis of resnet and googlenet models for malware detection. *Journal of Computer Virology and Hacking Techniques* 15, 1 (2019), 29–37.
- [65] KINGMA, D. P., AND WELLING, M. Auto-encoding variational bayes. *stat 1050* (2014), 1.
- [66] KOLMOGOROV, A. N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady Akademii Nauk* (1957), vol. 114, Russian Academy of Sciences, pp. 953–956.
- [67] KRAKOVSKÁ, A. Predictability improvement as a tool to detect causality. In *2017 11th International Conference on Measurement* (2017), IEEE, pp. 39–42.
- [68] KRAKOVSKÁ, A., AND HANZELY, F. Testing for causality in reconstructed state spaces by an optimized mixed prediction method. *Physical Review E* 94, 5 (2016), 052203.
- [69] KRAKOVSKÁ, A., AND JAKUBÍK, J. Implementation of two causal methods based on predictions in reconstructed state spaces. *Physical Review E* 102, 2 (2020), 022203.
- [70] KRAKOVSKÁ, A., JAKUBÍK, J., CHVOSTEKOVÁ, M., COUFAL, D., JAJCAY, N., AND PALUŠ, M. Comparison of six methods for the detection of causality in a bivariate time series. *Physical Review E* 97, 4 (2018), 042207.
- [71] KRAKOVSKA, A., JAKUBIK, J., CHVOSTEKOVA, M., COUFAL, D., JAJCAY, N., AND PALUS, M. Comparison of six methods for the detection of causality in a bivariate time series. *Physical Review E* (2018).

- [72] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.
- [73] LE, L., SINH, D., TUNG, L., AND LIN, B. P. A practical model for traffic forecasting based on big data, machine-learning, and network KPIs. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)* (Jan 2018), pp. 1–4.
- [74] LEE, J., BAHRI, Y., NOVAK, R., SCHOENHOLZ, S. S., PENNINGTON, J., AND SOHL-DICKSTEIN, J. Deep neural networks as gaussian processes. In *International Conference on Learning Representations* (2018).
- [75] LEE, S., KIM, J., JUN, J., HA, J., AND ZHANG, B. Overcoming catastrophic forgetting by incremental moment matching. *Advances in Neural Information Processing Systems (NIPS)* (2017).
- [76] LESHNO, M., LIN, V. Y., PINKUS, A., AND SCHOCKEN, S. Multilayer feed-forward networks with a nonpolynomial activation function can approximate any function. *Neural networks* 6, 6 (1993), 861–867.
- [77] LI, C., GUO, W., SUN, S., AL-RUBAYE, S., AND TSOURDOS, A. Trustworthy Deep Learning in 6G Enabled Mass Autonomy: from Concept to Quality-of-Trust KPIs. *IEEE Vehicular Technology Magazine* (2020).
- [78] LI, G., HOYDIS, J., CARVALHO, E., STIMMING, A., AND QIN, Z. Best Readings in Machine Learning in Communications. <https://www.comsoc.org/publications/best-readings/machine-learning-communications>, 2019.
- [79] LI, R., ZHAO, Z., ZHENG, J., MEI, C., CAI, Y., AND ZHANG, H. The learning and prediction of application-level traffic data in cellular networks. *IEEE Transactions on Wireless Communications* 16, 6 (June 2017), 3899–3912.

- [80] LI, X., AND OUYANG, G. Estimating coupling direction between neuronal populations with permutation conditional mutual information. *NeuroImage* 52, 2 (2010), 497–507.
- [81] LI, Y., YOSINSKI, J., CLUNE, J., LIPSON, H., AND HOPCROFT, J. E. Convergent learning: Do different neural networks learn the same representations? In *FE@NIPS* (2015), pp. 196–212.
- [82] LIANG, F., SHEN, C., YU, W., AND WU, F. Towards optimal power control via ensembling deep neural networks. *IEEE Transactions on Communications* (2019), 1–1.
- [83] LIU, Z., XU, Y., QIU, C., AND TAN, J. A novel support vector regression algorithm incorporated with prior knowledge and error compensation for small datasets. *Neural Computing and Applications* 31, 9 (2019), 4849–4864.
- [84] LOZANO, A., TULINO, A. M., AND VERDU, S. Optimum power allocation for parallel gaussian channels with arbitrary input distributions. *IEEE Transactions on Information Theory* 52, 7 (2006), 3033–3051.
- [85] LU, L., JIN, P., PANG, G., ZHANG, Z., AND KARNIADAKIS, G. E. Learning nonlinear operators via deepnet based on the universal approximation theorem of operators. *Nature Machine Intelligence* 3, 3 (2021), 218–229.
- [86] LU, Z., PU, H., WANG, F., HU, Z., AND WANG, L. The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems* (2017), pp. 6231–6239.
- [87] LUM, P. Y., SINGH, G., LEHMAN, A., ISHKANOV, T., VEJDEMO-JOHANSSON, M., ALAGAPPAN, M., CARLSSON, J., AND CARLSSON, G. Extracting insights from the shape of complex data using topology. *Scientific reports* 3 (2013), 1236.

- [88] LUNDBERG, S. M., AND LEE, S.-I. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (2017), pp. 4765–4774.
- [89] MA, B., GUO, W., AND ZHANG, J. A survey of online data-driven proactive 5g network optimisation using machine learning. *IEEE Access* 8 (2020), 35606–35637.
- [90] MA, B., GUO, W., AND ZHANG, J. A survey of online data-driven proactive 5g network optimisation using machine learning. *IEEE Access* 8 (2020), 35606–35637.
- [91] MACH, K. J., KRAAN, C. M., ADGER, W. N., BUHAUG, H., BURKE, M., FEARON, J. D., FIELD, C. B., HENDRIX, C. S., MAYSTADT, J.-F., O’LOUGHLIN, J., ET AL. Climate as a risk factor for armed conflict. *Nature* 571, 7764 (2019), 193–197.
- [92] MAO, Q., HU, F., AND HAO, Q. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Communications Surveys Tutorials* 20, 4 (Fourthquarter 2018), 2595–2621.
- [93] MEI, H., AND EISNER, J. M. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems* 30 (2017).
- [94] NAKAMURA, T., IMAMURA, M., MERCER, R., AND KEOGH, E. Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. In *IEEE International Conference on Data Mining (ICDM)* (2020), IEEE.
- [95] NEAL, R. M. Priors for infinite networks. In *Bayesian Learning for Neural Networks*. Springer, 1996, pp. 29–53.

- [96] NI, F., ZANG, Y., AND FENG, Z. A study on cellular wireless traffic modeling and prediction using elman neural networks. In *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)* (Dec 2015), vol. 01, pp. 490–494.
- [97] NIE, F., XIANG, S., LIU, Y., HOU, C., AND ZHANG, C. Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction. *Pattern Recognition Letters* 33, 5 (2012), 485–491.
- [98] PIAZZA, F., UNCINI, A., AND ZENOBI, M. Artificial neural networks with adaptive polynomial activation function.
- [99] PRABHU, R. S., AND DANESHRAJ, B. An energy-efficient water-filling algorithm for ofdm systems. In *2010 IEEE International Conference on Communications* (2010), IEEE, pp. 1–5.
- [100] QIAN, S., LIU, H., LIU, C., WU, S., AND SAN WONG, H. Adaptive activation functions in convolutional neural networks. *Neurocomputing* 272 (2018), 204–212.
- [101] RADFORD, A., METZ, L., AND CHINTALA, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
- [102] RAISSI, M., PERDIKARIS, P., AND KARNIADAKIS, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378 (2019), 686–707.
- [103] RAMONA, M., RICHARD, G., AND DAVID, B. Multiclass feature selection with kernel gram-matrix-based criteria. *IEEE transactions on neural networks and learning systems* 23, 10 (2012), 1611–1623.

- [104] RASMUSSEN, C. E. Gaussian processes in machine learning. In *Summer School on Machine Learning* (2003), Springer, pp. 63–71.
- [105] RAVI, S., AND LAROCHELLE, H. Optimization as a model for few-shot learning.
- [106] RIBEIRO, M. T., SINGH, S., AND GUESTRIN, C. ” why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (2016), pp. 1135–1144.
- [107] RIBEIRO, M. T., SINGH, S., AND GUESTRIN, C. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (2016), ACM, pp. 1135–1144.
- [108] RIBEIRO, M. T., SINGH, S., AND GUESTRIN, C. ”why should i trust you?”: Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), ACM, pp. 1135–1144.
- [109] ROGIN, J. Confirmed: Obama to “surge” 30,000 troops to afghanistan and “begin” withdrawal in july 2011. *Foreign Policy* (2009).
- [110] ROONEY, K. Barack obama’s first 100 days: A surge for afghanistan. *American Forces Press Service* (2009).
- [111] RUNGE, J. Detecting and quantifying causality from time series of complex systems.
- [112] SALEHYAN, I. Climate change and conflict: Making sense of disparate findings, 2014.
- [113] SALEHYAN, I. Climate change and conflict: Making sense of disparate findings. *Political Geography* (2014).

- [114] SCARDAPANE, S., SCARPINITI, M., COMMINIELLO, D., AND UNCINI, A. Learning activation functions from data using cubic spline interpolation. In *Italian Workshop on Neural Nets* (2017), Springer, pp. 73–83.
- [115] SCARDAPANE, S., VAN VAERENBERGH, S., TOTARO, S., AND UNCINI, A. Kafnets: Kernel-based non-parametric activation functions for neural networks. *Neural Networks 110* (2019), 19–32.
- [116] SCIANCALEPORE, V., SAMDANIS, K., COSTA-PEREZ, X., BEGA, D., GRAMAGLIA, M., AND BANCHS, A. Mobile traffic forecasting for maximizing 5G network slicing resource utilization. In *IEEE Conference on Computer Communications (INFOCOM)* (May 2017), pp. 1–9.
- [117] SHAO, J.-D., RONG, G., AND LEE, J. M. Learning a data-dependent kernel function for kpca-based nonlinear process monitoring. *Chemical Engineering Research and Design* 87, 11 (2009), 1471–1480.
- [118] SHCHUR, O., TURKMEN, A., JANUSCHOWSKI, T., AND GUNNEMANN, S. Neural temporal point processes: A review. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)* (2021).
- [119] SHEN, H., WANG, D., SONG, C., AND BARABÁSI, A.-L. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2014), vol. 28.
- [120] SHEN, Y., AND WANG, B. A fast learning algorithm of neural network with tunable activation function. *Science in China Series F: Information Sciences* 47, 1 (2004), 126–136.
- [121] SHEN, Y., WANG, B., CHEN, F., AND CHENG, L. A new multi-output neural model with tunable activation function and its applications. *Neural processing letters* 20, 2 (2004), 85–104.

- [122] SHORTEN, C., AND KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 1–48.
- [123] SHRIKUMAR, A., GREENSIDE, P., AND KUNDAJE, A. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 3145–3153.
- [124] SHRIKUMAR, A., GREENSIDE, P., AND KUNDAJE, A. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 3145–3153.
- [125] SIMONYAN, K., VEDALDI, A., AND ZISSERMAN, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).
- [126] SLETTEBAK, R. T. Don't blame the weather! climate-related natural disasters and civil conflict. *Journal of Peace Research* 49, 1 (2012), 163–176.
- [127] SOMUYIWA, S. O., GYORGY, A., AND GUNDÜZ, D. A reinforcement-learning approach to proactive caching in wireless networks. *IEEE Journal on Selected Areas in Communications* 36 (June 2018).
- [128] SRIVASTAVA, A. N., SCHUMANN, J., AND FISCHER, B. An ensemble approach to building mercer kernels with prior information. In *2005 IEEE International Conference on Systems, Man and Cybernetics* (2005), vol. 3, IEEE, pp. 2352–2359.
- [129] STOYAN, D., KENDALL, W. S., CHIU, S. N., AND MECKE, J. *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

- [130] SUGIHARA, G., MAY, R., YE, H., HSIEH, C.-H., DEYLE, E., FOGARTY, M., AND MUNCH, S. Detecting causality in complex ecosystems. *science* 338, 6106 (2012), 496–500.
- [131] SUGIHARA, G., MAY, R., YE, H., HSIEH, C.-H., DEYLE, E., FOGARTY, M., AND MUNCH, S. Detecting causality in complex ecosystems, 2012.
- [132] SUN, C., AND GUO, W. Forecasting wireless demand with extreme values using feature embedding in gaussian processes. *arXiv preprint arXiv:1905.06744* (2019).
- [133] SUN, C., SHRIVASTAVA, A., SINGH, S., AND GUPTA, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 843–852.
- [134] SUN, H., CHEN, X., SHI, Q., HONG, M., FU, X., AND SIDIROPOULOS, N. D. Learning to optimize: Training deep neural networks for wireless resource management. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (July 2017), pp. 1–6.
- [135] SUN, S. C., LI, C., WEI, Z., TSOURDOS, A., AND GUO, W. Scalable partial explainability in neural networks via flexible activation functions (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 15899–15900.
- [136] SUN, Y. Iterative RELIEF for feature weighting: algorithms, theories, and applications. *IEEE transactions on pattern analysis and machine intelligence* 29, 6 (2007), 1035–1051.
- [137] TENCH, S., FRY, H., AND GILL, P. Spatio-temporal patterns of ied usage by the provisional irish republican army. *European Journal of Applied Mathematics* 27, 3 (2016), 377–402.

- [138] THAI, P., ALAM, S., LILITH, N., AND NGUYEN, B. T. A computer vision framework using convolutional neural networks for airport-airside surveillance. *Transportation Research Part C: Emerging Technologies* 137 (2022), 103590.
- [139] TSONIS, A. A., DEYLE, E. R., YE, H., AND SUGIHARA, G. Convergent cross mapping: theory and an example. *Advances in nonlinear geosciences* (2018), 587–600.
- [140] URBAN, S., BASALLA, M., AND VAN DER SMAGT, P. Gaussian process neurons learn stochastic activation functions. *arXiv preprint arXiv:1711.11059* (2017).
- [141] VAHDAT, A., AND KAUTZ, J. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898* (2020).
- [142] VAN NES, E. H., SCHEFFER, M., BROVKIN, V., LENTON, T. M., YE, H., DEYLE, E., AND SUGIHARA, G. Causal feedbacks in climate change. *Nature Climate Change* 5, 5 (2015), 445–448.
- [143] VAN VEEN, H. J., SAUL, N., EARGLE, D., AND MANGHAM, S. W. Kepler mapper: A flexible python implementation of the mapper algorithm. *Journal of Open Source Software* 4, 42 (2019), 1315.
- [144] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [145] VECCI, L., PIAZZA, F., AND UNCINI, A. Learning and approximation capabilities of adaptive spline activation function neural networks. *Neural Networks* 11, 2 (1998), 259–270.
- [146] WANG, H., WU, X., HUANG, Z., AND XING, E. P. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings*

- of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8684–8694.
- [147] WANG, X., PIAO, S., CIAIS, P., FRIEDLINGSTEIN, P., MYNENI, R. B., COX, P., HEIMANN, M., MILLER, J., PENG, S., WANG, T., ET AL. A two-fold increase of carbon cycle sensitivity to tropical temperature variations. *Nature* 506, 7487 (2014), 212–215.
- [148] WANG, X., ZHOU, Z., XIAO, F., XING, K., YANG, Z., LIU, Y., AND PENG, C. Spatio-temporal analysis and prediction of cellular traffic in metropolis. *IEEE Transactions on Mobile Computing* (2018), 1–1.
- [149] WANG, Y., YANG, J., CHEN, Y., DE MAEYER, P., LI, Z., AND DUAN, W. Detecting the causal effect of soil moisture on precipitation using convergent cross mapping. *Scientific reports* 8, 1 (2018), 1–8.
- [150] WANG, Y., YAO, Q., KWOK, J. T., AND NI, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)* 53, 3 (2020), 1–34.
- [151] WEI, Z., GUO, W., LI, B., CHARMET, J., AND ZHAO, C. High-dimensional metric combining for non-coherent molecular signal detection. *IEEE Transactions on Communications* 68, 3 (2020), 1479–1493.
- [152] WEI, Z., LI, B., GUO, W., HU, W., AND ZHAO, C. On the accuracy and efficiency of sensing and localization for robotic. *IEEE Transactions on Mobile Computing* (2020), 1–1.
- [153] WEI, Z., LI, B., SUN, C., AND GUO, W. Sampling and inference of networked dynamics using log-koopman nonlinear graph fourier transform. *IEEE Transactions on Signal Processing* 68 (2020), 6187–6197.

- [154] WEISS, K., KHOSHGOFTAAR, T. M., AND WANG, D. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.
- [155] WEN, D., JIA, P., HSU, S.-H., ZHOU, Y., LAN, X., CUI, D., LI, G., YIN, S., AND WANG, L. Estimating coupling strength between multivariate neural series with multivariate permutation conditional mutual information. *Neural Networks* 110 (2019), 159–169.
- [156] WILLIAMS, C. K. Computing with infinite networks. In *Advances in neural information processing systems* (1997), pp. 295–301.
- [157] WILLIAMS, C. K., AND RASMUSSEN, C. E. *Gaussian Processes for Machine Learning*, vol. 2. MIT press Cambridge, MA, 2006.
- [158] WILSON, A. G. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge, 2014.
- [159] WILSON, A. G. *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge, 2014.
- [160] WU, K.-P., AND WANG, S.-D. Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. *Pattern Recognition* 42, 5 (2009), 710–717.
- [161] XIANG, L., GE, X., LIU, C., SHU, L., AND WANG, C. A new hybrid network traffic prediction method. In *IEEE Global Telecommunications Conference (GLOBECOM)* (Dec 2010), pp. 1–5.
- [162] XIAO, S., YAN, J., YANG, X., ZHA, H., AND CHU, S. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2017), vol. 31.

- [163] XU, F., LIN, Y., HUANG, J., WU, D., SHI, H., SONG, J., AND LI, Y. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE Transactions on Services Computing* 9, 5 (Sep. 2016), 796–805.
- [164] XU, Y., XU, W., YIN, F., LIN, J., AND CUI, S. High-accuracy wireless traffic prediction: A gp-based machine learning approach. In *GLOBECOM 2017-2017 IEEE Global Communications Conference* (2017), IEEE, pp. 1–6.
- [165] XU, Y., XU, W., YIN, F., LIN, J., AND CUI, S. High-accuracy wireless traffic prediction: A GP-based machine learning approach. In *IEEE Global Communications Conference (GLOBECOM)* (Dec 2017), pp. 1–6.
- [166] YANG, B., GUO, W., CHEN, B., YANG, G., AND ZHANG, J. Estimating mobile traffic demand using Twitter. *IEEE Wireless Communications Letters* 5, 4 (Aug 2016), 380–383.
- [167] YAO, X., AND LIU, Y. Towards designing artificial neural networks by evolution. *Applied Mathematics and Computation* 91, 1 (1998), 83–90.
- [168] YU, W., YANG, K., BAI, Y., XIAO, T., YAO, H., AND RUI, Y. Visualizing and comparing alexnet and vgg using deconvolutional layers. In *Proceedings of the 33rd International Conference on Machine Learning* (2016).
- [169] YUAN, X., HE, P., ZHU, Q., AND LI, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems* 30, 9 (2019), 2805–2824.
- [170] ZAI EL AMRI, W., REINHART, F., AND SCHENCK, W. Open set task augmentation facilitates generalization of deep neural networks trained on small data sets. *Neural Computing and Applications* 34, 8 (2022), 6067–6083.

- [171] ZAMMIT-MANGION, A., DEWAR, M., KADIRKAMANATHAN, V., AND SAN-GUINETTI, G. Point process modelling of the afghan war diary. *Proceedings of the National Academy of Sciences* 109, 31 (2012), 12414–12419.
- [172] ZHANG, B.-T., AND MUHLENBEIN, H. Evolving optimal neural networks using genetic algorithms with occam’s razor. *Complex systems* 7, 3 (1993), 199–220.
- [173] ZHANG, K., CHUAI, G., GAO, W., LIU, X., MAIMAITI, S., AND SI, Z. A new method for traffic forecasting in urban wireless communication network. *EURASIP Journal on Wireless Communications and Networking* 2019, 1 (Mar 2019), 66.
- [174] ZHANG, M., AND CHEN, M. Power allocation in multi-cell system using distributed deep neural network algorithm. In *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (Oct 2019), pp. 1–4.
- [175] ZHANG, X., ZHAO, Y., GUO, K., LI, G., AND DENG, N. An adaptive b-spline neural network and its application in terminal sliding mode control for a mobile satcom antenna inertially stabilized platform. *Sensors* 17, 5 (2017), 978.
- [176] ZHANG, Z., QIN, H., LIU, Y., WANG, Y., YAO, L., LI, Q., LI, J., AND PEI, S. Long short-term memory network based on neighborhood gates for processing complex causality in wind speed prediction. *Energy Conversion and Management* 192 (2019), 37–51.
- [177] ZHAO, L., WANG, Y., AND CHARGÉ, P. Efficient iterative water-filling power allocation method in MU-MIMO broadcast channels. In *Military Communications and Information Systems Conference* (Oct 2013), pp. 1–5.
- [178] ZHU, L., YU, F. R., WANG, Y., NING, B., AND TANG, T. Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 20, 1 (2019), 383–398.