Universiteit Leiden

Computer Science

Modelling Fish Occurrence in the Dutch Wadden Sea by using Bayesian Network Structure Learning Algorithm

Name: Rui Zhang

Student no: s1959662

Date: 15/04/2019

Company supervisor: Dr. Ir. Ghada El Serafy (Deltares)

1st university supervisor: Dr. Wojtek Kowalczyk (LIACS)

2nd university supervisor: Dr. Hao Wang (LIACS)

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)

Leiden University

Niels Bohrweg 1

2333 CA Leiden

The Netherlands

Abstract

As an essential component in the aquatic ecosystem, fish brings both economic and ecological benefits to the location where it occurs. The Dutch Wadden Sea is one of the largest intertidal area in the world, and its high biological diversity nourishes a wide variety of fish species that are specially adapted to this area. Modeling the fish occurrence in the Dutch Wadden Sea is meaningful for fish species conservation management. The occurrence of fish species is usually affected by the multiple influencial factors like climate change, nutrient dynamics, increased predation pressure, habitat deterioration and fisheries in isolation or cumulatively. The Bayesian Network (BN) model is ideal for modelling the influence of multiple factors within an uncertain domain. In this project, the BN models of fish occurrence in the Dutch Wadden Sea were automatically constructed by five different BN structure learning algorithms in GeNIe modeler, mainly based on in-situ measurement data. To make the learned BN structure more reasonable for domain experts, ecological domian knowledge was also involved in the BN structure learning process as several qualitive constrains. The experiment results show that the BN models can be effectively and reasonably constructed by the proposed methodology, and the learned models can also make proper predictions given evidence. The results also illustrate that the involvement of domain knowledge in the structure learning process improved the performance of original BN structure learning algorithms in different aspects. The results indicate that the BN structure learning algorithms are powerful and time-saving tools for the fish occurrence modelling in the Dutch Wadden Sea with the aid of domain knowledge. Thus, the proposed methodology could gather expertise and experience of both data scientist and ecologist together to improve the performance of the BN models.

Key words: Fish Occurrence, the Dutch Wadden Sea, Bayesian Network, Machine Learning, GeNIe modeler.

Acknowledgements

This thesis is the result of my master thesis project in Computer Science institute of Leiden University (LIACS). This project is also a part of my data science internship in Deltares from September 2018 to July 2019. During the internship, I had a very memorable time in Deltares to complete my graduation project. Hereby, I would like to thank several persons that offer me help and guidance during this master thesis project.

Firstly, I would like to thank Dr. Ir. Ghada El Serafy, who is my main supervisor in Deltares. It was she who provided me such an interestion topic for my graduation project and her appreciation inspired me to put more effort into this project. I also want to express my sincere gratitude to other researchers in her team, including Sonja Wanke and Anna Spinosa who gave me many valuable suggestions on the thesis writing, Alex Ziemba and Lorinc Meszaros who offered helpful advice when I was finishing the project. As I'm an amateur in the domain of ecology, their ecological knowledge gives me another point of view to refine my project.

Secondly, I also would like to thank my first and second supervisors in Leiden University, Dr. Wojtek Kowalczyk and Dr. Hao Wang. They gave me very helpful reading materials that provided me basic knowledge and indepth insight on Bayesian Network. Without their suggestions, this thesis cannot be as comprehensive as how it looks now.

Finally, thanks to the finantial support of my parents and Leiden Science China (LSC) scholarship, I can have the chance to study in the Leiden University.

Contents

List of Acronyms

BN	Bayesian Network
DAG	Directed Acyclic Graph
SVM	Support Vector Machine
ANN	Artificial Neural Network
JPD	Joint Probability Distribution
CPD	Conditional Probability Distribution
CPT	Conditional Probability Distribution Table
QSR	Wadden Sea Quality Status Report
DFS	The Dutch Demersal Fish Survey
NIOZ	Koninklijk Nederlands Instituut voor Onderzoek der Zee (Royal Netherlands Institute for Sea Research)
IMARES	Institute for Marine Resources & Ecosystem Studies
AWI	Alfred-Wegener-Institut
Chl-a	Chlorophyll-a
N/P ratio	The Nitrogen/Phosphorus ratio
TAN	Tree-Augmented Naïve Bayes
ANB	Augmented Naïve Bayes
PC	A Bayesian Network learning algorithm named after the abbrievation of the first names of its creators (P. Spirtes, C. Glymour)
SGS	A Bayesian Network learning algorithm named after the abbrievation of the last names of its creators (P. Spirtes, C. Glymour and R. Scheines)
EM	Expectation-Maximization

Table of Figures

List of Tables

1. Introduction

Nowadays, fish is one of the primary sources of animal protein for more than one billion people in the developing world, and millions of people depend on fisheries and aquaculture value chains such as processing or marketing. However, there still are major shortages on fish supplies in poor countries. Thus, improving the productivity and sustainability of fisheries and aquaculture is crucial to reducing hunger and poverty and the knowledge of the impact on fish occurrence can be quite helpful. On the other hand, as an essential component of the aquatic ecosystem, the occurrence of fish species also indicates the status of ecological balance in a specific area. Modelling the impact of different factors on fish occurrence provides people insight on how to develop a sustainable fishery approach that will help to protect the natural resources.

The study area, the Dutch Wadden Sea, lies between the Marsdiep close to Den Helder and the Dollard in Groningen. The region forms the transition between the estuaries and the North Sea and is rich in fish species specially adapted to the demanding environmental conditions [1]. Many fish species, including marine, estuarine and diadromous species, rely on the Wadden Sea for at least one of their life stages, benefitting from the high food availability and shelter from predators in the Dutch Wadden Sea [2].

As a consequence of global climate change and human activities such as industry, tourism and fishery, the impacts on marine environments are increasing dramatically and affect the properties of the ecosystem of the Dutch Wadden Sea in isolation or cumulatively [3]. In this case, the Bayesian Network is an ideal tool for assessing those impacts on fish occurrence because it is suitable for modelling the influence of multiple factors within an uncertain domain.

A Bayesian Network (BN) represents probabilistic relationships among multiple random variables via nodes and edges in a Directed Acyclic Graph (DAG). Unlike other machine learning algorithms such as Support Vector Machine (SVM) and Artificial Neural Networks (ANN), the structure of a Bayesian Network can be intuitively understood and interpreted into domain knowledge. Building a BN model for fish occurrence could benefit fish species conservation management by simulating different scenarios that help managers foresee the consequences before taking actions [4]. The intuitively understandable structure also enhances the communication between experts and non-experts.

In addition, the knowledge of domain experts sometimes could help enhance the performance of a model. However, most of data-oriented machine learning models are usually considered as a “black box” for experts of other domains because of the abstractive structure of those models. That means people can merely tune a few parameters, input the training data and wait for the results. Without knowing the principles of those models, it is difficult for experts to apply their knowledge to improve the reliability and accuracy of them. As the

structure of BN models can be easily interpreted into domain knowledge, it is also easy to modify the BN structure according to the experts' domain knowledge. Thus, this characteristic of BN models provides the opportunity to get experts' knowledge involved in its building process.

The building of BNs normally can be executed in two different ways. The first one is to build the BNs by hand based on experts' opinions. This is the most common way to build a BN model becausethe correctness of the model can be guaranteed under the guidance of domain experts. However, the acquisition of experts' knowledge sometimes can be quite time-consuming and in some domains, there are simply no sufficient knowledge of the domain. The second way to build a BN is to make use of graph theory and statistics knowledge and learn the structure and parameters of the BN from training data. In this case, experts' knowledge is no longer needed, which makes the building process more objective and efficient,but also unpredictable because of data uncertainty and unexpected behaviour of learning algorithms. In this project, a methodology considering both data and experts' knowledge was proposed to build the BN models.

This project aims to develop BNs trained from both data and experts' knowledge for modelling causal relationship among fish occurrence and other influencing factors. According to [2], the factors like climate change, nutrient dynamics, increased predation pressure, habitat deterioration and fisheries are often mentioned to be the potential drivers to the changing of fish occurrence. In this project, nutrient concentration, climate change, fisheries, tourism and other fish occurrence were considered as influencing factors when training the Bayesian network model.

Usually, there are two reasons for using a BN structure learning algorithm: density estimation and knowledge discovery [5]. Therefore, the main objective can be divided into two sub-objectives. For density estimation, the output model is expected to accurately predict the probability distribution of the number of fish occurrence given some evidence. For knowledge discovery, the goal is to extract some useful domain knowledge out of data by examining the structure of the learned network. For different sub-objectives, different metrics were used to evaluate the performance of BN structure learning algorithms.

The remaining part of this paper is organized as follows: Chapter 2 gives an introduction of basic concepts that were utilized in this project. The description of all the collected datasets for this project is given in Chapter 3. In Chapter 4, the methodology that combines in-situ measurement data and domain knowledge to train the BN models of fish occurrence in the Dutch Wadden Sea will be proposed and comprehensively explained. The results that illustrate the performance of each BN structure learning algorithm will be shown in Chapter 5. Chapter 6 gives extra discussion on how the BN structure learning algorithms perform in modelling fish occurrence in the Dutch Wadden Sea. Chapter 7 provides future perspective for the BN structure learning algorithms and finally the thesis will be concluded in Chapter 8.

2. Background

In this Chapter, several major concepts in this project will be introduced, including some basics of probability theory (Section 2.1), Bayes' Theorem and its application (Section 2.2), the basic concepts of Bayesian Network (Section 2.3) and the learning of Bayesian Network (Section 2.4), the description of the study area (Section 2.6) and the introduction and comment on the previous similar work in this area (Section 2.7).

2.1. Basics of Probability Theory

In probability theory, one of the key concepts is the random variable, which is a function that associates with each possible outcome in event space Ω a numerical quantity (typically a real number) that denotes the probability of the outcomes occurring as following [6]:

$$P(A = \alpha) = 0.1 \tag{2.1}$$

Here A is a random variable and α is one of the possible outcomes of this random variable. This formula means that the event α has 10% chance to happen among all the possible events of A. Those chances of all the different possible outcomes in a random variable are also called the probability distribution of the random variable. A probability distribution for a single random variable always satisfies the following conditions:

$$P(A = \alpha) > 0, for\ all\ \alpha \in \Omega \tag{2.2}$$

$$\sum_{\alpha \in \Omega} P(A = \alpha) = 1 \tag{2.3}$$

When considering the outcomes of multiple random variables, there are two ways to denote the possibility of multiple events occurring simultaneously. The joint probability distribution (JPD) of multiple random variables gives the probability that each random variable falls in any particular range or discrete set of values specified for that variable [7]. For example:

$$P(A = \alpha, B = \beta) = 0.3 \tag{2.4}$$

The formula above denotes that there is a 30% chance that event α and event β occur simultaneously. In this case, the event space Ω of a joint probability distribution contains all the possible combination of events that we are willing to assign probabilities. A JPD always satisfies the following conditions:

$$P(e) > 0, for\ all\ e \in \Omega \tag{2.5}$$

$$\sum_{e \in \Omega} P(e) = 1 \tag{2.6}$$

$$P(A = \alpha, B = \beta) = P(A = \alpha) \cdot P(B = \beta), if\ A\ and\ B\ are\ independent \tag{2.7}$$

The conditional probability distribution (CPD) gives the probability of an event (or a set of events) occurring on the condition of the other events happening. For example, given two random variables A and B:

$$P(A = \alpha \mid B = \beta) = 0.2 \tag{2.8}$$

Which means that the event α happens with 20% probability under the condition of the event β occurring. Formally, the definition of the conditional probability of α given β is shown by the following formula:

$$P(A = \alpha \mid B = \beta) = \frac{P(A=\alpha, B=\beta)}{P(B=\beta)} \tag{2.9}$$

Equation (2.9) expresses the relationship between JPD and CPD. In addition, the CPD also satisfies the following conditions:

$$P(e \mid c) > 0, for\ all\ e \in \Omega_e\ and\ c \in \Omega_c \tag{2.10}$$

$$\sum_{e \in \Omega_e} P(e \mid c) = 1 \tag{2.11}$$

2.2. Bayes' Theorem and Naïve Bayes

Through the definition of CPD in Equation (2.9), we can derive the chain rule of conditional probability as follows:

$$P(A = \alpha, B = \beta) = P(A = \alpha \mid B = \beta) \cdot P(B = \beta) = P(B = \beta \mid A = \alpha) \cdot P(A = \alpha) \tag{2.12}$$

Then, the formula of Bayes' theorem (Bayes' law or Bayes' rule) can be derived from Equation 2.12:

$$P(A = \alpha \mid B = \beta) = \frac{P(B=\beta \mid A=\alpha) \cdot P(A=\alpha)}{P(B=\beta)} \tag{2.13}$$

The Bayes' theorem shows a way to compute the conditional probability $P(A = \alpha \mid B = \beta)$ from its inversed version $P(B = \beta \mid A = \alpha)$. In Equation (2.13), $P(A = \alpha \mid B = \beta)$ is the posterior probability of hypothesis $A = \alpha$ given the evidence $B = \beta$. $P(A = \alpha)$ and $P(B = \beta)$ are called prior probabilities of the hypothesis and the evidence. $P(B = \beta \mid A = \alpha)$ is the probability of the evidence $B = \beta$ given hypothesis $A = \alpha$.

One of the most important applications of Bayes' theorem is the Naïve Bayes classifier, which was first introduced into the text retrieval community in the early 1960s [8]. The Naïve Bayes classifier works by calculating the conditional probability distribution of a target variable, given a set of attributes, then assigning the target variable to the class with the highest probability. All the Naïve Bayes classifiers are based on a strong independence assumption: all the attribute variables are independent of each other, given the target variable. From the graphical expression in Figure 1, all the attribute nodes provide evidence for the classification of a target node, but there is no causal relationship among attribute nodes.

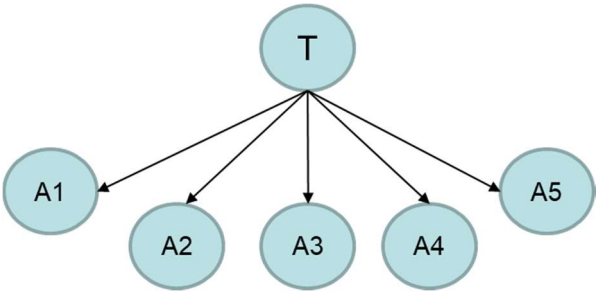


Figure 1: Graphical expression of Naïve Bayes classifier. Edge: causal relationship, T: target node (target variable), A1...A5: attribute nodes (attribute variables).

2.3. Bayesian Network

In reality, the strong independence assumption of the Naïve Bayes model is usually not satisfied. Therefore, the Bayesian Network was invented to model the probalistic relationships that violate the independence assumption, which provides a more flexible way to represent the dependency among random variables. In this case, Naïve Bayes model can be considered as a special form of Bayesian Network.

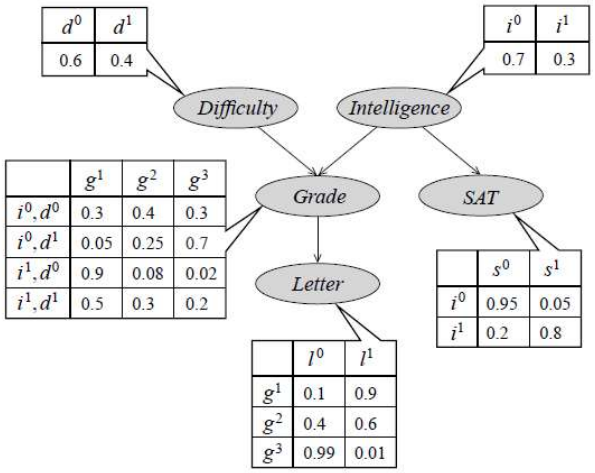


Figure 2: An example Bayesian Network: Student [5]

A Bayesian Network (BN), also known as Bayesian Belief –Network, decision network, Bayes model is one of the probabilistic graphical models that represent knowledge under uncertainty. BNs combine principals from graph theory, probability theory, computer science and statistics, and are popular in the statistics, the machine learning and the artificial intelligence domain [9]. Figure 2 shows a simple BN that demonstrates the probabilistic relationships among a student's intelligence (Intelligence), the student's SAT (a standardized test used for college admissions in the United States) score (SAT), the difficulty of the course (Difficulty), the grade that the student had in the school (Grade) and the chance of the student getting the reference letter from teachers (Letter). According to Figure 2, the BN model represents those probabilistic relationships via a Directed Acyclic Graph (DAG), whose nodes represent random variables, edges correspond to a causal relationship between variables. The nodes and edges are the structure of a BN. In addition, each node is

assigned a conditional probability distribution table (CPT) representing the detail of the probabilistic relationship among the node and its parent nodes numerically. The CPTs are also called the parameters of the BN model.

The BN model makes prediction just like Naïve Bayes model, the probability distribution of target variable nodes can be calculated given a set of evidences (features). For example, in Figure 2, given the evidence that the student's SAT score (SAT) is known as good ($S = s^1$), the conditional probability distribution of the target variable node, student's intelligence (Intelligence), can be calculated by using Bayes' theorem in Equation (2.13):

$$\begin{cases} P(I = i^0 \mid S = s^1) = P(S = s^1 \mid I = i^0) * P(I = i^0)/P(S = s^1) = 0.035/P(S = s^1) \\ P(I = i^1 \mid S = s^1) = P(S = s^1 \mid I = i^1) * P(I = i^1)/P(S = s^1) = 0.24/P(S = s^1) \\ P(I = i^0 \mid S = s^1) + P(I = i^1 \mid S = s^1) = 1 \end{cases}$$

By solving the equation set above, finally we have:

$$\begin{cases} P(I = i^0 \mid S = s^1) = 0.13 \\ P(I = i^1 \mid S = s^1) = 0.87 \end{cases}$$

The result shows that the student is more likely (87% chance) to be one of high intelligence ($I = i^1$), given the student had a good SAT score. The process shown above is also called inference of Bayesian Network.

The BNs can not only model the direct dependencies among random variables, but also indirect ones. From the student example in Figure 2, four kinds of triplet structures can be found in the BN:

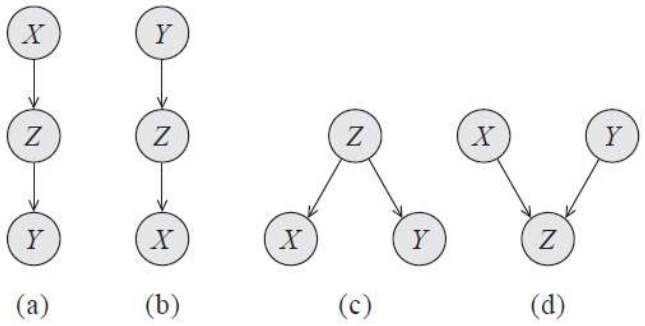


Figure 3: Four possible triplet structures from X to Y via Z: (a)Casual Trial, (b)Evidential Trial, (c)Common Effect, (d)Common Cause (v-structure)

1. Casual Trial (Difficulty → Grade → Letter): In this example, let us assume that the teacher gives the reference letter only relies on the grade that the student has. If the student chooses a harder course, it will be more difficult for him/her to get a good grade. Therefore, if the student's grade is not known, then the difficulty

of the course can influence the chance of getting the reference letter by changing the chance of the student having a good grade in that course. However, if the grade is known, the teacher has decided whether to give a reference letter or not, so the course difficulty can no longer influence the decision. In conclusion, on this trail the random variable "Difficulty" can influence "Letter" if and only if "Grade" is not observed. Formally, the conditional dependency can be expressed as: $(\boldsymbol{Difficulty} \perp \boldsymbol{Letter} \mid \boldsymbol{Grade})$.

2. Evidential Trial (Letter ← Grade ← Intelligence): This trail is the reversed version of the previous one. According to this trial, if the student's grade is unknown, the reference letter could provide some evidence for the student's intelligence. However, if the student's grade is already known, the reference letter can no longer provide any useful information for it and only relies on the student's grade. On this trail, "Letter" can influence "Intelligence" if and only if "Grade" is not observed, whose conditional dependency can be written as: $(\boldsymbol{Intelligence} \perp \boldsymbol{Letter} \mid \boldsymbol{Grade})$.

3. Common Effect (Grade ← Intelligence → SAT): In the student example, the students' intelligence can affect both their grades in the school and their SAT scores. In this case, both "Grade" and "SAT" provide evidence for the students' intelligence. If the intelligence is not given, the grade of the student in the school can influence the "belief" (i.e. probability distribution) of the student's SAT score by changing our "belief" of the student's intelligence, and vice versa. On the other hand, if the intelligence is known, the grade and SAT score cannot have any impact on each other for they both have already decided by their direct parent. In conclusion, "Grade" and "SAT" can influence each other if and only if "intelligence" is not observed, whose conditional dependency can be written as: $(\boldsymbol{Grade} \perp \boldsymbol{SAT} \mid \boldsymbol{Intelligence})$.

4. Common Cause (Difficulty → Grade ← Intelligence): The last triplet structure is different from the other three. In this example, the course difficulty and the students' intelligence can both influence the grades of the students. If the students' grades are not given, the course difficulty has nothing to do with the intelligence of those students. Because even a student chooses a difficult course, the student can have a bad grade on this course if he/she is not smart enough. However, if the grade of the student is given, it builds a reliable connection between the course difficulty and the student's intelligence. For example, the student who gets a good grade on a difficult course is more likely to be one of high intelligence. Therefore, in this structure, "Difficulty" and "Intelligence" can influence each other if and only if "Grade" is observed (or one of the descendants of "Grade" is observed), which can be written as: $(\boldsymbol{Difficulty} \perp \boldsymbol{Intelligence} \mid \boldsymbol{W}, \boldsymbol{Grade} \notin \boldsymbol{W})$. Apparently, the conditional dependency of this structure is different from that of others. In practice, it is necessary to pay more attention to this structure. So, it is given another name: v-structure.

The conditional dependency of each triplet structure can not only be explained intuitively by the specific example but also be formally proved with the help of axioms of probability theory. According to the Equation

(2.7), if two random variables A and B are independent, then their joint distribution can be written as: $P(A,B) = P(A) \cdot P(B)$.

For structure (a) in Figure 3, the joint distribution for X Y Z can be factorized as: $P(X,Y,Z) = P(X) \cdot P(Y \mid Z) \cdot P(Z \mid X)$. Then $P(X,Y) = \sum_Z P(X,Y,Z) = \sum_Z P(X) \cdot P(Y \mid Z) \cdot P(Z \mid X)$. According to the Equation (2.11), $\sum_Z P(Z \mid X) = 1$. Therefore, $P(X,Y) = P(X) \cdot P(Y \mid Z)$, which means that X and Y are not independent when Z is not given, corresponding to the conditional dependency ($\boldsymbol{X \perp Y \mid Z}$)**.** Moreover, the structure (b) is the same structure as (a), so the proof and conclusion are also identical.

The joint distribution of structure (c) can be factorized as: $P(X,Y,Z) = P(X \mid Z) \cdot P(Y \mid Z) \cdot P(Z)$. Then $P(X,Y) = \sum_Z P(X,Y,Z) = P(X \mid Z) \cdot P(Y \mid Z)$, which leads to the same conclusion above. Therefore, the first 3 structures share the same type of conditional dependency.

The joint distribution of structure (d) can be factorized as: $P(X,Y,Z) = P(X) \cdot P(Y) \cdot P(Z \mid X,Y)$. Then $P(X,Y) = \sum_Z P(X,Y,Z) = P(X) \cdot P(Y)$, which shows that X and Y are independent when Z is not given, corresponding to the conditional dependency ($\boldsymbol{X \perp Y \mid W, Z \notin W}$)**.**

2.4. Parameter Learning and Structure Learning for Bayesian Network

The next question is how to acquire a BN model. The most common way is to manually construct the network with the help of experts in the domain. That means the constructors estabilish edges and set the parameters (CPT) for BNs completely in accordiance with experts' opinions and knowledge, which guarantees the constructed BN model reasonable from the perspective of experts. However, as every coin has two sides, this subjective construction method may result a dilemma as experts canhave different opinions on the same concept sometimes. Moreover, such a method requires a large amount of experts' knowledge when building a massive network, and to acquire such amount of knowledge will not be a trivial task.

As the development of information technology, the acquisition of large amounts of data generated from the distribution we wish to model is often easier than that of human expertise. Therefore, a number of algorithms have been invented to learn BN models from training data, which avoids spending much time on consulting with domain experts and makes the BN construction process more efficient consequently. The model built in this way can objectively reflect the real situation if the data are processed properly. However, as the model is not directly derived from domain knowledge, unexpected results might be found in the network learned from data. The latest means that this method is able to discover new domain knowledge out of data, nevertheless the unexpected outcomes are not always welcome. In the trained networks, there might be some components that are inconsistent with people's intuition or experts' knowledge. The problem might be caused by two reasons: Firstly, because of the measurement error or malfunction of sensors, there is always some unexpected noise in the training dataset and too much noise will lead the learning process to a wrong way; Secondly,

different BN learning algorithms have their own standard to determine whether a pair of random variables are strongly related, and the judgement of the learning algorithms might also be different with human experts. The unpredictable output is the major problem of data-based construction method.

The Bayesian Network learning includes two parts: structure learning and parameter learning. In this project, the research mainly focuses on BN structure learning algorithms, but parameter learning is still a necessary step to build the complete BN model. The task of BN structure learning can be described as following: Assume that a set of data $\boldsymbol{D}$ is generated from a joint probability distribution $P^*(\chi)$ represented by some Bayesian network $\boldsymbol{G}^*$, the task is to find the best network $\boldsymbol{G}$ that resembles $\boldsymbol{G}^*$ from $\boldsymbol{D}$. In parameter learning, the network structure $\boldsymbol{G}$ is fixed, and the task is to estimate the CPTs in the Bayesian network that fit $\boldsymbol{D}$ the best.

2.5. Graphical Network Interface: GeNIe Modeller

Nowadays, various computer software tools are available to help build the probabilistic graphical models (including the BN model). In practice, people need to choose the most suitable one for their purpose. In this project, three of most commonly used software tools with the user interface, GeNIe modeller [10], Netica [11] and Hugin Expert [12], were selected as the candidate software tools. A general overview of those tools is shown in Table 1.

Table 1: General Comparison of three graphical network building tools with the user interface: GeNIe modeller, Netica and Hugin Expert

	GeNIe	**Netica**	**Hugin**
Latest Version	2.3 (Jan. 2019)	6.05 (Jun. 2018)	8.5 (May. 2017)
Built-in BN structure learning algorithms	1.Bayesian Search (K2) * 2.PC* 3.Greedy Thick Thinning* 4.Tree-Augmented Naïve Bayes (TAN) 5.Augmented Naïve Bayes 6.Naïve Bayes *Background Knowledge supported	1.Tree-Augmented Naïve Bayes (TAN)	1. PC 2. NPC 3. Greedy search-and-score algorithm 4. Chow-Liu tree 5. Rebane-Pearl polytree 6. Tree-Augmented Naïve Bayes (TAN)
Built-in BN parameter learning algorithms	1. EM	1. Counting 2. Expectation-Maximization (EM) 3. Gradient descent	1. Adaptation 2. EM 3. EM OOBN

Inference Algorithm	1. Clustering 2. Relevance-based decomposition 3. Polytree algorithm 4. Probabilistic Logic Sampling 5. Likelihood Sampling 6. Backward Sampling 7. AIS algorithm	1. Junction tree	1. Junction tree
Dynamic Bayesian Network support	Yes	Yes	Yes
API support	Java, C++, Python, .NET	Java, C, C++, C#, VB, Python	Java, COM, C++, .NET, Web Service, Python, ActiveX
Free version	Free to academic users (Reference required)	Limited in model size (no more than 15 nodes)	No Free Version

According to the comparison in Table 1, GeNIe modeller is the best choice to achieve the goal of this project. Firstly, six different BN structure learning algorithms are supported by GeNIe modeller while only one is supported by Netica. Although Hugin Expert also contains six structure learning algorithms, unfortunately, the free version of this software was not found that makes Hugin Expert not suitable for academic research.

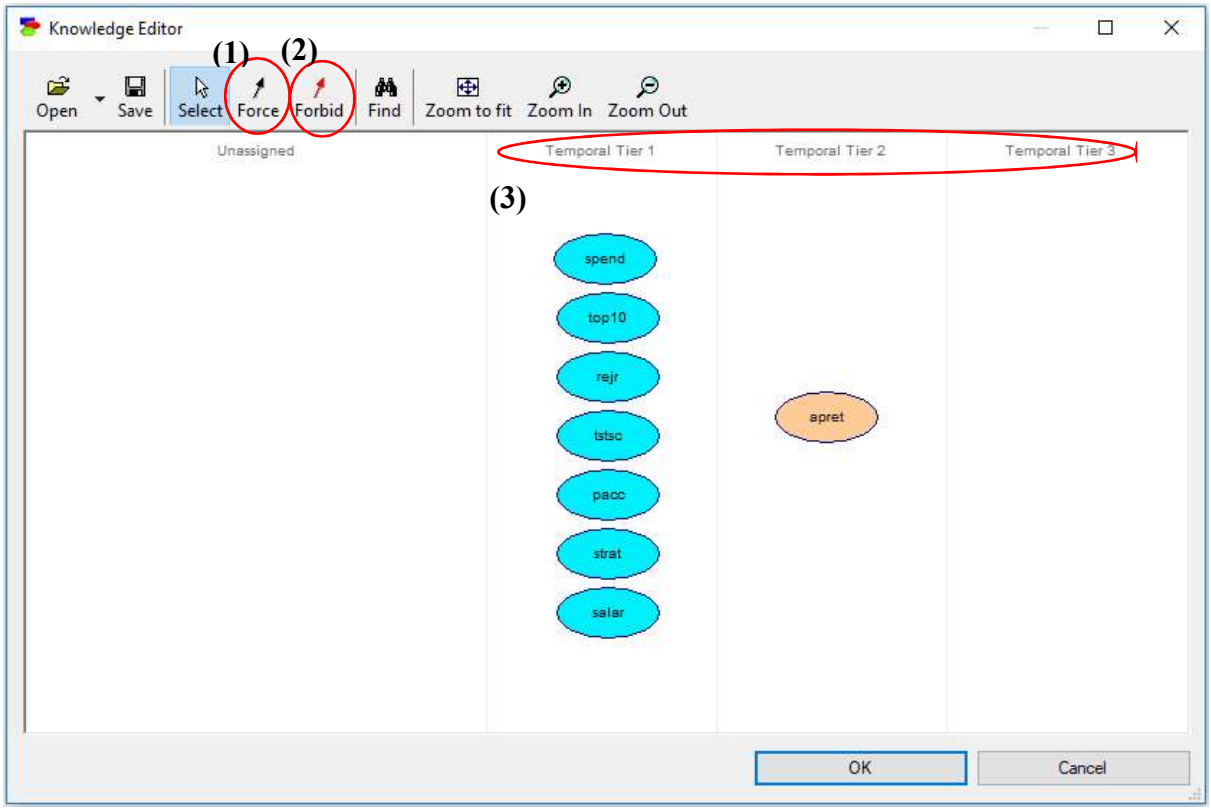


Figure 4: Background knowledge editor panel in GeNIe modeller, which shows three different types of background knowledge supported in GeNIe: (1) Force Arcs, (2) Forbid Arcs, (3) Temporal Tiers.

Another significant advantage of GeNIe modeller over other tools is that it supports using background knowledge to adjust the output BN model learned from data. By using this function of GeNIe, the goal of combining data and experts' knowledge to train a BN model can be easily achieved. Figure 4 shows how people input background knowledge to GeNIe modeller when learning the BN model from data. The interface of the knowledge editor panel shows that there are three kinds of background knowledge that can be read by GeNIe:

1. **Force Arcs:** these arcs are guaranteed to appear in the learned structure (unless those arcs cause a cycle).

2. **Forbid arcs:** these arcs are guaranteed to be absent in the learned structure.

3. **Temporal Tiers:** there will be no arcs from variables that occur later (in higher tiers) to nodes happening earlier (in lower tiers).

2.6. Study Area: The Dutch Wadden Sea

The Wadden Sea is a natural area between Den Helder in the Netherlands and Esbjerg in Denmark, which has been entitled "UNESCO World Heritage Site" since 2009 as the largest unbroken system of intertidal sand and mud flats in the world. The formation of the Wadden Sea started from 8000 BC when the lower parts of Pleistocene valleys of rivers (e.g. Elbe, Weser and Ems) were changed into tidal basins. Around 2000 BC, the sediment infill of the basins started to keep up with the decreasing sea level rise speed. As a consequence, the salt-marsh areas and coastal peat marshes increased in size at that time. Nowadays, with the systematic construction of modern dykes, the form of the Wadden Sea has become relatively stable. The North Sea water brings mud into the Wadden Sea with tides. The mud is mainly originated from several rivers or the erosion of the Canal Coast. The tidal flat and wetlands in the Wadden Sea create a high biological diversity, so the Wadden Sea is rich in species specially adapted to the demanding environmental conditions.

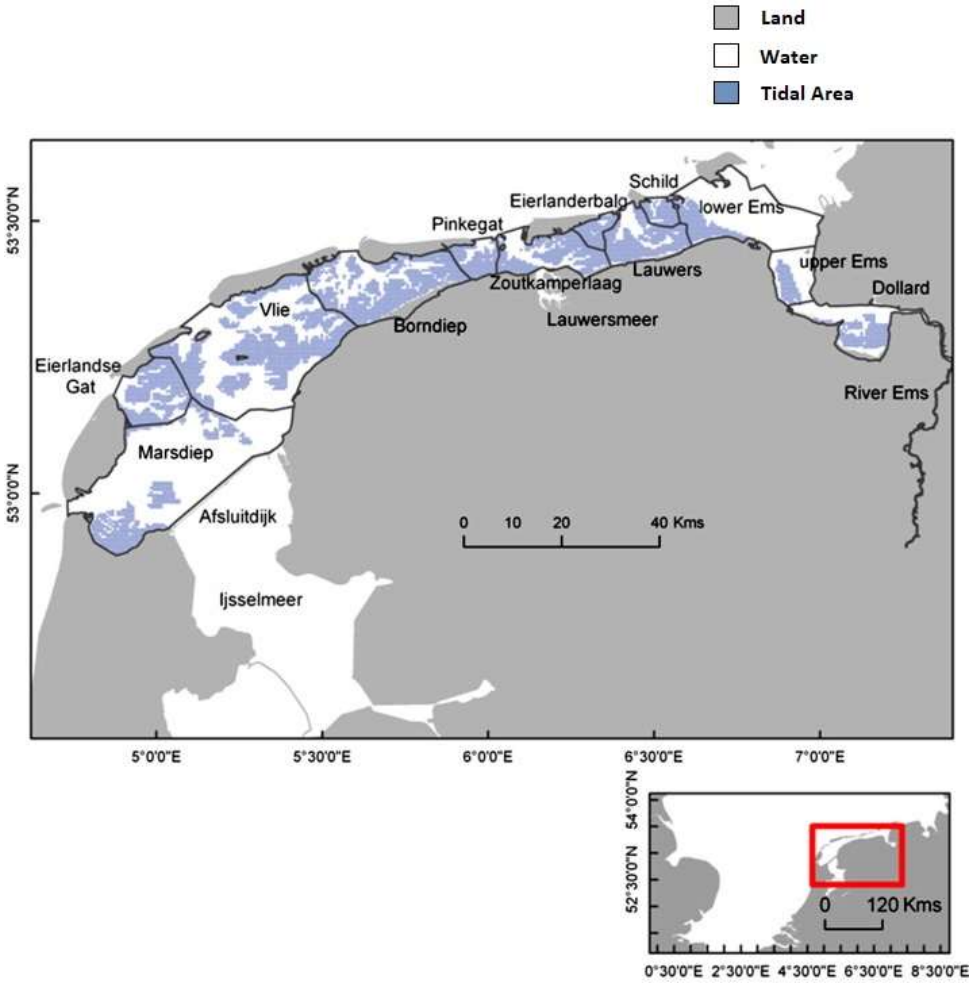


Figure 5 [13]: Case Study area: the Dutch Wadden Sea. The picture also shows in blue the distribution of tidal area in the Dutch Wadden Sea.

The Dutch Wadden Sea (see Figure 5) lies between the Marsdiep by Den Helder and the Dollard in Groningen. It covers a total surface area of 2,550 km². The region contains seven islands called West Frislan Islands, in which five islands are inhabited, and two are uninhabited, which decline in size moving from west to east. The beauty and diversity of those islands make this part of the Wadden Sea World Heritage a popular tourist attraction in the Netherlands. Those islands are also considered to be the barrier islands of the Dutch Wadden Sea that protect the coastline from storm damage and shelter habitats that are refuges for wildlife [14]. This area also forms the transition between estuaries and the marine environment. Therefore, both marine and estuaries fish species can be found in the study area. According to [2], the fish species of the Dutch Wadden Sea can be divided into three groups based on their habits:

1. Marine juveniles: Those marine fish species spend their juvenile stage in the Dutch Wadden Sea, benefitting from the high food availability and the lack of predators there, such as flatfish and other groundfish [15].

2. Estuarine residents: Those estuarine fish species spend almost the entire life stages in the Dutch Wadden Sea.

3. Diadromous: Those fish species inhabit the area as a route to either marine or freshwater spawning sites [16].

Figure 6: Fishes caught by a beam trawl in the western Dutch Wadden Sea (Photo: Ingrid Tulp, IMARES).

Currently, various fish monitoring programmes have been conducted to observe the status and trend of fish in the entire Wadden Sea (Figure 6). Table 2 shows information about those programmes provided by Wadden Sea Quality Status Report (QSR) 2016. From this table, the The Dutch Demersal Fish Survey (DFS), Royal Netherlands Institute for Sea Research (NIOZ) and Institute for Marine Resources & Ecosystem Studies (IMARES) programmes focus on the Dutch Wadden Sea area. Among those three programmes, the DFS covers the entire Dutch Wadden Sea, but it only collected data in September, which makes it impossible to observe the seasonality of fish occurrence through the data of this programme. The NOIZ and IMARES surveyed around the spring and autumn each year, and the duration of NOIZ programme is much longer than that of IMARES. Therefore, the data of NOIZ fish monitoring programme was chosen to train the BN models of this project (details in Fish Occurrence Dataset: NOIZ Fyke).

Table 2: List of fish monitoring programmes in the entire Wadden Sea included in the QSR 2016 [2].

Monitoring programme	Sampling period	Sampling areas	Years
DFS	Sep	The entire Dutch Wadden Sea	1970-2015
DYFS	Sep-Oct	The entire German Wadden Sea	1978-2015

AWI	Mar, Jul	The German Wadden Sea (East Frisia)	1993-2007
NIOZ	Mar-Jun, Sep-Oct	Western Dutch Wadden Sea	1960-2015
IMARES	Apr-Jun, Sep-Nov	Western Dutch Wadden Sea	2000-2015
Jade	Apr-Aug	The German Wadden Sea (Jade)	2005-2015
Oyster reefs	May, Jun, Sep	The German Wadden Sea (Jade)	2014
Salt marshes	monthly	The German Wadden Sea (Dithmarschen)	2015-2016
Schleswig-Holstein	Aug	The German Wadden Sea (Dithmarschen & North Frisia)	1991-2015
German estuaries	May, Sep-Oct	The German Wadden Sea (Ems, Weser, Elbe and Elder)	2000-2015
Danish rivers	By species	The Danish Wadden Sea	1975-2015

2.7. Related Work

In [3], Eelke Folmer designed a BN model for assessing the cumulative environmental impact of activity in the western Dutch Wadden Sea. The author focused on the "salt-mining" case and considered the influencing factors regarding 1. Geomechanics and mining technology, 2. Morphodynamics, 3. Economics and 4. Ecology. The prototype BN model is shown in Figure 7. The work provided an important reference on building a BN model for the Dutch Wadden Sea. It showed a way to construct the BN model that combines experts' knowledge and data from statistical analyses and simulation. However, this prototype model designed by Folmer (Figure 7) had several limitations.

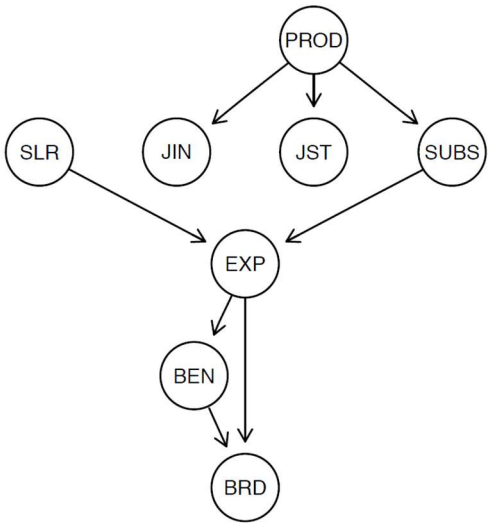


Figure 7: A prototype BN model for assessing cumulative effects of salt mining on the western Dutch Wadden Sea. SLR: Sea Level Rise; PROD: production; SUBS: Subsidence; EXP: exposure time; BEN: benthos; BRD: bird; JIN: Incidental jobs; JST: Structural jobs.

Although the prototype BN model was constructed combining both experts' domain knowledge and data, Folter utilized both kinds of information in a separate way. Some of the edges in the BN model were based on the information provided by a report [17], and others were made by data generated by physical models such as ASMITA [18] [19]. However, the analyses in the report were qualitative rather than quantitative, making it difficult to estimate the parameters (i.e. the CPTs) of those edges based on this report, which limited the ability of the constructed BN model to assess the impact quantitatively. Moreover, the prototype BN model also was not validated by in-situ data, which made the model not reliable from a data science point of view.

The methodology of the BN model construction in this report differes from Folmer's in the following aspects: **1. Data-oriented model**: The BN models were mainly trained by in-situ measurement data, and experts' knowledge only worked as auxiliary during structure learning. **2. Quantitative analysis**: All the CPTs were learned from in-situ measurement data. In other words, the constructed BN models were fully quantitative. **3. Performance evaluation:** All the constructed models were validated by in-situ measurement data from different aspects.

3. Datasets

Training dataset is the basis that determines the quality of learned BN models to build BN models with help of using learning algorithms. In this project, the following factors that impact the fish occurrence were considered: 1. the occurrence of major fish species in the Dutch Wadden Sea, 2. water quality variables such as dissolved oxygen, chlorophyll-a and water temperature etc., 3. human activities such as fishery and tourism. Therefore, data of those three groups of random variables have been collected from different databases and sources.

3.1. Fish Occurrence Dataset: NOIZ Fyke

As is mentioned in Section 2.6, NOIZ (Royal Netherlands Institute for Sea Research) operated the fish monitoring programme by setting a kom fyke trap at the entrance of the Marsdiep basin in the western Dutch Wadden Sea (see Figure 8 (E)). The kom fyke (see Figure 8 (B)) is a passive gear including a 200 m long leader and two chambers with a mesh-size of 10 * 10 mm. The fish monitoring with the kom fyke normally started around March or April and continued until October, because the trap would be removed in winter to prevent it from the damage by ice, and after 1971 less monitoring took place during summer due to fouling of the net and clogging by macroalgae.

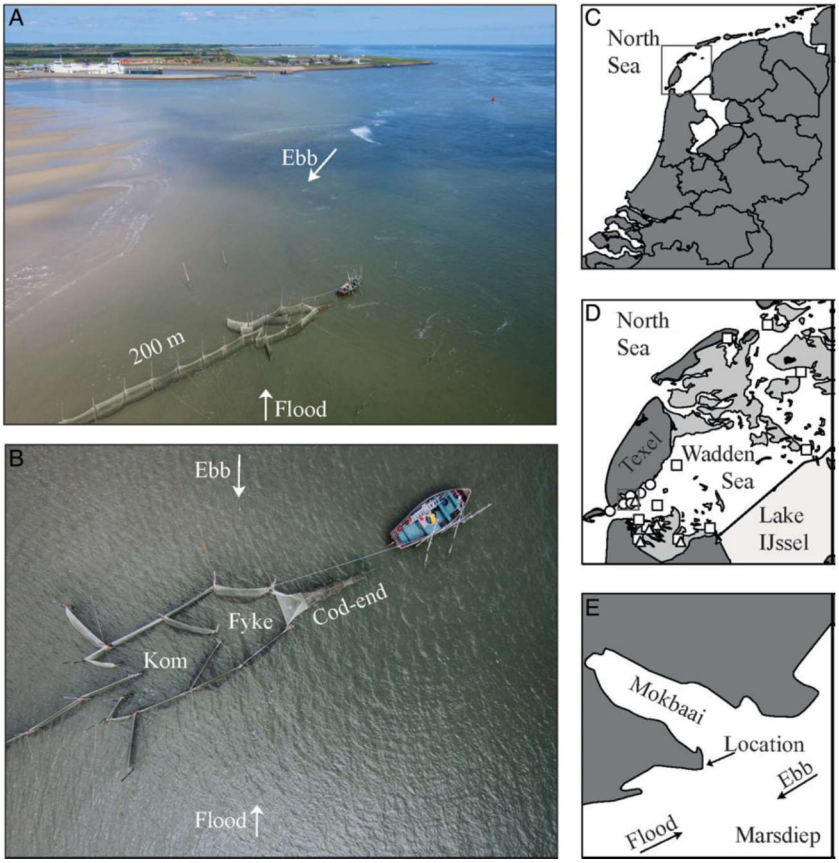
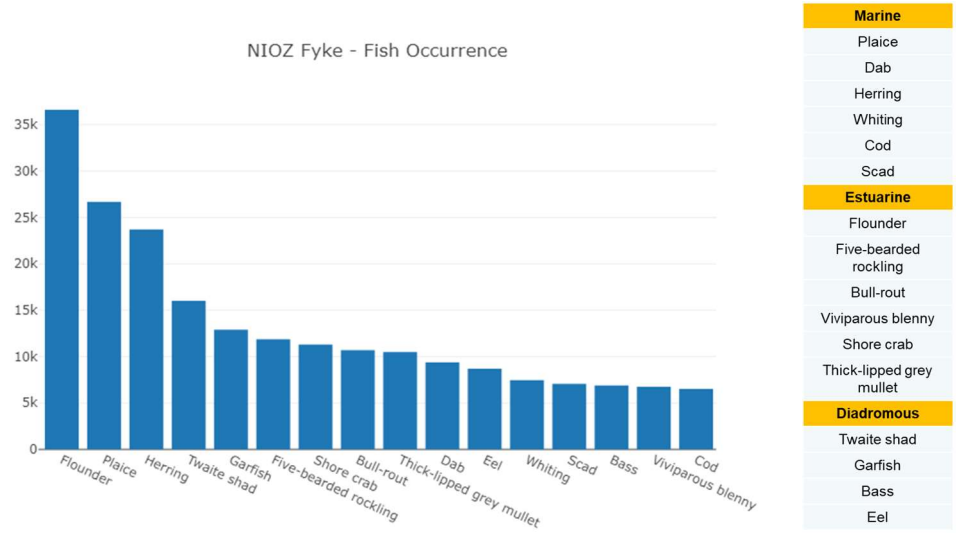


Figure 8: The NIOZ kom-fyke in the Dutch Wadden Sea. (A), (B): Aerial photograph showing the location and the design of the kom-fyke. (C), (D), (E): The map showing the location of NIOZ kom-fyke (Dark grey = land. Light grey = intertidal areas. White = water.) (Picture by Van Walraven et al. [20])

Large amount of data are crucial for training a reliable and generalised model, so those fish species which only occur occasionally should be excluded for the further analysis. Figure 9 (A) shows all the fish occurrences whose numbers surpass 5k in the NOIZ Fyke dataset (1960-2012), and those fishes are the major fish species in the western Dutch Wadden Sea. As is mentioned in the previous section (Study Area: The Dutch Wadden Sea), fish species found in the Dutch Wadden Sea can be roughly divided into three groups: 1. marine juvenile, 2. estuarine resident and 3. diadromous, and those terms are also called “functional guild” of fish species. Figure 9 (B) illustrates how the sixteen major fish species shown in (A) correspond to the three functional guilds, and at least one species was found for each guild.



(A) **(B)**

Figure 9: The major fish species whose occurrence was larger than 5k in the period of 1960-2012 in NOIZ Fyke dataset (A) and their functional guilds (B).

To make the constructed BN models comprehensive, all the three guilds need to be considered and analysed and to make the structure of the BN models clear; only a few fish species should be included as random variable nodes. Finally, the top 4 fish species, Flounder, Plaice, Herring and Twaite shad, were chosen for further study. According to Figure 9 (B), Flounder represents estuarine guild, Plaice and Herring represent marine guild, and Twaite shad represents the diadromous guild.

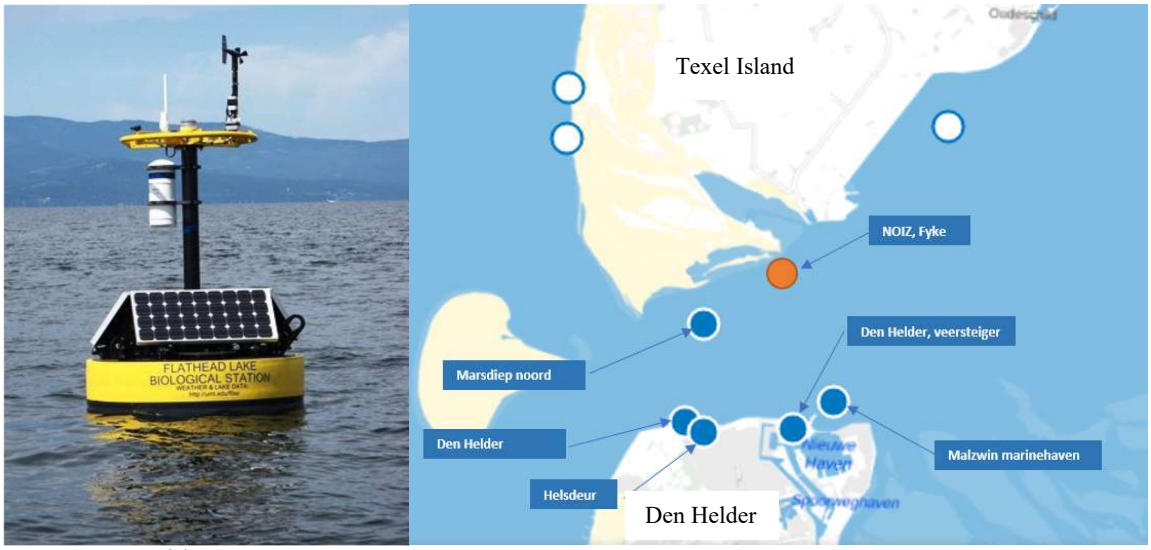
3.2. Water Quality Dataset: Waterinfo

The data on water quality were collected from the website named "Waterinfo" (http://waterinfo.rws.nl), which is the open data portal supported by Rijkswaterstaat. Rijkswaterstaat is part of the Dutch Ministry of Infrastructure and Water Management and responsible for the design, construction, management and maintenance of the main infrastructure facilities in the Netherlands. The institution owns multiple water stations (Figure 10 (a)) monitoring the water quality of rivers and sea all around the Netherlands.

The data on the website "Waterinfo" are distributed in two different ways. The data for the public are generally coloured based on the results of classification function with extra interpolation, and those data are allowed to be explored with look-back from 2 days up to 28 days. The data for the experts do not contain any classification and interpolation processes, but the look-back time is much longer than that for the public, and the expert's portal allows users to explore into more detailed water quality variables.

For the research purpose of this project, the data of high-level water quality variables with long time series were needed for training the BN models in association with NOIZ Fyke data. Therefore, water quality data for experts from water stations near the location of NOIZ Fyke were collected for further analysis. Figure 10 (b) shows the selected five water stations between Texel Island and Den Helder from which data were acquired. Most of the water quality data were measured at the Marsdiep Noord. As for water quality variables, the data

for eight variables were collected as they were the only measured variables in the locations shown in Figure 10 (b) related to fish occurrence. All those measured variables reflected the water quality status of the western Dutch Wadden Sea from two different aspects: 1. Climate change: Water Temperature, Water Level, Acidity. 2. Abundance of nutrients: Dissolved Oxygen, Total Phosphorus, Total Nitrogen, Chlorophyll-a, Floating Dust.



(a) (b)

Figure 10: (a) An example of biological water quality station (from the FLBS website). (b) The locations of selected water stations (blue points), unselected water stations (white points) and NOIZ Fyke (orange point).

3.3. Human Activities Dataset: Fishery & Tourism

Unlike constantly-measured variables in the water quality and fish occurrence groups, the databases of human activities are quite rare, especially for a relatively limited area like the Dutch Wadden Sea. For this reason, a different method has been deployed to acquire data on human activities, and the collected data focused on the topics of Fishery and Tourism in the Dutch Wadden Sea.

The data regarding fishery were provided by Eurostat (https://ec.europa.eu/eurostat/web/fisheries). Eurostat statistics on fisheries contain data for EU member states as well asIceland and Norway on: 1. Catches of fish products made by vessels in fishing regions; 2. Aquaculture production(marine and freshwater); 3. Landings of fishery products in ports; 4. Fishing fleet. In this project, the number of fishing fleets was considered as a random variable indicating human activities in the BN models. Therefore, the data of fishing fleets grouped by country, type of gear and engine power were downloaded from Eurostat. It should be noted that the database collected the annual total number of fleets for a whole country, which means that the granularity of the fishery dataset is much bigger than the water quality and fish occurrence datasets, both in time and space.

The data for tourism were found in the Annex “Tourism” of QSR 2016 [21]. The Dutch Wadden Sea coast can only be reached from the open sea and eight sluice passages. Thus, the number of boats through those

sluice passages was used to indicate the number of tourists around this area, which has been recorded since 1982. Unfortunately, the original source data for sluice passages could not be found on the open internet, so the exact data can only be estimated based on Figure 11 and Figure 12. In this case, an online tool called WebPlotDigitizer (https://automeris.io/WebPlotDigitizer/) was used to extract data from plots. Firstly, the annual total number of boats through sluice passages from 1996-2015 were extracted from Figure 11, and then from Figure 12 the seasonal trend of recreation boats was acquired in the same way. Finally, the monthly total number of boats through sluice passages for each year in the period of 1996-2015 was generated, and to introduce some randomness into this dataset, each monthly total number was added by a random number generated by a normal distribution (see Figure 13).

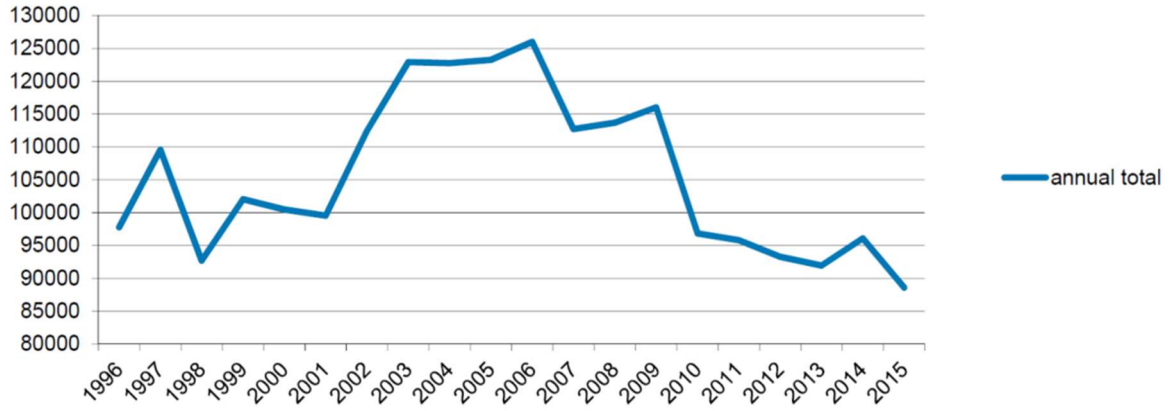


Figure 11: The number of sluice passages in the Dutch Wadden Sea, 1996-2015. (Source: Rijkswaterstaat Marjan Vroom, 2016.) [21]

Figure 12: Seasonality of sluice passages in the Dutch Wadden Sea, 2010-2015. (Source: Rijkswaterstaat, Marjan Vroom, 2016.) [21]

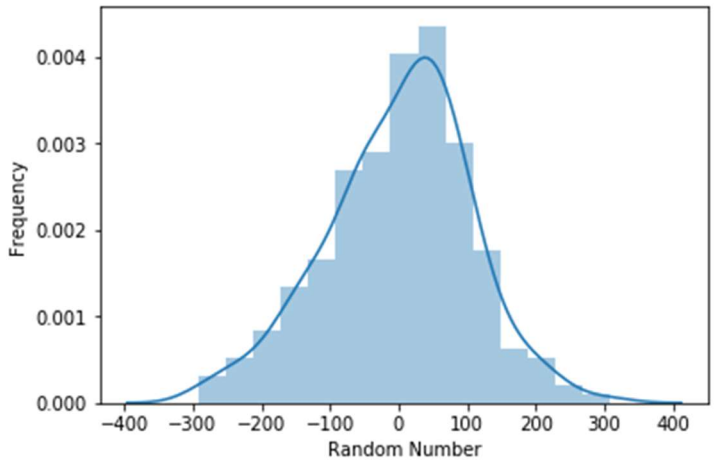


Figure 13: The distribution of random numbers added to the generated tourism data.

The two plots in Figure 14 illustrate that the yearly and monthly trends in the generated tourism dataset are almost identical to the trends in Figure 11 and Figure 12 proving that the generated dataset reflects the information of the in-situ measurement data.

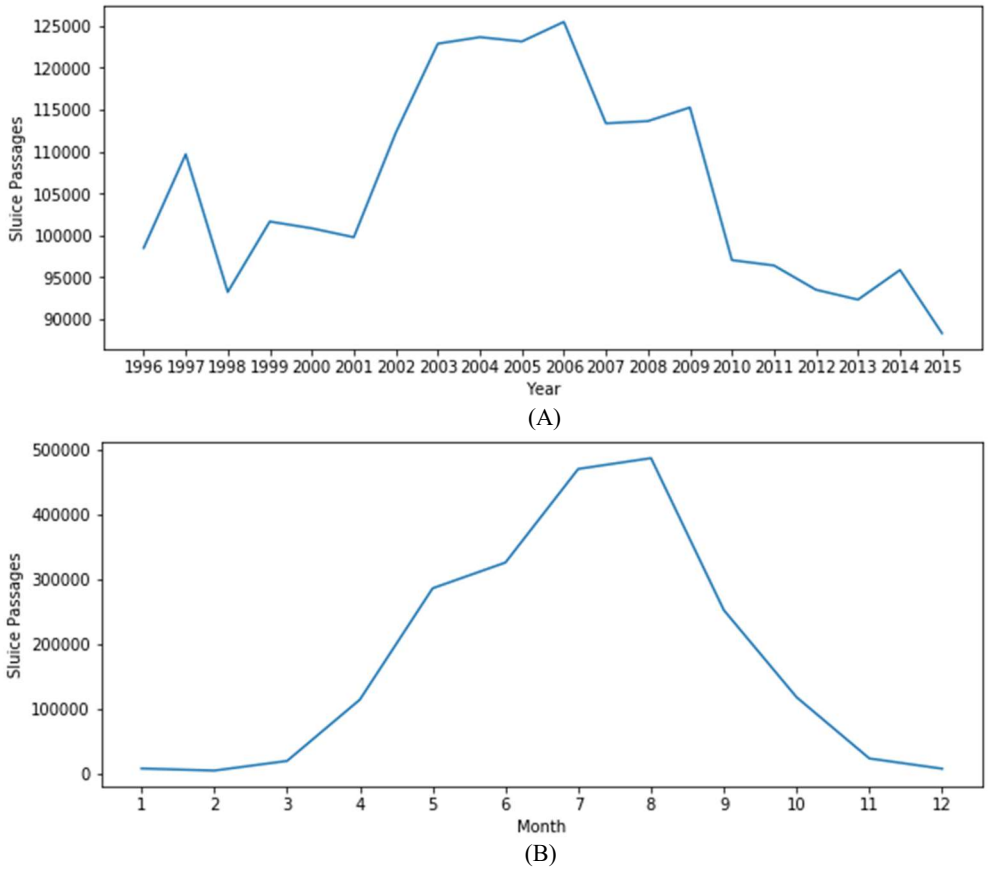


Figure 14: The yearly (A) and monthly (B) trend in the generated tourism dataset.

3.4. Summary

Table 3 summarizes all the datasets collected from different sources for the research purpose of this project. As it is shown in this table, most of the measurements for fish occurrence and water quality data were taken for one day, but the measurements of different variables were not necessarily taken on the same day. In addition, most of the measurements of nutrient variables such as Total Phosphorus were only taken in a single day of each month. Thus, when integrating all those datasets, the monthly average value of each variable was calculated, and the data measured from the same year and month were merged together.

It should also be noted that if the learning algorithm did not support training dataset with missing values, the tourism dataset will be the "shortest board on the bucket", since it covers the shortest period (1996-2015) among all those datasets. This means that all the data of other variables out of the period covered by tourism dataset should be abandoned to avoid missing values in the merged training dataset.

Table 3: The information of all the data collected for training the BN models.

Dataset	Categories	Location	Measurement Period	Measurement Interval	Number of Samples
NOIZ-Fyke	Fish Occurrence	Texel Island south (52.991400, 4.772483)	1960-2012	per day	273002
Waterinfo-Total Nitrogen	Water Quality	Marsdiep noord	1988-2014	per day	381
Waterinfo-Total Phosphorus	Water Quality	Marsdiep noord	1988-2014	per day	383
Waterinfo-Dissolved Oxygen	Water Quality	Helsdeur Malzwin marinehaven Marsdiep noord	1971-2014	per day	737
Waterinfo-Water Temperature	Water Quality	Den Helder Den Helder, veersteiger Helsdeur Malzwin marinehaven Marsdiep noord	1971-2015	per day per hour per 10 min	312863
Waterinfo-Acidity	Water Quality	Helsdeur Malzwin marinehaven Marsdiep noord	1971-2014	per day	778
Waterinfo-Chlorophyll-a	Water Quality	Marsdiep noord	1988-2014	per day	472

Waterinfo-Floating dust	Water Quality	Helsdeur Malzwin marinehaven Marsdiep noord	1973-2014	per day	746
Waterinfo-Water Level (w.r.t. Normal Amsterdam Level)	Water Quality	Den Helder	1960-2015	per 10 min per 3 hours	1786345
Eurostat-Fishing fleets by type of gear and engine power	Human Activity	EU countries, Iceland and Norway	1992-2017	per year	4102
QSR 2016 - The number of sluice passages in the Dutch Wadden Sea	Human Activity	The Dutch Wadden Sea	1996-2015	per year per month	240

4. Methodology

As mentioned in Section 2.4, the two traditional BN construction methods have their own advantages and disadvantages: the method based on experts' knowledge could guarantee the "conceptual correctness" of the model but cost much time on acquisition of human expertise; the method based on data could be time-saving but might cause the inconsistency between network structure and domain knowledge. Therefore, this project aims to explore a way to utilize both data and domain knowledge, try to overcome the disadvantages of both traditional methods.

An overview of the methodology to achieve the objective of this project is given in Figure 15. Firstly, in-situ measurement data collected from different sources were cleaned and pre-processed, then finally merged into one training dataset. To make the training dataset readable by the BN structure learning algorithm in GeNIe modeler, two pre-processing steps were needed: handling missing values and discretization, which will be discussed in Section 4.1. As for domain knowledge, it was collected from different works of literature written by experts in ecology, biology and ichthyology, or by consulting with experts directly. All the collected domain knowledge was qualitative rather than quantitative. Thus, the domain knowledge was not able to determine the BN models, but work as an aid in the structure training process after transferring it to GeNIe background knowledge (force arcs, forbid arcs and temporal tiers) which is mentioned in Section 2.5.The details will be introduced in Section 4.2.

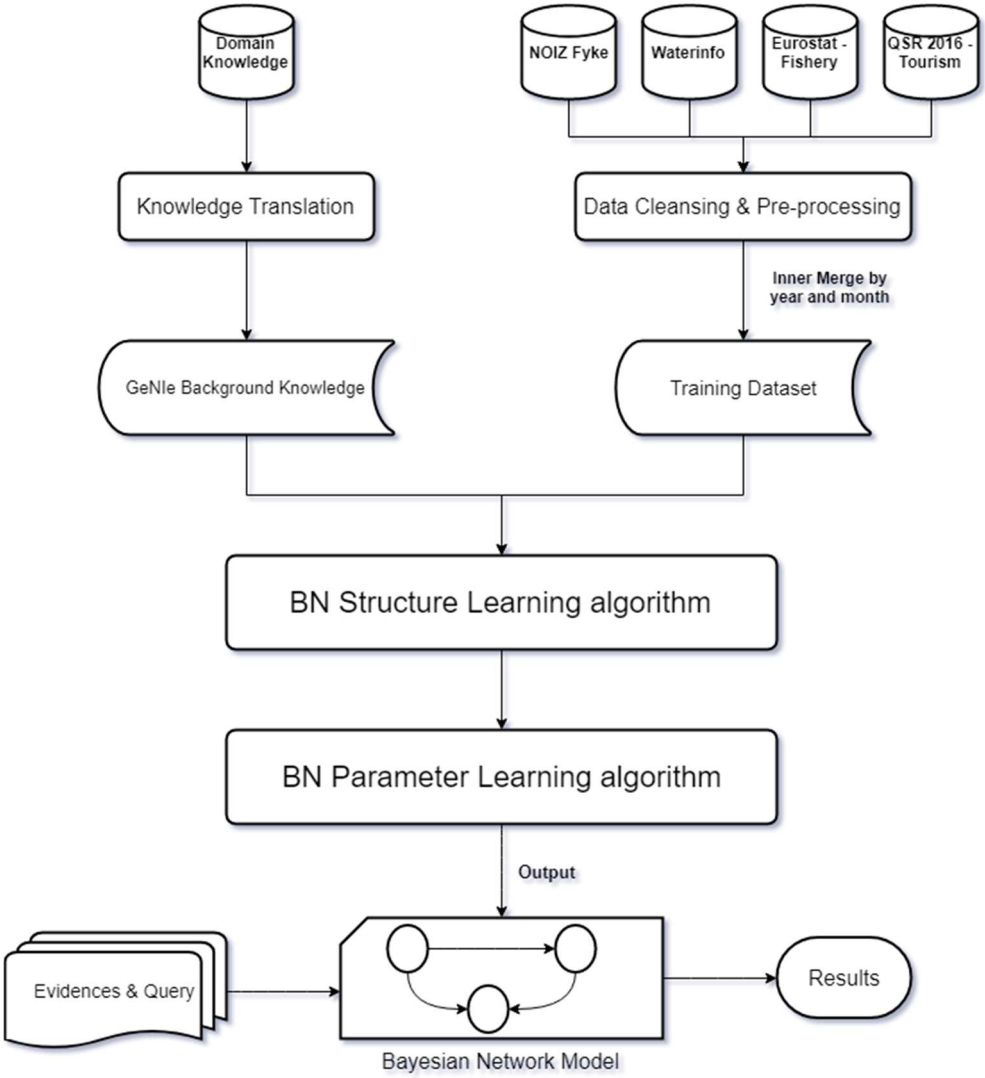


Figure 15: The work flow diagram of training BN models based on in-situ measurement data and experts' domain knowledge.

After all the data was pre-processed, the built-in structure learning algorithms in GeNIe started to run for training the BN models from the training dataset, with the aid of background knowledge (Section 4.3). Then in GeNIe modeller, the parameters of this BN model were automatically learned by the Expectation-maximization (EM) algorithm right after the structure learning algorithm completed its work (Section 4.4).

Finally, the output BN models needed to be evaluated and validated from different aspects. The metrics used for the evaluation and validation steps are introduced in detail in Section 4.5.

4.1. Data Pre-processing

All the BN structure learning algorithms in GeNIe modeller do not support training dataset with missing values, and most of them require the dataset to be discretized. Therefore, two pre-processing steps, discretisation and imputation of missing values, are needed before the training dataset can be read by the structure learning algorithms properly.

4.1.1. Imputation of Missing Values

The data in the training dataset were acquired from different databases and those databases covered different time periods as is shown in Table 3. Thus, when those data were merged together as a single training dataset, missing data would appear where the original database did not cover.

Figure 16 illustrates some of the commonly used methods to handle missing values. Generally, there are two ways to handle missing values in the dataset: simply deleting them or imputation. The deletion of missing data is usually the default in many statistical packages, but it may cause the heavy loss of data. As it is shown in Figure 17, strong correlations (i.e. absolute value greater than 0.4) can be found among water quality variables, thus makes it possible to use imputation method to handle missing values in water quality dataset, which means replacing each missing values with a reasonable estimated value based on the other observed values by using regression models.

The data missing in the water quality dataset was mainly caused by occasional sensors malfunction and different deploying date of sensors. Therefore, an obvious "monotone missing pattern" [22] was found in the water quality dataset (see Figure 18), and the completeness of each water quality variable is shown in Table 4. In this case, the missing values in different variables can be estimated sequentially, and the order was determined by the completeness and correlation relationship of each variable.

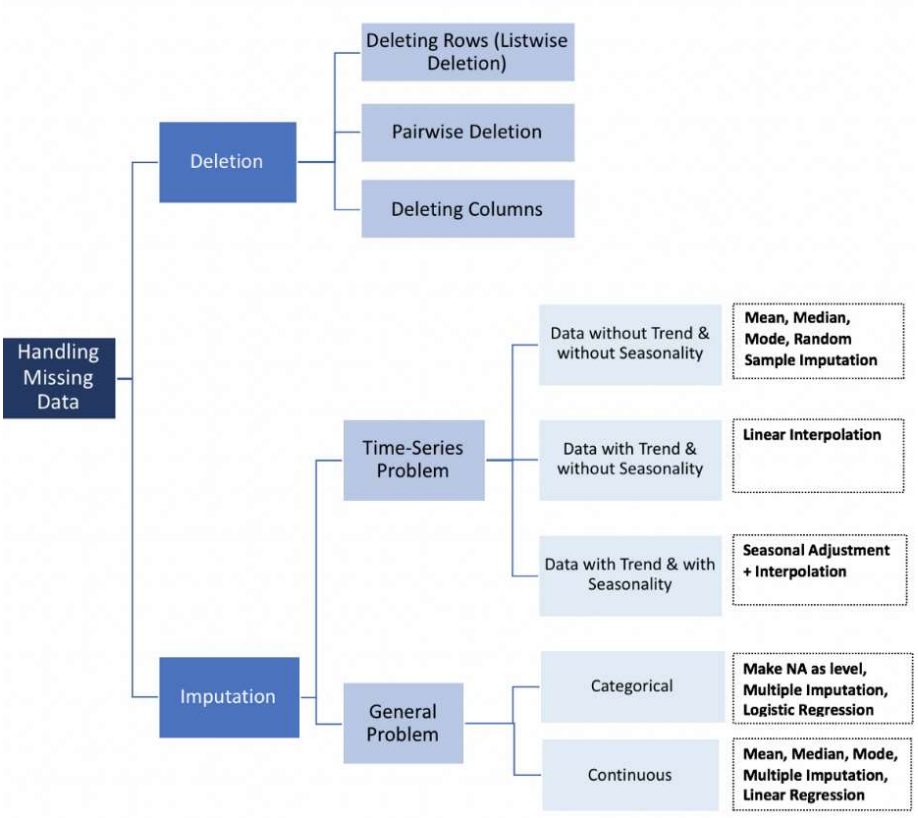


Figure 16: An overview of the most commonly used methods to deal with missing values. (Source: Alvira Swalin, 2018 [23])

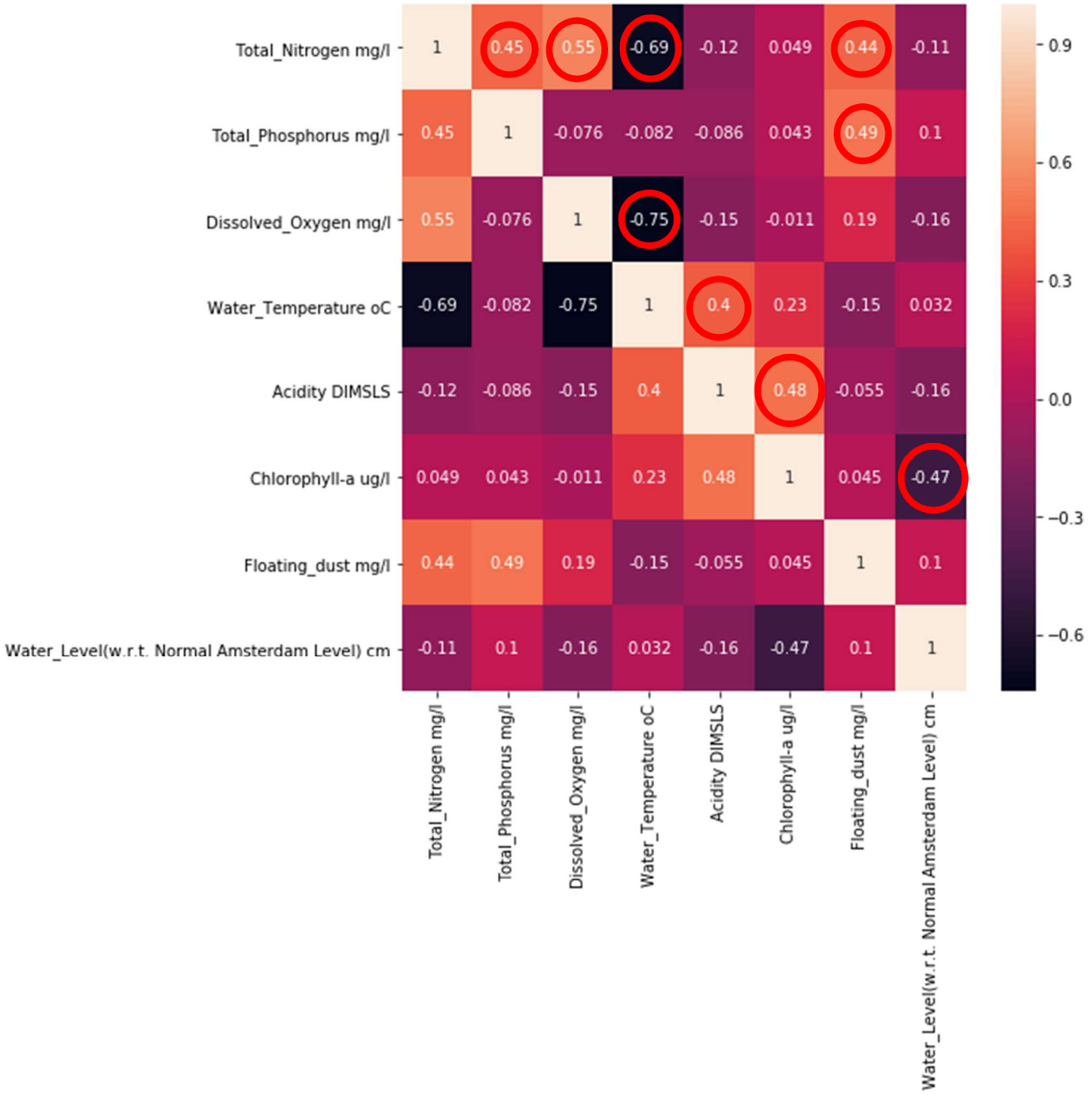


Figure 17: Correlation matrix of water quality variables. Strong correlations are marked by red circles.

The estimation sequence started with the variable "Water Level" with 100% completeness, and then "Chlorophyll-a" was estimated by "Water Level" because "Chlorophyll-a" has the strongest correlation with "Water Level" (see Figure 17). In each iteration, the variable to be estimated would be among variables having a strong correlation with the previously estimated variable (see red circles in Figure 17), and each variable was estimated by estabilishing the regression relationship with all the variables that had been estimated before it. Following the method mentioned above, the definitive order of estimation was: 1. Chlorophyll-a, 2. Acidity, 3. Temperature, 4. Dissolved Oxygen, 5. Total Nitrogen, 6. Total Phosphorus, 7. Floating Dust.

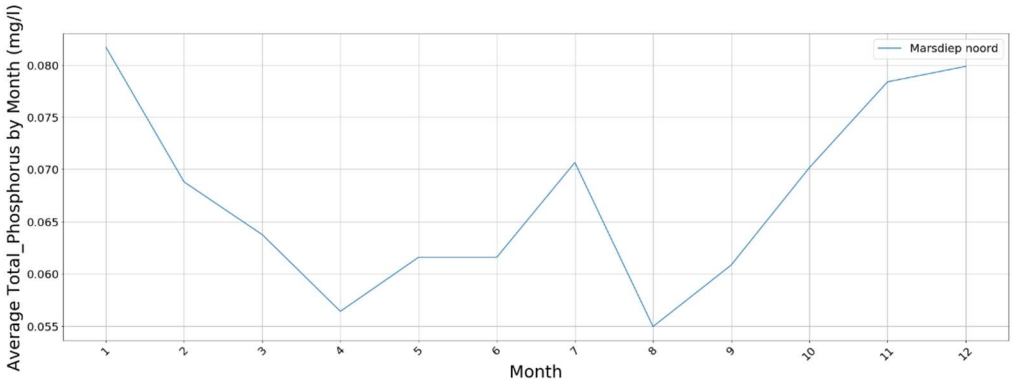
Missing monotone				Missing arbitrarily			
v1	v2	v3	v4	v1	v2	v3	v4
X	X	X	X	X	X	.	X
X	X	X	X	.	X	X	.
X	X	X	.	X	.	X	.
X	X	.	.	X	X	.	.
X	.	.	.	.	X	X	X

Figure 18: Illustration of monotone missing pattern and arbitrary missing pattern. [22] 'X' represents data and '.' represents missing values.

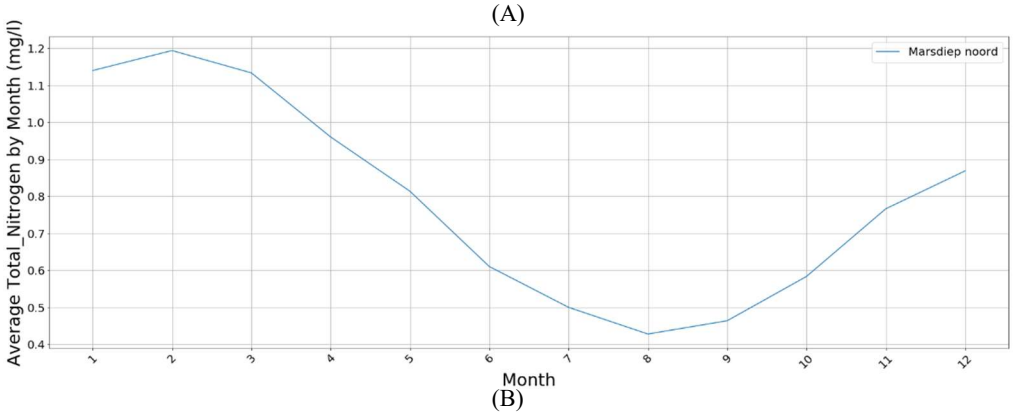
Table 4: The data completeness of each water quality variable in the period of 1960-2015.

Variable	Water Level	Acidity	Water Temperature	Dissolved Oxygen	Floating Dust	Total Nitrogen	Total Phosphorus	Chlorophyll-a
Completeness	100%	78%	78%	78%	75%	47%	47%	47%

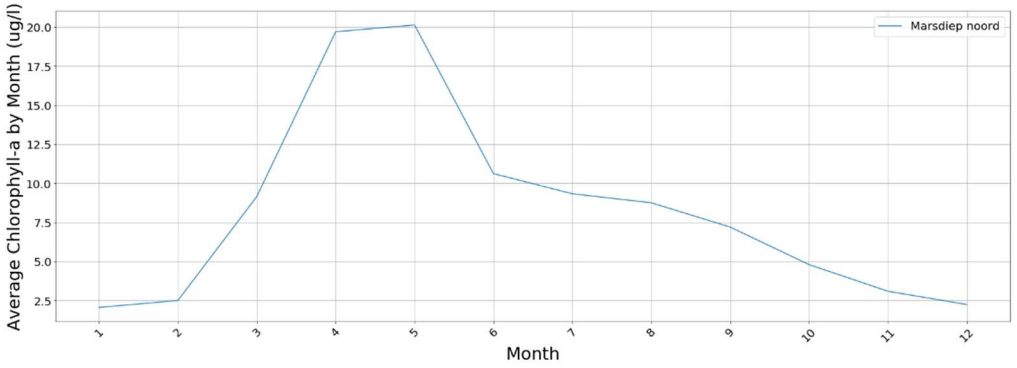
Furthermore, analysis results in Figure 19 show that seasonality can be found in the data of some water quality variables. For example, the surface water temperature in summer should be higher than that in the other seasons of the same year. It is inconsistent with the common sense if the highest water temperature in one year appears out of summertime in the estimated data. Therefore, the auxiliary variable "month" was involved as a predictor in the missing values estimation process of each water quality variable, to guarantee the seasonality was maintained within the estimated missing values. In Figure 20, it is clear to see that before "month" was included, the highest estimated water temperature in the year 1960 was found in October and the lowest one was found in December, which did not meet the seasonal pattern of water temperature in Figure 19 (D). After the variable "month" was included, the month with highest water temperature became July and the one with lowest water temperature became February, which are the more reasonable results according to Figure 19 (D).



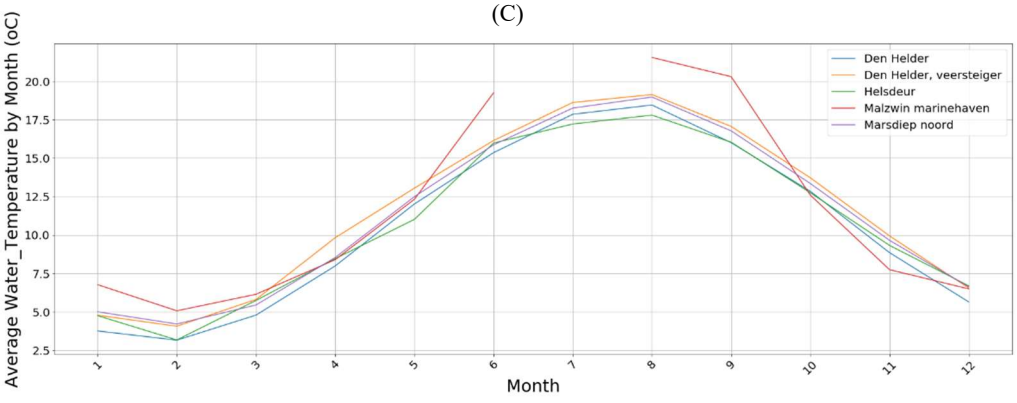
(A)



(B)



(C)



(D)

Figure 19: The seasonality within the water quality variables. (A) Total Phosphorus, (B) Total Nitrogen, (C) Chlorophyll-a, (D) Water Temperature.

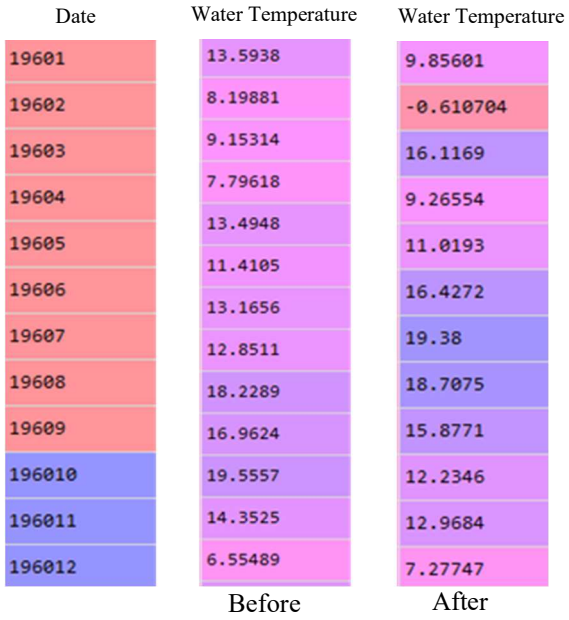


Figure 20: Samples of estimated missing values before and after "month" being involved in the estimation process.

As for the regression model, three different models were considered in this project: 1. Linear Regression, 2. Support Vector Regressor (SVR), 3. Random Forest Regressor. Their performance was compared in Table 5, and it shows that SVR is the best choice to estimate the missing value in the water quality dataset.

Table 5: The rooted mean squared error (RMSE) of three regression models when estimating the missing values of each water quality variable.

Water quality variable	**Linear**	**SVR**	**Random Forest**
Chlorophyll-a (ug/l)	7.21	2.88	2.59
Acidity (DIMSLS)	0.22	0.021	0.091
Water Temperature (oC)	4.91	0.27	0.61
Dissolved Oxygen (mg/l)	0.93	0.01	0.43
Total Nitrogen (mg/l)	0.23	0.0099	0.12
Total Phosphorus (mg/l)	0.031	0.0091	0.015
Floating dust (mg/l)	61.44	37.85	37.81

4.1.2. Discretization

All the data directly collected from databases were continuous data, but the BN structure learning algorithms can only use data with discrete values. Thus, discretization was applied to transfer continuous data into discrete counterpart. The core of the discretization steps is how to define the discrete classes of each random variable. The most common unsupervised ways to discrete datasets are Equal width method and Equal frequency method [24]. Equal width method divides the range of each variable into several intervals with identical width,

and Equal frequency method divides the range into intervals containing the equal number of sorted values. Figure 21 shows that the data discretized by Equal frequency method tend to be more balanced, and the balance of data is essential for the BN training process.

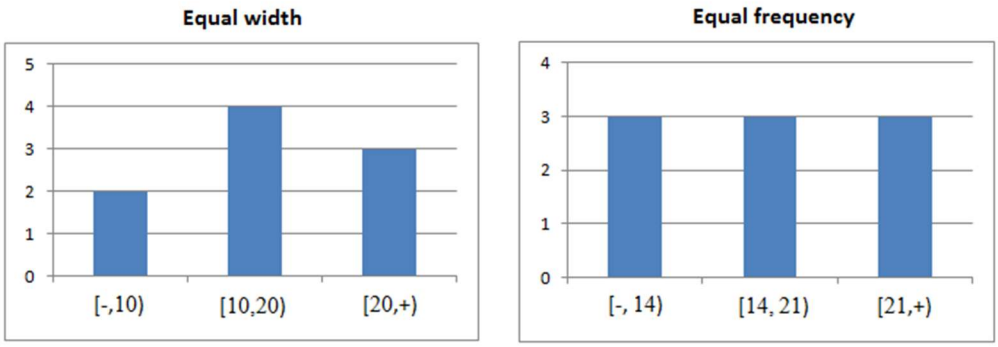


Figure 21: Equal width discretization vs Equal frequency discretization (Source: https://www.saedsayad.com/unsupervised_binning.htm)

However, most random variables in this project have their ecological meaning, so do their discrete classes. Thus, the definition of those discrete classes cannot only rely on statistical characteristics. The ecological meanings should also be considered.

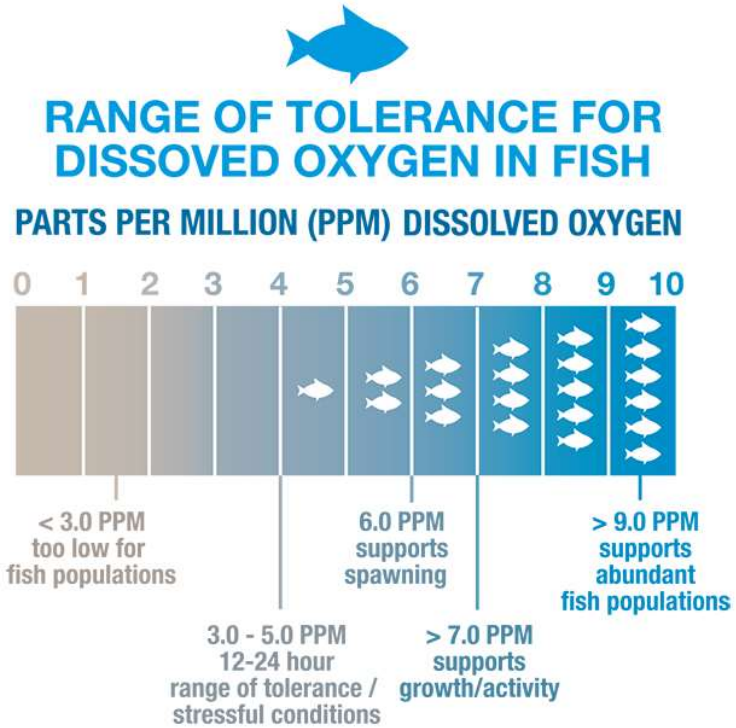


Figure 22: How dissolved oxygen affects aquatic life. [25] 1 ppm = 1 mg/l.

According to Figure 22 [25], the dissolved oxygen in the water has a huge impact on the life of fishes. It shows that the dissolved oxygen below 5.0 ppm (5.0 mg/l) is stressful for most of fish species. The image didn't

show how the dissolved oxygen higher than 10.0 ppm affects aquatic life, so it is assumed that this level is too high for fish species. The dissolved oxygen between 5.0 ppm and 10.0 ppm seems to be a normal range for fishes, and this range can be divided at 8.0 ppm to make the data more balanced.

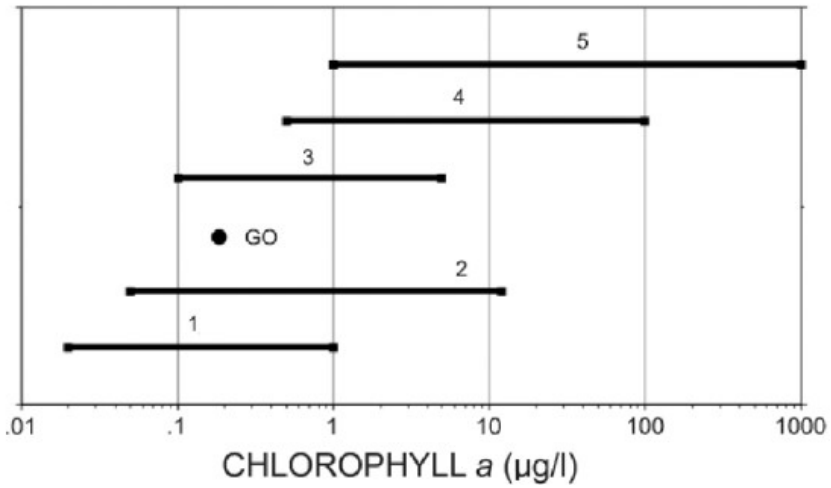


Figure 23: General ranges of chlorophyll-a concentrations for different ocean and coastal provinces: 1 - Sargasso Sea, Equatorial Pacific, Caribbean, 2 - California Current, 3 - Estuaries and Coastal Waters, 4 - North Atlantic, 5 - harmful algal blooms. GO = global ocean average of 0.19. [26]

According to Figure 23 [26], the normal range of chlorophyll-a in estuaries and coastal waters is between 0.1 µg/l and 6 µg/l. Therefore, the chlorophyll-a higher than 6 µg/l was defined as "high" concentration in the Dutch Wadden Sea, and the normal range was divided at 3µg/l to make the data balanced.

As for acidity, the scientific definition of neutral water is pH = 7, water with a pH < 7 is considered acidic and with a pH > 7 is considered basic or alkaline. In this project, pH = 7~7.6 was defined as neutral, pH = 7.6~8.4 was defined as moderately alkaline and pH > 8.4 was defined as strongly alkaline.

The details of discrete classes of all random variables refer to

Appendix A: Random variable nodes and discrete classes.

4.2. Background Knowledge

As shown in Figure 4, there are three kinds of background knowledge in GeNIe modeller: force arcs, forbid arcs and temporal tiers. They can work as an aid during the BN structure learning to make the output BNs more reasonable to domain experts and help to improve the performance of the output BN models. In this project, force arcs and temporal tiers were used to help build the BN models.

In reality, the interactions of all those random variables included in this project can be extremely complicated. Therefore, to make the learned BN structure more readable and understandable for both experts and non-experts, the following assumption was made: the human activity can affect fish occurrence directly, or indirectly by affecting water quality, the water quality only affects fish occurrence and fish occurrence cannot affect human activity and water quality backwards. In this case, the outline of the learned BN models would be like Figure 24. To make the BN models satisfied the assumption, three temporal tiers were used to control the direction of edges between two groups. The human activity variables, the water quality variables and fish occurrence variables were assigned to temporal tier 1, tier 2 and tier 3 respectively, making the learned BN structure consisitent with the outline structure in Figure 24.

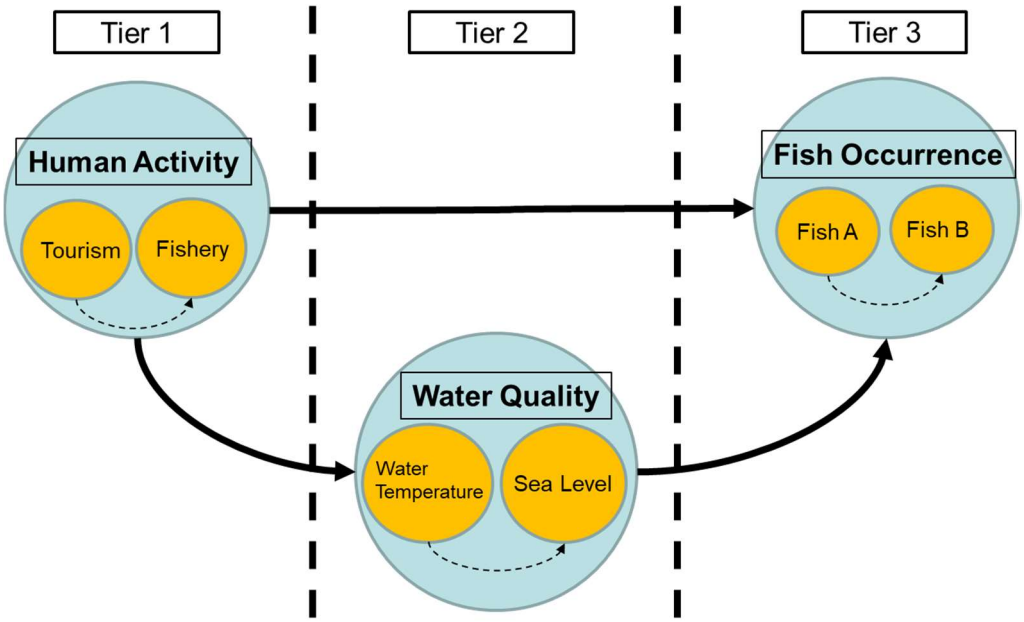


Figure 24: The outline of the BN models for fish occurrence in the Dutch Wadden Sea

If a pair of random variables were said to have a certain causal relationship by literature, their nodes would be forced to be connected by using force arcs. Figure 25 shows all the forced connection in the learned BN models in this project. Among all five kinds of domain knowledge, the first three kinds (a, b, c) are all about the relationships among water quality variables, and the last two of them (d, e) are about the relationships between water quality and fish occurrence. The following are the explanations of those forced connections.

a. Eutrophication: Eutrophication is a harmful phenomenon when a water body becomes overly enriched with nutrients causing the excessive growth of algae. This process may result in the depletion of dissolved oxygen and consequently endanger aquatic life. According to the research of [27], the low ratio of Nitrogen to Phosphorus (N/P ratio) indicates the increasing of eutrophication problem in the water body. Therefore, those nodes in Figure 25 were forced to be connected in that way.

b. Water Temperature to plant: Water temperature plays an important role in the growth of algae and other plants in the water by affecting the abilities of plants to absorb oxygen in the water [28] [29] [30].

c. Water Temperature to solubility: When water temperature increases, the gas solubility of water also increases because the average kinetic energy of the molecules that make up the solution also increases with temperature. However, more dissolved oxygen is present in water with a lower temperature compared to water with a higher temperature because of the solubility of a gas in a liquid is an equilibrium phenomenon [31].

d. Water Temperature to fish species: Water temperature may affect the migration of fresh water fishes and also cause changes of the occurrence of marine fishes [32] [33].

e. Dissolved Oxygen to fish species: Dissolved oxygen is the key to support all the life forms in the water, including all kinds of fish species [25].

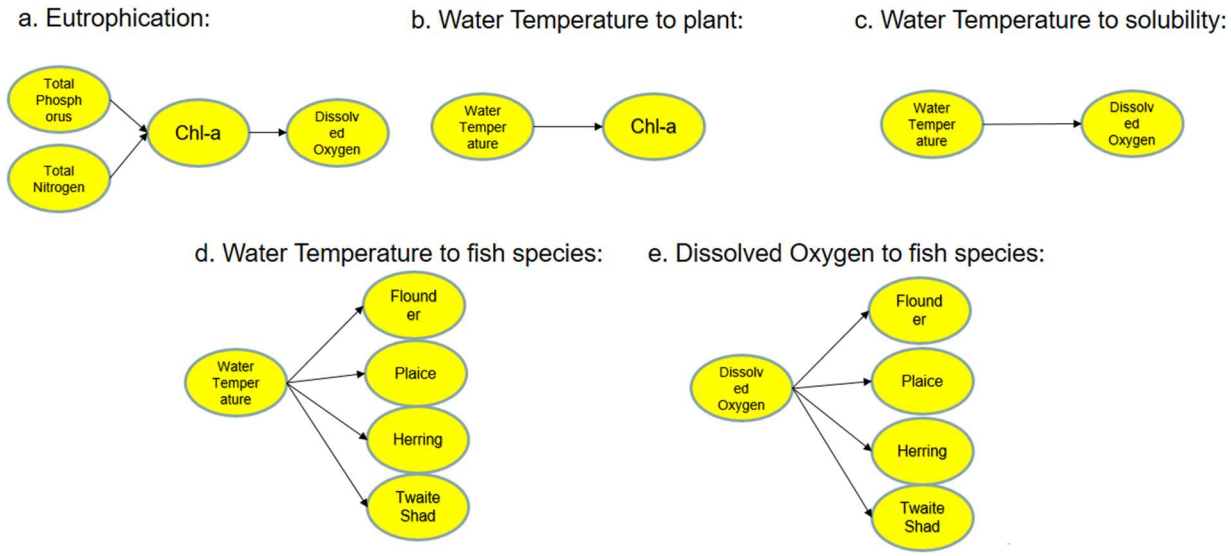


Figure 25: All the force arcs imported into the BN learning process. The yellow circles represent random variable nodes and arrows represent force arcs.

4.3. BN Structure Learning Algorithms

GeNIe modeller supports six different algorithms for BN structure Learning: 1. Bayesian Search, 2. PC, 3. Greedy Thick Thinning, 4. Tree-Augmented Naïve Bayes (TAN), 5. Augmented Naïve Bayes (ANB), 6.

Naïve Bayes. In this project, as a relatively simple and aged algorithm, Naïve Bayes was regarded as a baseline for other algorithms when comparing their performances.

4.3.1. Bayesian Search

Bayesian Search, also known as K2, was firstly introduced by Cooper & Herkovitz in 1992 [34], and is one of the most commonly used BN structure learning algorithms. The goal of this algorithm is to find the best BN structure that maximizes the Bayesian Score and it follows a hill climbing procedure with random start to avoid being trapped by local maximum (see Figure 26).

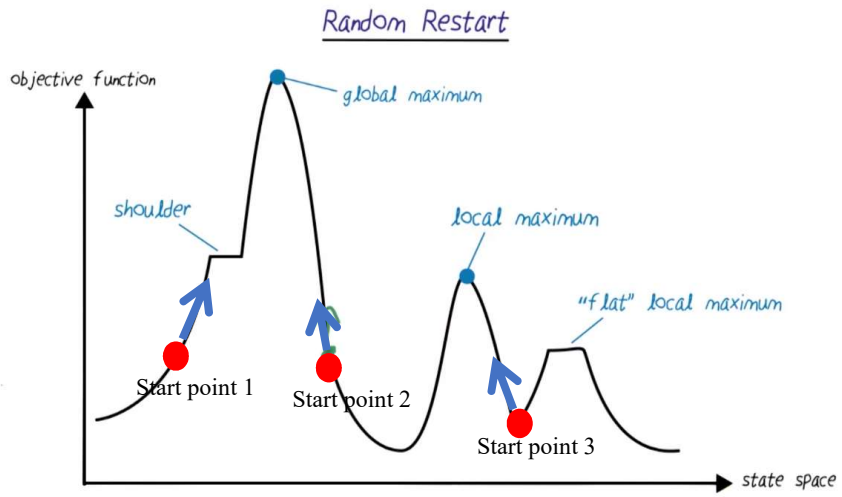


Figure 26: Hill climbing attempts to find a better solution by making an incremental change to the solution (the direction of blue arrows). However, the algorithm may only reach the local maximum with wrong start points (e.g. point 2, 3). The random restart will repeat hill climbing procedure with multiple random generated start points to find the global maximum [35].

The Bayesian Score measures the likelihood between training data and the BN structure by calculating the conditional probability of the BN structure G given the training data D:

$$P(G|D) = \frac{P(D|G)P(G)}{P(D)} \tag{4.1}$$

$P(G|D)$ cannot be calculated directly, so in equation (4.1), Bayes' theorem is used to convert the calculation of $P(G|D)$ to the calculation of $P(D|G)$. As $P(D)$ will be fixed once the training dataset is given, the formula of Bayesian Score can be written as follows:

$$score_B(G:D) = logP(D|G) + logP(G) \tag{4.2}$$

In equation (4.2), $logP(D|G)$ is called the Marginal likelihood, and $logP(G)$ is the structure-prior which remains constant for any structure in GeNIe modeller. The procedure of Bayesian Search can be described as the following:

1. Start with a graph without any edge.

2. Traverse all the possible connections until finding a connection that increases $score_B(G:D)$. Then add the connection into the graph.

3. Repeat step 2 until no connection can be found to make $score_B(G:D)$ increase. Record the graph and its corresponding Bayesian Score.

4. Repeat 2,3 with different traversing orders and choose the graph with the highest $score_B(G:D)$.

4.3.2. PC Algorithm

The PC algorithm was proposed by Spirtes et al. in 1993 [36], and it is actually a refined version of the SGS algorithm proposed by the same group of researchers [37]. Both algorithms are based on the same idea: the conditional dependency (d-separation) test. As shown in Figure 3, four different kinds of indirect relationships corresponding to two different types of conditional dependency.

The basic procedure of PC and SGS can be described as the following:

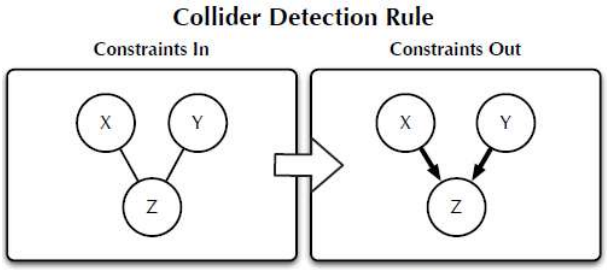
1. Start with a fully connected graph.

2. Run d-separation test for each pair of nodes $\boldsymbol{X}, \boldsymbol{Y}$. If there exists a set of nodes (or empty set) $\boldsymbol{W}$ making d conditional dependency $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{W})$ satisfied, which means they cannot be connected directly, then remove the edge between the pair of nodes.

3. If there exists a node $\boldsymbol{Z}$ between a pair of nodes $\boldsymbol{X}, \boldsymbol{Y}$, making conditional dependency $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{W}, \boldsymbol{Z} \notin \boldsymbol{W})$ satisfied, then add a v-structure into the graph. (see Figure 27 (A))

4. If the conditional dependency of $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ is not $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{W}, \boldsymbol{Z} \notin \boldsymbol{W})$, then $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ must not form a v-structure. In this case, if the edge has been oriented from $\boldsymbol{X}$ to $\boldsymbol{Z}$, the edge between $\boldsymbol{Y}$ and $\boldsymbol{Z}$ must be oriented from $\boldsymbol{Z}$ to $\boldsymbol{Y}$ to avoid forming a v-structure. (see Figure 27 (B))

(A)



(B)

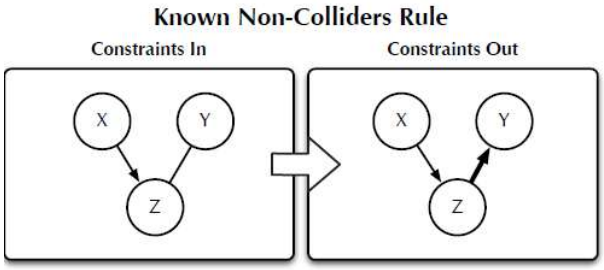


Figure 27: The orientation of undirected edges in the PC and SGS algorithm. (A) The creation of v-structure. (B) The avoidance of v-structure

The difference between PC and SGS is that, when running conditional dependency test, the SGS algorithm will try all the possible sets of nodes between the node pair $\boldsymbol{X}, \boldsymbol{Y}$, while the PC algorithm will only try the node sets with a limited number of nodes. The idea behind it is that the higher order conditional dependency test is less reliable because of the limitation of the sample data. For example, in this project, there're 25 random variables in total taking 3 values each, to run the conditional dependency test for two variables on all the possible sets of remaining nodes requires considering the relations among 3^{23} distinct states. However, only a small portion of those states will be instantiated even in quite large sample data. Therefore, the PC algorithm is much more efficient than SGS algorithm when dealing with a large number of nodes.

The advantage of these conditional-dependency-based algorithms like the PC and SGS is that they focus on finding indirect relationships between nodes rather than direct ones, that makes those algorithms able to extract more in-depth information from data. However, the PC algorithm tends to output an overcomplicated structure if the training dataset is small, bringing a lot of pressure to the parameter learning afterwards.

4.3.3. Tree-Augmented Naïve Bayes (TAN) and Augmented Naïve Bayes (ANB)

Tree-Augmented Naïve Bayes (TAN) and Augmented Naïve Bayes (ANB) both proposed by Friedman et al. in 1997 [38] try to augment the performance of Naïve Bayes classifier by adding connections among the attribute nodes (see Figure 28).

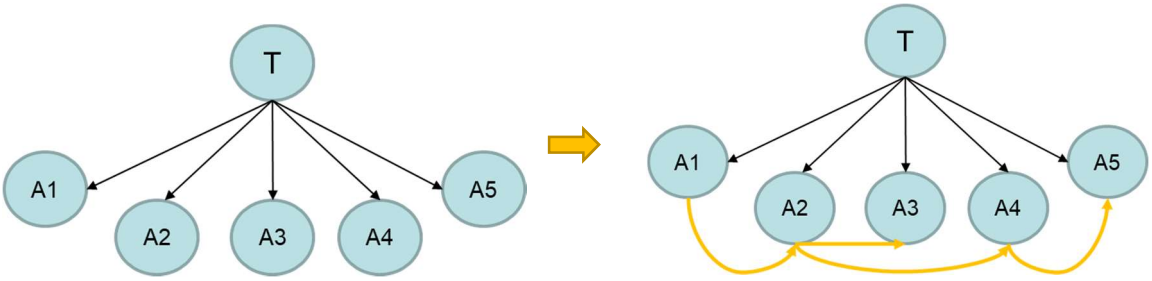


Figure 28: From Naïve Bayes classifier to TAN or ANB structure.

The core of TAN and ANB is to determine the dependency between attribute variables. Relative entropy, also known as mutual information, is used to measure the dependency, and the formula is as following:

$$I(X_i, X_j) = \sum_{X_i, X_j} \hat{P}(X_i, X_j) log \frac{\hat{P}(X_i, X_j)}{\hat{P}(X_i)\hat{P}(X_j)} \tag{4.3}$$

The procedure of TAN and ANB is described as following:

1. Start with a Naïve Bayes structure (see Figure 1).

2. Compute mutual information for each pair of attribute nodes as weight between them.

3. Build a maximum spanning tree among those attribute nodes.

4. Transform the resulting undirected tree to a directed one.

The difference between TAN and ANB is that TAN only allows each node to have maximum two parents and ANB does not have such limitation unless it is imposed by one of ANB algorithm's parameters (Max Parent Count).

The TAN and ANB are good at classification tasks. However, in TAN and ANB structure, only one variable can be set as a target, and the strong assumption is made that all the remaining variables are dependent on the target variable. Therefore, both algorithms don't support background knowledge in GeNIe.

4.3.4. Greedy Thick Thinning

In 1997, Cheng et al. [39] proposed a Greedy Thick Thinning algorithm, which is a more comprehensive method taking both direct dependency and conditional dependency into account. The direct dependency is measured by mutual information as TAN and ANB in Equation (4.3), and the conditional dependency is measured by conditional mutual information as following:

$$I(X_i, X_j \mid C) = \sum_{X_i, X_j, C} \hat{P}(X_i, X_j \mid C) log \frac{\hat{P}(X_i, X_j \mid C)}{\hat{P}(X_i \mid C)\hat{P}(X_j \mid C)} \tag{4.4}$$

As the name suggests, the procedure of this algorithm can be divided into three steps: "Greedy", "Thick" and "Thinning":

1. **Drafting**: Greedy search for a draft graph by adding edges between the node pairs whose mutual information is larger than a threshold ε.

2. **Thick**: Add edges when the pairs of nodes cannot be d-separated (by using conditional mutual information).

3. **Thinning**: Remove edges when the pairs of nodes can be d-separated (by using conditional mutual information).

4.4. BN Parameter Learning Algorithm

GeNIe modeler uses the Expectation-Maximization (EM) algorithm proposed by Lauritzen in 1995 [40] to learn parameters for a BN structure (also called "pattern" in GeNIe). In general, this algorithm estimates the BN parameters from data by maximizing the EM loglikelihood, which measures the probability of the data D given a model $\langle G, \hat{\theta}_G \rangle$:

$$score_L\left(\langle G, \hat{\theta}_G \rangle : D\right) = logP\left(\hat{\theta}_G \middle| D\right) \tag{4.5}$$

where G denotes the structure of a BN model and $\hat{\theta}_G$ is the estimated parameters of G. During parameter learning, the structure G is fixed, thus G can be ignored when calculating EM loglikelihood. This metric seems to be similar to Bayesian Score in Equation (4.2). The difference between them is that the Bayesian Score measures the likelihood between the BN structure without parameters and dataset, while the EM loglikelihood measures the likelihood between the whole BN model (i.e. with parameters) and dataset.

Besides, the algorithm can also learn parameters from data with missing values. When calculating the EM loglikelihood, the algorithm will assume that all the missing values are filled with all their possible values in the dataset.

The procedure of the EM parameter learning algorithm can be described as following:

1. Initialize the parameters randomly.

2. Calculate $score_L(G:D)$, if there're missing values in dataset D, fill them with all possible values.

3. Update the parameters to make $score_L(G:D)$ increase.

4. Repeat 2,3 until converge (i.e. the difference of parameters between two adjacent iterations is smaller than a threshold).

4.5. Validation Methods

After all the learning processes are finished, validation steps are necessary to test the performance of the BN models trained by different algorithms.

The main objective of this project can be divided into two sub-objectives: knowledge discovery and density estimation of fish occurrence, and different metrics will be used to evaluate the ability of the BN models to solve those two sub-objectives respectively.

For knowledge discovery, the BN models are expected to extract useful information from data as much as possible, so the models should fit the training dataset rigorously. In this case, EM loglikelihood is an ideal metric to measure how much the model fits the training dataset. Besides, different ecological scenarios were also introduced to the BN models to see if the reactions of those models were correct according to domain knowledge.

For density estimation, the BN models can be regarded as classification models being able to predict the future. In this case, the generalization of those models becomes very important. Cross-validation accuracy is a reliable metric to measure the generalization ability of each model. In GeNIe modeler, k-fold cross validation means to train the parameters of the structure using k-1 equally-divided parts of the dataset and test the model using kth part, then repeat the same procedure k times until all parts being used as test dataset. The cross-validation accuracy is the average classification accuracy among all the k experiments.

Furthermore, in order to evaluate the reliability of the learned BN models, the confidence interval was estimated based on bootstrap sampling [41]. However, GeNIe modeler does not support the confidence interval calculation as well as training the parameters of BN models using bootstrap samples. Therefore, the work needs to be done in Python environment, and the procedure is shown in Figure 29. Firstly, the GeNIe network file was imported into Python by "PySmile" wrapper. Then a Python package for building probabilistic models called "Pomegranate" was used to train the parameters of the same BN structure using hundreds of bootstrap samples in the batch. After that, all those BN models with the same structure and different parameters were queried by the same set of evidences and the same target. Finally, the confidence interval was estimated by taking quantiles of all the results answered by those models.

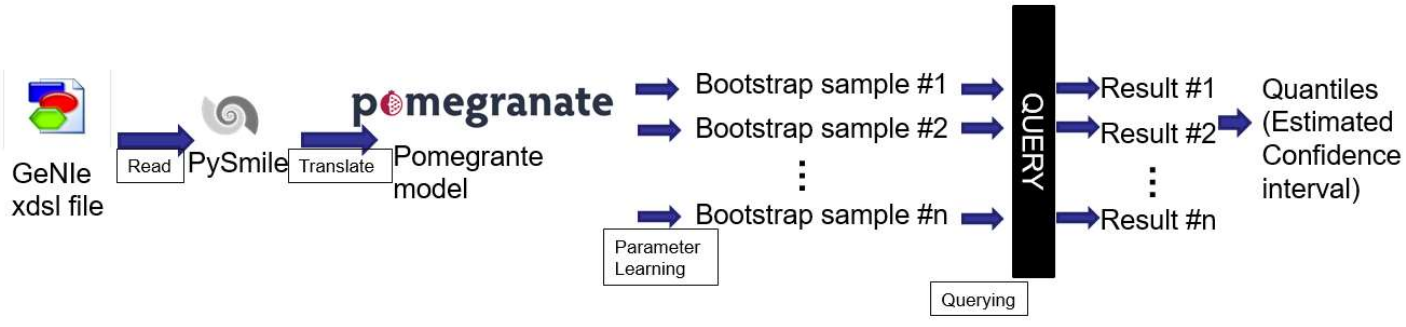


Figure 29: The work flow of confidence interval estimation for GeNIe BN models using Python package ***"Pomegranate"****.*

5. Results

In this project, the BN models for fish occurrence in the western Dutch Wadden Sea was trained by six different BN structure learning algorithms and those algorithms were compared using the following metrics: running time, network structure complexity (number of nodes & edges), EM loglikelihood and cross-validation accuracy. Besides, Bayesian Search, PC and Greedy Thick Thinning algorithm were run with and without the aid of background knowledge respectively, to demonstrate how background knowledge works in the BN structure training process.

In the experiment, two versions of Fish Occurrence BN model were learned using different training dataset (see Figure 30): one only contains the random variables in water quality and fish occurrence group, and the other contains the random variables in all three groups. The latter model was trained by fewer samples than the former one because of the lack of human activity data.

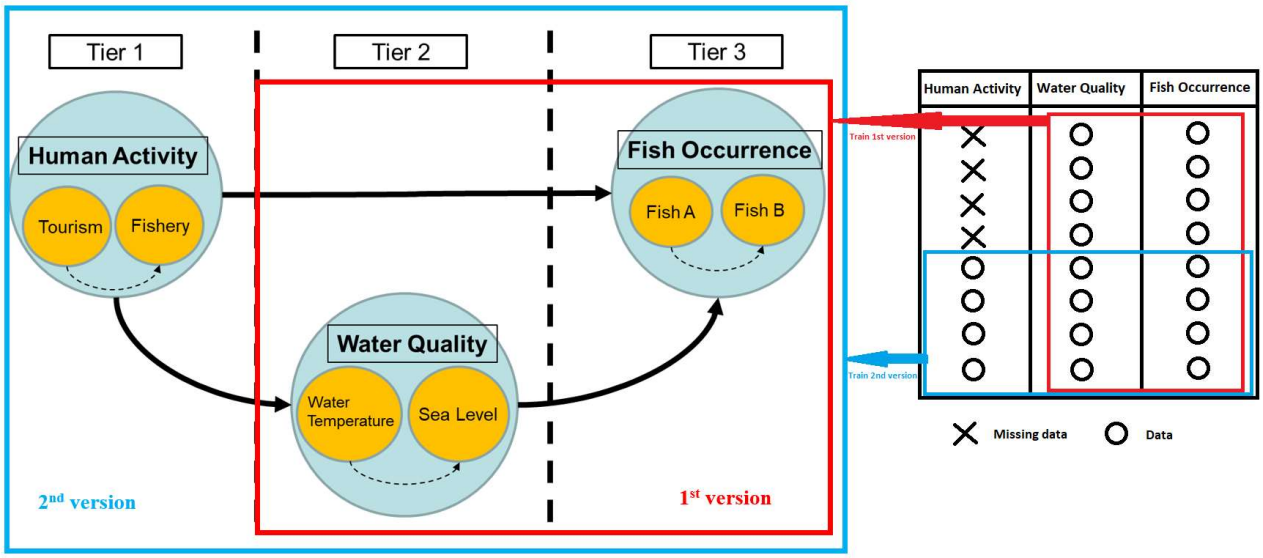


Figure 30: Illustration of two versions of BN model involved in the experiment

At last, the confidence interval was estimated as discussed in Section 4.6, to measure the reliability of the prediction from the learned BN models under a certain scenario. The range of confidence interval also indicates the robustness of those BN models.

The experiment environment is as following:

CPU: Intel Core i5-3360M (2.8GHz)

RAM: 8 GB

Software: GeNIe modeler (2.3 academic), pySmile (1.2.1 academic), Spyder (Python 2), Pomegranate (0.10.0)

5.1. BN Models for Water Quality and Fish Occurrence

In this section, the BN models contain the random variables in water quality and fish occurrence groups. Thus, those BN models aim to discover the influence of water quality on major fish species in the western Dutch Wadden Sea. In this case, those models were trained by 635 monthly averaged samples originated from "Waterinfo" and "NOIZ Fyke" database, in the period of 1960-2012. The parameter setting of each algorithm is shown in Table 6. Most of the parameters were default setting in GeNIe except "Sample size" parameter in Bayesian Search algorithm. This parameter controls the number of samples that take part in the Bayesian score calculation. This parameter was set to 300 (default 50) to make the output BN structure reflect the pattern of more data. Table 7 compares the performance of each BN structure learning algorithm on this task. It should be noted that the running time in this table includes both structure learning time and parameter learning time, and 4-fold cross-validation was used to test the classification accuracy of each algorithm.

Table 6: The parameter setting for training BN models based on water quality and fish occurrence datasets.

Algorithm	**Parameter setting**
Bayesian Search	Iterations: 20, Max parent count: 8, Sample size: 300, Link probability: 0.1, Prior link probability: 0.001, Seed: 2
PC	Max adjacency: 8, Significance: 0.05
TAN	Class variable: Flounder, Seed: 2
ANB	Class variable: Flounder, Feature selection: no, Max parent count: 8, Iterations: 20, Sample size: 300, Link probability: 0.1, Prior link probability: 0.001, Seed: 2

Greedy Thick Thinning	Max parent count: 8

Table 7: The summary of the performance of each BN structure learning algorithm for training BN models based on water quality and fish occurrence datasets.

Algorithm	**Running time**	**Network Edges (Nodes)**	**EM Loglikelihood**	**Cross-validation Accuracy**
Bayesian Search (no knowledge)	0.188s	17(12)	-6059.66	Overall = 0.58 Flounder = 0.55 Plaice = 0.57 Herring = 0.63 Twaite shad = 0.57
Bayesian Search (with knowledge)	0.266s	27(12)	-5781.57	Overall = 0.59 Flounder = 0.59 Plaice = 0.58 Herring = 0.63 Twaite shad = 0.57
PC (no knowledge)	0.251s	23(12)	-5780.93	Overall = 0.54 Flounder = 0.53 Plaice = 0.51 Herring = 0.57 Twaite shad = 0.54
PC (with knowledge)	0.282s	29(12)	-5626.34	Overall = 0.58 Flounder = 0.57 Plaice = 0.57 Herring = 0.62 Twaite shad = 0.55
TAN	0.109s	21(12)	-5920.53	Overall = 0.59 Flounder = 0.59 Plaice = 0.56 Herring = 0.63 Twaite shad = 0.57
ANB	0.344s	30(12)	-5725.25	Overall = 0.58 Flounder = 0.58 Plaice = 0.58 Herring = 0.63 Twaite shad = 0.55
Greedy Thick Thinning (no knowledge)	0.11s	16(12)	-5950.42	Overall = 0.59 Flounder = 0.60 Plaice = 0.58 Herring = 0.65 Twaite shad = 0.53
Greedy Thick Thinning (with knowledge)	0.14s	22(12)	-5909.08	Overall = 0.59 Flounder = 0.58 Plaice = 0.58 Herring = 0.63 Twaite shad = 0.57
Naïve Bayes	0.11s	11(12)	-6500.04	Overall = 0.58 Flounder = 0.58 Plaice = 0.57 Herring = 0.62 Twaite shad = 0.55

According to the result in Table 7, among all the BN structure learning algorithm (except Naïve Bayes), TAN and Greedy Thick Thinning are the most time-saving choices, and both of them had slightly better performance on overall classification accuracy than Naïve Bayes (TAN and Greedy Thick Thinning had 0.59 cross-validation accuracy while Naïve Bayes had 0.58 accuracy). Therefore, those two algorithms are the optimal solution for the density estimation or prediction problem in this case.

As for knowledge discovery problem, the PC algorithm is able to reach higher EM loglikelihood by adding fewer edges than Bayesian Search and ANB algorithms. Comparing the performance of Bayesian Search with background knowledge and PC without background knowledge, it is possible to see that, Bayesian Search reached a comparable EM loglikelihood by adding four more edges compared to PC. Even though the ANB algorithm added more edges than PC with background knowledge, the EM loglikelihood of ANB algorithm was still less than that of PC. However, the PC algorithm had the worst classification accuracy among all the algorithms in GeNIe without the aid of background knowledge. Through that, the PC algorithm itself is not good at solving classification problem, and background knowledge can help it perform much better on classification and prediction task.

In addition, background knowledge could stably improve the EM loglikelihood of Bayesian Network, PC and Greedy Thick Thinning algorithm, but did not necessarily improve the overall cross-validation accuracy of them.

Finally, the confidence intervals were estimated by using bootstrap samples to validate the reliability of BN models trained by different algorithms.In the testing scenario, it is given that N/P ratio was critically low, and the occurrence of Flounder was observed in the end. In other words, the evidence was "Total_Phosphorus" = "High" and "Total_Nitrogen" = "Low", and the target node was "Flounder". In Table 8, the number outside brackets is the median and the numbers inside brackets denote the estimated 90% confidence interval.

According to the result in Table 8, Bayesian Search, PC and Greedy Thick Thinning tend to have very similar results in this scenario, and the ranges of confidence intervals of those algorithms are all around 0.10 ~ 0.15. However, the results of TAN and ANB are quite different from those of the other algorithms, and the confidence intervals of those two algorithms based on Naïve Bayes are dramatically large, which indicates that the estimated probability distribution from TAN and ANB were unreliable. Although TAN and ANB performed well in cross-validation, obviously they failed to make a reliable prediction when the statuses of only a few nodes were known like this scenario.

Table 8: 90% confidence interval of predicted probability of each occurrence level of Flounder when Total Phosphorus is high and Total Nitrogen is low. The parameters were trained by 400 samples (around 2/3 of total samples) in each of 100 iterations.

Algorithm	Flounder = "High"	Flounder = "Medium"	Flounder = "Low"
Bayesian Search (no knowledge)	0.38 (0.30, 0.47)	0.46 (0.37, 0.54)	0.14 (0.10, 0.20)
Bayesian Search (with knowledge)	0.39 (0.34, 0.44)	0.40 (0.34, 0.46)	0.19 (0.15, 0.26)
PC (no knowledge)	0.35 (0.30, 0.43)	0.40 (0.34, 0.46)	0.23 (0.19, 0.27)
PC (with knowledge)	0.39 (0.33, 0.46)	0.42 (0.33, 0.48)	0.17 (0.12, 0.23)
TAN	0.50 (0.20, 0.77)	0.33 (0.08, 0.72)	0.11 (0, 0.25)
ANB	0.47 (0.04, 0.69)	0.35 (0.08, 0.67)	0.19 (0.02,0.46)
Greedy Thick Thinning (no knowledge)	0.38 (0.32, 0.44)	0.43 (0.35, 0.50)	0.17 (0.12, 0.23)
Greedy Thick Thinning (with knowledge)	0.39 (0.33, 0.46)	0.42 (0.35, 0.49)	0.17 (0.13, 0.22)

5.2. BN Models for Human Activity, Water Quality and Fish Occurrence

Next, attempts were made to get the human activity dataset involved. Because the tourism data collected in this project only covered the period of 1996-2015, the BN models in this section were trained by monthly averaged 192 samples during this period.

The first attempt was made to train the BN models including all the random variables in

Appendix A: Random variable nodes and discrete classes. As shown in the table, there are 12 variables indicating the number of fishery vessels with different engine powers. One of the output BN structure is shown in Figure 31. From this structure, it is clear to see that, all the nodes for fishery vessels were strongly connected to each other and only one node ("Fishery_vessel_KW25_74") was connected to the node ("Total_Phosphorus") out of the human activity group. That means most of the nodes for fishery vessels should have little impact on other nodes.

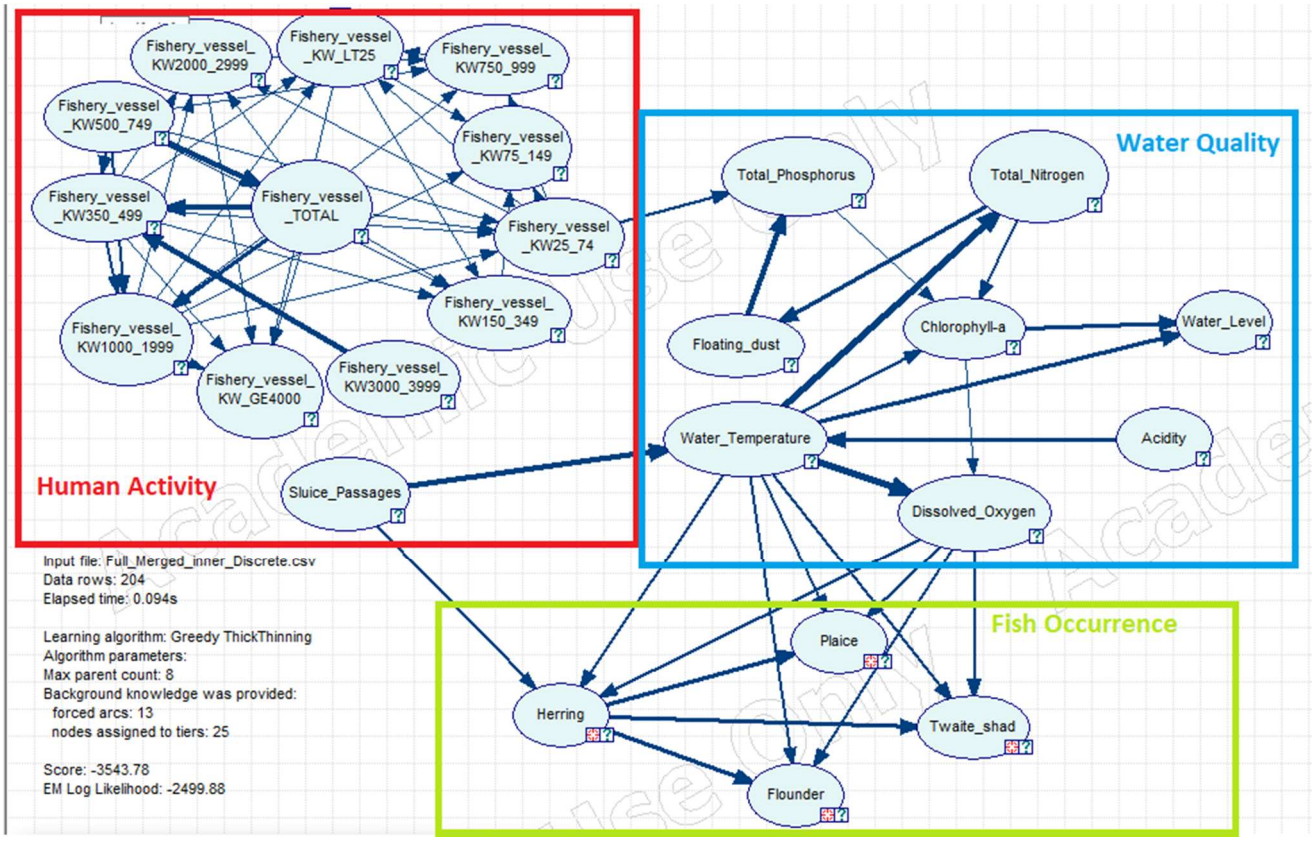


Figure 31: The first draft BN model trained by Greedy Thick Thinning including all the random variables of Human activity (red), Water quality (blue) and Fish occurrence (green). The width of edges indicates the strength of influence.

Thus, during the second attempt, all the random variables for fishery vessels other than "Fishery_vessel_KW25_74" were abandoned to make the structure clear and understandable. The parameter setting and the performance of each structure learning algorithm during the second attempt is shown in Table 9 and Table 10, respectively. In this experiment, most of the parameters remained default. The "Sample size" of Bayesian Search was set to 100 for the total number of training samples decreased to 192, and the "Significance" of PC algorithm in this experiment was set to 0.1 (default 0.05) to make nodes more likely to be connected, because smaller "Significance" caused some of the nodes were isolated in the output structure.

Table 9: The parameter setting for training BN models based on human activity, water quality and fish occurrence datasets.

Algorithm	Parameter setting
Bayesian Search	Iterations: 20, Max parent count: 8, Sample size: 100, Link probability: 0.1, Prior link probability: 0.001, Seed: 2
PC	Max adjacency: 8, Significance: 0.1
TAN	Class variable: Flounder, Seed: 2
ANB	Class variable: Flounder, Feature selection: no, Max parent count: 8, Iterations: 20, Sample size: 100, Link probability: 0.1, Prior link probability: 0.001, Seed: 2
Greedy Thick Thinning	Max parent count: 8

Table 10: The summary of the performance of each BN structure learning algorithm for training BN models based on human activity, water quality and fish occurrence datasets.

Algorithm	Running time	Network Edges (Nodes)	EM Loglikelihood	Cross-validation Accuracy
Bayesian Search (no knowledge)	0.218s	18(14)	-2105.64	Overall = 0.62 Flounder = 0.55 Plaice = 0.60 Herring = 0.73 Twaite shad = 0.59
Bayesian Search (with knowledge)	0.375s	25(14)	-2066.76	Overall = 0.62 Flounder = 0.49 Plaice = 0.63 Herring = 0.75 Twaite shad = 0.62

PC (no knowledge)	0.156s	37(14)	-2049.08	Overall = 0.59 Flounder = 0.51 Plaice = 0.58 Herring = 0.70 Twaite shad = 0.57
PC (with knowledge)	0.171s	42(14)	-1919.61	Overall = 0.63 Flounder = 0.57 Plaice = 0.59 Herring = 0.73 Twaite shad = 0.62
TAN	0.062s	25(14)	-2045.07	Overall = 0.65 Flounder = 0.57 Plaice = 0.65 Herring = 0.78 Twaite shad = 0.60
ANB	0.281s	26(14)	-2070.58	Overall = 0.64 Flounder = 0.55 Plaice = 0.63 Herring = 0.79 Twaite shad = 0.58
Greedy Thick Thinning (no knowledge)	0.063s	20(14)	-2064.22	Overall = 0.61 Flounder = 0.50 Plaice = 0.62 Herring = 0.71 Twaite shad = 0.60
Greedy Thick Thinning (with knowledge)	0.078s	25(14)	-2051.35	Overall = 0.61 Flounder = 0.49 Plaice = 0.64 Herring = 0.76 Twaite shad = 0.55
Naïve Bayes	0.047s	13(14)	-2489.21	Overall = 0.62 Flounder = 0.52 Plaice = 0.64 Herring = 0.76 Twaite shad = 0.58

According to Table 10, PC still had the highest EM loglikelihood among all the algorithms in this experiment, but because the parameter “Significance” increased, the output structure became overcomplicated with 37 edges and 14 nodes. TAN had the highest cross-validation accuracy in this experiment. However, the cross-validation accuracy of Greedy Thick Thinning was worse than the baseline, Naïve Bayes algorithm, even with the aid of background knowledge. The main reason was that Greedy Thick Thinning failed to classify the occurrence of Flounder compared to other algorithms. Especially for TAN, ANB and Naïve Bayes, those three algorithms set the node Flounder as the class variable, which means that all other variables were forced to connect to the class variable. Therefore, Greedy Thick Thinning might fail to find some key connections that were essential to classify the occurrence of Flounder in the Dutch Wadden Sea, because of the lack of training samples.

As for the role of background knowledge played, it still helped improve the EM loglikelihood for all the three algorithms which support background knowledge and improve the cross-validation accuracy of PC

algorithm a lot just like in the previous experiment. Thus, in conclusion, the PC algorithm benefitted the most from the aid of background knowledge.

In this section, to estimate the confidence interval of each algorithm, the same scenario was used to test the BN models asSection 5.1 , and the same proportion of training samples were selected to train the parameters in each iteration. According to Table 11, TAN and ANB still had a much wider confidence interval than other algorithms had. Moreover, Compared with the result in Table 8, the confidence intervals estimated in this experiment were generally wider (around 0.15 ~ 0.25) those in the previous experiment in Section 5.1(around 0.10 ~ 0.15), which indicates that the model trained by fewer samples was less reliable.

Table 11: 90% confidence interval of predicted probability of each occurrence level of Flounder when Total Phosphorus is high and Total Nitrogen is low. The parameters were trained by 120 samples (around 2/3 of total samples) in each of 100 iterations.

Algorithm	Flounder =“High”	Flounder =“Medium”	Flounder =“Low”
Bayesian Search (no knowledge)	0.27 (0.20, 0.35)	0.36 (0.29, 0.43)	0.34 (0.26, 0.45)
Bayesian Search (with knowledge)	0.32 (0.22, 0.41)	0.41 (0.29, 0.54)	0.25 (0.14, 0.37)
PC (no knowledge)	0.22 (0.14, 0.35)	0.40 (0.28, 0.53)	0.37 (0.23, 0.48)
PC (with knowledge)	0.32 (0.29, 0.37)	0.35 (0.32, 0.41)	0.31 (0.27, 0.34)
TAN	0.20 (0.04, 0.51)	0.54 (0.31, 0.73)	0.22 (0.03, 0.40)
ANB	0.17 (0.07, 0.40)	0.51 (0.32, 0.67)	0.28 (0.12, 0.45)
Greedy Thick Thinning (no knowledge)	0.34 (0.24, 0.44)	0.42 (0.32, 0.50)	0.22 (0.11, 0.36)
Greedy Thick Thinning (with knowledge)	0.30 (0.21, 0.41)	0.42 (0.30, 0.52)	0.27 (0.15, 0.40)

5.3. Scenario Testing

One of objectives in this project is to discover knowledge from data by using BN structure learning algorithm. That means the learned BN structure should be able to demonstrate some meaningful scenarios to both expert and non-expert audiences. As the BN models trained by PC algorithm had the highest EM Loglikelihood, those models were tested by ecological scenarios to see how good the PC algorithm works on extracting useful information from data without the aid of background knowledge.

Firstly, a relatively simple scenario was used to test the BN structure trained by PC including the random variables of water quality and fish occurrence group (the 1st version in Section 5.1). In this scenario, the

focus was on the effect of water temperature on dissolved oxygen. The result is shown in Figure 32, in which the probability of the "very high" dissolved oxygen concentration was decreasing with the raise of water temperature. As mentioned in section 4.2, the colder water can hold more oxygen than warmer water because of equilibrium phenomenon. Thus, the learned BN model demonstrated the correct phenomenon in this scenario

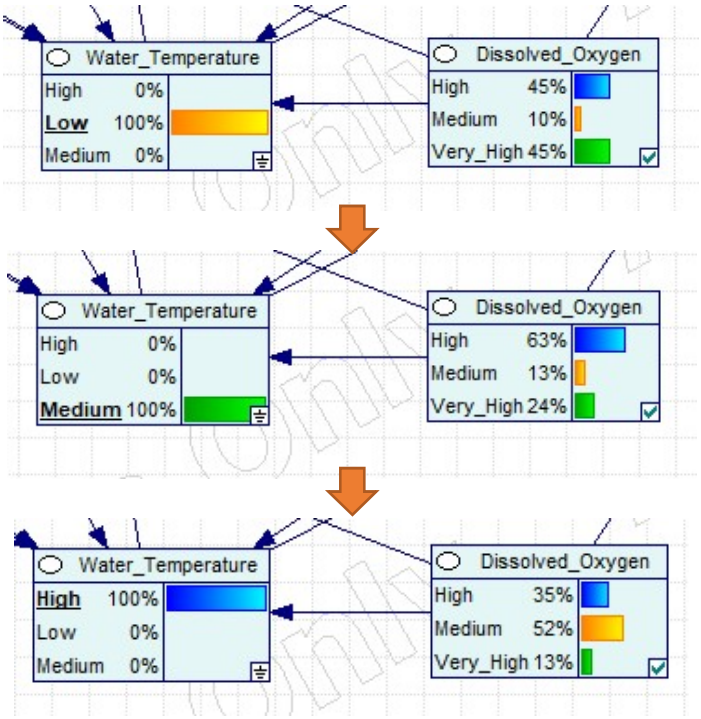


Figure 32: The change of probability distribution of Dissolved Oxygen concentration when Water Temperature changes. (PC, 1st version, no background knowledge)

Then, let's take a look at a more complicated scenario: eutrophication phenomenon. The eutrophication happens when the N/P ratio gets too low causing the excessive growth of algae and eventually deplete the oxygen in the water. This phenomenon was simulated by giving evidence of "Total_Phosphorus" = "High" and "Total_Nitrogen" = "Low" to the learned BN model. According to the result in Figure 33, in the learned BN model, when the concentration of total phosphorus was set to "High" and that of total nitrogen was set to "Low", the probability of "High" chlorophyll-a concentration increased by 10% indicating the growth of algae and aquatic plants, and the probability of "Medium" dissolved oxygen (which is the lowest concentration of dissolved oxygen recorded in the training dataset) increased significantly indicating the depletion of oxygen in the water. Thus, the learned BN model correctly simulated the eutrophication phenomenon even without the aid of background knowledge.

The test's results in Figure 32 and Figure 33 show that PC algorithm can really extract some meaningful information out of data. Moreover, the BN models learned by other BN structure learning algorithms sometimes can also demonstrate the similar results in the two scenarios above. Thus, it can be concluded

that the BN models trained by structure learning algorithms really contain some valuable information for further research and demonstration.

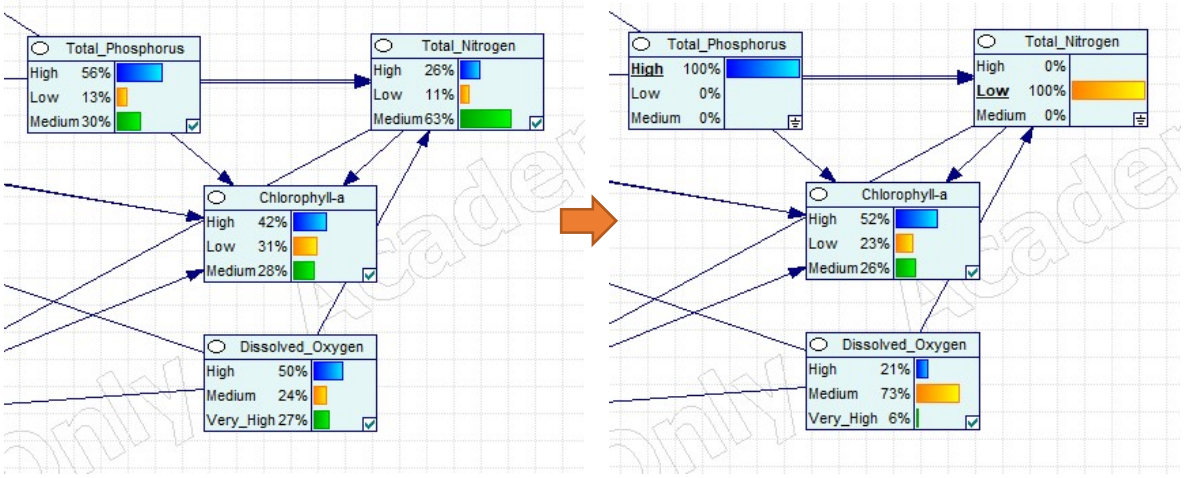


Figure 33: The demonstration of eutrophication phenomenon in the learned BN model (PC, 1st version, no background knowledge)

6. Discussion

During the experiment, the drawback and limitation of BN models were found. This Chapter gives the possible causes of some drawbacks according to further analysis of the experiment results.

6.1. The Limitation of Bayesian Network as Classifier

The BN models are regarded as classification models, When those models are used to predict the future possibilities given a certain scenario. However, there are other models which can also do the same task, such as Support Vector Machine (SVM) and Random Forest. They are among the most commonly used classification algorithms, so it is interesting to take a look at how the BN models trained by structure learning algorithm work on classification tasks compared with those commonly used classification models.

Therefore, three of structure learning algorithms which have highest overall cross-validation accuracy in Table 7 were picked up to compare with SVM and Random Forest. To make the result comparable, SVM and Random Forest were also tested by 4-fold cross validation process using the training dataset including water quality and fish occurrence variables.

According to Table 12, it is shown that both SVM and Random Forest have higher overall cross-validation accuracy and higher accuracy for each single fish species than all the learned BN models. Especially for Flounder and Plaice, the accuracy of SVM and Random Forest is 10% ~ 20% higher than that of BN models. To figure out what causes the difference, let's take a look at the details of the classification results for Flounder and Plaice in Table 13.

Table 12: The comparison of classification accuracy among BN learning algorithms and other classification algorithms.

Algorithms	**Bayesian Search (with knowledge)**	**TAN**	**Greedy Thick Thinning (with knowledge)**	**Support Vector Machine (SVM)**	**Random Forest**
Cross-validation accuracy	Overall = 0.59 Flounder = 0.59 Plaice = 0.58 Herring = 0.63 Twaite shad = 0.57	Overall = 0.59 Flounder = 0.59 Plaice = 0.56 Herring = 0.63 Twaite shad = 0.57	Overall = 0.59 Flounder = 0.58 Plaice = 0.58 Herring = 0.63 Twaite shad = 0.57	Overall = 0.69 Flounder = 0.76 Plaice = 0.75 Herring = 0.68 Twaite shad = 0.58	Overall = 0.69 Flounder = 0.77 Plaice = 0.71 Herring = 0.70 Twaite shad = 0.61

According to the result in Table 13, the classification accuracy of BN models might be too sensitive to the imbalance of training samples. The most significant phenomenon can be found in the classification result of Plaice. All the three BN models trained by different algorithms completely failed to classify the "High" concentration class of Plaice, whose training samples are the smallest among all the classes for Plaice. Only two out of 150 samples were correctly classified in this class by the BN models trained by Bayesian Search and Greedy Thick Thinning algorithm. However, SVM and Random Forest worked much better on this task.

Table 13: The cross-validation classification accuracy for each class in Flounder and Plaice. The numbers in the brackets denote (correctly classified samples / total samples).

	Bayesian Search (with knowledge)	**TAN**	**Greedy Thick Thinning (with knowledge)**	**Support Vector Machine (SVM)**	**Random Forest**
Flounder "High"	0.42 (79/187)	0.44 (83/187)	0.37 (70/187)	0.60 (113/187)	0.63 (119/187)
Flounder "Medium"	0.67 (151/223)	0.57 (128/223)	0.69 (155/223)	0.90 (201/223)	0.95 (213/223)
Flounder "Low"	0.65 (149/226)	0.75 (170/226)	0.65 (149/226)	0.76 (173/226)	0.72 (164/226)
Plaice "High"	0.01 (2/150)	0.17 (26/150)	0.01 (2/150)	0.72 (109/150)	0.50 (76/150)
Plaice "Medium"	0.87 (204/234)	0.67 (159/234)	0.87 (204/234)	0.85 (201/234)	0.94 (221/234)
Plaice "Low"	0.65 (165/252)	0.69 (175/252)	0.65 (165/252)	0.66 (168/252)	0.61 (156/252)

The reason is that the BN models are generative models while SVM and Random Forest are discriminative models. For generative models, when a sample x needs to be classified, the conditional probability $\boldsymbol{P(C_X \mid x)}$ for each class $\boldsymbol{C_X}$ will be calculated by Bayes' Theorem, and assign x to the $\boldsymbol{C_X}$ that maximize $\boldsymbol{P(C_X \mid x)}$. The formula to calculated $\boldsymbol{P(C_X \mid x)}$ is as following:

$$\boldsymbol{P(C_X \mid x) = \frac{P(x \mid C_X) \cdot P(C_X)}{P(x)}} \tag{6.1}$$

In Equation (6.1), the small number of training samples for one class $\boldsymbol{C_X}$ will cause a small $\boldsymbol{P(C_X)}$. That will make the sample less likely to be assigned to the class $\boldsymbol{C_X}$ even though $\boldsymbol{P(x \mid C_X)}$ is large. Therefore, the balance of training data is more important for generative model than for discriminative model. In other words, the generative models like the BN model are not suitable to classify imbalanced data.

6.2. The Randomness of BN Structure Learning Algorithms

During the experiments, it was noticed that the output of BN structure learning algorithms had some randomness. The randomness was brought by different reasons for different algorithms. For Bayesian Search algorithm, the randomness was brought by the different random restart points. In the experiments, the parameter "Seed" was set to 2 to keep the output of Bayesian Search unique. Figure 34 shows how EM Loglikelihood of BN models learned by Bayesian Search algorithm changes when random seed varied from 1 to 10. The change of EM Loglikelihood indicates the change of BN structure. Thus, the choice of random seed actually affects the performance of the learned BN models a lot. The randomness can be reduced by introducing more iterations, but it will add extra running time of this algorithm.

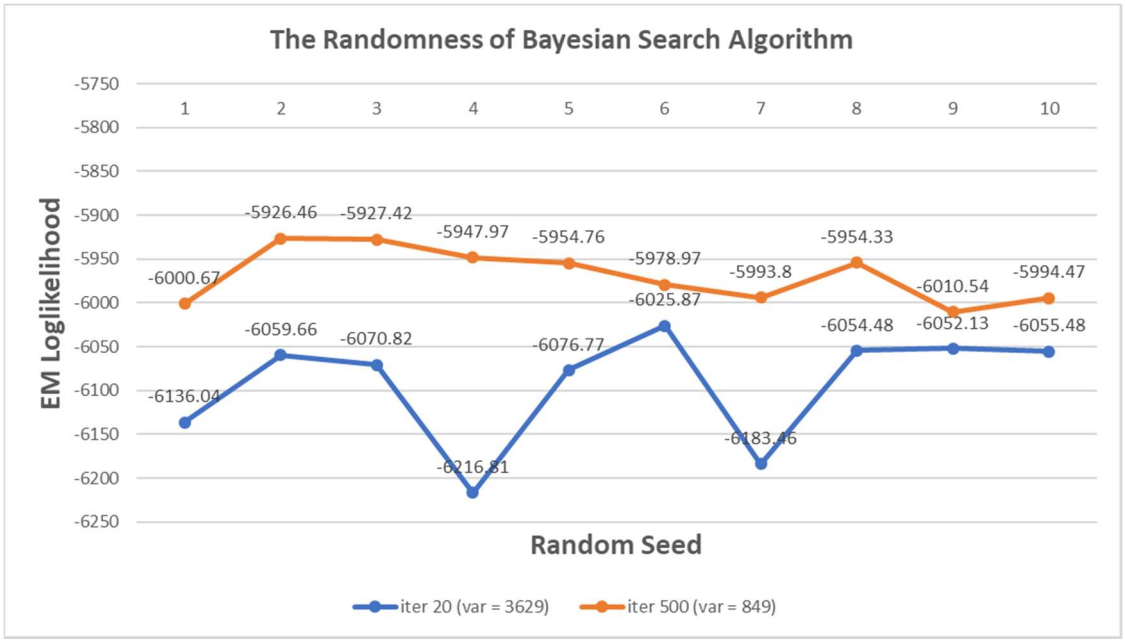


Figure 34: The EM loglikelihood of BN models learned by Bayesian Search algorithm with different random seeds.

In the PC algorithm, the orientation of undirected edges only happens when the algorithm needs to create a v-structure or avoid a v-structure. Thus, at the end of the structure learning process, there are still many undirected edges left as illustrated in Figure 35. The randomness of PC algorithm was brought by those undirected edges. To create a legal BN structure, the structure in Figure 35 must be converted to a DAG, which means that those undirected edges must be oriented without creating a cycle. However, the orientation way is not unique.

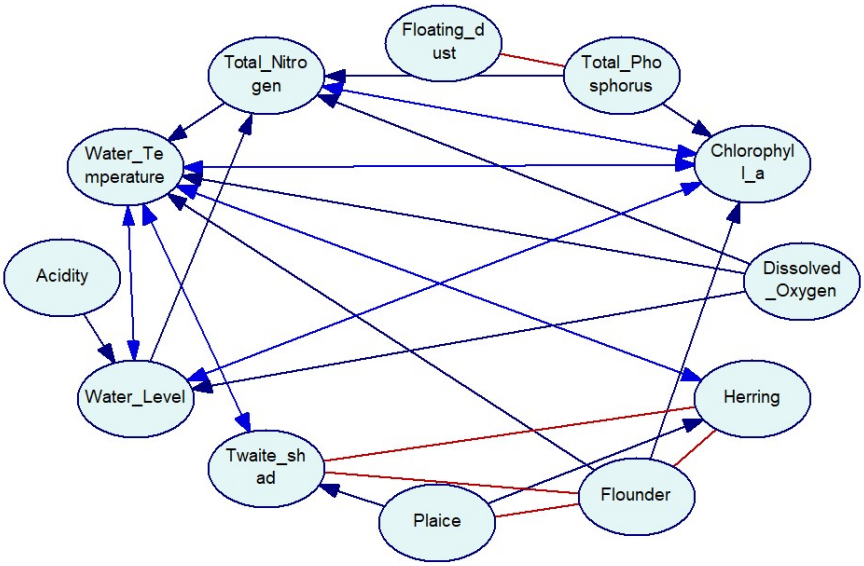
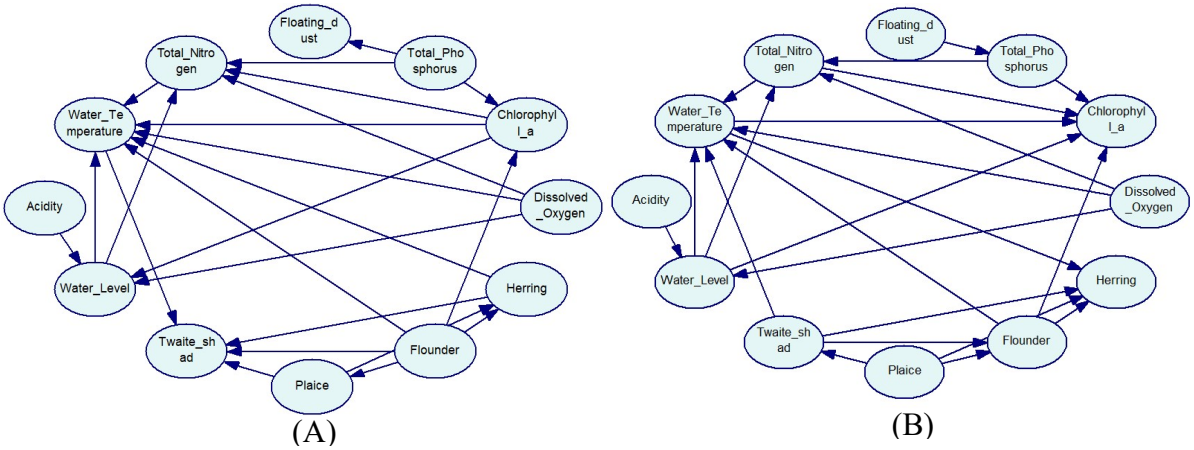


Figure 35: The original structure trained by PC algorithm. In this structure, there're several edges with no direction (Red edges) and some edges directing both sides (light blue edges). Before it becomes a legal BN structure, all the undirected edges should be oriented properly to form a DAG.

The following experiment demonstrated to what extent the orientation way can affect the performance of final BN structure. The five red edges (edges with no direction) in Figure 35 were oriented in two ways in the experiment: A. Left Orientation: Twaite_shad ← Flounder, Twaite_shad ← Herring, Plaice ← Flounder, Flounder → Herring, Total_Phosphorus ← Floating_dust; B. Right Orientation: Twaite_shad → Flounder, Twaite_shad → Herring, Plaice → Flounder, Flounder → Herring, Total_Phosphorus → Floating_dust. According to Figure 36, when orientation way changes, the EM loglikelihood and the cross-validation accuracy of final BN structure also change. Therefore, the BN structure output by PC algorithm is not a determined optimum. The performance of the structure will fluctuate with the variation of orientation way.

As for TAN, ANB, because the mutual information $I(X_i, X_j)$ is a symmetrical metric (i.e. $I(X_i, X_j) = I(X_j, X_i)$), the direction of some edges also cannot be surely defined. However, the constrains of both algorithms are relatively strong, which largely reduces their randomness. Thus, the outputs of both algorithms are not affected by the choice of random seed very much.



	A. Left Orientation structure	B. Right Orientation structure
EM Loglikelihood	-5815.85	-5780.94
Cross-validation accuracy	Overall = 0.55 Flounder = 0.55 Plaice = 0.53 Herring = 0.57 Twaite shad = 0.54	Overall = 0.54 Flounder = 0.53 Plaice = 0.51 Herring = 0.57 Twaite shad = 0.54

Figure 36: The difference between two BN structures trained by PC algorithm with different orientations.

7. Future Works

Section 6.2 has shown the randomness of BN structure learning algorithms in GeNIe, causing output instability. One of the main reason is that the metrics used to measure the strength of the relationship between two random variables are symmetrical. For example, given two randoms variables X_i, X_j, In TAN, ANB and Greedy Thick Thinning, the relative entropy $I(X_i, X_j)$ should be calculated to determine whether the two variables are strongly related. However, as the relative entropy is a symmetrical metric, meaning $I(X_i, X_j) = I(X_j, X_i)$. Thus the metric can only determine whether the two random variable nodes should be connected but can not judge the direction of the edge. Consequently, the algorithms based on such metric must randomly set the direction of some edges at the end of structure learning process. To make the output BN structure unique, an asymmetrical metric needs to be used in the learning process. The "Strength of influence" proposed by Koiter [42] is an ideal asymmetrical metric to measure the significance of a single direction connection. The basic idea of "Strength of influence" is to calculate the distance between different conditional distributions of one variable given one of its parents.

For example, Figure 37 shows two example conditional distributions $P(Y|X)$. The conditional distribution (A) shows that X does not have any influence on Y, because the value of X does not affect the distribution of Y, and the Euclidean distance between two conditional distributions $P(Y|X = 0)$ and $P(Y|X = 1)$ is $E([0.98,0.02],[0.98,0.02]) = 0$. The conditional distribution (B) shows that X can affect the distribution of Y, and the Euclidean distance between two conditional distributions $P(Y|X = 0)$ and $P(Y|X = 1)$ in this case is $E([0.8,0.2],[0.5,0.5]) = 0.42$. Thus, if the "Strength of influence" is used as the metric during the structure learning process, X and Y with a conditional distribution in (B) will be more likely to be connected from X to Y. It should be noted that the distance is a symmetrical metric, but when the direction of the edge between X and Y changes, the conditional distribution between them will also change with it. Thus, the "Strength of influence" itself is not a symmetrical metric.

		X	
		0	1
Y	0	0.98	0.98
	1	0.02	0.02

(A)

		X	
		0	1
Y	0	0.8	0.5
	1	0.2	0.5

(B)

Figure 37: The example of conditional distributions between random variable X and Y. Each random variable can take two values 0 or 1.

During the machine learning process, having a large dataset is crucial to learn a generalized model. However, in reality, large dataset is quite scarce. In practice, people use data augmentation techniques to generate more data based on the existing data to enlarge the training dataset for a better performance. The

data augmentation is usually applied to images and the commonly used methods include flip, rotation, scale and adding noise. Recently, the generative adversarial network (GAN) [43] and Variational AutoEncoder (VAE) [44] are becoming more and more popular as a data augmentation technique. As the BN structure learning algorithm can estimate the probability distribution from samples, the learned BN model can also be used to generate new data based on existing dataset. Thus, the BN structure learning algorithm has great potential to be a powerful data augmentation technique.

Nowadays, serious games are applied to environmental management field to help future researchers and professionals obtain first-hand experience on practical environmental sustainability challenges and to promote awareness about sustainable resource planning and management [45]. Utilizing BN models trained from in-situ measurement data can bring uncertainty and reality to serious games that makes the games more appealing and educational.

8. Conclusion

In this project, five different BN structure learning algorithms were applied to train the BN models for fish occurrence in the western Dutch Wadden Sea from in-situ measurement data with the aid of background knowledge. Two training datasets including different sets of random variables and different amounts of samples were made to train the BN models separately. One dataset including only part of random variables about water quality and fish occurrence has more samples, the other dataset including all the random variables about human activity, water quality and fish occurrence has fewer samples. Based on those two datasets, two versions of BN models were learned.

The result shows that the PC algorithm is the best choice to extract useful information from data (i.e. knowledge discovery), while TAN and ANB are good at making predictions(i.e. density estimation). However, TAN and ANB will fail to make a reliable prediction when only the values of few nodes are given. By comparing the confidence intervals between two different versions of BN models, the BN models trained by more samples could make more reliable predictions. The aid of background knowledge in this project helped to improve the likelihood between the BN model and the data, but it cannot always improve the classification accuracy significantly.

The experiment also shows that the limitation of generative models like BN as classifier. The BN model is too sensitive to the imbalance of training dataset when making the prediction. Thus, to train a BN model as a good classifier, the training dataset should be strictly balanced (especially for target variables).

Some of BN structure learning algorithms involve randomness in their learning process, yielding uncertainty of their output BN models. For Bayesian Search algorithm, the randomness can be reduced by running the algorithm with more iterations. For the algorithms based on certain metrics to decide which nodes to be connected, using an asymmetric metric like “strength of influence” can solve the randomness problem.

In conclusion, this project proposed a methodology to learn a BN model from both data and experts’ knowledge, and the BN models built by this methodology have been proved to be useful for extracting meaningful knowledge out of data and making the prediction in the future.

References

[1] "Wadden Sea - UNESCO World Heritage Centre," UNESCO, 2009. [Online]. Available: https://whc.unesco.org/en/list/1314.

[2] I. Tulp, L. Bolle, H. Haslob, P. de Vries, N. Jepsen, J. Scholle and H. van der Veer, "Fish," in *Wadden Sea Quality Status Report 2017*, Wilhelmshaven, Germany, Common Wadden Sea Secretariat, 2017.

[3] E. Folmer, "The Utility of Bayesian Belief Network for analysis of cumulative effects in the Dutch Wadden Sea," Waddenacademie, 2016.

[4] D. Heckerman, "A Tutorial on Learning with Bayesian Networks," Microsoft Research Advanced Technology Division, Redmond, 1996.

[5] D. Koller and N. Friedman, Probablistic Graphical Models: Principles and Techniques, Cambridge, Massachusetts: The MIT Press, 2009.

[6] J. Blitzstein and J. Hwang, Introduction to Probability, CRC Press, 2014.

[7] K. I. Park, Fundamentals of Probability and Stochastic Processes with Applications to Communications, Springer, 2018.

[8] M. E. Maron, "Automatic Indexing: An Experimental Inquiry," *Journal of the ACM,* vol. 8, no. 3, pp. 404-417, 1961.

[9] I. Ben Gal, "Bayesian Networks," in *Encyclopedia of Statistics in Quality and Reliability*, John Wiley & Sons, 2007.

[10] "BayesFusion," BayesFusion, LLC, 2019. [Online]. Available: http://www.bayesfusion.com/.

[11] "Norsys - Netica Application," Norsys Software Corp., 2019. [Online]. Available: https://www.norsys.com/netica.html.

[12] "Hugin Expert," HUGIN EXPERT A/S , 2019. [Online]. Available: www.hugin.com.

[13] T. Compton, S. Holthuijsen, A. Koolhaas, A. Dekinga, J. Ten Horn, J. Smith, Y. Galama, M. Brugge, D. van der Wal, J. van der Meer, H. van der Veer and T. Piersma, "Distinctly variable mudscapes:

Distribution gradients of intertidal macrofauna across the Dutch Wadden Sea," *Journal of Sea Research,* vol. 82, pp. 103-116, 2013.

[14] A. P. Oost, C. Winter, P. Vos, F. Bungenstock, R. Schrijvershof, B. Röbke, J. Bartholdy, J. Hofstede, A. Wurpts and A. Wehrmann, "Geomorphology," in *Wadden Sea Quality Status Report 2017*, Wilhelmshaven, Germany, Common Wadden Sea Secretariat, 2017.

[15] H. van der Veer, R. Berghahn, J. Miller and A. Rijnsdorp, "Recruitment in flatfish, with special emphasis on North Atlantic species: Progress made by the Flatfish Symposia," *ICES Journal of Marine Science,* vol. 57, pp. 202-215, 2000.

[16] M. Elliott, A. Whitfield, I. Potter, S. Blaber, D. Cyrus, F. Nordlie and T. Harrison, "Fish and Fisheries 8," in *The guild approach to categorizing estuarine fish assemblages: a global review.* , 2007, pp. 241-268.

[17] S. Hulscher, P. Meire, G. Rienstra and J. Urai, "Position paper Zoutwinning onder de Waddenzee," 2016.

[18] N. C. KRAUS, "Reservoir model of ebb-tidal shoal evolution and sand bypassing.," *Journal of Waterway, Port, Coastal and Ocean Engineering,* vol. 126, pp. 305-313, 2000.

[19] M. J. F. STIVE, Z. B. WANG, M. CAPOBIANCO, P. RUOL and M. C. BUIJSMAN, "Morphodynamics of a tidal lagoon and the adjacent coast.," in *DRONKERS & SCHEFFERS, eds. Physics of Estuaries and Coastal Seas*, Rotterdam, Balkema, 1998, pp. 397-407.

[20] L. Van Walraven, V. T. Langenberg, R. Dapper, J. I. Witte, A. F. Zuur and H. W. Van der Veer, "Long-term patterns in 50 years of scyphomedusae catches in the western Dutch Wadden Sea in relation to climate change and eutrophication," *JOURNALOF PLANKTON RESEARCH,* vol. 37, pp. 151-167, 2015.

[21] J.-B. Bjarnason, W. Günther and H. Revier, "Tourism," in *Wadden Sea Quality Status Report*, Wilhelmshaven, Germany, Common Wadden Sea Secretariat, 2017.

[22] M. Soley-Bori, "Dealing with missing data: Key assumptions and methods for applied analysis," Boston University, Boston, 2013.

[23] A. Swalin, "How to Handle Missing Data – Towards Data Science," Towards Data Science Inc., 31 January 2018. [Online]. Available: https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4. [Accessed 21 May 2019].

[24] S. Kotsiantis and D. Kanellopoulos, “Discretization Techniques: A recent survey,” *GESTS International Transactions on Computer Science and Engineering,* vol. 32, no. 1, pp. 47-58, 2006.

[25] B. Oram, “Water Research Center - Dissolved Oxygen in Water, Streams, Watershed,” B.F. Environmental Consultants Inc., 2014. [Online]. Available: https://www.water-research.net/index.php/dissovled-oxygen-in-water. [Accessed 22 May 2019].

[26] J. Schalles, “OPTICAL REMOTE SENSING TECHNIQUES TO ESTIMATE PHYTOPLANKTON CHLOROPHYLL a CONCENTRATIONS IN COASTAL,” *Remote Sensing of Aquatic Coastal Ecosystem Processes: Science and Management Applications,* pp. 27-79, 2006.

[27] Y. Zhang, C. Song, L. Ji, Y. Liu, J. Xiao, X. Cao and Y. Zhou, “Cause and effect of N/P ratio decline with eutrophication aggravation in shallow lakes,” *Science of the Total Environment,* vol. 627, pp. 1294-1302, 2018.

[28] P. A. White, J. Kalff, J. B. Rasmussen and J. M. Gasol, “The effect of temperature and algal biomass on bacterial production and specific growth rate in freshwater and marine habitats,” *Microbial Ecology,* vol. 21, no. 1, pp. 99-118, 1991.

[29] T. W. Davis, D. L. Berry, G. L. Boyer and C. J. Gobler, “The effects of temperature and nutrients on the growth and dynamics of toxic and non-toxic strains of Microcystis during cyanobacteria blooms,” *Harmful Algae,* vol. 8, no. 5, pp. 715-725, 2009.

[30] L. Guilioni, J. Wéry and J. Lecoeur, “High temperature and water deficit may reduce seed number in field pea purely by decreasing plant growth rate,” *Functional Plant Biology,* vol. 30, no. 11, pp. 1151-1164, 2003.

[31] M. I. Badran, “Dissolved oxygen, chlorophyll a and nutrients: seasonal cycles in waters of the Gulf Aqaba, Red Sea,” *Aquat. Ecosys. Health Manage,* vol. 4, no. 2, pp. 139-150, 2001.

[32] N. Jonsson, “Influence of Water Flow, Water Temperature and Light on Fish Migration in Rivers,” *Nordic Journal of freshwater research,* vol. 66, pp. 20-35, 1991.

[33] S. A. Murawski, “Climate Change and Marine Fish Distributions: Forecasting from Historical Analogy,” *Transactions of the American Fisheries Society,* vol. 122, no. 5, pp. 647-658, 1993.

[34] G. F. Cooper and E. Herskovits, “A Bayesian method for the induction of probabilistic networks from data,” *Machine Learning,* vol. 9, no. 4, pp. 309-347, 1992.

[35] T. Tran, "AIND 6," A Medium Corporation, 5 4 2017. [Online]. Available: https://medium.com/@baotrungtn/aind-6-fccc24729f0a. [Accessed 30 6 2019].

[36] P. Spirtes, C. Glymour and R. Scheines, Causation, Prediction, and Search, Cambridge: The MIT Press, 1993.

[37] P. Spirtes and C. Glymour, "An Algorithm for Fast Recovery of Sparse Causal Graphs," *Social Science Computer Review,* vol. 9, no. 1, pp. 62-72, 1991.

[38] N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian Network Classifiers," *Machine Learning,* vol. 29, no. 2, pp. 131-163, 1997.

[39] J. Cheng, D. A. Bell and W. Liu, "An Algorithm for Bayesian Belief Network Construction from Data," in *IN PROCEEDINGS OF AI & STAT'97*, 1997, pp. 83-90.

[40] S. L. Lauritzen, "The EM algorithm for graphical association models with missing data," *Computational Statistics & Data Analysis,* vol. 19, no. 2, pp. 191-201, 1995.

[41] M. Egmont-Petersen, A. Feelders and B. Baesens, "Confidence intervals for probabilistic network classifiers," *Computational Statistics & Data Analysis,* vol. 49, pp. 998-1019, 2005.

[42] J. Koiter, "Visualizing Inference in Bayesian Networks," TUDelft, Delft, 2006.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Networks," *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014),* p. 2672–2680, 2014.

[44] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint,* no. arXiv:1312.6114, 2013.

[45] K. Madani, T. W. Pierce and A. Mirchi, "Serious games on environmental management," *Sustainable Cities and Society,* vol. 29, pp. 1-11, 2017.

Appendix A: Random variable nodes and discrete classes

Name	Category	Values	Note
Sluice Passages	Human Activity (Tourism)	Low: (0, 1096] Medium: (1096, 13471] High: (13471, 30000] (unit: number of times)	Total number of sluice passages per month in the Dutch Wadden Sea area indicates the number of tourists.
Fishery_vessel_KW_LT25	Human Activity (Fishery)	Low: (100, 144] Medium: (144, 163] High: (163, 300] (unit: number of ships)	Total number of active fishery ships with engine power less than 25 KW per year.
Fishery_vessel_KW25_74	Human Activity (Fishery)	Low: (80, 100] Medium: (100, 109] High: (109, 130] (unit: number of ships)	Total number of active fishery ships with engine power between 25 KW and 74 KW per year.
Fishery_vessel_KW75_149	Human Activity (Fishery)	Low: (85, 102] Medium: (102, 151] High: (151, 200] (unit: number of ships)	Total number of active fishery ships with engine power between 75 KW and 149 KW per year.
Fishery_vessel_KW150_349	Human Activity (Fishery)	Low: (260, 270] Medium: (270, 286] High: (286, 310] (unit: number of ships)	Total number of active fishery ships with engine power between 150 KW and 349 KW per year.
Fishery_vessel_KW350_499	Human Activity (Fishery)	Low: (30, 40] Medium: (40, 47] High: (47, 50] (unit: number of ships)	Total number of active fishery ships with engine power between 350 KW and 499 KW per year.
Fishery_vessel_KW500_749	Human Activity (Fishery)	Low: (46, 49] Medium: (49, 51] High: (51, 55] (unit: number of ships)	Total number of active fishery ships with engine power between 500 KW and 749 KW per year.
Fishery_vessel_KW750_999	Human Activity (Fishery)	Low: (5, 17] Medium: (17, 26] High: (26, 30] (unit: number of ships)	Total number of active fishery ships with engine power between 750 KW and 999 KW per year.
Fishery_vessel_KW1000_1999	Human Activity (Fishery)	Low: (70, 96] Medium: (96, 119] High: (119, 160] (unit: number of ships)	Total number of active fishery ships with engine power between 1000 KW and 1999 KW per year.
Fishery_vessel_KW2000_2999	Human Activity (Fishery)	Low: (3, 11] Medium: (11, 28] High: (28, 30] (unit: number of ships)	Total number of active fishery ships with engine power between 2000 KW and 2999 KW per year.
Fishery_vessel_KW3000_3999	Human Activity (Fishery)	Low: 1 High: 2 (unit: number of ships)	Total number of active fishery ships with engine power between 3000 KW and 3999 KW per year.
Fishery_vessel_KW_GE4000	Human Activity (Fishery)	Low: (7, 8] Medium: (8, 10] High: (10, 13] (unit: number of ships)	Total number of active fishery ships with engine power greater than 4000 KW per year.
TOTAL	Human Activity (Fishery)	Low: (800, 841] Medium: (841, 994] High: (994, 1200]	Total number of active fishery ships per year.

		(unit: number of ships)	
Total Nitrogen	Water Quality (Nutrient)	Low: (0, 0.5] Medium: (0.5, 0.9] High: (0.9, 2] (unit: mg/l)	Daily average value per month
Total Phosphorus	Water Quality (Nutrient)	Low: (0, 0.05] Medium: (0.05, 0.07] High: (0.07, 0.4] (unit: mg/l)	Daily average value per month
Dissolved Oxygen	Water Quality (Nutrient)	Medium: (5, 8] High: (8, 10] Very High: (10, 16] (unit: mg/l)	Daily average value per month
Chlorophyll-a	Water Quality (Nutrient)	Low: (0, 3] Medium: (3, 6] High: (6, 70] (unit: ug/l)	Daily average value per month
Floating Dust	Water Quality (Nutrient)	Low: (0, 20] Medium: (20, 37] High: (37, 200] (unit: mg/l)	Daily average value per month
Acidity	Water Quality (Climate Change)	Neutral: (7, 7.6] Moderately alkaline: (7.6, 8.4] Strongly alkaline: (8.4, 9] (unit: ph)	Daily average value per month
Water Temperature	Water Quality (Climate Change)	Low: (-1, 7] Medium: (7, 15] High: (15, 30] (unit: oC)	Daily average value per month
Water Level	Water Quality (Climate Change)	Low: (-30, -5] Medium: (-5, 5] High: (5, 40] (unit: cm)	The relative water level compared to normal Amsterdam level (N.A.P.)
Flounder	Fish Occurrence	Low: [0, 1] Medium: (1, 4] High: (4, 20] (unit: number of occurrence)	Estuarine resident Daily average occurrence per month
Plaice	Fish Occurrence	Low: [0, 1] Medium: (1, 4] High: (4, 20] (unit: number of occurrence)	Marine juvenile Daily average occurrence per month
Herring	Fish Occurrence	Low: [0, 1] Medium: (1, 2] High: (2, 20] (unit: number of occurrence)	Marine juvenile Daily average occurrence per month
Twaite shad	Fish Occurrence	Low: [0, 1] Medium: (1, 2] High: (2, 20] (unit: number of occurrence)	Diadromous species Daily average occurrence per month