# Network Analysis Methods for Smart Inspection in the Transport Domain

Gerrit Jan de Bruin

# Network Analysis Methods for Smart Inspection in the Transport Domain

Gerrit Jan de Bruin

Human Environment and Transport
Inspectorate
*Ministry of Infrastructure*
*and Water Management*

Universiteit
Leiden

The work in this book was funded by the Ministry of Infrastructure and Water Management through Leiden University. The author used facilities at the Leiden Centre of Data Science, the Leiden Institute of Advanced Computer Science, and the Innovation and Data Lab of the Human Environment and Transport Inspectorate (part of the Ministry of Infrastructure and Water Management).

SIKS Dissertation Series No. 2023-21
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.



Typesetting & Figures: LaTeX, `Gephi`, `diagrams.net`, `Matplotlib`, and `Seaborn`.
All software used is open-source.
Open access: https://scholarlypublications.universiteitleiden.nl/
Cover design: Wolters Vormgeving
Printed by: Gildeprint, Enschede, the Netherlands, https://www.gildeprint.nl/
ISBN: 978-94-6419-967-3

# Network Analysis Methods for Smart Inspection in the Transport Domain

## Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op donderdag 16 november 2023
klokke 13.45 uur

door

Gerrit Jan de Bruin
geboren te Amersfoort
in 1993

**Promotor:**
Prof.dr. H.J. van den Herik

**Copromotores:**
Dr. F.W. Takes
Dr. C.J. Veenman

**Promotiecommissie:**
Prof.dr. H.C.M. Kleijn
Prof.dr. T.H.W. Bäck
Dr. W.A. Kosters
Prof.dr. J.N. Kok (Universiteit Twente)
Prof.dr. N.V. Litvak (Universiteit Twente)

# Preface

This thesis is part of an extensive collaboration between the Dutch Ministry of Infrastructure and Water Management (I&W) and Leiden University. My first personal encounter with I&W was during my master's study in Analytical Chemistry. I was determined to work on a subject with societal relevance, leading me to Jasper van Vliet, who worked at the Inspectie Leefomgeving en Transport (ILT), part of I&W. The aim of my master thesis was to conduct efficient compliance monitoring of cargo ship's fuel by using the information from chemical sensors. The thesis made it to a letter to the parliament [88, 89]. Afterward, Jasper invited me to participate in a new Ph.D. project of the Ministry. In this project, the ambition is to arrive at intelligence-led vehicle inspection by risk assessments. Two research directions were launched to explore the risk assessment of vehicles: (1) the application of machine learning techniques and (2) the application of network science techniques.

During my Ph.D. research, many developments occurred related to the topics of the thesis. I would like to mention two specific events that have had an impact on my research. The first one is the introduction of the General Data Protection Regulation (GDPR) in 2018. This law requires the use of transparent models that allow for an explanation of the results achieved. The new law led to increased awareness of the importance of fair data and fair models. The second event is the upset of the Dutch childcare benefits ("De Toeslagenaffaire"). In 2019 it became painfully clear how things can go wrong when authorities are (1) relying on biased data and (2) using models that are not validated fairly. I will address the two points (biases and non-validated models) in my thesis, although I work with non-personal data. They are in particular relevant for the proposed procedure to implement a smart inspection of cargo ships in Chapter 6.

Working both at Leiden University and at the ILT allowed me to interact with the wonderful world of academia and to stay in close contact with a governmental organization that makes a big impact by ensuring safe transportation and reducing the environmental pollution in the Netherlands. This enriching combination has helped me to create this thesis, for which I am grateful.

Gerrit Jan de Bruin, Utrecht, March 8th, 2023

# Contents

# List of Abbreviations

**AA** Adamic-Adar ..... 24, 29, 30, 32
**Aminer** Arnetminer ..... 35
**ANPR** Automatic Number-Plate Recognition ..... 13, 68, 73, 82
**AP** Average Precision ..... 57, 60, 61
**AUC** Area Under the Receiver Operating Characteristic Curve ..... 33, 36, 38, 45, 57, 76, 103–105, 108–110, 113
**CN** Common Neighbors ..... 24, 29, 30, 32, 56
**DBLP** Digital Bibliography & Library Project ..... xix, 30, 31, 34, 37, 39, 43, 44
**FOC** "Flag Of Convenience" ..... 98
**GC** Giant Component ..... xx, 6, 9, 10, 57, 71, 72, 80, 81, 83, 84, 87–89, 91, 106, 107
**GDPR** General Data Protection Regulation ..... vii
**GPS** Global Positioning System ..... 66
**HPLP** High-Performance Link Prediction ..... 55
**I&W** Infrastructure and Water Management ..... vii, 13, 82
**ILT** Inspectie Leefomgeving en Transport (English: Human Environment and Transport Inspectorate) ..... vii, viii, 1, 2, 5
**IMO** International Maritime Organization ..... 98
**JC** Jaccard Coefficient ..... 24, 30, 32
**KONECT** KOblenz NEtwork CollecTion ..... 35
**MoU** Memorandum of Understanding ..... 94, 98, 113
**NIR** New Inspection Regime ..... 94–96
**NL** the Netherlands ..... 85
**NN** Number of Neighbors ..... 56
**NSI** Netherlands Shipping Inspectorate ..... 2
**PA** Preferential Attachment ..... 24, 30, 32
**PF** PropFlow ..... 56
**PSC** Port State Control ..... 94

**RDW** Rijksdienst voor het Wegverkeer (English: Netherlands Vehicle Authority) ..... 85
**ROC** Receiver Operating Characteristic ..... xv, xx, 33, 76, 77
**SCAFF** Splitting Criterion Area under the curve for Fairness ..... 103, 106
**SNAP** Stanford Network Analysis Project ..... 35
**SP** Shortest Paths ..... 56
**WiFi** Wireless Fidelity ..... 66
**WTF** Weighted Temporal Features ..... 32
**XGBoost** eXtreme Gradient Boosting ..... 36, 38, 39, 56, 59

# List of Definitions

# List of Figures

# List of Tables

# 1 Introduction

Transportation supports our modern global economy like never before. Millions of vehicles, such as ships, planes, trains, and trucks, allow for truly worldwide trade [180], for most humans increasing welfare to levels previously thought to be unreachable [169]. However, the global transportation system also has its challenges; several dangers may come with the modern way of transporting goods and people, such as (1) environmental pollution, (2) culpable accidents, and (3) labor exploitation [88]. Reducing the severe risks involved is of utmost importance. Policy makers have recognized the need to limit transportation risks; therefore, national laws and international treaties have been developed to make transportation as safe and clean as possible [55]. The mere existence of laws and treaties does not immediately eliminate all of these dangers because vehicle operators may choose not to comply with legislation. Therefore, government inspectors periodically check vehicles to ensure compliance. Examples of noncompliant dangerous behavior include lack of safety training and disregarding rest periods (dangerous to humans) or lack of waste treatment (dangerous to, e.g., the environment and wildlife). Inspectorates have the job of ensuring compliance in the transport domain.

In the Netherlands, it is the responsibility of the Human Environment and Transport Inspectorate, in Dutch "Inspectie Leefomgeving en Transport (ILT)", to inspect vehicles and their operators. The inspectorate monitors 160 different policy issues and takes enforcement action when necessary [90]. Examples of issues are (1) the quality of fuel used in vehicles, (2) working conditions for transport personnel, and (3) illegal dumping of waste. Well-functioning inspectorates make a country a healthier, happier, cleaner, more prosperous, and safer place to live [136].

The remaining part of this introductory chapter is structured as follows. We start by exploring smart vehicle inspection in Section 1.1. At the end of this section, we introduce our contribution in the form of automated techniques that help ensure smart vehicle inspection. In Section 1.2, we introduce networks, a powerful model for achieving this task. Section 1.3 dives into one specific representation of a network where temporal

information is available, i.e., the temporal network. After that, in Section 1.4, we investigate the prediction of new network links as an approach to better understand the network's dynamics. Subsequently, in Section 1.5, we focus on the characteristics of the data used throughout the thesis, being transport networks. In Section 1.6, we provide the problem statement and research questions. The research methodology is presented in Section 1.7. Finally, an overview of the thesis is provided together with our contributions in Section 1.8.

## 1.1 Smart vehicle inspection

A major challenge for inspectorates is achieving maximum compliance towards legislation with finite inspection capacity [136]. For example, the cargo shipping industry is responsible for around 80% of global trade movements [190]. Historically, shipping inspectorates selected a random sample from all ships entering a port, such that all ships have an equal probability of being inspected. As a result, an inspectorate with a limited number of inspectors would only sporadically encounter noncompliant behavior at a ship, assuming that noncompliance is rare. Hence, ship owners might think there is no need to comply with legislation because noncompliant behavior is unlikely to be noticed. It can result in neglecting safety procedures and, therewith, more dangerous behavior.

Many inspectorates are limited in the number of inspectors they can employ. In the Netherlands, the Netherlands Shipping Inspectorate (NSI, part of ILT) can only inspect twelve ships per week [87, 146], while over 500 merchant ships arrive weekly in the port of Rotterdam alone [160]. Therefore, many inspectorates (including Netherlands Shipping Inspectorate (NSI)) are looking for innovative methods to maximize compliance and thereby minimize riskful behavior. One way of doing so is by improving the assessment procedure of vehicles for inspection so that more time can be spent on noncompliant vehicles. Traditionally, rule-based systems are considered to this end. The rules in these systems are based on expert knowledge. In this work, we consider the use of historical data to obtain better assessments of vehicles. Inspectorates performing data-driven assessment for inspections, as defined in Definition 1, are more likely to find noncompliant vehicles and are thus more effective in detecting dangerous behavior [45, 86, 137]. In this thesis, the terms "inspectorate" and "inspections" will be used solely in the context of the inspection of vehicles and their operators.

> **Definition 1.** *Data-driven assessment for inspection* is the process that uses (historical) data to determine what entities are likely associated with noncompliant behavior and thus need an inspection.

Taking the assessment procedure of vehicles for inspection one step further means that we not only make data-driven assessments (which may still involve human decisions) but require the assessment to be done in a so-called *smart* way, as detailed in Definition 2.

> **Definition 2.** *Smart inspection* is performed when a data-driven approach is taken to assess vehicles likely associated with noncompliant behavior in an accurate, automated, fair, and interpretable way.

Doing smart inspection ensures that vehicle owners are motivated to comply with legislation because they know that noncompliance will likely result in inspections and subsequent fines or legal consequences.

We briefly explain what we consider an (1) accurate, (2) automated, (3) fair, and (4) interpretable assessment in this paragraph and describe the last three aspects in more detail in the following subsections. While an accurate assessment is a logical consequence of an adequately performing machine learning model, the other three aspects deserve further elaboration.

- An *accurate* assessment is an assessment that closely matches the true outcome.
- An *automated* assessment is performed without human intervention and can automatically adjust to new data.
- A *fair* assessment does not discriminate towards sensitive characteristics.
- In an *interpretable* assessment, the entire approach, including how it arrives at an assessment, is clear to humans.

## Automated assessment

Ideally, vehicle assessment for inspection should be performed in an *automated* manner, considering many vehicles in a limited time, with little time-consuming human intervention. This moves away from the classically considered rule-based approach, in which solely human intelligence is used. In the current work, we consider *machine learning* methods for the assessment process. Machine learning is the process of learning (or equivalently, training) a model from examples of data represented by characteristic features [17, 76, 137]. The learned model can then make predictions about new (unseen) data. Features refer to characteristic properties of the examples provided to the machine learning algorithm. Engineering these features is an essential step in machine learning and can significantly affect the performance of a model. In the case of transport vehicles, features include vehicle characteristics, such as country of registration or maximal transport capacity. A machine learning model should be validated (Definition 3) and tested (Definition 4) to make sure to assess its performance.

> **Definition 3.** *Model validation* is the process of evaluating the performance and reliability of (possibly multiple) models on unseen data to select the best-performing model [76].

> **Definition 4.** *Model testing* is the final process of evaluating the performance of a model on unseen data after the model is fully trained [76].

Model validation and testing are done by dividing the examples into disjoint sets of data, usually the (1) train, (2) validation, and (3) test set [76]. A machine learning algorithm

then uses (1) the train set to learn the model, (2) the validation set to perform model validation, and (3) the test set to do model testing.

The power of machine learning models lies in their ability to easily identify trends and patterns in the data that are too complex for humans to find. Moreover, and especially useful in our setting, machine learning models can handle more vehicles than humans. The assessment of vehicles for inspection is thus ideally performed by a machine learning model.

### Fair assessment

Assessment of vehicles should be performed in a fair manner (called fair assessment) to prevent discriminatory use of sensitive features. In our setting, sensitive features are properties of vehicles that the model should not consider as features for the model because of, e.g., legal restrictions or ethical considerations. An example specific to the transport domain is the registration country of a vehicle. There are at least two reasons why it is undesirable to use the registration country.

First, some countries are subject to more rigorous inspections than others. Therefore, historical data can be biased toward certain registration countries. This bias can occur when the inspection process is not standardized across all countries, leading to unequal levels of scrutiny. Second, vehicle operators themselves can initiate changes in the country of registration, thereby influencing the assessment outcome [131].

Now that we have established that one or more sensitive features can be present in the data, we mention two ways to limit the use such of sensitive information and thus arrive at a more fair model for the assessment of vehicles.

The first way is to disregard any sensitive information altogether. A clear advantage is that the sensitive information itself cannot be used to make a prediction. A significant drawback is that the sensitive information may correlate with non-sensitive information, resulting in the indirect use of sensitive information [13]. The second way is to use models that can produce fair assessments by special treatment of sensitive information, further detailed in Definition 5 below.

> **Definition 5.** A *fair model* produces assessments that do not discriminate towards characteristics of the example that are deemed sensitive.

Fair models minimize the negative outcome for sensitive groups by so-called decorrelation of assessments with sensitive information [75, 96, 157, 214]. Decorrelation is the reduction of correlation between sensitive information and the predicted outcome of a model. Moreover, recently developed methods allow users to tune the fairness-performance trade-off by controlling the level of decorrelation with the sensitive information. As such, these models can prevent sensitive information from being used in the assessment from being exploited, ensuring similar outcomes for the sensitive and non-sensitive groups. They urge careful consideration of the balance between performance on the one hand

and the restricted use of sensitive information on the other hand. In Chapter 6, we use a fair model and describe how fairness can be quantified.

The country of registration is not the only feature that may be deemed sensitive. In general, vehicle operators can manipulate static administrative information to arrive at a more favorable risk assessment. Examples of *static administrative* information include insurance company, vehicle's type, size, and construction year. In contrast, *behavioral* information, which is dynamic in nature, is more resilient to this type of manipulation. A good example of behavioral information used in our work is spatiotemporal information about the itineraries of vehicles. Expectedly, behavioral information, and not administrative information, is more indicative of riskful behavior.

Multiple ways to take behavioral information of a vehicle over time exist, such as time series analysis [73] and reinforcement learning [184]. Our work explores the use of *networks* (see Definition 7). Multidisciplinary studies repeatedly show that network-driven approaches can often reveal otherwise hidden complex patterns and properties that signal meaningful phenomena in the real world [6, 10, 21, 129, 183]. In this work, networks enable us to explicitly model vehicle relations, considering interactions between these vehicles as part of the national or global transportation system [164, 208]. In Section 1.2, we further explore and define the necessary network concepts and properties relevant to the transport domain.

## Interpretable assessment

A challenge of most commonly used machine learning models is that their predictions are difficult to understand. It can be hard for humans to comprehend how multiple factors affect the inner workings of machine learning methods. As a result, people may perceive limited transparency [121]. Governmental organizations that motivate how they make their decisions and what data underpins these decisions (called interpretable assessment) are more trusted by society [198]. Hence, *interpretable* machine learning models (Definition 6) should be preferred in the inspection domain [122].

> **Definition 6.** An *interpretable model* is a model that allows humans to understand (1) what procedures were followed to make the model, (2) the inner workings of the model, and (3) how the model arrives at its predictions [8, 120, 123].

## Our contribution towards smart vehicle inspection

While clearly within reach, full implementation of smart inspection has yet to be achieved. In the case of the ILT, a desire to become more data-driven has been expressed; however, this needs to be sufficiently translated into inspection practice. This thesis examines how data on vehicle behavior can be leveraged to better understand contemporary problems in the transport domain, focusing on the smart inspection of vehicles (Definition 2). In

particular, we model vehicle behavior by making use of networks. In addition to addressing several fundamental problems related to the analysis of networks, we use networks modeling vehicle behavior in machine learning approaches for the accurate, automated, and fair assessment of vehicles. By doing so, we (1) provide a novel approach toward the assessment of vehicles for smart inspection and (2) obtain a better understanding of the dynamics of the global transportation system. Ultimately, our findings will contribute to a safer and healthier environment [136].

## 1.2 Networks

We deem networks to be a suitable data model to capture complex patterns in the behavior of vehicles, with the ability also to capture temporal aspects (as further discussed in Section 1.3). We start by defining networks and related concepts and subsequently discuss seven commonly observed properties of networks useful for understanding the data modeled by these networks. These properties are leveraged in the data-driven approach taken in this thesis toward the accurate, automated, fair, and interpretable assessment of vehicles, i.e., smart inspection.

The field of research that, in a general sense, concerns itself with methods for discovering knowledge from real-world systems modeled as networks is referred to as *network science* [10]. We define a network in Definition 7.

> **Definition 7.** A *network* is a set of entities called nodes combined with a set of edges (or, equivalently, links) that connect pairs of nodes.

Nodes connected by an edge are said to be *adjacent* and are also called *neighbors*. For the remainder of the introduction, we assume that the edges in the network are *undirected* and *unweighted*. Some concepts slightly change when considering *directed* edges; this will be explained in the relevant chapters where needed. A node's *degree* is its number of neighbors. Nodes with a large degree are also called *hubs* and are often deemed to have a central role in the network. Two nodes are connected when there exists a *path* between these nodes; a path is a sequence of edges linking a series of nodes.

A *component* is a subset of nodes and edges for which it holds that (1) there is a path between all pairs of nodes in the component and (2) it is not part of any larger component. A network can consist of multiple components. With respect to the components, we introduce three new concepts. First, we frequently analyze the largest component of a network, commonly referred to as the *Giant Component (GC)*. Second, in a component, the *shortest path* is a path which uses a minimum number of edges to connect a pair of nodes. The length of the shortest path (called *distance*) equals the number of nodes involved minus one. Thus, two adjacent nodes have a distance of one $(2-1)$ to each other. Third, the *diameter* is the maximum distance between any pair of nodes in a component.

## Social networks

A typical type of network often investigated is the *social* network, which is studied in many different disciplines, such as psychology, sociology, and mathematics. A node marks a person in these networks, while an edge indicates (for example) acquaintance. A figurative sketch of a so-called *ego network* of person D is given in Figure 1.1. An ego network consists of the individual node, its immediate neighbors and the edges connecting those neighbors. Node D has a degree of six and is part of three *triangles* (1: nodes A, B, D, 2: nodes B, C, D, and 3: D, E, G). A triangle is formed when three nodes are fully connected, i.e., have three edges between them. Like in many real-world networks, the ego network shown in Figure 1.1 contains extra contextual *node attributes*. In the figure, gender or profession (both indicated by the outfit) are examples of node attributes. The type of acquaintance (work, sport, or housemate) is considered an *edge* attribute. We can identify many types of networks in the real world, including information networks [108, 139], the aforementioned social networks [51, 119, 173, 174], technical networks [148], and transport networks [16, 94]. In the latter type of network, nodes are vehicles. We will explain this type of network in detail in Section 1.5.

In Subsections 1.2.1 to 1.2.4, we discuss seven common concepts to get a comprehensive understanding of networks. These concepts will prove relevant in the remainder of the thesis. Although we explain these concepts using examples from social networks, the discussed measures can be applied to any network.

### 1.2.1 Assortativity

The first concept is assortativity, defined in Definition 8.

> **Definition 8.** *Assortativity* refers to the inclination of nodes to link with other nodes that share similar (or dissimilar) characteristics [127].

The numeric value of assortativity is equal to the correlation coefficient (i.e., the Pearson coefficient) of the characteristics of linked nodes. Positive values for this measure indicate that neighboring nodes share a similar characteristic [10]. A value of zero indicates that there is no assortativity. Negative values indicate that nodes share dissimilar characteristics.

In social networks, the *degree* assortativity is usually observed. A positive value indicates that most people are acquainted with people with a similar number of friends [130]. It is well-known that celebrities often befriend other celebrities (i.e., the hubs in social networks). More specifically, marriages often occur between famous people, much more often than we would expect based on chance alone [10]. Generally, this strong degree assortativity is present in many more types of networks. We show a small example of assortativity in Figure 1.2; the two nodes with the highest degree (indicated in black) link to other nodes with a relatively large degree. In turn, the nodes with the smallest

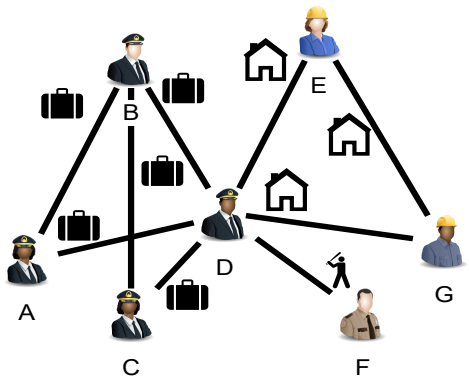



Figure 1.1: An example of an ego network (here part of a social network).

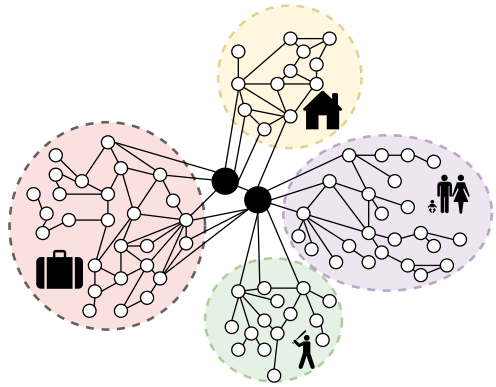


Figure 1.2: Community structure of a social network.

degree mainly connect to nodes with low degrees. Social networks generally also show assortativity in terms of age and race [22].

In contrast, in degree *disassortative* networks, nodes with a high degree are more likely linked to nodes with a lower degree. An example of a degree disassortative network is the topology of the internet [132], where hubs (servers, also called autonomous systems) frequently link to low-degree nodes (individual machines). The degree disassortative nature of such a network has consequences for how resilient the internet is towards failures. A breakdown of a limited number of hubs can prove disruptive to the overall connectivity of the network. Exactly this happened in recent times; major outages of the internet occurred in November 2020 [68], July 2021 [199], and June 2022 [182], because some hubs failed.

In Chapter 2, the measure of degree assortativity is used to characterize the structure of networks. Moreover, assortativity is used in Chapter 5 to understand truck behavior.

### 1.2.2 Clustering

The second concept commonly observed in real-world networks is *clustering*. Particularly in social networks, people tend to organize themselves in tightly knitted groups, so-called cliques [80, 197]. The *clustering coefficient* quantifies the extent to which clustering is present.

In particular, the *node clustering coefficient* quantifies the fraction of triangles that exist compared to how many triangles could exist between a node's neighbors. For example, in Figure 1.1, we observe that node D is part of three triangles. When all neighbors of node D are connected, there are $6 \cdot 5/2 = 15$ triangles, meaning that the local clustering coefficient of node D is $3/15 = 0.2$.

In Chapters 2, 5 and 6, the clustering coefficient is used to characterize the structure of networks.

### 1.2.3 Community structure

The third concept is that numerous networks possess a clear *community* structure. Communities are groups of nodes that are densely linked amongst each other but sparsely linked with other communities [64]. While defined above purely based on network structure, it is repeatedly observed that communities correspond to nodes sharing some property in real-world settings. In the social network depicted in Figure 1.2, each community is indicated by a colored area and a pictogram indicating the correspondence with the social groupings by household, interest, neighborhood, or profession.

Many methods for *community detection* exist [58, 76, 106]. In community detection, the goal is to optimally split the network into communities. When contextual information is known, it may be utilized to find the right communities. However, often we wish to find communities in an automated way by only using the network topology. A way to achieve this is by optimization of the so-called *modularity* measure [19, 186], which is typically computed as the difference between the actual number of edges within a community and the expected number of edges within the communities assuming the connections between the nodes were randomly created [25]. Suppose the optimization process obtains a high modularity value. In that case, nodes within the discovered communities are more connected to each other than they are to nodes in other communities. Hence, a strong community structure is likely present. Likewise, when a low modularity value is obtained, this often indicates that the network does not have a strong community structure [128].

In Chapter 5, assortativity (as discussed in Subsection 1.2.1) is used to understand the community structure of a network.

### 1.2.4 Giant Component, sparseness, small-world, and scale-free properties

We continue with four more common concepts frequently observed in real-world networks. These concepts (numbered 4, 5, 6, and 7) relate to the macro-scale of a network, meaning they can only be observed when considering the overall structure of a network.

4. **Large Giant Component.** Real-world networks often exhibit a Giant Component (GC) spanning the vast majority of all nodes. Throughout this work, we frequently use the GC to ensure all nodes are connected.
5. **Sparseness.** Real-world networks are typically *sparse*, meaning that from all the pairs of nodes that could be linked, relatively few links exist [10]. The sparseness of links in networks has implications for predicting new links, which we will discuss further in Section 1.4.
6. **Small-world.** Small-world networks are networks where nodes can typically reach each other using a shortest path of small length [119, 197]. The *average path length* or *average distance* of a component is equal to the average length of a *shortest* path (i.e., the distance) between all pairs of nodes [10]. We characterize networks using

the average shortest path length in the GC in Chapters 4 and 5. The GC in small-world networks tend to have relatively low average distances, even if the overall component is large in terms of the number of nodes and edges [6]. The significance of small-world networks is that they can provide efficient communication between distant nodes while maintaining local connectivity and resilience to node failures.

7. **Scale-freeness.** Many real-world networks are believed to be scale-free, which means many nodes have a relatively low degree, and few nodes have a very high degree. The degrees of nodes in a scale-free network thus lack a characteristic scale, making the degree distribution "scale-free" [195]. Therefore, the notion of scale-free networks is closely related to the presence of hubs. There is some controversy [82, 92] as to whether scale-free networks occur frequently [11, 12, 15, 195] or not [26]. Part of this discussion can be traced back to how closely the degree distribution resembles a power law distribution, lognormal distribution, or other types of skewed distributions [26]. Some scholars consider real-world networks universally scale-free, regardless of the domain of the network and the identity of the nodes [12]. The scale-free structure of many networks also has implications for predicting new links, which we will discuss further in Section 1.4.

## 1.3 Temporal networks

So far, we have assumed that networks are static, meaning we assume that *all* edges exist at some point in time. Real-world networks usually evolve and are therefore better modeled by a *temporal* network, which we defined in Definition 9.

> **Definition 9.** A *temporal network* is a network in which the edges are associated with a timestamp or time interval [38, 83].

The edges of a temporal network are consequently defined by (1) the source, (2) the target, and (3) an edge attribute containing temporal information on edge formation. In this work, we consider only temporal networks of which the edges are formed at a specific point in time, thus where the third edge attribute in Definition 9 is a timestamp. We do not consider any edge removal. Temporal networks allow for a more in-depth study of the growth mechanisms of a network [5, 52]. For example, a growth process known as preferential attachment can lead to the emergence of aforementioned scale-free networks. *Preferential attachment* is the process where new edges are preferentially linked with nodes that are hubs (i.e., have a high degree) at the time of edge formation [12]. The process will result in a feedback loop in which hubs increase their large degree even further, causing an increasingly skewed degree distribution. Generally, we differentiate between two types of temporal networks [81], viz. networks with (1) persistent relations and (2) discrete events.

First, we have temporal networks modeling *persistent relations* between the nodes. An example of such relations can be found in *acquaintance networks*, where an edge connects

two people if they are acquainted with each other in some way (such as friendship, kinship, or a professional relation) [119]. At most, one edge exists between two nodes in the network, and those edges are assumed to be present indefinitely, i.e., they appear but do not disappear.

Second, we can consider temporal networks modeling *discrete events* [133]. Multiple edges between a pair of nodes can exist, each with its associated timestamp. A *communication network* is an example of a temporal network containing discrete events. Like social networks, the nodes are people, but now the edges consist of communication events, such as calls or messages. Two persons can communicate often; thus, each edge has a distinct timestamp, and many edges may exist between the same two nodes.

## 1.4 Link prediction

An important task in network science is *link prediction*. It has numerous applications in real-world scenarios, such as spam mail detection in communication networks or friend recommendations in online social networks. The link prediction task is defined differently for varying purposes. In the broadest definition, the task is to predict which links exist between two nodes in a network. These links may be unobserved or even missing. In this definition, link prediction [114] can be employed on *static* networks (meaning no time information is present). Therefore, we call this task *missing* link prediction. However, in this work, we are interested in the *temporal* aspect of this task. Hence, we define link prediction in Definition 10.

> **Definition 10.** *Link prediction* is the task of predicting which links will appear in the future [10, 62].

Link prediction, as defined above, requires the use of temporal networks because links that appear later in time need to be known to train the model. Therefore, it is sometimes also called *temporal* link prediction.

Commonly, the link prediction task is formulated as a machine learning problem. The examples provided to the model consist of all pairs of nodes that are not adjacent in a current network snapshot, being the network consisting of all edges up to a certain point in time. The machine learning model aims to predict whether each currently unconnected pair of nodes is linked in a future network snapshot. Multiple types of features can be utilized to perform this task [54].

An example type is the *similarity-based* feature type, which considers how similar the surrounding network structure of two nodes is. Two typical similarity-based features are (1) the number of *unique* neighbors and (2) the number of *common* neighbors of both nodes. To explain the workings of these features, let us consider the well-known Zachary karate club social network [144, 213], depicted in Figure 1.3. It is a network of different karate club members, with links marking social interactions outside the club.

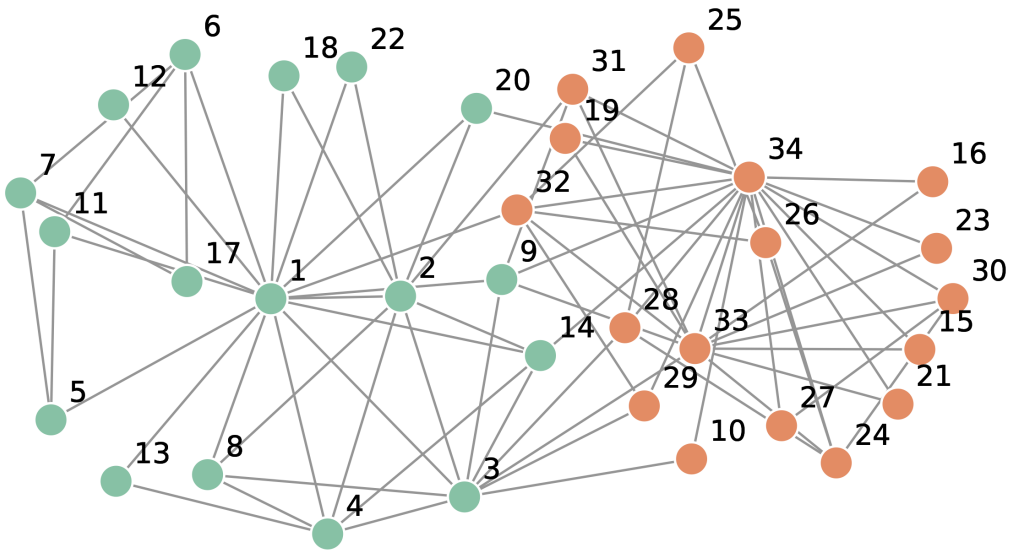



Figure 1.3: The well-known Zachary karate club social network.

Members 16 and 23 have a high similarity, as they both have a degree of two, and all their neighbors are shared. A clear advantage of the machine learning approach towards link prediction is that (1) multiple types of features (provided as input to the model) can be conveniently combined to arrive at a well-performing model [63] and (2) the approach is interpretable when simple topological network features are used [54].

In the remainder of this section, we mention two challenges of link prediction. These challenges will be addressed in Chapters 2 and 3.

The first challenge is that most works in the literature do not consider temporal information associated with the network's edges. Thereby, they ignore the evolution of the network observed so far. Using time-aware measures can improve prediction, but it ignores an essential dichotomous aspect of many temporal networks, namely that two types of temporal networks exist. The two types were discussed earlier in Section 1.3: (1) networks where edges are *persistent relations* and (2) networks where edges mark *discrete events*. We recall that temporal networks with discrete events may contain multiple edges between nodes, each having its own timestamp. This type of temporal network allows the evolution of edges between a pair of nodes to be exploited in the link prediction task.

In Chapter 2, we will show that we can improve link prediction performance when accounting for these discrete events.

The second challenge is that the validation (Definition 3) and the testing (Definition 4) of link prediction models are two nontrivial tasks often overlooked in existing work. It is only after applying proper model validation and testing that we may have sufficient confidence in applying a model in the real world. In particular, validation and testing

can identify *overfitting*, see Definition 11, which causes a too-optimistic performance estimation and, therefore, in machine learning, the well-known warning is that overfitting will reduce the validity of a study.

> **Definition 11.** *Overfitting* happens when a model matches the training data too closely, and the model is not working well on new, unseen data [167].

Obtaining a hold-out validation and using a general test set is impossible for network data because network data is, by definition, "related". Returning to the Zachary karate club social network in Figure 1.3, it is generally agreed that the nodes can be divided into two communities, which is indicated by the color of the node. A rigorous approach would be to sample the green nodes for model learning and the red ones for model validation when applying missing link prediction on the network. However, the ego networks of some nodes, in particular, node 9 and 10, are severely altered when such a sampling step is performed. These alterations could happen at a large scale in real-world networks, making the resulting link prediction model unusable. It is even more problematic for measures based on distance that use more global information beyond a node's direct neighborhood.

In Chapter 3, we explore two different splitting strategies in an attempt to discover how to perform adequate model validation on a collection of real-world temporal networks.

## 1.5 Transport networks

As explained in Section 1.1, our research examines how we can leverage transport networks to better understand vehicle behavior. We distinguish two different types of network data used throughout this work, being (1) *co-driving trucks* (Definition 12) in Chapters 4 and 5 and (2) *cargo ship networks* (Definition 13) in Chapter 6. Both datasets have national or even international coverage and systematically record nearly all vehicles for a specific period and location. Big datasets like these allow for a complete overview of all transport of that specific type. In both cases, the study of the temporal network aspects allows us to understand the behavior of the trucks (and ships) in relation to all other trucks (and ships). Below, we briefly describe (1) these two transport networks and (2) what we seek to understand from them.

### Truck co-driving networks

The Ministry of I&W gathers movements of trucks by Automatic Number-Plate Recognition (ANPR) systems; see Figure 1.4. The systems monitor any vehicle that passes, although some data may be missing, for example, because of misread license plates or avoidance of the cameras. Subsequently, it registers details such as license plate, country of registration, hazardous substances, length, weight, speed, and of course, the time of registration. By


Figure 1.4: The Weigh-In-Motion system.

exploring the data, we aim to learn and better understand what factors influence the trucks to do co-driving, an activity which we define in Definition 12 (and detail further in Chapter 4).

> **Definition 12.** *Co-driving* is the process where two trucks are observed at the same location within a very short time window. *Systematic* co-driving occurs when two trucks drive together frequently (e.g., more than once).

To investigate the process of truck co-driving, we consider the so-called truck co-driving network. We construct a temporal network from all systematic co-driving events by considering every truck as a node, linking two trucks when they show systematic co-driving behavior. Each temporal edge is thus characterized by the two trucks it links and the time period of the systematic co-driving event. The location of the co-driving activity that occurred is included as a spatial edge attribute. This spatiotemporal network is a particular extension of the temporal network, as both time and spatial information are available.

The truck co-driving networks have our interest for two reasons. First, we are interested in the properties of the co-driving network and the comparison with networks from other domains (e.g., social networks). Second, we want to know what communities of trucks are present in the network and what factors contribute to the formation of these communities. Chapters 4 and 5 provide a complete account of our research using the co-driving trucks dataset.

Ultimately, we mention two societal advantages of understanding truck co-driving behavior. First, understanding truck co-driving behavior can help reduce traffic congestion [187]. Moreover, co-driving and therewith platooning trucks can optimize fuel usage because of the aerodynamic drag reduction.

### Cargo ship network

The second set of data comprises all port calls of sea-going cargo ships in Europe, including the times of entrance into and departure from the port. These are collected from each port's administrative systems. All inspectorates have access to the same dataset in Europe, and the data can thus be used for smart ship inspection. We capture the behavior of the ships in relation to other ships by considering this dataset as a network. Deriving features from this network allows us to incorporate more information in a machine learning model than we would otherwise capture from the static data. The construction of a so-called cargo ship network (Definition 13) allows for extracting meaningful information for a machine learning model identifying noncompliant behavior of ships.

> **Definition 13.** The *cargo ship network* is a temporal network of all movements of cargo ships between ports [94]. The departure and arrival ports and the time of departure characterize edges.

This network is spatiotemporal as well. The edges have a temporal attribute indicating when the movement occurred. Unlike the truck co-driving network, each node is associated with a location. Relevant to our setting is that the inspectorate keeps records of all ships where noncompliances have been found, which can be used as node attributes and ultimate labels in a machine learning model. The entire study of the cargo ship network is presented in Chapter 6.

## 1.6 Problem statement and research questions

This section will describe our problem statement and research questions. As explained at the beginning of this introduction, smart inspections (Definition 2) are essential to ensuring a healthy and clean environment. In our work, we consider four aspects (see Section 1.1) of smart inspection and aim to handle them. It lead us to the following problem statement.

> **Problem statement:** *How can network science methods leverage behavioral data for smart inspection of vehicles?*

We subdivide the problem statement into five research questions. The first two questions address fundamental network science challenges, and the last three address more applied questions in the transportation domain. Below we describe the background and rationale behind this subdivision, i.e., our research strategy.

It has previously been observed that not all networks perform similarly in the missing link prediction task (e.g., [63]). In addition, literature so far has not extensively dealt with the relation between *network structure* and *performance* in the (temporal) link prediction task (Definition 10). These two observations leads us to Research question 1.

**Research question 1:** *What is the relation between network structure and model performance in link prediction?*

Let us now turn to the validation of link prediction models. A common approach to model validation (Definition 3) and testing (Definition 4) on tabular data is to use a hold-out set, i.e., a separate test set to evaluate the model's performance. Such a hold-out set is impossible to obtain for network data because all data are inherently related (see Section 1.4). If the hold-out criterion is not met, it can result in overfitting (Definition 11). We therefore formulate Research question 2 as follows.

**Research question 2:** *How can we obtain accurate estimates of the performance of link prediction models by using adequate splits into train, validation and test sets?*

Having posed our research questions addressing fundamental network science challenges, we now consider the research questions addressing smart vehicle inspection.

Our exploration of smart vehicle inspection starts by considering the case of the co-driving of trucks. We want to learn what factors contribute to truck co-driving, for reasons explained in Section 1.5. We do so by exploration of the co-driving network, arriving at Research question 3.

**Research question 3:** *How do network structure and vehicle attributes relate to co-driving behavior?*

We continue with the analysis of the truck co-driving network. For the inspectorate, it is interesting to understand (1) which groups of truck operators show frequent co-driving behavior and (2) what brings the truck operators in these groups together. When inspectorates want to change the behavior of truck operators, they can target specific communities via targeted communication. The question is which community detection model (and what parameter setting) yields the best partitioning into communities to do so. We explore the use of node attribute information to find such an optimal partitioning in Research question 4.

**Research question 4:** *How can node attribute information be exploited to automatically create a good partitioning of a co-driving network into communities?*

Finally, we proceed to the smart cargo ship inspection. We use information from the cargo ship network to improve the fair assessment of cargo ships for inspections, allowing us to answer Research question 5.

**Research question 5:** *How can ship behavior be utilized to enable smart inspection of cargo ships?*

Answering these five research questions allows us to deepen our understanding of machine learning methods on network data. The other way around, it improves our understanding of the effects of information on connectivity and relatedness of individual entities, i.e., the network aspect, on machine learning tasks. In turn, this knowledge can

improve the understanding of the behavior of different vehicles. Ultimately, it may enable smart inspection of vehicles, thereby maximizing the impact of new regulations for a sustainable planet.

## 1.7 Research methodology

We answer the five research questions by the following research methodology, consisting of six phases:

1. We establish the *context* of the question at hand.
2. We collect *relevant literature*.
3. We establish *preliminaries* and set up experiments.
4. We determine what *data* is available and what properties does this data possesses.
5. We *report and discuss* the findings of the experiments.
6. We provide a *conclusion* and suggest *future work*.

Answering the five research questions allows us to formulate an answer to the problem statement in Chapter 7.

## 1.8 Thesis overview and contributions

Below, we first provide an overview of the thesis and then indicate which research questions are answered within each chapter and what methodology was used.

We can differentiate three topics that our research covers: (1) machine learning, (2) network science, and (3) smart vehicle inspection. Each chapter relates to at least two topics. In Figure 1.5, we present a diagram we coin as a “ranked classification diagram”. It is a Venn diagram with each chapter assigned to one of the three topics above. The ranking aspect comes from the following; a chapter is more closely related to a topic when put nearer the corresponding circle.

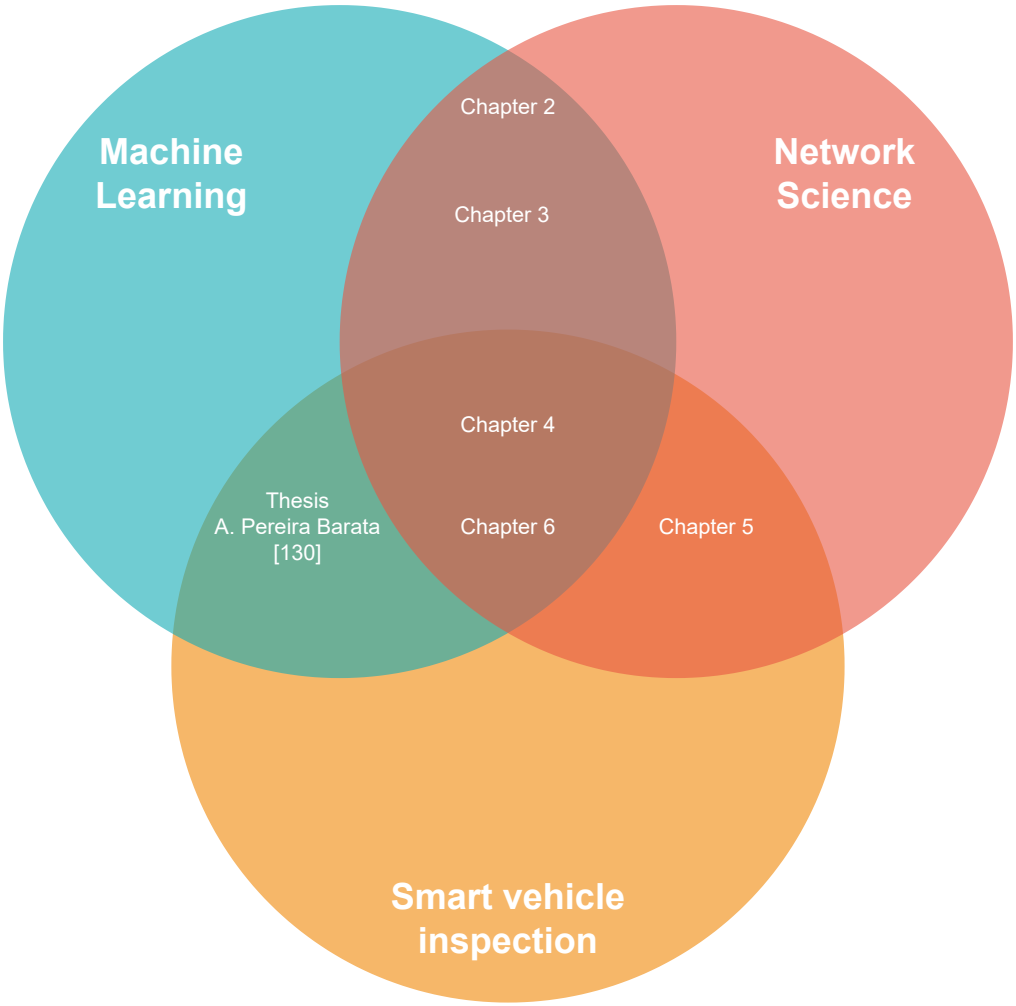



Figure 1.5: The relation between this thesis's three topics and chapters (indicated by a "ranked classification diagram").

## Contributions

- In Chapter 2, we address Research question 1. It starts with the topics of machine learning and network science. A large corpus of publicly available temporal networks is gathered. Link prediction is applied to all of them, and the link prediction performance and properties of the temporal networks are systematically investigated. The content of this chapter is based on the work described in:

  G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Supervised temporal link prediction in large-scale real-world networks." *Social Network Analysis and Mining* 11, 80 (2021). DOI: 10.1007/s13278-021-00787-3.
- Chapter 3 is devoted to Research question 2. Topics covered in this chapter are again machine learning and network science. The topic of smart vehicle inspection is not directly covered, but the evaluation of link prediction strategies is important when used in smart vehicle inspection. We also use a corpus of publicly available temporal networks gathered in this work. Different strategies for link prediction are assessed and evaluated. The content of this chapter is based on the work described in:

  G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Experimental evaluation of train and test split strategies in link prediction." In: *Proceedings of the 9th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 994. Springer, 2021, pages 79–91. DOI: 10.1007/978-3-030-65351-4_7.
- Chapter 4 is answering Research question 3, thereby covering the topics of network science, machine learning, and smart vehicle inspection. The chapter considers the construction of the truck co-driving network. We analyze the properties of the network and apply link prediction to the network to understand the (social) processes underlying the co-driving behavior. The content of this chapter is based on the work described in:

  G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding dynamics of truck co-driving networks." In: *Proceedings of the 8th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 882. Springer, 2020, pages 140–151. DOI: 10.1007/978-3-030-36683-4_12.
- Chapter 5 addresses Research question 4. It covers the topics of network science and smart vehicle inspection. A new approach to community detection using assortativity is proposed and applied to the truck co-driving network. The content of this chapter is based on the work described in:

  G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding behavioral patterns in truck co-driving networks." In: *Proceedings of the 7th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 813. Springer, 2018, pages 223–235. DOI: 10.1007/978-3-030-05414-4_18.

- Chapter 6 provides an answer to Research question 5. It brings together all the topics: network science, machine learning, and smart vehicle inspection. We provide an approach to smart cargo ships inspection. A comprehensive analysis is made of the fairness and performance of the model. The content of this chapter is based on the work described in:

  G. J. de Bruin, A. Pereira Barata, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Fair automated assessment of non-compliance in cargo ship networks." *EPJ Data Science* 11, 13 (2022). DOI: 10.1140/epjds/s13688-022-00326-w.

Chapter 7 concludes the thesis with answers to the research questions and the problem statement. Possible future research directions are provided as well.

## Cooperation

It deserves to be noted that the work by Antonio Pereira Barata was part of the same project as this thesis and thus is also concerned with machine learning and smart vehicle inspection [153, 157]. His work focused on methods for assessing the impact of missing data in the truck registration data (see Section 1.5), as well as machine learning methods for *tabular data*, whereas this thesis focuses on methods for better understanding *network data* in relation to smart vehicle inspection.

# 2

# Supervised link prediction in large-scale temporal networks

Missing link prediction is a well-studied technique for inferring the missing edges between two nodes in a static representation of a network. In temporal networks, such as modern-day social networks, the temporal information associated with each link can be used to predict *future links* between thus far unconnected nodes, thereby enabling *temporal* link prediction (Definition 10). In the continuation of the thesis, this is referred to as link prediction. In this chapter, we address Research question 1, which reads as follows.

> **Research question 1:** *What is the relation between network structure and model performance in link prediction?*

The chapter presents a systematic investigation of link prediction, making use of 26 temporal, structurally diverse, real-world networks ranging from thousands to millions of nodes and links. We analyze for each network the relationship between (1) the typology and (2) the obtained link prediction performance. Meanwhile, we employ well-established topological features.

The current chapter corresponds to the following publication:
G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Supervised temporal link prediction in large-scale real-world networks." *Social Network Analysis and Mining* 11, 80 (2021). DOI: 10.1007/s13278-021-00787-3

## 2.1 Link prediction

Link prediction is a frequently employed method within the broader field of social network analysis [10]. Many critical real-world applications exist in a variety of domains. Two examples are the prediction of (1) missing links between pages of Wikipedia and (2) users that are friends on an online social network [101]. As mentioned in Section 1.4, link prediction is often defined as predicting missing links based on the currently observable links in a network [114]. Many real-world networks have temporal information on when the edges were created [50]. Such *temporal networks* are also called dynamic or evolving networks (see also Definition 9). They open up the possibility of doing link prediction (contrasting with the aforementioned missing link prediction). The availability of temporal information means that we can infer future edges between two nodes as opposed to only predicting missing links [110]. For instance, in friendship networks, link prediction may (1) facilitate friend recommendations and (2) predict who will form new friendships in the future.

Existing work on link prediction is typically performed on one or a handful of specific networks, making it challenging to examine the generalizability of the approaches used [115]. This chapter provides the first large-scale empirical study of link prediction on 26 different large-scale and structurally diverse temporal networks originating from various domains. In doing so, we provide a systematic investigation of *how* temporal information is best used in link prediction.

We illustrate how the performance of social networks will likely be higher. Because they have a higher density than other networks, nodes have more common neighbors. Thus a given instance may provide more information to the link prediction model. By using this example, we demonstrate that it is essential to understand the relationship between the network's structural characteristics and the performance of link prediction features.

A common approach in link prediction is to learn a classifier that utilizes multiple features to classify which links are missing or, in case of temporal link prediction, will appear in the future. Features are typically computed for every pair of nodes that is not (yet) connected, based on the topology of the network [101]. These topological features essentially calculate a similarity score for a node pair, where a higher similarity signals a higher likelihood that this pair of nodes should be connected. Commonly used topological features in machine learning include Common Neighbors (CN), Adamic-Adar (AA), Jaccard Coefficient (JC), and Preferential Attachment (PA) (Subsection 2.4.1A). These features clearly relate to the structural position of the nodes in the network. Previous work has suggested a straightforward approach to taking the temporal evolution into account in topological features [35, 189]. We describe the process of obtaining the set of temporal topological features in Subsection 2.4.1B. The benefit of using such a set of features is that they are well-established and interpretable. Moreover, recent work has

shown that in a supervised classifier, the topological features perform as well as other features that are less interpretable and more complex [63]. A further comparison with other features is provided in Section 2.2.

As we have seen, previous studies ignore that two types of temporal networks can be distinguished (see also Section 1.3): networks with *persistent relationships* and networks with *discrete events* [133]. The example of friendship networks, as mentioned earlier, contains edges marking persistent relationships that occur at most once for related persons. In the case of discrete event networks, an edge marks a discrete event (e.g., a communication) at an associated timestamp, representing a message sent from one person to another. In contrast to networks with persistent relationships, multiple edges can occur between two persons in discrete event networks. So far, previous studies have ignored that each link is not of the same type. In our approach, we address this literature gap by what we coin *past event aggregation*. This allows us to take both types of temporal links into account, where all information of two-faceted past interactions (i.e., persistent and discrete) are incorporated into the temporal topological features.

Finally, the temporal topological features implicitly assume so-called edge-centered temporal behavior. This suggests that phenomena at the level of links determine the evolution of the network. Here, we may challenge the usual assumption that the temporal aspect is merely caused by the activity of nodes, being the decision-making entities in the network. At this point, we remark that the nodes are operating somewhat independently of the structure of the remainder of the network [78]. To investigate whether the assumption on the temporal aspect holds, we compare (1) temporal topological features with (2a) features consisting of static topological features and/or (2b) features capturing temporal node activity. By testing this distinction on the 26 different temporal networks, we are able to better understand whether the temporal aspect is best captured by considering edge-centered or node-centered temporal information.

Below we sum up the four contributions of this chapter.

1. To the best of our knowledge, we are one of the first to present *a large-scale empirical study of link prediction on various networks*. In total, we examine the performance of a link prediction model on 26 structurally diverse networks, varying in size from a few hundred to over a million nodes and edges.
2. We analyze possible relations between structural network properties and the observed performance in link prediction. We find that *networks with degree disassortativity* (see Subsection 1.2.1), signaling frequent connections between nodes with different degrees, *show better performance in link prediction*.
3. We show that the performance of link prediction can significantly be improved by *taking multiple past interactions* between two nodes *into account*.
4. To understand the relation between *node-centered* and *edge-centered* temporal behavior, the information networks used in this study stand out, as they appear to have more node-centered temporal behavior.

The remainder of this chapter is structured as follows. In Section 2.2, we further elaborate on related work. Section 2.3 provides the preliminaries of this chapter, leading up to a formal definition of link prediction. We continue with the research methodology in Section 2.4. It will be followed by describing the temporal networks in Section 2.5. In Section 2.6, we report on the four experiments and their results. In Section 2.7, the conclusion is presented, together with an outlook.

## 2.2 Related work on link prediction

Although much literature is available on link prediction, we found that attention to *temporal* networks and how to apply link prediction to them is relatively limited. Some reviews have been published. They are pointing out the various approaches toward link prediction [49, 50]. Consequently, we will start with an exploration of four types of approaches presented therein.

First, probabilistic models require (1) additional node or edge attributes to obtain satisfactory performance (which hinders a generic approach to all networks) or (2) techniques that do not scale to larger networks [101] (rendering them unusable for the larger networks used in the study).

Second, approaches such as matrix factorization, spectral clustering [168], and deep learning approaches, such as DeepWalk [158] and Node2Vec [70], all try to find a lower-dimensional representation of the temporal network and use the obtained representation as a basis for link prediction. Apart from hindering the generic approach desired in this work, the need for interpretability of lower-dimensional representations usually is a significant problem in domains where the model needs to be interpreted by law, such as in medicine or businesses dealing with personal information [84]. For example, in Chapter 4, we will examine the driving patterns of trucks in a so-called truck co-driving network, where trucks are connected when they frequently drive together. When an inspectorate uses gathered network information to predict which trucks should be inspected for possible misconduct, truck drivers may legally have the right to know why they were selected. Since we aim to provide approaches toward link prediction that apply to any scientific domain, we disregard approaches that learn a lower-dimensional representation.

Third, in the time series forecasting approach, the temporal network is divided into multiple snapshots [71, 134, 135, 161, 177]. For each of these snapshots, static topological features are learned. The topological features of a future network snapshot are learned using forecasting, thereby enabling link prediction. This approach does scale well to larger networks and is interpretable. However, it is unclear into how many snapshots the temporal network should be divided and whether the number of snapshots should remain constant across all networks used, again, hindering a truly generic approach.

Fourth, we focus on temporal topological features [34, 189]. Recent work has suggested that using topological features in supervised learning may outperform more complex features learned from a lower-dimensional representation of the temporal network [63]. Section 2.4 provides further details on this concept. The topological features are provided to a supervised link prediction classifier. Many different machine learning algorithms are known to work well in link prediction. Commonly used classifier algorithms include logistic regression [133, 161], support vector machines [3, 135], $k$-nearest neighbors [3, 34, 35], and random forests [32–35, 63, 135]. We report performances using the logistic regression classifier. This classifier provides the four following benefits, (1) it allows an intuitive explanation of how each instance is classified [17], (2) the classifier is relatively simple and hence interpretable [123], (3) the classifier scales well to larger networks, and (4) good results are achieved without parameter optimization [133].
To sum up, in contrast to earlier works on link prediction, which has been applied on only a handful of networks [18, 34, 35, 71, 124, 133–135, 161, 168, 177, 178, 189], we apply link prediction on a structurally diverse set of 26 large-scale, real-world networks. We aim to do so using a generic, scalable, and interpretable approach.

## 2.3 Preliminaries

This section describes the notation used in this chapter in Subsection 2.3.1. In Subsection 2.3.2, we explain the various network properties and measures used in this chapter. Finally, in Subsection 2.3.3, we formally describe the link prediction task.

### 2.3.1 Notation

In this chapter, we use the following notation for the link prediction task (see Definition 10).

An undirected, temporal network $G_{[t_a,t_b]}(V, E)$ consists of a set of nodes $V$ and edges (or, equivalently, links) $E = \{(u, v, t_i) \mid u, v \in V \wedge t_a \leq t_i \leq t_b\}$ that occur between timestamps $t_a$ and $t_b$.

Networks with discrete events, where multiple events can occur between two nodes, can be seen as a multigraph, where multi-edges exist: links between the same two nodes but with different timestamps [69]. In this work, the removal of edges is not considered since this information is unavailable for most temporal networks.

A static representation of the underlying network is needed to compare static and temporal features (Section 2.4). The static, simple graph is obtained from the temporal network by collapsing multi-edges into a single edge. The graph's number of nodes (also called the size) is $n = |V|$, and the number of edges is $m = |E|$. For convenience in later definitions, $N(u)$ is the set of all neighbors of node $u \in V$. The size of the set, i.e., $|N(u)|$,

is the number of neighbors of node $u$, which is in a simple graph equal to the degree of node $u$. In case of a multigraph, $|E(u)|$ is the degree of node $u$.

### 2.3.2 Real-world network properties and their measures

Several properties exist that characterize the macro-scale of a network [10]. These properties guide us in exploring how the structure relates to the link prediction performance. In this work, we use at least the following five properties: (1) the number of nodes and edges (not explained below), (2) average clustering coefficient, (3) degree assortativity, (4) density, and (5) diameter. Each property is derived from the underlying static graph.

- **Average clustering coefficient:** The average clustering coefficient (see also Subsection 1.2.2) is given by $C = n^{-1} \sum_{u \in V} 2L_u / \big(|N(u)| \cdot \big(|N(u)| - 1\big)\big)$, when $|N(u)| - 1 > 1$. $L_u$ represents the number of edges between the neighbors of node $u$. Highly clustered networks are often observed in the real world and particularly in social networks.
- **Degree assortativity:** It is often observed that nodes do not connect to random other nodes but instead connect to similar ones (see also Subsection 1.2.1). For instance, degree assortativity is observed in social networks, meaning that nodes often connect to other nodes with a similar degree. We can measure the degree assortativity of a network by calculating the Pearson correlation coefficient, $\rho$, between the degree of nodes at both ends of all edges [126] (see also Definition 8). In case low-degree nodes more frequently connect with high-degree nodes, the obtained value is negative.
- **Density:** The density of a network indicates what fraction of the pairs of nodes are connected. For networks of the same size, higher density means that the average degree of nodes is higher, which has implications for the overall structural information available to the link prediction classifier. For a network with $m$ edges and $n$ nodes, the *density* is equal to $2m/n(n-1)$.
- **Diameter:** The diameter is the largest distance observed between any pair of nodes (see also Section 1.2). The distance is measured in terms of the number of nodes in the path between the pair of nodes. This property, together with density, captures how well-connected a network is.

### 2.3.3 The goal of a supervised link prediction model

The goal of a supervised link prediction model is to predict for unconnected pairs of nodes in the temporal network $G_{[t_{q=0}, t_{q=s}]}$ whether they will connect in an evolved interval $[t_{q=s}, t_{q=1}]$ where $q$ marks the $q$-th percentile of observed timestamps in the network and $0 < s < 1$. Hence, timestamps $t_{q=0}$ and $t_{q=1}$ mark the time associated with the first and last edge in the network, respectively. Moreover, timestamp $t_{q=s}$ marks the time used to split the network into two intervals. The examples provided to the supervised link prediction model are pairs of nodes that are not connected in $[t_{q=0}, t_{q=s}]$. For each example $(u, v)$ in the dataset, a feature vector $x_{(u,v)}$ and binary label $y_{(u,v)}$ is

provided to the supervised link prediction model. The label for each pair of nodes $(u, v)$ is $y_{(u,v)} = 1$ when it will connect in $[t_{q=s}, t_{q=1}]$ and $y_{(u,v)} = 0$ otherwise. Because parameter $s$ determines the number of considered nodes, it affects the class imbalance encountered in the supervised link prediction; values close to $1$ result in a larger number of node pairs to consider while limiting the number of positives.

The features used in the supervised link prediction model are only allowed to use the information of network $G_{[t_{q=0}, t_{q=s}]}$, preventing any leakage from nodes that will connect in the evolved time interval $[t_{q=s}, t_{q=1}]$. Note that the temporal information contained in the network is used for two purposes; (1) it allows to split the network into two temporal intervals, and (2) it is used in feature engineering to model temporal evolution.

## 2.4 Chapter research methodology

This section explains the research methodology used in this chapter. It can be seen as an addition to the general research methodology described in Section 1.7. We emphasize described features used in supervised link prediction. We start by explaining the different sets of features in Subsection 2.4.1. We then present a novel research method, i.e., an intuitive approach to incorporate information on past interactions in the case of discrete event networks. Additionally, in Subsection 2.4.2, we discuss the supervised link prediction model.

### 2.4.1 Features

We explain three types of features in this section. First, the static topological features are provided in Subsection 2.4.1A. Second, the temporal topological features are given in Subsection 2.4.1B. Finally, the node activity features are specified in Subsection 2.4.1C.

#### 2.4.1A Static topological features

We use four common static topological features, forming the feature vector for each candidate pair of nodes $(u, v)$. These features are computed on the static graph underlying the temporal network, as defined in Subsection 2.3.1. Below we define each of them.

- **Adamic-Adar (AA):** The $AA$ feature considers all common neighbors, favoring nodes with low degrees [1].

$$AA_{\text{static}}(u, v) = \sum_{z \in N(u) \cap N(v)} 1/\log \left|N(z)\right| \text{ with } \left|N(z)\right| > 1 \tag{2.1}$$

- **Common Neighbors (CN):** The $CN$ feature equals the number of common neighbors of two nodes.

$$CN_{\text{static}}(u, v) = |N(u) \cap N(v)| \tag{2.2}$$

- **Jaccard Coefficient (JC):** The $JC$ feature is similar to the $CN$ feature but normalizes for the number of unique neighbors of the two nodes.

$$JC_{\text{static}}(u,v) = |N(u) \cap N(v)| / |N(u) \cup N(v)| \text{ with } |N(u) \cup N(v)| > 0 \qquad 2.3$$

- **Preferential Attachment (PA):** The $PA$ feature considers that nodes with a high degree are more likely to make new links than nodes with a lower degree (see also Section 1.3).

$$PA_{\text{static}}(u,v) = |N(u)| \cdot |N(v)| \qquad 2.4$$

#### 2.4.1B Temporal topological features

The temporal topological features are extended versions of the static topological features presented above in Subsection 2.4.1A. The construction of these features then requires three steps, namely:
**Step I.** Temporal weighting
**Step II.** Past event aggregation
**Step III.** Weighted topological features
The resulting feature vector for a given pair of nodes, after applying the three steps, consists of all possible combinations of three different temporal weighting functions (exponential, linear, square root), eight different past event aggregations (see below under step II), and four different weighted topological features ($AA$, $CN$, $JC$, $PA$). Thus, for discrete event networks, the feature vector is of length $3 \cdot 8 \cdot 4 = 96$, and for networks with persistent relationships, it is of length $3 \cdot 4 = 12$.

**Step I: Temporal weighting**
The topological features need weighted edges (see Step III), while the networks used in this study have edges with an associated timestamp. In the temporal weighting step, we obtain these weights with the help of a methodology described by Tylenda *et al.* [189]. The temporal weighting functions are provided in Equations 2.5 to 2.7. In these functions, a numeric timestamp $t_i$ is converted to a weight $w$. Note that $t_{\min}$ and $t_{\max}$ denote the earliest and latest observed timestamp over all edges of the considered network.

In Figure 2.1, the behavior of the different weighting functions is shown when applied to the DBLP network [109]. It is further described in Section 2.5. The exponential weighting function (Equation 2.6) assigns a higher weight to more recent edges than the linear (Equation 2.5) and square root (Equation 2.7) functions. In contrast, the square root function assigns higher weights to older edges than the linear and exponential functions. When the weights of older edges become close to zero, these edges are discarded by the weighted topological features. To prevent the edges from far in the past are discarded completely, we bound the output of each weighting function between a positive value $\ell$ and $1.0$ ($\ell$ stands for lower bound), with $0 \geq \ell < 1$.

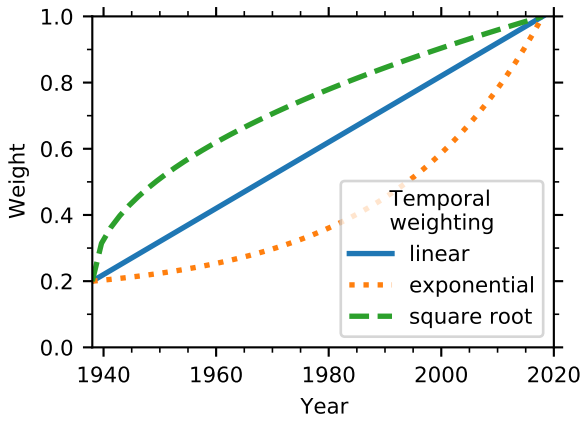



Figure 2.1: Mapping of three weighting functions for the DBLP network.

$$w_{\text{linear}} = \ell + (1-\ell)\frac{t_i - t_{\min}}{t_{\max} - t_{\min}} \tag{2.5}$$

$$w_{\text{exponential}} = \ell + (1-\ell) \cdot \frac{\exp\left(3 \cdot (t_i - t_{\min})/(t_{\max} - t_{\min})\right) - 1}{\mathrm{e}^3 - 1} \tag{2.6}$$

$$w_{\text{square root}} = \ell + (1-\ell) \cdot \sqrt{(t_i - t_{\min})/(t_{\max} - t_{\min})} \tag{2.7}$$

**Step II: Past event aggregation**

In the case of networks with discrete events, each multi-edge has an associated weight after the previous temporal weighting step. To allow the weighted topological features to be computed, we need to obtain a single weight for each node pair, capturing the past activity of these nodes. For this purpose, we propose to obtain the weight by using eight different aggregation functions. All eight functions use as input a set containing all weights of past events. The following functions are used: (1) the zeroth, (2) first, (3) second, (4) third, (5) fourth quantile, and the (6) sum, (7) mean, and (8) variance of all past weights. Utilizing these as summary statistics, we capture the different types of linkage in networks that occur in the real world. For example, it may matter whether interaction occurred often, far away in the past, or recently. The aggregation functions aim to capture different temporal behaviors. Quantile functions bin the set of weights, a common feature-engineering step. Taking the mean, sum, and variance of the set of weights allows the model to capture more complex trends. An example of these complex trends is the so-called bursty behavior, which is often observed in real-world data [9].

**Step III: Weighted topological features**
In Equations 2.8 to 2.11, the Weighted Temporal Features ($WTF$) are presented, which are taken from Bütün *et al.* [35]. In these equations, $WTF(u, v)$ denotes the weight obtained for a given pair of nodes $(u, v)$ after edges have been temporally weighted and, in case of networks with discrete events, events have been aggregated.

$$AA_{\text{temporal}}(u, v) = \sum_{z \in N(v) \cap N(y)} \frac{WTF(u, z) + WTF(v, z)}{\log\left(1 + \sum\limits_{x \in N(z)} WTF(z, x)\right)} \tag{2.8}$$

$$CN_{\text{temporal}}(u, v) = \sum_{z \in N(u) \cap N(v)} WTF(u, z) + WTF(v, z) \tag{2.9}$$

$$JC_{\text{temporal}}(u, v) = \sum_{z \in N(u) \cap N(v)} \frac{WTF(u, z) + WTF(v, z)}{\sum\limits_{x \in N(u)} WTF(u, x) + \sum\limits_{y \in N(v)} WTF(v, y)} \tag{2.10}$$

$$PA_{\text{temporal}}(u, v) = \sum_{u \in N(x)} WTF(u, x) \cdot \sum_{v \in N(y)} WTF(v, y) \tag{2.11}$$

### 2.4.1C Node activity features

The goal of the node activity features is to capture node-centered temporal activity. To this end, we create the node activity features in the following three steps: (1) temporal weighting, (2) aggregation of node activity, and (3) combining node activity. These steps are explained below. The feature vector for a given pair of nodes consists of all combinations of three different temporal weighting functions, seven aggregation functions applied to the node activity, and four combinations of the node activity. It results in a feature vector of length $3 \cdot 7 \cdot 4 = 84$.

**Step 1. Temporal weighting.** The temporal weighing method is the same as used in feature engineering of the temporal weighted topological features (see Subsection 2.4.1B and Figure 2.1).

**Step 2. Aggregation of node activity.** The weights from all edges adjacent to the node under investigation are collected for each node. We obtain a fixed feature vector for each node by aggregating using the following seven functions: (1) the zeroth, (2) first, (3) second, (4) third, (5) fourth quantile, and (6) sum and (7) mean of the node activity vector (here the variance of all node weights is suppressed because some nodes have only one edge, rendering the variance undefined). Similar to the engineering of the temporal topological features, these aggregations capture different kinds of activity that a node may exhibit. Nodes show bursty activity patterns in some networks [78].

**Step 3. Combining node activity.** To take the activity obtained in the previous two steps of both nodes under consideration into account, we use four different combination functions. These four functions are (1) absolute difference, (2) minimum, (3) maximum, and (4) sum. By doing this, we obtain the *node activity feature vector*.

### 2.4.2 Supervised link prediction

The features discussed in Subsection 2.4.1 serve as input for a supervised machine learning model that predicts whether a pair of currently disconnected nodes will connect in the future (see Subsection 2.3.3). Here we use the logistic regression classifier. It was chosen because of its simplicity, overall good performance on this type of task, and its explainability (see Section 2.2). We did not consider optimizing parameters because it is outside the scope of the current work.

In theory, a number quadratic in the number of nodes (i.e., the node pairs) could be selected as input for the classifier, with positive instances being node pairs that connect in the future. This would result in a significant class imbalance. To address the imbalance and, at the same time, limit the computation time needed to train the model, we reduce the number of node pairs given as input to the classifier by the following two steps.

**Step 1.** Pairs of nodes are only selected if they are exactly the same distance apart [111]. In our study, which involves large networks, the selection is limited to include only pairs of nodes with a distance of two.

**Step 2.** Pairs of nodes are sampled by replacement such that 10,000 will connect (positive instances) and 10,000 will not connect (negative instances). By following this sampling procedure, we obtain a balanced set of examples that do not require further post-processing and can be used directly by the classifier. In practice, the train set for the logistic regression classifier is obtained using stratified sampling, taking 75% of all examples. The remaining instances are used as a test set. Because we do not optimize any parameters of the logistic regression classifier, no validation set is used.

Analogously to previous work [50], we measure the classifier's performance on the test set utilizing the Area Under the ROC Curve (AUC). The AUC only considers the ranking of each score obtained for each pair of nodes provided to the logistic regression classifier. It makes the measured performance robust to cases where the applied threshold on the scores is chosen poorly. An AUC of $0.5$ signals random behavior, i.e., no classifier performance. A perfect performance is obtained when the AUC equals $1$, which is highly unlikely in practical settings.

## 2.5 Data and the statistics used

Our experiments are performed on a structurally diverse and large collection of 26 temporal networks. The networks can be categorized into three domains: social, information, and technological. The distinction of networks in these three domains is taken from other network repositories [104, 107]. In Table 2.1, some common structural properties of these datasets are presented (see Subsection 2.3.2 for properties and measures). It is apparent from Figure 2.2, showing the relation between the number of nodes and edges for each of the 26 datasets, that the selected networks span a broad range in size.

Table 2.1: Summary statistics of the 26 temporal networks (sorted by number of nodes). (The following abbreviations and symbols are used in the heading of the columns; D.a.: Degree assortativity, A.c.c: Average clustering coefficient, ø: Diameter. In the column "Domain", Technological is abbreviated to Tech. and Information to Inf. The column label provides an abbreviated name of the specific dataset. The full names are in the references. For * and **, see Subsection 2.6.4.)

| | Label | Domain | Edge type | Nodes ($n$) | Edges ($m$) | Density | D.a. | A.c.c. | ø | Ref. |
|---|---|---|---|---|---|---|---|---|---|---|
| | emails | Social | persistent | 167 | 82,927 | $2 \cdot 10^{-1}$ | 0.15 | 0.59 | 5 | [118] |
| ** | UC | Inf. | persistent | 899 | 33,720 | $2 \cdot 10^{-2}$ | 0.10 | 0.07 | 6 | [139] |
| | EU | Social | persistent | 986 | 332,334 | $3 \cdot 10^{-2}$ | 0.05 | 0.41 | 7 | [211] |
| | Dem | Social | persistent | 1,891 | 39,264 | $2 \cdot 10^{-3}$ | −0.15 | 0.21 | 8 | [200] |
| | bitA | Social | event | 3,683 | 22,650 | $2 \cdot 10^{-3}$ | −0.15 | 0.17 | 10 | [103] |
| | bitOT | Social | event | 5,573 | 32,029 | $1 \cdot 10^{-3}$ | −0.15 | 0.16 | 14 | [103] |
| | chess | Inf. | event | 6,050 | 21,163 | $1 \cdot 10^{-3}$ | 0.36 | 0.05 | 13 | [104] |
| | HepTh | Inf. | persistent | 6,798 | 290,597 | $9 \cdot 10^{-3}$ | 0.08 | 0.77 | 11 | [106] |
| | HepPh | Inf. | persistent | 16,959 | 2,322,259 | $8 \cdot 10^{-3}$ | 0.17 | 0.61 | 8 | [106] |
| * | Condm | Social | persistent | 17,218 | 88,090 | $4 \cdot 10^{-4}$ | 0.29 | 0.64 | 19 | [112] |
| | SX-MO | Social | persistent | 24,818 | 506,550 | $6 \cdot 10^{-4}$ | −0.05 | 0.31 | 9 | [143] |
| | D-rep | Social | event | 30,398 | 87,627 | $2 \cdot 10^{-4}$ | 0.02 | 0.01 | 12 | [46] |
| | Rbody | Tech. | persistent | 35,010 | 265,491 | $2 \cdot 10^{-4}$ | 0.03 | 0.18 | 11 | [102] |
| | Rtit | Tech. | persistent | 53,018 | 510,787 | $1 \cdot 10^{-4}$ | −0.01 | 0.18 | 17 | [102] |
| | FB-w | Social | event | 55,387 | 335,708 | $2 \cdot 10^{-4}$ | −0.02 | 0.12 | 16 | [194] |
| | FB-l | Social | event | 55,387 | 335,708 | $2 \cdot 10^{-4}$ | 0.22 | 0.12 | 16 | [194] |
| | Enron | Social | persistent | 87,273 | 1,148,072 | $8 \cdot 10^{-5}$ | 0.22 | 0.12 | 14 | [97] |
| | loans | Inf. | event | 89,269 | 3,394,979 | $8 \cdot 10^{-4}$ | −0.04 | 0.00 | 8 | [163] |
| | trust | Social | event | 114,467 | 717,667 | $9 \cdot 10^{-5}$ | −0.07 | 0.13 | 14 | [165] |
| | Wiki | Social | persistent | 116,836 | 2,917,785 | $3 \cdot 10^{-4}$ | −0.06 | 0.36 | 10 | [25] |
| | D-v | Inf. | event | 139,409 | 3,018,197 | $3 \cdot 10^{-4}$ | −0.21 | 0.14 | 4 | [79] |
| | SX-AU | Social | persistent | 159,316 | 964,437 | $4 \cdot 10^{-5}$ | −0.10 | 0.11 | 13 | [143] |
| | SX-SU | Social | persistent | 194,085 | 1,443,339 | $4 \cdot 10^{-5}$ | −0.08 | 0.12 | 12 | [143] |
| | D-f | Social | event | 279,374 | 1,729,983 | $4 \cdot 10^{-5}$ | −0.05 | 0.09 | 18 | [79] |
| | AMin | Social | persistent | 855,165 | 23,787,273 | $9 \cdot 10^{-6}$ | 0.16 | 0.61 | 22 | [215] |
| | DBLP | Social | persistent | 1,824,701 | 29,487,744 | $5 \cdot 10^{-6}$ | 0.15 | 0.63 | 23 | [109] |

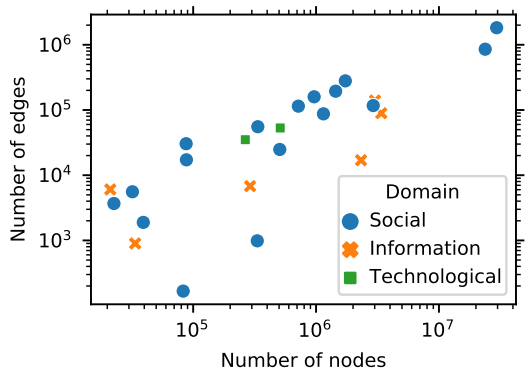



Figure 2.2: Number of nodes and edges of the 26 temporal networks. (The horizontal and vertical axes have logarithmic scaling.)

Also, for each network, it is indicated whether the edges mark persistent relationships or discrete events. In the latter case, the network has a multigraph structure, which requires preprocessing as discussed in Subsection 2.4.1B. We observe seventeen networks showing degree disassortative behavior, meaning high-degree nodes tend to connect to low-degree nodes more frequently. The other nine networks show the opposite behavior. We do not observe any significant relation between the domain of a network and its degree assortativity or any other global property of the network.

A total of 21 networks were obtained from the KOblenz NEtwork CollecTion (KONECT) repository [104]. Four networks (EU, Rbody, Rtit, and trust, see Table 2.1) were obtained from the Stanford Network Analysis Project (SNAP) repository [107]. The Arnetminer (Aminer) network was obtained directly from http://www.cn.aminer.org/data. The last column in Table 2.1 references the work where each network is introduced for the first time. Any directed network is converted into an undirected network by ignoring the directionality. In originally signed networks, we use only positive edges.

## 2.6 Experiments

In Subsection 2.6.1, we start with the experimental setup. Then, the structure follows the four experiments described in four separate subsections. In the first experiment (Subsection 2.6.2), we examine *the performance of link prediction* on 26 networks. The second experiment (Subsection 2.6.3) continues with analyzing *the relation between structural network properties and the performance in link prediction*. In the third experiment (Subsection 2.6.4), we show the results of past event aggregation to *the link prediction in networks with discrete events*. We finish with the fourth experiment (Subsection 2.6.5) with a *comparison between node-centered and edge-centered temporal behavior*.

### 2.6.1 Experimental setup

The research methodology to obtain examples and labels that serve as input for the classifier has been briefly explained in Subsection 2.4.2. We need to determine the value $s$ for each network. Around two-thirds of the edges are commonly used for extraction of features [3, 34, 35, 112], and hence we choose $s = 2/3$.

The first step in the creation of temporal topological and node activity features is to assign temporal weight to each edge. In Subsection 2.4.1B, Step I, parameter $\ell$ is introduced to prevent discarding old edges in the temporal weighting method. Based on earlier work [189], we set $\ell = 0.2$, giving minimal weight to links far away in the past while still sufficiently discounting these older links.

In the four experiments, we use four sets of features. These feature sets, which are indicated by capital Roman numerals, are as follows. They are defined in Subsection 2.4.1.

I Static topological (as defined in Subsection 2.4.1A)

II-A Temporal topological (as defined in Subsection 2.4.1B)

II-B Temporal topological without past event aggregation (like Subsection 2.4.1B but skipping Step II and using only the last occurring event)

III Static topological + node activity (Subsection 2.4.1B + Subsection 2.4.1C)

Standardizing features by subtracting the mean and scaling the variance to unit is standard practice. The logistic regression classifier provided in the Python scikit-learn package [149] is used. Although the goal of this work is not to extensively compare machine learning classifiers, in Subsection 2.6.2 results on the performance in terms of AUC obtained using two other commonly used classifiers, random forests [149] and eXtreme Gradient Boosting (XGBoost) [41] are presented. For almost all datasets, similar relative performance is observed.

The code used in this research is publicly available [28]. It uses the Python language and the packages NetworkX [72] for network analysis, scikit-learn [149] for the machine learning pipeline, and the Scipy ecosystem [193] for some of the feature engineering and statistical tests. The C++ library teexGraph [185] was used to determine the diameter of each network. The package versions and all dependencies can be found in the repository.

### 2.6.2 Experiment 1: Improvement of prediction performance with temporal information

We examine whether *temporal information improves the overall link prediction performance*. Baseline performance is obtained by ignoring temporal information, using static topological features (Feature set I). In contrast, temporal topological features (Feature set II-A) are used to obtain link prediction performance utilizing temporal information.

The results of this comparison are presented in Figure 2.3 and Table 2.2. (Feature sets II-B and III are used in later experiments.) They indicate that using temporal information improves the prediction performance of new links, i.e., performance reported in column

Table 2.2: Link prediction performance of the 26 temporal networks. (The following sets of features are used: I: Static topological features; II-A: Temporal topological features with past event aggregation; II-B: Temporal topological features without past event aggregation; and III: Static topological + node activity features.)

| | Label | Domain | Edge type | Nodes ($n$) | AUC | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | I | II-A | II-B | III |
| | emails | Social | multi | 167 | 0.864 | 0.921 | 0.852 | 0.902 |
| ** | UC | Information | multi | 899 | 0.731 | 0.893 | 0.744 | 0.873 |
| | EU | Social | multi | 986 | 0.839 | 0.873 | 0.811 | 0.849 |
| | Dem | Social | multi | 1,891 | 0.920 | 0.944 | 0.919 | 0.938 |
| | bitA | Social | simple | 3,683 | 0.868 | 0.945 | 0.945 | 0.940 |
| | bitOT | Social | simple | 5,573 | 0.821 | 0.947 | 0.947 | 0.939 |
| | chess | Information | simple | 6,050 | 0.665 | 0.735 | 0.735 | 0.736 |
| | HepTh | Information | multi | 6,798 | 0.757 | 0.835 | 0.776 | 0.819 |
| | HepPh | Information | multi | 16,959 | 0.828 | 0.879 | 0.834 | 0.868 |
| * | Condm | Social | multi | 17,218 | 0.688 | 0.760 | 0.706 | 0.728 |
| | SX-MO | Social | multi | 24,818 | 0.859 | 0.944 | 0.909 | 0.933 |
| | D-rep | Social | simple | 30,398 | 0.837 | 0.866 | 0.866 | 0.865 |
| | Rbody | Technological | multi | 35,010 | 0.880 | 0.905 | 0.854 | 0.890 |
| | Rtit | Technological | multi | 53,018 | 0.903 | 0.931 | 0.906 | 0.925 |
| | FB-w | Social | simple | 55,387 | 0.762 | 0.809 | 0.809 | 0.788 |
| | FB-l | Social | simple | 55,387 | 0.762 | 0.803 | 0.803 | 0.775 |
| | Enron | Social | multi | 87,273 | 0.847 | 0.912 | 0.873 | 0.909 |
| | loans | Information | simple | 89,269 | 0.786 | 0.947 | 0.947 | 0.946 |
| | trust | Social | simple | 114,467 | 0.889 | 0.936 | 0.936 | 0.937 |
| | Wiki | Social | multi | 116,836 | 0.864 | 0.936 | 0.896 | 0.939 |
| | D-v | Information | simple | 139,409 | 0.933 | 0.941 | 0.941 | 0.939 |
| | SX-AU | Social | multi | 159,316 | 0.937 | 0.970 | 0.959 | 0.970 |
| | SX-SU | Social | multi | 194,085 | 0.921 | 0.965 | 0.946 | 0.961 |
| | D-f | Social | simple | 279,374 | 0.891 | 0.926 | 0.926 | 0.924 |
| | AMin | Social | multi | 855,165 | 0.725 | 0.849 | 0.804 | 0.816 |
| | DBLP | Social | multi | 1,824,701 | 0.704 | 0.826 | 0.743 | 0.786 |

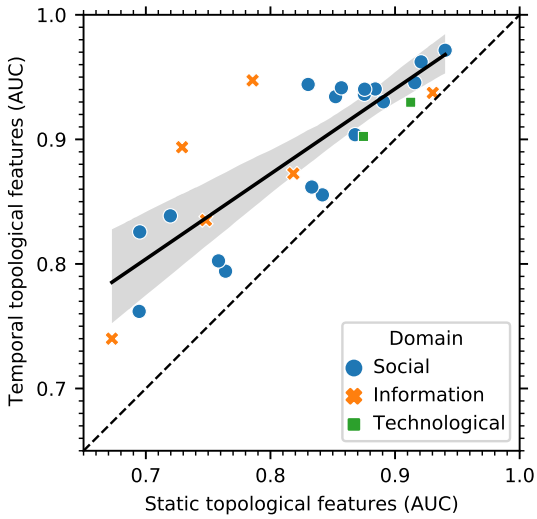



Figure 2.3: Link prediction performance of the 26 temporal networks. (The black line and gray band indicates the best linear regression fit and its 95% confidence interval, respectively.)

"II-A" is always higher than that in "I". So, every network performs better when temporal topological features are used. The average improvement in performance is $0.07 \pm 0.04$ (mean $\pm$ standard deviation).

For some networks, performance improves considerably when temporal information is used in the prediction. For example, the loans network has a mediocre baseline performance of $0.79$, but a high performance of $0.95$ is observed when temporal information is employed. From the results of this experiment we may conclude that the performance improvement can be related to the network's structure. Next, the relation between the structural properties of networks and the performance in link prediction is explored.

### Choice of classifier

The logistic regression classifier was used for interpretability (see Definition 6), as further discussed in Section 2.2. In Table 2.3, we provide, for each of the datasets as introduced in Table 2.1, the performance in terms of AUC obtained using two other commonly used classifier algorithms, being the random forest [149] and XGBoost [41] algorithms, with default parameters. For almost all datasets, similar relative performance is observed. We continue the other experiments (Subsections 2.6.3 to 2.6.5) using the logistic regression classifier.

Table 2.3: Link prediction performance with past event aggregation (Feature set II-A, see experimental setup in Section 2.6).

| | Label | Logistic Regression | Random Forest | XGBoost |
|---|---|---|---|---|
| | emails | 0.921 | 0.951 | 0.955 |
| ** | UC | 0.893 | 0.942 | 0.946 |
| | EU | 0.873 | 0.953 | 0.942 |
| | Dem | 0.944 | 0.984 | 0.981 |
| | bitA | 0.945 | 0.974 | 0.974 |
| | bitOT | 0.947 | 0.973 | 0.967 |
| | chess | 0.735 | 0.833 | 0.830 |
| | HepTh | 0.835 | 0.867 | 0.856 |
| | HepPh | 0.879 | 0.816 | 0.798 |
| * | Condm | 0.760 | 0.875 | 0.870 |
| | SX-MO | 0.944 | 0.959 | 0.959 |
| | D-rep | 0.866 | 0.973 | 0.976 |
| | Rbody | 0.905 | 0.944 | 0.938 |
| | trust | 0.936 | 0.971 | 0.969 |
| | Rtit | 0.931 | 0.948 | 0.946 |
| | FB-w | 0.809 | 0.769 | 0.772 |
| | FB-l | 0.803 | 0.769 | 0.772 |
| | Enron | 0.912 | 0.973 | 0.970 |
| | loans | 0.947 | 0.941 | 0.943 |
| | WikiC | 0.936 | 0.979 | 0.981 |
| | D-v | 0.941 | 0.908 | 0.910 |
| | SX-AU | 0.970 | 0.990 | 0.990 |
| | SX-SU | 0.965 | 0.981 | 0.982 |
| | D-f | 0.926 | 0.977 | 0.977 |
| | AMin | 0.849 | 0.872 | 0.865 |
| | DBLP | 0.826 | 0.919 | 0.923 |
| | mean | 0.892 | 0.925 | 0.923 |

### 2.6.3 Experiment 2: Structural network properties and link prediction performance

In the second experiment, we examine and discuss *which structural properties are associated with high link prediction performance*. We do so by exploration of the Pearson correlation coefficient, $\rho$, with the link prediction performance obtained.

In Figure 2.4, the Pearson correlations between the performance in link prediction and the structural network properties discussed in Section 2.5 are presented. Most properties show a modest correlation with the link prediction performance. However, a significant negative correlation is found between the degree assortativity of a network and the prediction performance of new links using static topological features ($p = 3 \cdot 10^{-6}$) and temporal topological features ($p = 5 \cdot 10^{-7}$). It means that strong disassortative behavior in networks, where nodes of low degree are more likely to connect with nodes of high degree, show better performance in link prediction. The relation between degree assortativity and the link prediction performance is shown in more detail in Figure 2.5.

We observe a negative correlation between degree assortativity and link prediction performance. It can be explained as follows. In real-world networks, low-degree nodes typically vastly outnumber the high-degree nodes. However, nodes far exceeding the average degree, so-called hubs, are also relatively often observed in real-world networks [10]. In degree disassortative networks, the numerous low-degree nodes connect more frequently with hubs than other low-degree nodes. For these low-degree nodes, the preferential attachment feature will provide higher scores for candidate nodes having a high degree. Therefore, the supervised model can use this information to perform better.

From Figure 2.5, we also learn that the temporal topological features have an even stronger correlation ($\rho = -0.82$) than the static topological features ($\rho = -0.78$). A possible explanation is that the temporal features can determine which nodes will grow to active hubs, linking to many low-degree nodes. This information would be lost in a static network representation. From the results of the experiment as shown in Figure 2.5, we may conclude that the temporal topological features likely capture relevant temporal behavior.

#### Degree-preserving rewiring

By performing assortative and disassortative degree-preserving rewiring, we further substantiate that disassortative networks indeed show higher link prediction performance. Utilizing simulation, we modified many network datasets from Table 2.1 using assortative and disassortative degree-preserving rewiring, following an approach similar to the one proposed in [191]. In particular, we aim to retain the local clustering properties by not selecting two edges at random, but rather selecting two edges that are close to each other, ensuring that there are not too many triangles and, in addition to that, clustering is destructed, as this is a determining feature in link prediction.

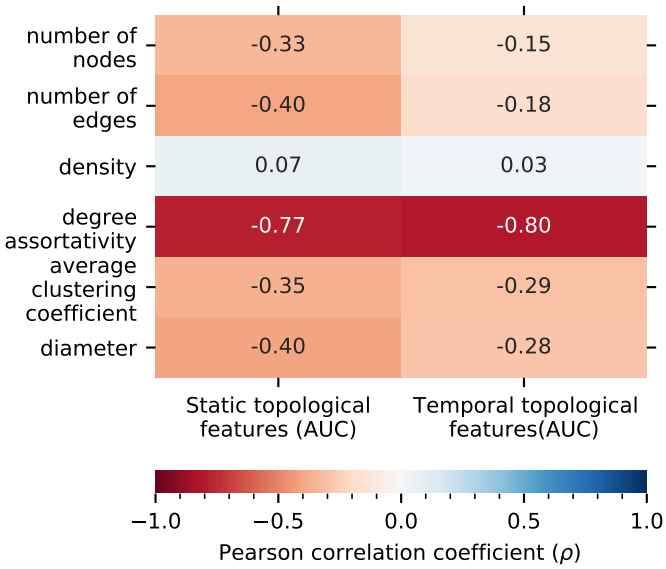



Figure 2.4: Correlations between network properties and performance (in a classifier learned only with static [Feature set I] and with temporal topological features [Feature set II-A]).

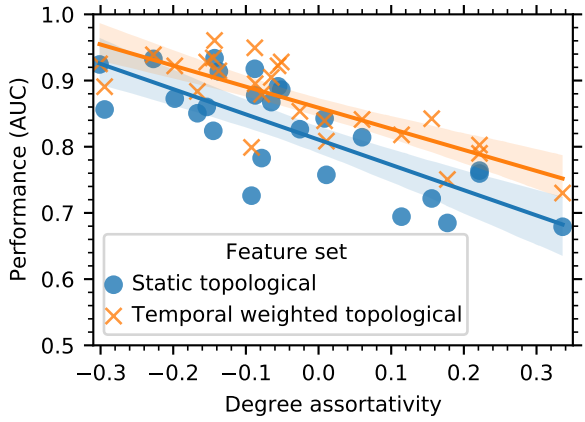



Figure 2.5: Degree assortativity and link prediction performance (in a classifier learned only with static topological features [Feature set I] and temporal weighted topological features [Feature set II-A]). (The lines indicate the relation between the network's degree assortativity and the classifier's performance, the band indicates the 95% confidence interval between the two.)

The research methodology, which we repeat several times (explained below), consists of five steps.

1. An edge $(u, v)$ is randomly selected.
2. We randomly select a node $x$ from the neighborhood of $u$.
3. We sample a node $y$ connected to $x$ but not to $u$ or $v$. At this time, pairs of nodes $(u, v)$ and $(x, y)$ are connected while the link $(v, y)$ is absent.
4. We determine from the pairs of nodes $(u, v)$, $(v, y)$, and $(x, y)$ which node pair has a maximum difference in degree.
5. We rewire the edges such that this pair with a maximum difference in degree becomes connected, see below.
   (a) Node pair $(v, y)$ has the maximum difference in degree, and there is no gain in assortativity by rewiring any edges,
   (b) Node pair $(u, v)$ has the maximum difference in degree, and by moving all edges (recall, there can be multiple links between two nodes) between $(u, v)$ to $(v, y)$, the assortativity is increased.
   (c) Node pair $(x, y)$ has the maximum difference in degree, and by moving all edges between $(x, y)$ to $(v, y)$, the assortativity is increased.

In case we want to perform disassortative degree-preserving rewiring, we consider in Steps 4 and 5 the node pair with the smallest difference in degree. The five steps are repeated $0.2 \cdot m$ times, with increments of $0.2 \cdot m$, until $m$.

The resulting degree assortativity values of the rewired networks can be found in Table 2.4. We observe that degree disassortative rewiring (compared to assortative rewiring) is associated with a more significant change in the degree assortativity.

We list the percentual increase in performance for both disassortativity and assortativity rewired datasets in Table 2.5. In both cases, we observe higher performance for disassortativity rewired networks, which confirms that dissassortative networks show higher link prediction performance.

Table 2.4: Degree assortativity of networks after rewiring (from degree disassortative rewiring [up to $-100\%$] to degree assortative rewiring [up to $100\%$]).

| | Label | $-100\%$ | $-80\%$ | $-60\%$ | $-40\%$ | $-20\%$ | $0\%$ | $20\%$ | $40\%$ | $60\%$ | $80\%$ | $100\%$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | emails | 0.01 | 0.01 | 0.07 | 0.09 | −0.00 | 0.15 | 0.14 | 0.18 | 0.16 | 0.09 | 0.19 |
| ** | UC | −0.05 | −0.03 | −0.02 | 0.01 | 0.06 | 0.10 | 0.14 | 0.17 | 0.18 | 0.21 | 0.23 |
| | EU | 0.23 | 0.16 | 0.36 | 0.34 | 0.15 | 0.05 | 0.12 | 0.11 | 0.10 | −0.18 | −0.11 |
| | Dem | −0.21 | −0.21 | −0.16 | −0.14 | −0.14 | −0.15 | −0.06 | −0.00 | 0.06 | 0.09 | 0.13 |
| | bitA | −0.25 | −0.24 | −0.22 | −0.19 | −0.17 | −0.15 | −0.10 | −0.04 | 0.01 | 0.10 | 0.22 |
| | bitOT | −0.23 | −0.22 | −0.20 | −0.17 | −0.16 | −0.15 | −0.11 | −0.07 | −0.02 | 0.04 | 0.14 |
| | chess | −0.17 | −0.14 | −0.05 | 0.04 | 0.18 | 0.36 | 0.52 | 0.62 | 0.69 | 0.74 | 0.78 |
| | HepTh | −0.18 | −0.13 | −0.08 | −0.03 | 0.03 | 0.08 | 0.18 | 0.31 | 0.46 | 0.57 | 0.61 |
| | HepPh | −0.11 | −0.07 | −0.02 | 0.04 | 0.10 | 0.17 | 0.26 | 0.35 | 0.43 | 0.48 | 0.52 |
| * | Condm | −0.04 | 0.00 | 0.05 | 0.11 | 0.20 | 0.29 | 0.42 | 0.53 | 0.59 | 0.62 | 0.63 |
| | SX-MO | −0.24 | −0.21 | −0.17 | −0.13 | −0.09 | −0.05 | 0.02 | 0.09 | 0.16 | 0.22 | 0.29 |
| | D-rep | −0.19 | −0.16 | −0.12 | −0.08 | −0.04 | 0.02 | 0.13 | 0.29 | 0.46 | 0.56 | 0.64 |
| | Rbody | −0.11 | −0.09 | −0.06 | −0.03 | 0.00 | 0.03 | 0.07 | 0.10 | 0.12 | 0.13 | 0.15 |
| | Rtit | −0.11 | −0.09 | −0.07 | −0.05 | −0.04 | −0.02 | 0.04 | 0.09 | 0.14 | 0.14 | 0.13 |
| | FB-w | −0.12 | −0.09 | −0.06 | −0.00 | 0.08 | 0.22 | 0.43 | 0.61 | 0.71 | 0.77 | 0.81 |
| | FB-l | −0.12 | −0.09 | −0.06 | −0.00 | 0.08 | 0.22 | 0.43 | 0.61 | 0.71 | 0.77 | 0.81 |
| | Enron | −0.14 | −0.11 | −0.09 | −0.07 | −0.05 | −0.04 | 0.01 | 0.03 | 0.06 | 0.08 | 0.09 |
| | loans | −0.20 | −0.17 | −0.14 | −0.12 | −0.09 | −0.07 | −0.02 | 0.06 | 0.22 | 0.47 | 0.61 |
| | trust | −0.26 | −0.23 | −0.19 | −0.14 | −0.09 | −0.01 | 0.13 | 0.33 | 0.52 | 0.64 | 0.70 |
| | Wiki | −0.08 | −0.08 | −0.07 | −0.06 | −0.06 | −0.06 | −0.04 | −0.03 | −0.02 | −0.01 | 0.00 |
| | D-v | −0.27 | −0.26 | −0.24 | −0.23 | −0.21 | −0.21 | −0.20 | −0.16 | −0.06 | 0.13 | 0.31 |
| | SX-AU | −0.25 | −0.22 | −0.20 | −0.17 | −0.13 | −0.10 | −0.06 | −0.01 | 0.03 | 0.08 | 0.13 |
| | SX-SU | −0.16 | −0.15 | −0.13 | −0.11 | −0.10 | −0.08 | −0.05 | −0.03 | −0.00 | 0.03 | 0.07 |
| | D-f | −0.13 | −0.12 | −0.10 | −0.09 | −0.07 | −0.05 | 0.02 | 0.18 | 0.47 | 0.64 | 0.71 |
| | AMin | 0.01 | 0.03 | 0.05 | 0.07 | 0.11 | 0.16 | 0.21 | 0.24 | 0.28 | 0.30 | 0.33 |
| | DBLP | 0.01 | 0.03 | 0.05 | 0.07 | 0.11 | 0.15 | 0.21 | 0.26 | 0.30 | 0.33 | 0.36 |

Table 2.5: Link prediction performance after rewiring.

| | Label | $-100\%$ | $-80\%$ | $-60\%$ | $-40\%$ | $-20\%$ | $20\%$ | $40\%$ | $60\%$ | $80\%$ | $100\%$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | emails | $-0.074$ | $-0.107$ | $-0.106$ | $-0.096$ | $-0.103$ | $-0.131$ | $-0.126$ | $-0.136$ | $-0.138$ | $0.024$ |
| ** | UC | $-0.311$ | $-0.266$ | $-0.270$ | $-0.356$ | $-0.297$ | $-0.312$ | $-0.388$ | $-0.373$ | $-0.303$ | $-0.083$ |
| | EU | $-0.061$ | $-0.119$ | $-0.088$ | $-0.084$ | $-0.074$ | $-0.070$ | $-0.106$ | $-0.067$ | $-0.107$ | $-0.109$ |
| | Dem | $-0.152$ | $-0.162$ | $-0.134$ | $-0.171$ | $-0.105$ | $-0.130$ | $-0.124$ | $-0.123$ | $-0.169$ | $-0.021$ |
| | bitA | $-0.259$ | $-0.243$ | $-0.267$ | $-0.280$ | $-0.245$ | $-0.309$ | $-0.373$ | $-0.413$ | $-0.390$ | $-0.052$ |
| | bitOT | $-0.252$ | $-0.263$ | $-0.264$ | $-0.308$ | $-0.325$ | $-0.376$ | $-0.395$ | $-0.353$ | $-0.371$ | $-0.014$ |
| | chess | $-0.317$ | $-0.349$ | $-0.368$ | $-0.377$ | $-0.410$ | $-0.406$ | $-0.403$ | $-0.281$ | $-0.382$ | $0.036$ |
| | HepTh | $-0.142$ | $-0.189$ | $-0.202$ | $-0.234$ | $-0.276$ | $-0.248$ | $-0.249$ | $-0.220$ | $-0.177$ | $-0.020$ |
| | HepPh | $-0.162$ | $-0.193$ | $-0.208$ | $-0.213$ | $-0.226$ | $-0.234$ | $-0.201$ | $-0.177$ | $-0.137$ | $-0.034$ |
| * | Condm | $-0.243$ | $-0.252$ | $-0.269$ | $-0.294$ | $-0.344$ | $-0.273$ | $-0.263$ | $-0.252$ | $-0.243$ | $-0.095$ |
| | SX-MO | $-0.161$ | $-0.167$ | $-0.179$ | $-0.187$ | $-0.178$ | $-0.205$ | $-0.212$ | $-0.194$ | $-0.200$ | $-0.015$ |
| | D-rep | $-0.416$ | $-0.445$ | $-0.506$ | $-0.586$ | $-0.332$ | $-0.233$ | $-0.202$ | $-0.187$ | $-0.167$ | $-0.006$ |
| | Rbody | $-0.162$ | $-0.169$ | $-0.187$ | $-0.178$ | $-0.182$ | $-0.219$ | $-0.220$ | $-0.243$ | $-0.248$ | $0.015$ |
| | Rtit | $-0.136$ | $-0.124$ | $-0.132$ | $-0.144$ | $-0.132$ | $-0.156$ | $-0.191$ | $-0.198$ | $-0.188$ | $0.031$ |
| | FB-w | $-0.240$ | $-0.250$ | $-0.239$ | $-0.239$ | $-0.242$ | $-0.251$ | $-0.273$ | $-0.291$ | $-0.326$ | $0.084$ |
| | FB-l | $-0.246$ | $-0.253$ | $-0.257$ | $-0.244$ | $-0.232$ | $-0.236$ | $-0.266$ | $-0.291$ | $-0.325$ | $0.096$ |
| | Enron | $-0.165$ | $-0.171$ | $-0.177$ | $-0.191$ | $-0.188$ | $-0.211$ | $-0.214$ | $-0.228$ | $-0.200$ | $0.004$ |
| | loans | $-0.347$ | $-0.413$ | $-0.459$ | $-0.333$ | $-0.300$ | $-0.230$ | $-0.215$ | $-0.229$ | $-0.265$ | $-0.024$ |
| | trust | $-0.198$ | $-0.215$ | $-0.216$ | $-0.253$ | $-0.246$ | $-0.300$ | $-0.301$ | $-0.264$ | $-0.205$ | $0.012$ |
| | Wiki | $-0.003$ | $-0.211$ | $-0.218$ | $-0.243$ | $-0.296$ | $-0.446$ | $-0.407$ | $-0.378$ | $-0.336$ | $-0.029$ |
| | D-v | $0.097$ | $-0.011$ | $-0.019$ | $-0.044$ | $-0.073$ | $-0.077$ | $-0.047$ | $-0.047$ | $-0.043$ | $0.017$ |
| | SX-AU | $-0.276$ | $-0.281$ | $-0.279$ | $-0.280$ | $-0.287$ | $-0.408$ | $-0.440$ | $-0.445$ | $-0.468$ | $-0.005$ |
| | SX-SU | $-0.244$ | $-0.265$ | $-0.272$ | $-0.309$ | $-0.302$ | $-0.389$ | $-0.397$ | $-0.408$ | $-0.392$ | $-0.002$ |
| | D-f | $-0.170$ | $-0.202$ | $-0.227$ | $-0.263$ | $-0.292$ | $-0.325$ | $-0.295$ | $-0.278$ | $-0.213$ | $0.012$ |
| | AMin | $-0.278$ | $-0.292$ | $-0.292$ | $-0.310$ | $-0.320$ | $-0.385$ | $-0.337$ | $-0.375$ | $-0.372$ | $-0.095$ |
| | DBLP | $-0.335$ | $-0.331$ | $-0.330$ | $-0.358$ | $-0.361$ | $-0.431$ | $-0.357$ | $-0.443$ | $-0.427$ | $-0.046$ |
| | mean | $-0.202$ | $-0.229$ | $-0.237$ | $-0.253$ | $-0.245$ | $-0.269$ | $-0.269$ | $-0.265$ | $-0.261$ | $-0.012$ |

### 2.6.4 Experiment 3: Enhancement of performance with past event aggregation

In this subsection we address Experiment 3. Two sets of features are used to examine *how networks with different types of temporal information can be exploited in link prediction to improve link prediction performance*. Feature set (II-A) is constructed with past event aggregation, allowing the use of the information contained in all discrete events. Feature set (II-B) considers only the last occurring edge between two nodes, ignoring past events.

The performance obtained with these two sets of features is reported in Table 2.2. The two sets of features yield the same results for networks with persistent edges because the networks do not contain past events. In Figure 2.6, we show the difference between the two performances of the networks with discrete events in more detail.

From the results of the experiment, we may conclude that networks with discrete events perform better when aggregating past events. The result is broadly interesting for link prediction research, as the derived feature modification steps can be inserted into any topological network feature aiming to capture the similarity of nodes in an attempt to predict their future connectivity. Interestingly, we observe significant differences in the performance improvement of past event aggregation for each discrete event network.

On the one hand, we observe networks with only minor improvement when past events are aggregated. For example, the Condense matter scientific collaboration network (Condm, indicated with * in Tables 2.1 to 2.3) shows only a minor improvement of $0.706$ to $0.760$ AUC. A possible explanation is that temporal information of discrete events has limited use since it takes time to develop a successful collaboration.

On the other hand, the UC Irvine message network (indicated with ** in Tables 2.1 to 2.3) shows an improvement in AUC from $0.744$ to $0.893$. The improvement might be caused by the more variable nature of messages, which take only a short time to establish. In that case, the feature set with past event aggregation might provide higher scores to pairs of actively messaging nodes.

### 2.6.5 Experiment 4: Comparison of node- and edge-centered link prediction

Earlier, in Subsection 2.6.2 we examined whether temporal topological features improve link prediction performance. These features assume edge-centered temporal behavior. Now, in Experiment 4, we compare *the performance of edge-centered features with features that assume node-centered temporal behavior*. The link prediction of edge-centered features is done with Feature set II-A, and node-centered features are contained in Feature set III.

The results of both feature sets are presented in Table 2.2 and more detailed in Figure 2.7. We observe a strong correlation ($\rho = 0.92$, $p = 0.009$) between obtained performances using both feature sets. It suggests that most networks' temporal aspects can be modeled using either node-centered or edge-centered temporal features.

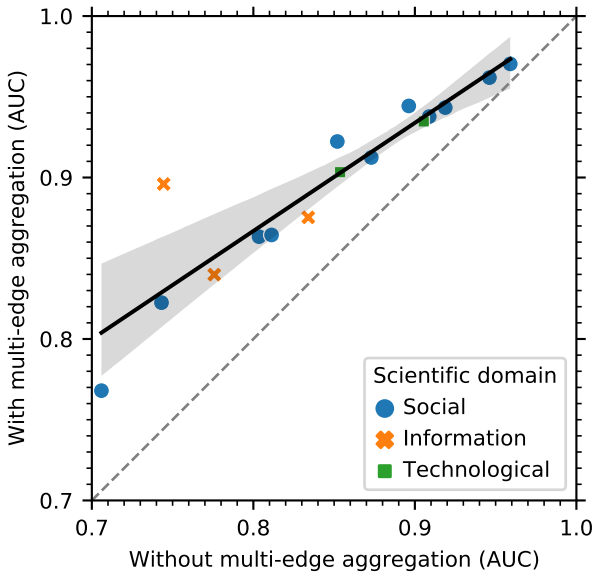



Figure 2.6: Link prediction performance with (Feature set II-A, $y$-axis) and without (Feature set II-B, $x$-axis) past event aggregation.

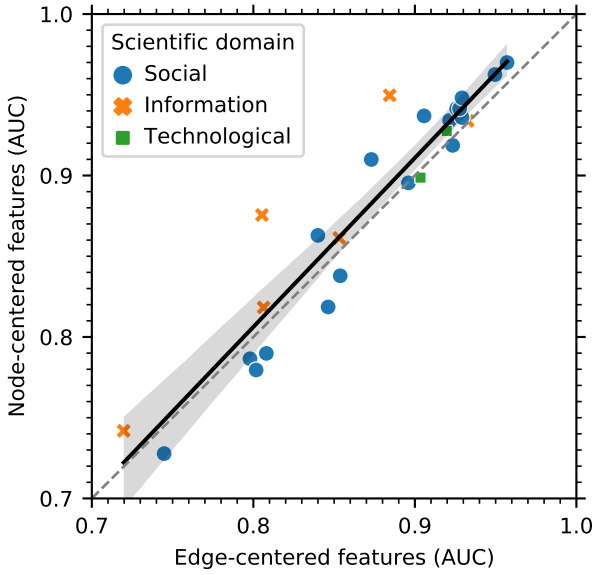



Figure 2.7: Link prediction performance of node-centered features (Feature set III, $x$-axis) and edge-centered features (Feature set II, $y$-axis). (The dotted line indicates equal performance. The solid black line indicates the best fit using linear regression. All networks are shown.)

A new indicator for further research is that for the four information networks, the performance of the node-centered features is higher than edge-centered features. This finding hints that in information networks, temporal behavior may be node-centered. Given this study's low number of information networks, further research should be conducted on a more extensive set of information networks to verify this finding.

We analyze the link performance only on pairs of nodes at a distance of two; different findings may be observed using more global features of node similarity are used. Notwithstanding this limitation, based on the current results, we may conclude that both node- and edge-centered features in supervised link predictions can achieve high performance.

## 2.7 Chapter conclusions and outlook

This chapter addressed Research question 1: "Wat is the relation between network structure and model performance in link prediction?" We performed a large-scale empirical study of link prediction using various structurally diverse networks. Moreover, we aimed to demonstrate the benefit of past event aggregation, allowing us to take the rich interaction history of nodes into account in predicting their future linking activity. This study resulted in four findings, that substantiate the relation between network structure and model performance in link prediction.

- Supervised link prediction performance is consistently higher when temporal information is considered (Subsection 2.6.2).
- The performance in link prediction appears related to the global structure of the network (Subsection 2.6.3). Most notably, degree disassortative networks perform better than degree assortative networks.
- We proposed an approach to deal with event-based links by aggregating information from multiple past interactions (Subsection 2.6.4). It increases the performance of link prediction. The derived feature modification steps can be inserted into any topological feature, potentially improving the performance of any supervised link prediction.
- We showed that in four information networks, features capturing node activity and static topological features outperform features that consider edge-centered temporal information, suggesting that the temporal mechanisms in these networks may reside with the nodes (Subsection 2.6.5).

### Chapter outlook

The next step of this work may be to analyze networks originating from different domains. It appears that publicly available networks from other domains, such as biological, economic, and transportation networks, typically do not contain temporal information [63]. However, it would be interesting to investigate whether the findings presented in this chapter also hold for these types of networks.

# 3

# Performance of split strategies in link prediction

In Chapter 2, we have explored supervised machine learning towards the link prediction task. To train, validate (Definition 3) and test (Definition 4) supervised machine learning models, we need disjoint and independent splits of data. However, nodes in a real-world network are inherently related to each other. Therefore, separating candidate links into these disjoint sets is impossible. This challenge leads to Research question 2, which reads as follows.

> **Research question 2:** *How can we obtain accurate estimates of the performance of link prediction models by using adequate splits into the train, validation, and test set?*

In this chapter, we will evaluate two approaches to split data in link prediction: the *random* split and the *temporal* split. We will compare their performances on six large network datasets.

The current chapter corresponds to the following publication:
G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Experimental evaluation of train and test split strategies in link prediction." In: *Proceedings of the 9th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 994. Springer, 2021, pages 79–91. DOI: 10.1007/978-3-030-65351-4_7

## 3.1 Machine learning methods on networks

Machine learning has emerged as a powerful instrument for analyzing all kinds of datasets. Here, we focus on supervised learning, which is well established when using non-relational (i.e., tabular) data. However, supervised machine learning on network data is challenging because obtaining an independent train, validation, and test set is nontrivial [74]. A common type of machine learning in networks is link prediction, where the goal is to predict whether a link will be formed in some future state of an evolving network (see Section 1.4 and Definition 10). In recent years, there has been an increasing interest in link prediction; hence, several review papers on this topic exist, e.g., [4, 101, 114].

A crucial first step in machine learning in networks is engineering the features. Here we assume that the network topology data can be converted into features with potentially helpful information for a predictive model. Established approaches for feature engineering in link prediction are based on (1) similarity, (2) probabilistic and maximum likelihood, and (3) dimensionality reduction [101]. Following our approach in Chapter 2, we will focus on the similarity-based approach. In this approach, pairs of nodes (candidates for links to be formed in the future) are assigned scores according to their similarity. We will exclusively use topological features to assess similarity, so we can apply the feature engineering to networks where no additional information is available about the nodes. The similarity-based approach provides at least three benefits. First, similarity-based features provide more accurate results compared to embedding techniques [63]. Second, the similarity-based approach provides easily explainable features compared to other techniques [114]. Third, most features are obtained at relatively low computational costs for the more extensive networks used in this study [114].

The similarity-based approach brings us to the main problem addressed in this chapter. For proper validation and testing in any machine learning task, instances belonging to the train set (on which the model will be trained) should be *disjoint* and *independent* of features belonging to the validation and test set. Since many dependencies usually exist between nodes in a network, it is a challenging task to achieve. We should seriously take into consideration that obtaining a *dependent* validation and test set possibly results in too-optimistic performance measurements (or, equivalently, overestimating the so-called generalization performance of the model [76]). According to Ghasemian *et al.*, it still needs to be determined how common machine learning steps, such as cross-validation and model selection methods, extend from non-relational to network data [62].

Assessment of the performance in supervised machine learning is essential for at least two reasons. The first reason is the selection of an appropriate model. It is possible to construct completely different models for a particular task, ranging from entirely different classifiers to identical models with other (hyper)parameters. Of course, we prefer to select a model with the best generalization performance on a dataset independent of the

train set. Here we aim at an independent validation set so that we can assess the extent to which overfitting takes place (Definition 11). The second reason for assessing model performance is to estimate the prediction error on new, unseen data. The performance should be assessed (1) by using the test data that is not used in any part of training the model, or (2) in choosing the right hyper-parameters or selecting a model [76]. Our research evaluates the differences in collecting independent datasets to examine a classifier's generalizability score. In our procedure, we perform a split only once, so two datasets are obtained (for instance, train and test data). Here we remark that our research methodology can easily be extended to obtain an independent third set (for instance, for validation purposes).

This work contributes to making an *in-depth* comparison of two approaches to splitting network data into two disjoint sets. Here, we aim to contribute to a better evaluation of performance estimation in link prediction and will answer Research question 2.

The remainder of this chapter is structured as follows. Related work is discussed in Section 3.2. Our research methodology (a formal description of the link prediction problem) is presented in Section 3.3. Relevant properties of the six temporal networks are presented in Section 3.4. Section 3.5 features information about the experimental setup. Then, Section 3.6 is concerned with the precise description of the experimental setup, the results, and a discussion of these results. Conclusions and future work are provided in Section 3.7.

## 3.2 Related work on validation of link prediction models

Only a relatively small body of literature is directly concerned with splitting a network dataset into disjoint and independent sets to evaluate the performance for machine learning purposes. We start our exploration with literature on performance estimation *in general* before we focus on prediction tasks in networks.

One of the causes of too-optimistic performance estimation is what is often described as "test set re-use" [167]. A well-known example is the $p$-hacking problem [91]. In short, $p$-hacking is the application of many different models to the same data in search of a statistically significant result with a sufficiently high $p$-value. This misuse can increase the probability that applied research findings are false. More specifically to data-driven research, too-optimistic performance estimations are suspected in Kaggle competitions [167]. In these online competitions, participants get the same dataset and compete for the best classifier performance on some predictive tasks without access to the test data. However, Kaggle allows users to repeatedly probe test data to obtain a continuously better performance of a submitted model. It is argued that this would lead to too-optimistic results [53]. However, the optimistic results were experimentally only observed to a limited extent [167].

Returning to machine learning on networks, Ghasemian *et al.* [62] investigated under- and overfitting networks. They examined (1) the performance of missing link prediction and (2) the so-called link description task as a diagnostic to evaluate the general tendency of such algorithms to under- and overfit. Hence, it is remarked that the authors defined the link prediction task differently since they do not necessarily include temporal information about the edges (see also Section 1.4). Hence, they removed a fraction of edges from a network and employed a machine learner to find the removed links from all pairs of nodes that are not connected anymore. The link description problem is different, as explained below. A network is sampled, but now the machine learner's task is finding the remaining edges of the sampled network from all pairs of nodes. The previously mentioned authors explain that (1) no algorithm can excel at both the link prediction and link description task and (2) that these tasks force an algorithmic trade-off, like the bias-variance trade-off in non-relational data [76]. In our work, we want to bring the notion of overfitting from non-relational data to relational data. While Ghasemian *et al.* focus on overfitting caused by the bias-variance trade-off [62], we investigate the too-optimistic estimation of generalization performance caused by test set reuse in networks.

## 3.3 Chapter research methodology

This section will start with a formal description of the link prediction problem in Subsection 3.3.1. In Subsection 3.3.2, we explain how we split the data into disjoint and separate sets for the link prediction classifier. Subsection 3.3.3 continues with the description of two types of features. In Subsection 3.3.4 we provide information about the classifier. Finally, in Subsection 3.3.5, we explain the performance metrics used.

### 3.3.1 Link prediction

The link prediction task is similarly defined as in Chapter 2. The temporal, potentially undirected, network (see Definition 9) $G = (V, E)$ consists of a set of nodes $V$ and edges $(u, v, t_i) \in E$ connecting nodes $u, v \in V$ with time $t_i \geq t_a$. Time $t'$ indicates the time of the first edge occurring in $G$. Parallel edges with different timestamps can exist.

Since the network is temporal, we can construct snapshots of network $G$ for a given time interval. We denote such a snapshot with $G_{[t_a,t_b]} = (V_{[t_a,t_b]}, E_{[t_a,t_b]})$ with $E_{[t_a,t_b]}$ being a set consisting only of edges occurring between $t_a$ and $t_b$ (with $t_a < t_b$) and $V_{[t_a,t_b]}$ the nodes taking part in these edges. Now assume that we make two such snapshots, $G_{[t_a,t_b]}$ and $G_{[t_b,t_c]}$ from two time intervals $[t_a, t_b]$ and $[t_b, t_c]$ with $t_a < t_b < t_c$. The evolution of a temporal network is shown in Figure 3.1a.

The task for the supervised binary link prediction classifier (explained in Subsection 3.3.4) is to predict from $G_{[t_a,t_b]}$ whether a pair of nodes will connect in $G_{[t_b,t_c]}$.

Hence, the input for the classifier is all pairs of nodes $X_{[t_a,t_b]} = \left(V_{[t_a,t_b]} \times V_{[t_a,t_b]}\right) \setminus E_{[t_a,t_b]}$ (see also Figure 3.1b). The network $G_{[t_a,t_b]}$ needs to be sufficiently "mature" so that the underlying static topology is well captured [112]. Hence we call the period of time $[t_a, t_b]$ the *maturing interval*. Subsequently, we call the time period $[t_b, t_c]$ the *probing interval*. For every pair of nodes $x_i \in X_{[t_a,t_b]}$, we probe whether the couple is present in the probing interval (indicated by $y_i = 1$) or not (denoted $y_i = 0$). The entire procedure is summarized in Figure 3.1. In Figure 3.1b, the instances considered in the classifier are shown. Positive instances ($y_i = 1$) are shown in solid green lines, while negatives ($y_i = 0$) are shown in red dashed lines.

## 3.3.2 Splitting strategies

After describing the general procedure of link prediction, we need a strategy to separate the pairs of nodes into different disjoint and independent sets for the classifier. Below, we will explain two dominant methods to split the data [3, 112]. Applying a temporal split is more complicated than the random split due to the various parameters involved. However, a temporal split prevents, to a greater extent, the reuse of the node and edge set information from the test set in training.

### 3.3.2A Random split

In the random split, the train and test sets are obtained by randomly splitting instances from a single probing phase. The validation set was omitted in our research (see Section 3.1). The random split method is, e.g., used in [112]. The entire procedure consists of three steps (see also Figure 3.2a).

1. We obtain all pairs of nodes disconnected during the maturing phase, $X_{[t_a,t_b]}$.
2. We determine for each of these pairs of nodes whether they connect (the value of $y_i$) in the probing phase $E_{[t_b,t_c]}$, as shown in Equation 3.1.

$$y_i = \begin{cases} 1 & \text{if } x_i \in E_{[t_b,t_c]} \\ 0 & \text{if } x_i \notin E_{[t_b,t_c]} \end{cases} \text{ for } x_i \in X_{[t_a,t_b]} \tag{3.1}$$

3. The pairs of nodes $X_{[t_a,t_b]}$ are separated into two disjoint sets $X^{\text{train}}_{[t_a,t_b]}$ and $X^{\text{test}}_{[t_a,t_b]}$ such that $X^{\text{train}}_{[t_a,t_b]} \cup X^{\text{test}}_{[t_a,t_b]} = X_{[t_a,t_b]}$ and $X^{\text{train}}_{[t_a,t_b]} \cap X^{\text{test}}_{[t_a,t_b]} = \emptyset$.

### 3.3.2B Temporal split

In the temporal split two consecutive probing phases obtain a train and a test set from two different time intervals. The temporal split method is for example used in [3]. As the name states, it takes the temporal aspect into account. More specifically, in the temporal split method two disjoint datasets are obtained by applying the probing phase on two

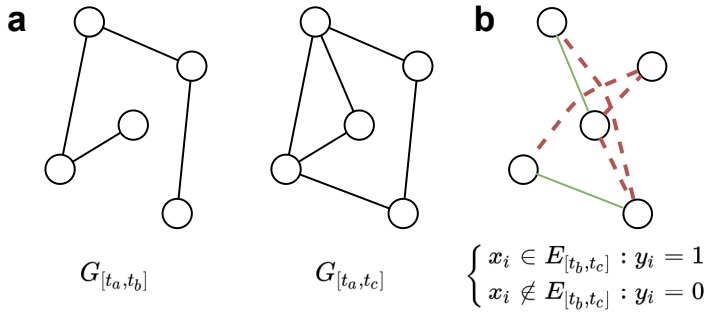
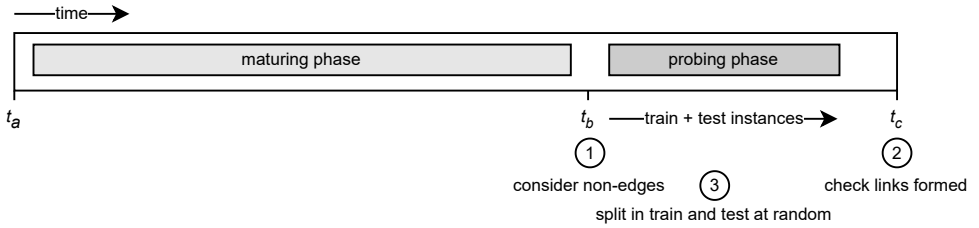
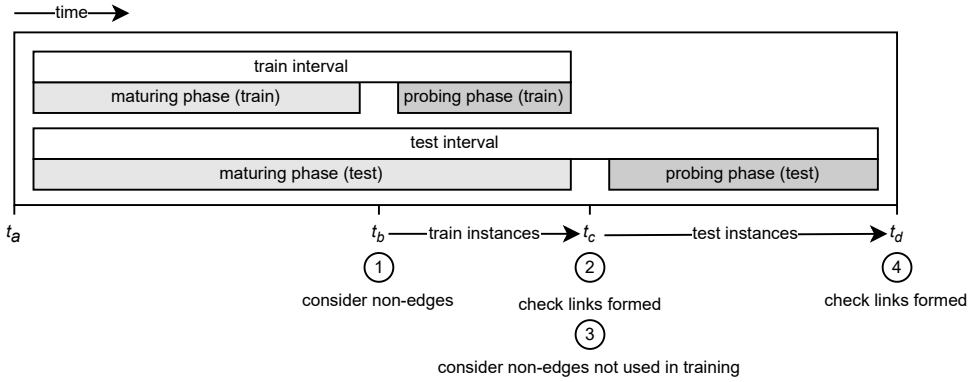



Figure 3.1: Procedure to obtain instances for the binary link prediction.




(a) Random split.




(b) Temporal split.

Figure 3.2: Two different strategies exist to obtain disjoint and independent sets.

consecutive snapshots called the training interval $[t_b, t_c]$ and test interval $[t_c, t_d]$. The four steps of this process are shown schematically in Figure 3.2b.

The train set is constructed in the first two steps as follows.

1. We consider every node pair that is not connected in the maturing phase of the train interval $X_{[t_a,t_b]}$.
2. For each node pair, we determine whether it will connect in the probing phase of the train interval, like Equation 3.1.

The test set is constructed similarly to the train set in Step 3 and 4.

3. We consider every node pair that is not connected in the maturing phase of the test interval $X_{[t_a,t_c]}$ and not used in the probing phase of the train interval.
4. We determine whether each pair of nodes connects in the probing phase of the test interval, as shown in Equation 3.2.

$$y_i = \begin{cases} 1 & \text{if } x_i \in E_{[t_c,t_d]} \\ 0 & \text{if } x_i \notin E_{[t_c,t_d]} \end{cases} \text{ for } x_i \in X_{[t_a,t_c]} \text{ and with } t_c < t_d \qquad 3.2$$

### 3.3.3 Features

As input for a classifier, we need a feature representation for every pair of nodes $x_i \in X$. As discussed in Section 3.1, we use the well-established similarity-based approach, where the feature for each pair of nodes $x_i = (u, v)$ consists of a particular score for each feature $S_{\text{feature}}(u, v)$. These scores are based solely on topological properties intrinsic to the network and not on contextual information [112, 125]. Hence, the features do not need any node information. Nodes with similar scores and thus a high similarity are more likely to connect. The score is either neighbor-based (similarity in local properties of the two nodes) or path-based (quasi-local or based on global properties of the two nodes) [40, 101]. We use the so-called High-Performance Link Prediction (HPLP) feature set defined in [112], as these are known to obtain good performance while limiting the number of features. The features can be separated into two types of features; the neighbor-based (Subsection 3.3.3A) and path-based (Subsection 3.3.3B) features. The features differ from those used in Chapter 2, where we used features that could be temporally extended to take past interactions into account (see Section 2.1).

In directed networks, we differentiate between (1) the neighbors connecting to node $u$, indicated by $N_{\text{in}}(u)$, and (2) the neighbors to which node $u$ connects, $N_{\text{out}}(u)$. Likewise, we differentiate also between the in-degree and out-degree of node $u$, $|E_{\text{in}}(u)|$ and $|E_{\text{out}}(u)|$, respectively.

#### 3.3.3A Neighbor-based features

Neighbor-based features take only the direct neighbors of the two nodes under consideration into account. Below we provide definitions of three concepts useful in subsequent feature definitions.

- **Number of Neighbors (NN)** is determined differently for directed and undirected networks. For *directed* networks, we use (1) the number of neighbors connecting to nodes $u$ and $v$ and (2) the number of nodes connected by $u$ and $v$. Hence, we get four features: $S_{\text{NN-in-}u}(u,v) = |N_{\text{in}}(u)|$; $S_{\text{NN-in-}v}(u,v) = |N_{\text{in}}(v)|$; $S_{\text{NN-out-}u}(u,v) = |N_{\text{out}}(u)|$; and $S_{\text{NN-out-}v}(u,v) = |N_{\text{out}}(v)|$. For the *undirected* case, the same score for pairs of nodes $(u,v)$ and $(v,u)$ is desired, and there is no difference between the number of nodes connecting from or to node $u$. Hence, we report both the maximum and minimum for a given pair of nodes, i.e., $S_{\text{NN-min}}(u,v) = \min(|N(u)|, |N(v)|)$ and $S_{\text{NN-max}}(u,v) = \max(|N(u)|, |N(v)|)$.
- **Degree (D)** is defined similarly, except that the *number of edges* is considered. For *directed* networks, we obtain again four features, viz. $S_{\text{D-in-}u}(u,v) = |E_{\text{in}}(u)|$; $S_{\text{D-in-}v}(u,v) = |E_{\text{in}}(v)|$; $S_{\text{D-out-}u}(u,v) = |E_{\text{out}}(u)|$; and $S_{\text{D-out-}v}(u,v) = |E_{\text{out}}(v)|$. For *undirected* networks, we obtain the maximum and minimum degree of nodes $u$ and $v$; $S_{\text{D-min}}(u,v) = \min(|E(u)|, |E(v)|)$; and $S_{\text{D-max}}(u,v) = \max(|E(u)|, |E(v)|)$.
- The **Common Neighbors (CN)** for a given pair of nodes is calculated by $S_{\text{CN}}(u,v) = |N(u) \cap N(v)|$. For *directed* networks, the score is calculated by considering the nodes that are connected from nodes $u$ and $v$, i.e., $S_{\text{CN}}(u,v) = |N_{\text{out}}(u) \cap N_{\text{out}}(v)|$.

#### 3.3.3B Path-based features

Path-based features take into account the paths between the two nodes under consideration. Since many paths exist, these features are computationally more expensive than neighbor-based ones. Below we provide the features with their definitions.

- **Shortest Paths (SP)**, $S_{\text{SP}}(u,v)$, indicates the number of shortest paths that run between nodes $u$ and $v$.
- **PropFlow (PF)**, $S_{\text{PF}}(u,v)$, corresponds to the probability that a restricted random walk starting from node $u$ and ends at node $v$ within $\ell$ steps [112]. We use the commonly applied value of $\ell = 5$ [112]. We collapse the network with multiple edges (occurring at different timestamps) to a weighted network where the weight equals the number of parallel edges between two nodes. Higher weights result in a higher transition probability for the random walk. This method is known for potentially obtaining scores for pairs of nodes $(u,v)$ that are different from those obtained for the pair $(v,u)$. This observation even holds for pairs in the undirected case [209]. Hence, we use the mean of the scores obtained for the pairs of nodes $(u,v)$ and $(v,u)$ in the undirected case.

### 3.3.4 Tree-based gradient boost classifier

We used a tree-based gradient boost learner for our classifier, as these are known to perform well in classification tasks [76]. The Python implementation of XGBoost was used [41]. This classifier has various hyper-parameters. While extensive hyper-parameter

tuning is beyond the scope of this chapter, we cross-validate two important hyper-parameters, viz. maximum depth of tree and class weights.

### 3.3.5 Performance metric: Average Precision

Link prediction is associated with *extreme class imbalance*, lower bounded by the number of nodes in the network [112]. Ideally, performance metrics used to evaluate the classifier should be robust against this class imbalance. The commonly encountered AUC lacks this robustness [111, 209] and is therefore not used. We are particularly interested in correctly predicting positives without losing precision, i.e., keeping the number of false positives low, and without losing recall, i.e., making sure we find all true positives. The Average Precision (AP) metric equals the weighted mean of precisions achieved at each threshold in the precision-recall curve. It is well-suited for our case.

## 3.4 Properties of the six temporal networks

Since our research aims to split the network into different snapshots based on time, temporal networks are needed. In this work, we use six different temporal networks that are (1) spanning a broad range of different domains, (2) publicly available, and (3) sufficiently large. The properties of these networks are shown in Table 3.1. We mention for each network whether it is directed, the number of nodes, the number of edges, the density, the mean distance ($\bar{d}$), and the diameter (Ø). The density, mean distance, and diameter were calculated on the underlying static network, i.e., the network without parallel edges. Below, we briefly discuss the six datasets used in this work. Except for the Condmat network, all datasets were obtained from KONECT [104].

1. The *Ask Ubuntu* network is an online contact network [143]. The snapshot of the network that we used was obtained in 2017. Ask Ubuntu is a community-driven question-and-answer site dedicated towards Ubuntu; it is derived from StackExchange,

Table 3.1: Summary statistics of the six temporal networks. (Edges and nodes in the GC are indicated between brackets. The mean distance between nodes is given in column $\bar{d}$, and column Ø indicates the diameter of the networks.)

| # | dataset | directed | nodes | (GC) | edges | (GC) | density | $\bar{d}$ | Ø |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Ask Ubuntu | ✓ | 159,316 | (96%) | 964,437 | (100%) | $4.0 \times 10^{-5}$ | 3.9 | 13 |
| 2 | Condmat | ✗ | 17,218 | (88%) | 88,090 | (100%) | $3.7 \times 10^{-4}$ | 6.3 | 19 |
| 3 | Digg | ✓ | 30,398 | (98%) | 87,627 | (100%) | $1.9 \times 10^{-4}$ | 4.7 | 12 |
| 4 | Enron | ✓ | 87,273 | (97%) | 1,149,072 | (100%) | $7.9 \times 10^{-5}$ | 4.9 | 14 |
| 5 | Slashdot | ✓ | 51,083 | (100%) | 140,778 | (100%) | $9.0 \times 10^{-5}$ | 4.5 | 17 |
| 6 | Stack Overflow | ✓ | 2,601,977 | (99%) | 63,497,050 | (100%) | $8.7 \times 10^{-6}$ | 3.9 | 11 |

a network of question-and-answer websites on topics in diverse fields. The nodes are the users, and a direct edge is created when a user replies to another user's message. These interactions can consist of an answer to another user's question, comments on another user's answer, and comments on another user's comments. Each edge is annotated with the time of interaction.

2. The scientific co-authorship dataset *Condmat* entails condensed matter physics collaborations from 1995 to 2000[1]. The undirected temporal network is made by adding an edge between all authors of a publication [111]. For each edge, the date of the publication connecting these authors is used. We observe that the number of authors per paper increases over time. It may cause varying performance in link prediction for different temporal snapshots. We deemed this outside the current research scope.
3. The *Digg* network is a communication network and contains the reply network of the social news website Digg from November 2009 [79]. Each node in the network is a person, and each edge connects the user replying to the reply receiver. Each reply is annotated with the time of that interaction.
4. The *Enron* dataset is a communication network and contains over a million emails sent between employees of Enron between 1999 and 2003 [97]. A directed edge from the sender to the recipient is added for each email.
5. The *Slashdot* website is a English tech news website that allows users to place a comment on each page, and shows where users can start a threaded discussion [65, 159]. The period during which the data was crawled covered August 2005 to August 2006 [85]. The communication network is constructed from these threads where users are nodes, and replies are edges, annotated with the answer time.
6. Like Ask Ubuntu, the *Stack Overflow* network is collected from StackExchange and can be considered an online contact network [143]. Nodes are users; directed edges represent interactions annotated with the exchange time.

## 3.5 Experimental setup

In Section 3.3 we explained the research methodology of our experiment. However, a few parameters need to be addressed explicitly to run the link prediction task (cf. Subsection 3.3.1) used in the experiment. We discuss them in the sections below. First, we discuss the *selection of node pairs* using their *distance* in Subsection 3.5.1. Second, in Subsection 3.5.2, we discuss the *choice of the time intervals* for the maturing and probe phases for both the *random* and *temporal* split (Subsection 3.3.2). Third, we continue in Subsection 3.5.3 by discussing the *number of pairs of nodes* used for *training* and *testing*. Fourth, the *performance is improved* by optimizing *the class weight* and *the value of the maximum tree depth* in Subsection 3.5.4. The class weight and the maximum tree depth

[1]The data was obtained from https://github.com/rlichtenwalter/LPmade

are called hyper-parameters. Fifth, we explain in Subsection 3.5.5 how *multiple snapshots* from a network are *constructed* for robustness checks.

### 3.5.1 Distance selection

The task of link prediction is computationally intensive for larger networks because there are $\left|\left(V_{[t_a,t_b]} \times V_{[t_a,t_b]}\right) \setminus E_{[t_a,t_b]}\right|$ instances. A way to reduce the number of instances and to reduce class imbalance, is to only consider pairs of nodes at a limited distance of each other in the network [111]. For our distance selection, we consider only pairs of nodes at a distance of two in our network.

### 3.5.2 Time intervals

The choice of the time intervals used for the maturing and probing phase in both the random and temporal split can affect the obtained results. To allow *fair* comparisons between the random and temporal split, the probing phase of the test interval should contain a number of edges similar to the probing phase of the training interval, i.e., $\left|E_{[t_a,t_b]}\right| \approx \left|E_{[t_c,t_d]}\right|$. Moreover, we need values that are consistent for the various networks. Timestamps of $t_b$, $t_c$, and $t_d$ were set so that the proportion of edges in the maturing and probing phase are approximately similar to the settings in [112].

For the Condmat network, this results in a ratio $\left|E_{[t_a,t_b]}\right| : \left|E_{[t_b,t_c]}\right|$ approximately equal to $5:1$. The number of edges are then $\left|E_{[t_a,t_b]}\right| \approx 50000$ and $\left|E_{[t_b,t_c]}\right| \approx \left|E_{[t_c,t_d]}\right| \approx 10000$.

### 3.5.3 Training and testing

In the case of random splitting, the instances $X_{[t_a,t_b]}$ should be split into two disjoint and independent sets, as explained in Subsection 3.3.2A. A 75% of the instances are used for training and the remainder for testing, i.e., $\left|X^{\text{train}}_{[t_a,t_b]}\right| = 3\left|X^{\text{test}}_{[t_a,t_b]}\right|$.

### 3.5.4 Improved performance

Below, we report the choices made regarding two hyper-parameters used in the XGBoost algorithm to improve performance. First, we adjusted the *class weights* of the positive instances to equal the total weight of the positive and negative samples. In a five-fold cross-validation setting applied to the training data, we determined for each network separately whether the adjusted class weights improve performance on the train set. Second, we determined the optimal *maximum tree depth* using the same five-fold cross-validation.

### 3.5.5 Robustness checks

The experimental setup of the *robustness checks* is as follows. We repeat the full procedure of selecting time intervals (see Subsection 3.5.2) on the Ask Ubuntu network ten times. Ten non-overlapping snapshots are obtained by shifting intervals such that each next interval starts ($t_a$) at the end of the previous interval ($t_c$ for random split, $t_d$ for temporal split). The robustness is then checked by comparing the Average Precision performance of the random split with that of the temporal split.

## 3.6 Results of the two different splitting strategies

The Average Precision (AP) (see Subsection 3.3.5) score of the classifiers for the six networks with the random split and temporal split method is shown in Table 3.2. This metric shows significant performance differences between the random and temporal split. The performance of the temporal split is for all networks lower than the random split. It may indicate that the random split provides an overly optimistic indication of the performance value. Furthermore, the difference between the random and temporal splits varies widely between the networks, indicating that the extent to which the test set is reused varies by network. Notably, the AP of the Ask Ubuntu network drops by 80%, demonstrating that the test set reuse can be extensive.

### Robustness checks

We checked the robustness of the findings by following the procedure (ten times performing the full procedure, as outlined in Subsection 3.5.5). We find an AP of $0.025 \pm 0.009$ (mean$\pm$standard deviation) when using the random split, while an AP of only $0.0061 \pm 0.0016$ is found for the temporal split. The different AP curves are shown in Figure 3.3. From our results, we may conclude that the random split precision-recall curves dominate their temporal counterparts in all snapshots.

Table 3.2: Link prediction performances for different split strategies (applied on the six temporal networks using the AP metric).

| # | dataset | random split | temporal split |
|---|---|---|---|
| 1 | AskUbuntu | 0.023 | 0.0046 |
| 2 | Condmat | 0.012 | 0.0048 |
| 3 | Digg | 0.0043 | 0.0014 |
| 4 | Enron | 0.016 | 0.012 |
| 5 | Slashdot | 0.0076 | 0.0021 |
| 6 | Stack Overflow | 0.0029 | 0.0013 |

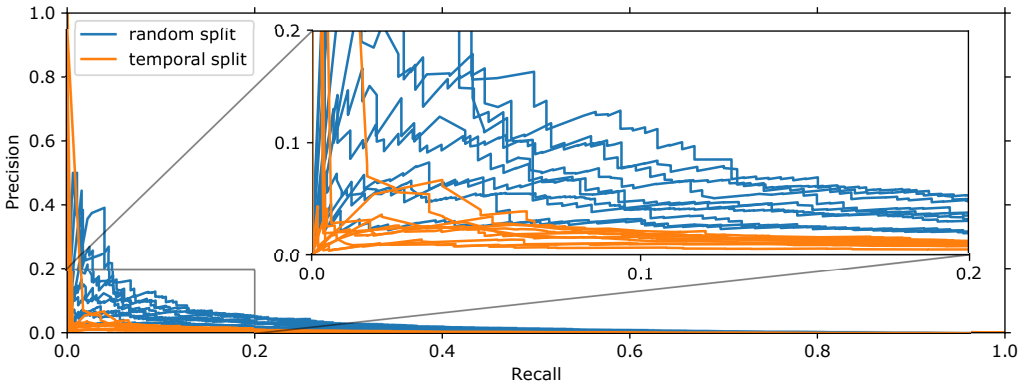



Figure 3.3: Precision-recall curves of the AskUbuntu network for robustness checks. (Performed on ten different snapshots.)

## 3.7 Chapter conclusion and outlook

In the present chapter, we analyzed two different ways of splitting data viz. into disjoint and independent sets in network data for training, validation, and testing of link prediction models. The results indicate that the *random split* consistently obtains higher performance estimates than the *temporal split*.

So, we are able to answer Research question 2: "How can we obtain accurate estimates of the performance of link prediction models by using adequate splits into train, validation, and test sets?". The answer is: "We obtain accurate estimates of the link prediction performance by using the temporal split, because the alternative, the random split, shows signs of overfitting." Based on our experiments we may conclude that the temporal split method provides more accurate estimates of the link prediction model performance.

### Chapter outlook

While the procedure of the temporal split prevents using the same temporal information of a given node, it still allows the same node to be used in multiple sets. Future work should devise more rigorous strategies to ensure to a further extent that the train, validation, and test set are disjoint and independent. Further research should be conducted to establish the relation between the extent of overfitting and the (domain of the) network.

# 4

# Understanding dynamics of truck co-driving networks

In this chapter, we move from the investigation of a generic network science problem towards the transportation domain by investigating the *behavior* of trucks and their drivers using a link prediction approach. Social links may exist between trucks, e.g., because their drivers work for the same company. We call the process where two trucks follow the same route at the same time *co-driving* (Definition 12). It means that the trucks are potentially *socially linked*.

Understanding *truck co-driving behavior* is important because co-driving can have a positive environmental impact. We aim to increase our understanding and will investigate the so-called *co-driving network*, extracted from a spatiotemporal dataset encompassing millions of truck measurements passing eighteen different highway locations in the Netherlands. It leads us to Research question 3, which reads as follows.

**Research question 3:** *How do network structure and vehicle attributes relate to co-driving behavior?*

We explore a *link prediction* approach to understand the (social) processes underlying the co-driving behavior. By investigating the importance of different types of features (e.g., vehicle attributes) provided to the link prediction algorithm, we learn step by step the relation between network structure and co-driving behavior.

The current chapter corresponds to the following publication:
G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding dynamics of truck co-driving networks." In: *Proceedings of the 8th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 882. Springer, 2020, pages 140–151. DOI: 10.1007/978-3-030-36683-4_12

## 4.1 Co-driving network

In the literature, many published studies concerning social network analysis use spatiotemporal data. This often allows enriching the analysis with meaningful insights into social processes. Much of the research performed so far used either GPS [44, 142], WiFi [171] or calls from mobile phones [196] to study social processes. In this study, we will analyze 19 million truck movements.

The goal is to study social phenomena among truck drivers to understand why truck drivers are engaged in so-called *co-driving behavior* with other drivers. In simple terms, co-driving is the activity where two trucks drive together, i.e., are frequently at the same place simultaneously. Here we assume a direct and natural relation between a truck and its driver, meaning that a truck driver only drives one truck and the same driver always drives the truck. Some strict selection criteria ensure that only *intentional* (or similarly, *systematic*) co-driving activity is investigated (see also Definition 12). The criteria are explained in Subsection 4.4.3.

Co-driving behavior is known to have a potentially positive impact on the environment through optimizing logistics and consequently reducing fuel use [188]. Hence, an improved understanding of co-driving behavior may stimulate co-driving behavior. Moreover, innovative forms of transportation, such as autonomous driving, may have significant implications for this behavior.

We construct a so-called *co-driving network* from the data at our disposal. The nodes of the network are trucks. A link is made when the two trucks frequently show intentional co-driving behavior. Other related work on similar data will focus on communities and static properties of the co-driving network, see Chapter 5 and [30].

This chapter aims to learn the relation between *the structure of the co-driving network* and *vehicle characteristics*. To this end, we use a link prediction approach [96]. More precisely, we develop a machine learning classifier that predicts whether two nodes that are so far unconnected, do connect. We then use a future snapshot of the network to check whether the pair of nodes did connect. Subsequently, we investigate the feature importance of each type that occurs in the link prediction classifier. The measure of importance allows us to understand what is assessed as important by the classifier, and thus what aspects are contributing to co-driving behavior. The features used can be categorized into four different types of features.

1. *Neighborhood* features relate to the local embedding in the co-driving network.
2. *Node* features relate to static meta-information of trucks.
3. *Path* features describe distance-related properties of truck pairs based on the global structure of the network.
4. *Spatiotemporal* features consider locations and periods.

The overall structure of this chapter coincides with the research methodology (see Section 1.7). We start with the introduction of the co-driving network in Section 4.1. In

Section 4.2, relevant work is provided on analyzing dynamics in social networks, including spatiotemporal data. Section 4.3 describes the spatiotemporal truck data. Section 4.4 reports how a co-driving network is constructed from the data. In this section, we also discuss the characteristics of the obtained network. Section 4.5 provides our research methodology for the experiments at hand, i.e., a formal description of the link prediction approach. It also explains how the different features are constructed from both the data and the obtained network. Section 4.6 outlines the experimental setup, demonstrates the performance of the link prediction approach, and assesses the importance of the features. Finally, in Section 4.7 we arrive at the conclusions and suggestions for future work.

## 4.2 Relevant related work on dynamics in networks

From the substantial body of related work on spatiotemporal data, we have selected three approaches frequently used to study dynamics in networks at the level of individual nodes. These three different approaches have in common that they all try to understand the underlying social network by studying node attributes available in the data.

First, Sekara *et al.* use sensors to measure proximity of students [175]. The authors show that when high-resolution data is available (both in time and location), groups of interacting nodes can be observed instantaneously. Hence, making sense of individual node attributes using network measures can be performed directly. For example, the authors show that the students may explore new locations in groups during the weekend, while the groups tend to be at the exact location.

Secondly, Kossinets and Watts analyze e-mail data gathered from students and employees at a university [100]. Unlike our truck data, e-mail data does not contain spatial information. In contrast and as an addition, this work collects and analyzes different node attributes such as professional status, gender, and age.

Finally, Wang *et al.* analyze the mobility patterns by tracking the mobility and interactions of millions of mobile phone users [196]. A social network is constructed from phone calls, where users are connected when they communicate. Three contributions from this literature are mentioned below.

1. The authors have established that spatial trajectories of two users strongly correlate when they are close in the social network.
2. Mobility features have a high predictive power concerning which nodes will connect; the prediction power is comparable to the power of network proximity features.
3. Link prediction performance can be improved by exploiting network proximity and mobility features.

Here, we remark that we have used a similar link prediction approach in our work. In addition, we have adapted findings from other related works [100, 175] by (1) distinguishing between weekends and weekdays and (2) using both network and static attributes.

## 4.3 Truck mobility data

Data collection of truck mobility data occurred at eighteen different locations throughout the Netherlands between 2016 and 2018. Every truck passing these locations is registered using an ANPR system. The data is obtained by the same systems as used in Chapter 5. At some locations, the registration systems faced an unexpected downtime. Only registrations from six out of eighteen systems have been considered to ensure a sufficiently valid range of data. These systems are located near the port of Rotterdam. Furthermore, registrations with low-quality data have been removed, such as (1) invalid characters in license plates and (2) non-existing countries.

We remark that the aforementioned quality selections have reduced the total number of registrations from 18,678,420 to 9,202,764. The monthly variation in truck registrations is provided in Figure 4.1, where we show for each of the 25 months (from January 2016 to February 2018) how many trucks are registered. We remark that the number of registrations after applying the quality selections is more stable over time. In Figure 4.2 the histogram of the number of registrations per truck is shown (note that both axes have logarithmic scales). For example, we see that about 1 million trucks are registered only once. More importantly, we see that the distribution of the number of registrations per truck remains similar after data selection.

## 4.4 The co-driving network

In Subsection 4.4.1, we start with three relevant concepts and two criteria to arrive at a procedure to obtain intentional truck co-driving events. Then we describe how the co-driving network is constructed from these events in Subsection 4.4.3. Subsection 4.4.4 continues with statistics of the acquired network to compare these to other social networks.

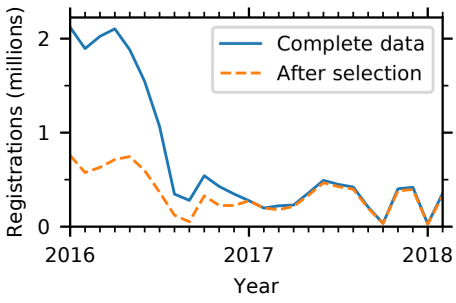



Figure 4.1: Monthly variations in truck registrations.

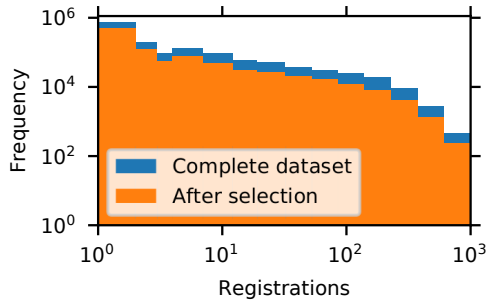



Figure 4.2: Histogram of number of registrations per truck.

### 4.4.1 Procedure to obtain intentional co-driving events

We will now provide the procedure to obtain intentional co-driving events (see Definition 12) with the help of three relevant concepts: (1) dataset of all registrations, (2) a co-driving event, and (3) an intentional co-driving event.

**Concept 1.** Our *dataset of all registrations* (as mentioned in Section 4.3) is denoted by $\mathcal{D}$. We use $\mathcal{D}_u$ to refer to all registrations $x_i$ in dataset $\mathcal{D}$ from truck $u$ with license plate $lp_i = u$. More formally, $\mathcal{D}_u = \{x_i \in \mathcal{D} : lp_i = u\}$.

**Concept 2.** A *co-driving event* $(u, v, t_i)$ happens when two registrations $x_i \in \mathcal{D}_u$ and $x_j \in \mathcal{D}_v$ from trucks $u$ and $v$ exist with the same location $loc_i = loc_j$ at time $t_i$ provided that they have at most $\Delta t = t_j - t_i$ (with $t_j < t_i$) seconds between them (see Subsection 4.5.2).

**Concept 3.** A co-driving event may occur *randomly* or *intentionally*. The following two criteria ensure that only intentional co-driving events are studied.

**Criterion 1: Sufficient small time interval.** The two registrations $x_i \in \mathcal{D}_u$ and $x_j \in \mathcal{D}_v$ from trucks $u$ and $v$ should exist with at most $\Delta t \leq \Delta t_{\text{max}}$ seconds apart. (Seconds will be further in this thesis be abbreviated by $\mathrm{s}$.) In Subsection 4.4.2, we will briefly discuss why we set the $\Delta t_{\text{max}}$ parameter to $8\,\mathrm{s}$. It ensures that trucks should be sufficiently close to each other when intentionally co-driving by setting a maximal time interval between two co-driving trucks.

**Criterion 2: At least two separate co-drive events.** To prevent a random co-driving event is marked as an intentional co-driving event, we require at least two separate co-driving events between trucks $u$ and $v$. Moreover, these two separate events should occur with at least two hours difference, i.e., two co-driving events exist, $(u, v, t_i)$ and $(u, v, t_j)$, for which holds that $|t_i - t_j| \geq 2\,\mathrm{h}$. With the latter requirement, we ensure that the two co-driving events originate from different truck journeys (we assume that in $2\,\mathrm{h}$, trucks are either outside the Netherlands or driving on the next journey).

### 4.4.2 Determining maximal time interval between co-driving trucks

In Criterion 1 above, we introduced the $\Delta t_{\text{max}}$ parameter, determining the maximal time interval between two co-driving trucks. We mentioned that $\Delta t_{\text{max}} = 8\,\mathrm{s}$ is deemed appropriate. We will now explain why.

There is a trade-off. High values will select a large share of random co-driving events, while low values will omit intentional co-driving behavior. We present three considerations when determining the value of $\Delta t_{\text{max}}$.

**Consideration 1.** Figure 4.3 shows the distribution of the time gap between two co-driving events. On the horizontal axis, we see the time gaps in whole seconds; the vertical axis denotes the relative frequency of that time gap (altogether the frequencies add up to 1). We note that distinct behavior is shown for random (yellow) and

intentional (blue) co-driving events. Intentionally co-driving trucks drive closer together than randomly co-driving trucks. We further note that the time gap in intentional co-driving trucks peaks at around $\Delta t = 2\,\mathrm{s}$ and is close to the $\Delta t = 1.3\,\mathrm{s}$, which is considered a minimum safe driving gap between two trucks [116]. After $\Delta t = 8\,\mathrm{s}$ the relative frequency of intentional co-driving trucks becomes similar to that of randomly co-driving trucks. This may indicate that only random co-driving events are selected as intentional co-driving from this value onward.

**Consideration 2.** Figure 4.4 shows the distribution of the number of trucks driving between two trucks involved in intentional co-driving for various values ($\Delta t_{\max} = 4,\ 8,\ 16$ and $32\,\mathrm{s}$). The horizontal axis denotes the number of trucks, and the vertical axis the cumulative relative frequency of that number of trucks driving between the co-driving pair. For values between $\Delta t = 4$ and $8\,\mathrm{s}$, we observe that virtually all trucks drive with at most one truck between them. Higher values result in a non-negligible probability that more than two trucks are driving between the two co-driving trucks. It is unlikely that trucks are intentionally co-driving when more than two trucks drive between these trucks because it is harder to coordinate routing. This is the case for values of $\Delta t_{\max} = 16\,\mathrm{s}$.

**Consideration 3.** We rationalize that *intentionally following* a truck is only possible when a maximum of a couple hundred meters between the two trucks exists. Provided that trucks in our data drive typically at a speed of around $20\,\mathrm{m\,s^{-1}}$, reasonable values for $\Delta t_{\max}$ should be at most $20\,\mathrm{s}$ to $30\,\mathrm{s}$.

The considerations above have led us to properly select intentional co-driving behavior for further analysis in this chapter.

### 4.4.3 Network construction

After applying the two criteria to select intentional co-driving events, the temporal network $G = (V, E)$ is constructed. In this network, the nodes are the trucks $u, v \in V$ that frequently show intentional co-driving behavior (have at least one edge). The links of

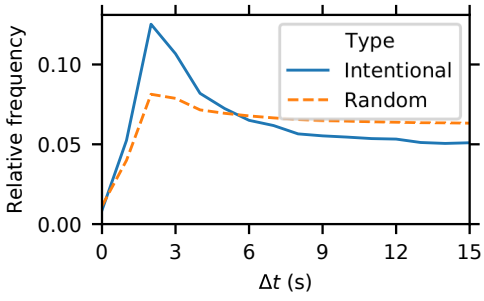



Figure 4.3: Frequency distribution of $\Delta t$ for both intentional and random co-driving.

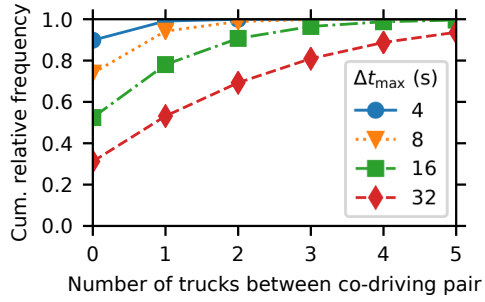



Figure 4.4: Number of trucks driving between a pair of co-driving trucks.

this network consist of the obtained co-driving events $(u, v, t_i) \in E$ between those trucks. We note that multiple links $(u, v, t_i)$ exist between two nodes $u$ and $v$ with different $t_i$ due to Criterion 2 (see Subsection 4.4.1) to select only intentional co-driving. We refer to the number of links between $u$ and $v$ as $w_{u,v}$, with $w_{u,v} \geq 2$ as a result of the two criteria discussed above. When no links exist between $u$ and $v$, the weight $w_{u,v}$ equals $0$.

### 4.4.4 Network statistics

In Table 4.1, we summarize nine statistical properties calculated from our obtained network. All these statistics are explained in Section 1.2. The degree distribution of *each* node is shown in Figure 4.5a. We show the node strength distribution in Figure 4.5b. The vertical axis denotes the frequency of the (a) number of neighbors (degree) and (b) *node strength* of all nodes in the truck co-driving network. The node strength of a node is equal to the sum of the weights of the nodes connected to that node.

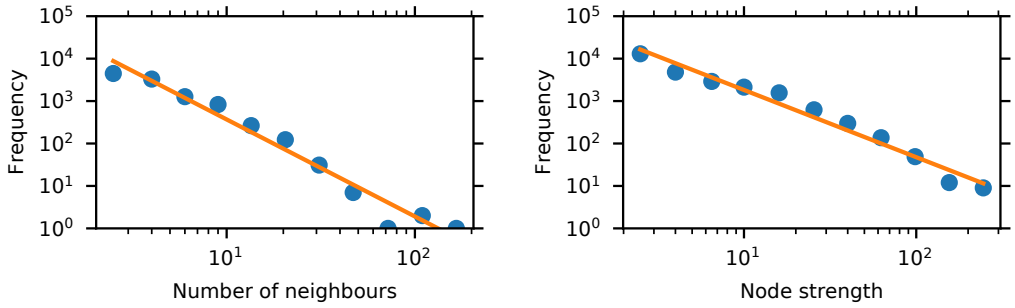
Our network is remarkably similar to other (social) networks. We find the following common properties [10, 12, 197] (see Section 1.2).

- A *Giant Component* is present that spans most nodes and edges (cf. item 4 in Subsection 1.2.4).
- *Sparseness* of edges, with only 0.2‰ of possible pairs of nodes being connected (cf. item 5 in Subsection 1.2.4).
- *Power-law behavior* in both the degree and weight distribution as seen in Figures 4.5a and 4.5b (cf. item 7 in Subsection 1.2.4).
- A relatively *low average path length* (cf. item 6 in Subsection 1.2.4).

Because our network is remarkably similar to other networks, we may conclude that the network construction is successful. In Sections 4.5 and 4.6, we will search for complex relationships between truck drivers that can be understood by investigating the obtained truck co-driving network.

Table 4.1: Nine statistical properties of the co-driving cargo truck network.

| Property | Value |
|---|---|
| Number of nodes | $25,553$ |
| Number of links | $73,059$ |
| Number of connected node pairs | $27,986$ |
| Fraction nodes in Giant Component | $62\%$ |
| Fraction links in Giant Component | $79\%$ |
| Density | $2.2 \times 10^{-4}$ |
| Power law exponent $\gamma$ | $3.3$ |
| Average shortest path length | $7.8$ |
| Diameter | $24$ |




(a) Degree distribution.

(b) Strength distribution.

Figure 4.5: (a) Degree and (b) strength distribution of co-driving cargo truck network. (Note the logarithmic axes.)

## 4.5 Chapter research methodology

This section presents the methodology used in this chapter for the analysis of the dynamics of the co-driving network. We start with a description of the proposed link prediction approach in Subsection 4.5.1. The features are provided in Subsection 4.5.3. In Subsection 4.5.4 we discuss the setup of the classifier. Finally, we provide the measures taken to reduce the observed class imbalance in Subsection 4.5.4.

### 4.5.1 Link prediction

We start by describing link prediction (see also Definition 10). We tailor similar notations used in the Chapters 2 and 3 to the problem at hand.

The link prediction problem is as follows. Given a network observed at a time interval $[t_a, t_b]$ (with $t_a < t_b$), the link prediction classifier needs to predict newly formed links in the network at an evolved time interval $[t_b, t_c]$ (with $t_b < t_c$). In doing so, the classifier can use present information to predict future links. The input of this classifier is a feature matrix $X$, which is based on a network $G_{[t_a,t_b]} = (V_{[t_a,t_b]}, E_{[t_a,t_b]})$ with $E_{[t_a,t_b]} = \{(u, v, t_i) \in E : t_a \leq t_i \leq t_b\}$ and $V_{[t_a,t_b]}$ the nodes taking part in these edges. The feature vector is calculated for each candidate node pair that is not linked (yet) in $G_{[t_a,t_b]}$: $X_{[t_a,t_b]} = \left(V_{[t_a,t_b]} \times V_{[t_a,t_b]}\right) \setminus E_{[t_a,t_b]}$. To ensure that all features are well-defined, we consider only pairs of nodes where both nodes are in the GC of $G_{[t_a,t_b]}$. The **target** of the classifier, $y$, denotes for a node pair whether a link is present in the evolved network:

$$y_{u,v} = \begin{cases} 0 \text{ if } (u, v, t_i) \notin E \\ 1 \text{ if } (u, v, t_i) \in E \end{cases} \quad \text{for some } t_b < t_i < t_c$$

We note that only the link formation is to be predicted; we do not aim to predict the weight of the link. Accordingly, the prediction can be seen as a *supervised binary classification*.

### 4.5.2 Features

Below, we explain the composition of the feature vector used for each candidate truck pair $(a, b)$. In Table 4.2, we present *all* 52 features used by the link prediction classifier. The various truck properties will be explained in Subsection 4.5.2A and the spatiotemporal information in Subsection 4.5.2B. All features used can be categorized into four types. We describe each of them in more detail below.

- **Neighborhood features.** These consider relevant operations related to the ego-network (see Section 1.2) properties of the nodes of the candidate pair. The neighborhood of a node is defined by $N(a) = \{v \in V : (a, v, t_i) \in E \text{ for some } t_i\}$. The strength of a node is the summed weight of every link connected to a node, $s_a = \sum_{u \in V} w_{a,u}$.
- **Node features.** These are constructed from information available about the trucks, see Subsection 4.5.2A.
- **Path features.** These relate to the macro-scale properties of the network (Subsection 1.2.4). We consider only the shortest path length in this chapter.
- **Spatiotemporal features.** These relate to the spatial and temporal behavior of the trucks, see Subsection 4.5.2B.

#### 4.5.2A Node features

The ANPR system determines the license plate and country ($country_u$) of each truck $u$ passing by the system. We use $\mathcal{D}_u$ to denote all registrations $x_i$ available of truck $u$ (as explained in Subsection 4.4.1). The registration systems are also equipped with sensors to measure the length ($length_i$), mass ($mass_i$), and the number of vehicle axes ($axes_i$) of each truck. These measurements may slightly differ between registrations. Therefore, we calculate the averages shown in Table 4.3 for each truck in the network.

The $driving\_hours$ and $weekend\_driver$ features are calculated because they are known to vary between trucks operating in different industrial sectors. The actual driving hour $t_i$ (h) is subtracted by $12\,\mathrm{h}$ and the absolute value is taken, such that it is a measure whether a truck $u$ drives at day (resulting in low values for $driving\_hours_u$, or night resulting in high values for $driving\_hours_u$).

#### 4.5.2B Spatiotemporal features

The spatial-temporal features aim to capture the truck pair's spatial and temporal behavior under consideration. We do so by counting the number of registrations in different periods. We consider periods of one week, one month, and one year. These periods are

Table 4.2: The features (of truck pair $a$ and $b$) of the link prediction classifier and their importance. (The importance of each feature is calculated using the Gini importance, see Subsection 4.5.3.)

| Index | Feature | Type | Feature importance |
|---|---|---|---|
| $X_1$ | $truck\_country\,(a) = truck\_country\,(b)$ | node | 0.005 |
| $X_2$ | $truck\_axes\,(a) + truck\_axes\,(b)$ | node | 0.006 |
| $X_3$ | $\lvert truck\_axes\,(a) - truck\_axes\,(b)\rvert$ | node | 0.008 |
| $X_4$ | $truck\_length\,(a) + truck\_length\,(b)$ | node | 0.017 |
| $X_5$ | $\lvert truck\_length\,(a) - truck\_length\,(b)\rvert$ | node | 0.040 |
| $X_6$ | $truck\_mass\,(a) + truck\_mass\,(b)$ | node | 0.016 |
| $X_7$ | $\lvert truck\_mass\,(a) - truck\_mass\,(b)\rvert$ | node | 0.030 |
| $X_8$ | $driving\_hours\,(a) + driving\_hours\,(b)$ | node | 0.016 |
| $X_9$ | $\lvert driving\_hours\,(a) - driving\_hours\,(b)\rvert$ | node | 0.030 |
| $X_{10}$ | $weekend\_driver\,(a) + weekend\_driver\,(b)$ | node | 0.014 |
| $X_{11}$ | $\lvert weekend\_driver\,(a) - weekend\_driver\,(b)\rvert$ | node | 0.019 |
| $X_{12}$–$X_{19}$ | $last\_week_\ell\,(a+b)$ for $\ell = 1, ..., 8$ | spatio-temporal | $0 - 0.027$ |
| $X_{20}$–$X_{27}$ | $last\_month_\ell\,(a+b)$ for $\ell = 1, ..., 8$ | spatio-temporal | $0 - 0.057$ |
| $X_{28}$–$X_{45}$ | $last\_year_\ell\,(a+b)$ for $\ell = 1, ..., 8$ | spatio-temporal | $0.010 - 0.060$ |
| $X_{46}$ | $\lvert N\,(a)\rvert + \lvert N\,(b)\rvert$ | neighborhood | 0.117 |
| $X_{47}$ | $\Big\lvert \lvert N\,(a)\rvert - \lvert N\,(b)\rvert \Big\rvert$ | neighborhood | 0.013 |
| $X_{48}$ | $\lvert N\,(a) \cup N\,(b)\rvert$ | neighborhood | 0.093 |
| $X_{49}$ | $\lvert N\,(a) \cap N\,(b)\rvert$ | neighborhood | 0.021 |
| $X_{50}$ | $s_a + s_b$ | neighborhood | 0.056 |
| $X_{51}$ | $\lvert s_a - s_b\rvert$ | neighborhood | 0.017 |
| $X_{52}$ | shortest path length in $G$ | path | 0.111 |

Table 4.3: Overview of available truck information.

| Property | | Description | Type |
|---|---|---|---|
| $truck\_country_u$ | | country of registration | string |
| $truck\_axes_u$ | $\underset{x_i \in \mathcal{D}_u}{\text{Median}}\ axes_i$ | number of axes | $\mathbb{Z}$ |
| $truck\_length_u$ | $\underset{x_i \in \mathcal{D}_u}{\text{Median}}\ length_i$ | length | $\mathbb{R}$ |
| $truck\_mass_u$ | $\underset{x_i \in \mathcal{D}_u}{\text{Median}}\ mass_i$ | mass | $\mathbb{R}$ |
| $driving\_hours_u$ | $\underset{x_i \in \mathcal{D}_u}{\text{Mean}}\ \lvert t_i(\text{h}) - 12\text{h}\rvert$ | usual driving hours | $[0, 12]$ |
| $weekend\_driver_u$ | $\underset{x_i \in \mathcal{D}_u}{\text{Mean}} \begin{cases} 0 \text{ if } t_i = \text{weekday} \\ 1 \text{ if } t_i = \text{weekend} \end{cases}$ | fraction driving in weekend | $[0, 1]$ |

chosen to cover a broad window of possible relevant periods. As an example, for feature $last\_day_\ell\,(a+b)$ registrations are counted for trucks $a$ and $b$ at location $\ell$ at the last day before the considered time.

### 4.5.3 Classifier

A *random forest* classifier is used to do link prediction. We choose this classifier because random forests are known to generalize well on unseen data. Our task is to determine the importance of each feature [42, 76].

We now discuss the setup of the classifier. The random forest classifier we used contains 128 decision trees. Larger values usually bring no significant performance gain [140]. Each decision tree is trained on a randomly drawn selection of variables. The number of randomly drawn features equals the square root of the total number of variables, a typical value used in classification [162].

Random sampling with replacement from the data *increases randomness* for each decision tree. The splitting criteria of the nodes are determined by considering the Gini impurity reduction as discussed in [76]. The random forest classifier allows obtaining the *feature importance* by determining the Gini impurity reduction for splitting nodes with a certain feature [76]. We recall that the feature importance is essential, as it enables us to understand the network dynamics by predicting new truck co-driving behavior.

Subsequently, we use the out-of-bag sample of each tree to estimate the performance of the random forest [76, 162]. We then assess the optimal value for the depth of the decision trees in the random forest. The classifier's performance is calculated on the test set, which is a 10% random sample of the data only used for this purpose.

### 4.5.4 Class imbalance

It is well-known that real-world network link prediction classifiers come with a large class imbalance [196], caused by sparseness of edges (see Subsection 4.4.4). The performance of the random forest classifier may drop if there is a large class imbalance. To overcome this limitation, we use the following two measures.

1. We adjusted the weights of the positive instances so that the total weight of the positive and negative samples are equal.
2. We consider only truck pairs where both trucks are involved in co-driving events in the last two months before time $\tau$. It will reduce the number of considered truck pairs. The class imbalance is also reduced because many truck pairs registered recently have a higher probability of co-driving.

## 4.6 Experimental setup and results

The setup of the experimental parameters are briefly discussed in Subsection 4.6.1. The results of the link prediction classifier are discussed in Subsection 4.6.2.

### 4.6.1 Experimental parameter setup

We set the value of $\tau$ such that half of the edges are formed. We experimentally found that with this value of $\tau$, the class imbalance is reduced while ensuring that at least a thousand truck pairs are present that will link. The class imbalance is $1 : 61,000$, meaning there is one positive instance for every 61,000 negative instances. Taking the two measures noted in Subsection 4.5.4 reduces the class imbalance to $1 : 15,000$, which improves link prediction performance. Nevertheless, even with this parameter setup, it is still a highly imbalanced set of instances.

Furthermore, we found an optimal maximum depth of three for the decision trees in the random forest using out-of-bag sampling (see Subsection 4.5.3).

For reproducibility purposes, we mention that the random forest is used as implemented in Python sci-kit learn 0.21.2 [149].

### 4.6.2 Results

We report the trade-off between true and false positives to assess the classifier's accuracy. The relation between these two values is shown in Figure 4.6 using the well-known Receiver Operating Characteristic (ROC) curve [140]. The AUC is $0.84$, meaning the classifier can accurately predict whether links will appear. The performance is sufficiently high, and therefore, we continue with the analysis of the feature importance observed.

In Table 4.2, the feature importance is presented. The features are shown for each of the four types (neighborhood, node, path, and spatiotemporal features) in Figure 4.7.

We observe that the neighborhood feature ($X_{46}$) scores highest with a feature importance of $0.117$, closely followed by the single path feature ($X_{52}$) with an importance of $0.111$. The two neighborhood features with the highest scores are $X_{46}$ and $X_{48}$, with an importance of $0.117$ and $0.093$, respectively. These features provide the sum of the node pairs' degrees and the union of their neighborhoods, respectively. Both the spatiotemporal and node features score lower, with a maximum feature importance of only $0.060$ and $0.040$, respectively.

Since the features based on network metrics (neighborhood and path) have higher feature importance, we may conclude from our experiments that the network view (i.e., the structure of the data in the network) on the data is helpful.

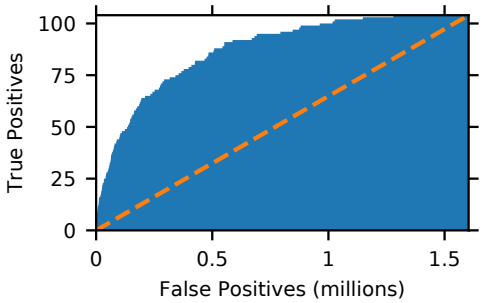



Figure 4.6: The ROC curve of the random forest link prediction classifier.

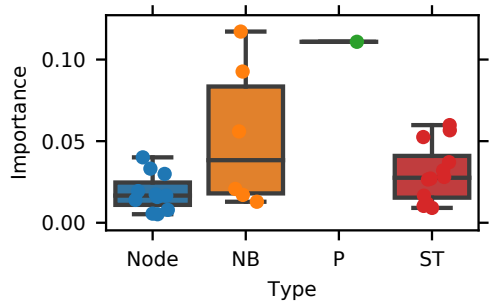



Figure 4.7: The Gini feature importance of the various feature sets. (NB, P, and ST are the neighborhood, path, and spatiotemporal features, respectively.)

## 4.7 Chapter conclusions and outlook

In this chapter, we addressed Research question 3: "How do network structure and vehicle attributes relate to co-driving behavior?" We compared four sets of features in a link prediction model applied to the co-driving network. By comparing the importance of the different types of features, we observe different abilities in predicting new links. From our experiments, we may conclude that features based on network measures, particularly the neighborhood feature and path feature to a lesser extent, can explicate the dynamics of the studied co-driving network. This means that the network perspective we have adopted in analyzing the spatiotemporal dataset of truck co-driving in the Netherlands has seriously contributed to our comprehension of co-driving behavior. Our second conclusion is that the link prediction approach is a viable method for analyzing spatiotemporal datasets that contain social behavior. Our answer to Research question 3 reads: "The network structure, and especially the ego-network structure of the nodes, relate strongly to co-driving behavior. The same is the case for spatiotemporal information about the truck itineraries. Vehicle attributes show a smaller relation to co-driving behavior."

### Chapter outlook

An exciting angle for future work is to use a similar approach to predict which nodes will turn inactive, i.e., will not form any new links. It will result in a substantially smaller set of candidate nodes for the link prediction algorithm. Finally, future work could focus on interpreting and applying the knowledge gained to actually stimulate co-driving behavior, which may in turn facilitate reductions in the fuel use of trucks.

# 5

# Understanding behavioral patterns in truck co-driving networks

This chapter consists of two distinct research steps. The first step explores methods for *detecting communities* within truck co-driving networks. The second step investigates methods for understanding the relations of these communities with assortativity (cf. Definition 8). These steps allow us to better understand the behavioral patterns in truck co-driving networks. Understanding how to stimulate co-driving in turn may help to reduce traffic congestion and optimize fuel usage as a result of reduced aerodynamic drag.

The driving force behind edges in the truck co-driving network is analyzed in terms of assortativity. Moreover, we aim to understand the community structure of the truck co-driving network. We propose a novel metric, *the average maximal community assortativity metric*, to arrive at an understanding of the network community structure through assortativity.

The current chapter builds on the insights gained in the previous chapter, where we focused on assessing the evolution of co-driving networks. In this chapter, we address Research question 4, which reads as follows.

**Research question 4:** *How can node attribute information be exploited to automatically create a good partitioning of a co-driving network into communities?*

The current chapter corresponds to the following publication:
G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding behavioral patterns in truck co-driving networks." In: *Proceedings of the 7th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 813. Springer, 2018, pages 223–235. DOI: 10.1007/978-3-030-05414-4_18

## 5.1 Truck co-driving network

In this chapter, we use network approaches to investigate what attributes lead to a group of truck drivers showing co-driving behavior. To do so, we use (1) network community detection [58] as well as (2) various metrics related to assortativity (also known as mixing patterns, see [126]).

We analyze a *unique* dataset gathered over one year, detailing the presence of at least two million trucks in the Netherlands (see Subsection 5.3.1 for a description of the data). We investigate the spatiotemporal data as a so-called *co-driving network*, wherein the nodes represent trucks (cf. Chapter 4). Trucks that are *co-driving* are observed at the same location within a very short time window. Those pairs of co-driving trucks that occur a certain number of times (e.g., more than once) are defined as *systematic co-driving trucks* (see Definition 12). In the co-driving network, the edges represent this systematic co-driving behavior. We will explain the construction of this network in Section 5.3.

The results of this work contribute to topics related to understanding human behavior, autonomous driving, and environmental sustainability. Using network metrics, we aim to derive *what attributes* may influence the decision of truck drivers to drive together systematically. The findings can be helpful for research on innovative forms of transportation, such as autonomous driving. We mention two possible benefits: (1) co-driving trucks can save up to 15% on fuel due to reduced aerodynamic drag [188] and (2) co-driving trucks reduce traffic congestion. It highlights the potential environmental implications of understanding co-driving behavior.

The co-driving network turned out to have at least three properties that are often encountered in real-world networks. First, the network has a significant *Giant Component* (GC, see item 4 in Subsection 1.2.4), which contains 37,858 nodes (trucks) and the majority of the co-driving links of the network. Second, the *average shortest path length* (cf. item 6 in Subsection 1.2.4) in the network is around nine edges, which, given a large number of nodes, is relatively tiny and hints at a small-world-like structure [119]. Third, our co-driving network is *scale-free* (cf. item 7 in Subsection 1.2.4), i.e., the degree distribution follows a power law [10]. We also investigate to what extent the network has a highly modular structure (cf. Subsection 1.2.3), meaning that a clear partitioning into communities exists.

As we will note in Subsection 5.3.1, we have access to additional node attributes (see Subsection 1.2.1). It allows us to study assortativity (Definition 8), which (1) enables insights into what attributes contribute to the network structure and (2) more importantly, explains co-driving behavior. Subsequently, we will use the node attributes to comprehend the communities better. With this knowledge, we aim to understand how local groups of co-driving trucks emerge and contribute to the global network structure. Furthermore, the proposed approach for understanding community detection results using assortativity

is broadly applicable in other networks, providing a methodological contribution to the field.

The remainder of this chapter is organized as follows. After discussing related work in Section 5.2, Section 5.3 explains how the network was constructed from the raw data. Section 5.4 is concerned with the proposed approach and techniques to understand the network structure. Then, Section 5.5 provides details on the results obtained. Conclusions and suggestions for future work are provided in Section 5.6.

## 5.2 Related work on understanding behavioral patterns from networks

We start with an important contribution by Barrat and Cattuto [14], in which face-to-face contacts were recorded for twenty-second intervals using measurement infrastructure at several social settings. One of the results was that aggregated network topology and temporal behavioral properties are strongly related.

Second, Barrat and Cattuto showed that community detection could make a sensible partitioning of the network that was explainable by node attributes. Our study employs a similar approach, where the network topology and community structure are explained by the properties of the individual nodes and their assortative linking patterns.

Third, in a more recent study, Kassarnig *et al.* [95] handed over a thousand phones to students who agreed to have their communication and spatiotemporal activities traced. The work showed that network metrics (such as academic performance of peers, centrality, and the fraction of low and high-performing peers) are more informative indicators of university performance than node attributes indicating an individual's characteristics such as personality, class attendance, and the Facebook activity level. It underpins the value of network metrics compared to classical data aggregates.

Fourth, research by da Cunha and Gonçalves [43] on the Brazilian Federal Police criminal intelligence network used network science techniques to uncover behavioral patterns amongst criminals. Similar to our data, their network also featured a significant Giant Component (GC) and a degree distribution that follows a power law. Their observed low density and high average shortest path length were explained as "no trust among thieves". Additionally, Cunha and Gonçalves showed that their GC had a highly modular structure, which was explained by the necessity of (1) being *efficient in running criminal activities* within the group while (2) at the same time also being *obscure to the outside world*.

Throughout this chapter, we will employ *community detection* and *node attributes* in a way comparable to those in the works mentioned above, aiming to extract behavioral insights. To the best of our knowledge, the work of this chapter is the first to investigate the phenomenon of truck co-driving using network science methods and techniques.

## 5.3 Network construction

This section explains the network construction. We start with the characteristics of the data in Subsection 5.3.1. In Subsection 5.3.2, we describe how we selected systematic co-driving events. We continue with the co-driving network and its node attributes in Subsection 5.3.3. Subsection 5.3.4 reports two validation metrics to confirm that we selected the right value of a parameter. Finally, Subsection 5.3.5 details a regional co-driving network and its additional node attributes.

### 5.3.1 Truck observation data

The data is obtained from an ANPR system[1]. The Dutch Infrastructure and Water Management Ministry maintains the system. The data contains over 16,000,000 observations of trucks passing at a measurement system. These systems are situated at evenly distributed locations in the Netherlands. For each observation, the following data was available:

- license plate (serving as a unique identifier);
- location $\ell$ (either one of seventeen highway locations);
- lane $h$, indicating which of the (at most two) lanes the truck was in;
- speed $v$ (in $\mathrm{km\,h^{-1}}$);
- timestamp $t$ at a $10\,\mathrm{ms}$ resolution; and
- *country* (using the ISO-2 country code).

We note that a slightly different dataset was used compared to Chapter 4. In this chapter, we retained observations of all locations and used only data available when performing the calculations in 2018.

We briefly mention two insights from the truck observation data. First, the frequency distribution of how often each distinct truck (identified by its license plate) is measured is given in Figure 5.1a. The horizontal axis denotes the number of measurements per truck and the vertical axis indicates the corresponding probability. The distribution is highly skewed to the lower values, meaning that most trucks are only measured a few times. There appears to be a truncated power law present. Second, the interval distribution between two successive measurements of the same truck at the same location is shown in Figure 5.1b. It demonstrates how most trucks that return have a diurnal pattern, visible from the peaks at multiples of $24\,\mathrm{h}$. Similarly, a weekly pattern is present. This figure indicates that most individual trucks have regular driving patterns.

### 5.3.2 Selection of systematic co-driving events

In the co-driving network, nodes are trucks, and edges represent systematically co-driving trucks Definition 12. We follow the same selection procedure as used in Subsection 4.4.3.

[1] See https://international.fhwa.dot.gov/pubs/pl07028 for details on this system.

We employ the following three criteria to determine which truck pairs are systematically co-driving together.

1. Trucks $a$ and $b$ should be at the same place, i.e., their location is identical, so $\ell_a = \ell_b$ (a *co-occurrence*).
2. Moreover, the *co-driving* trucks should be so in a time window of at most $\Delta t_{\text{max}}$, so $|t_a - t_b| \leq \Delta t_{\text{max}}$.
3. Finally, *systematically co-driving* trucks are those co-driving trucks $(a, b) \in E$ occurring at least $\Theta > 0$ times.

Thus, to derive the co-driving network, we must set parameters $\Delta t_{\text{max}}$ and $\Theta$.

We derive the right parameter setting for $\Delta t_{\text{max}}$ in a data-driven manner below. In Figure 5.2, network characteristics are shown for increasing values of $\Delta t_{\text{max}}$. Definitions of these metrics, all common in the field of network science, can be found in [10]. Recall that a high value for $\Delta t_{\text{max}}$ will increase the probability that a pair of co-occurring trucks is added by chance. Therefore we choose to keep the value relatively low, namely at $\Delta t_{\text{max}} = 8\,\text{s}$. At this value, the density of the resulting network is lowest, while the GC's size compared to the full network (in terms of both nodes and edges) has become stable. Other network metrics, such as the GC's diameter and average shortest path length, also stabilize around this value, as seen in Figure 5.2d.

We expect the probability that two trucks randomly co-drive twice is negligible. Therefore, we identify non-random and, thus, systematic co-driving by setting $\Theta = 2$.

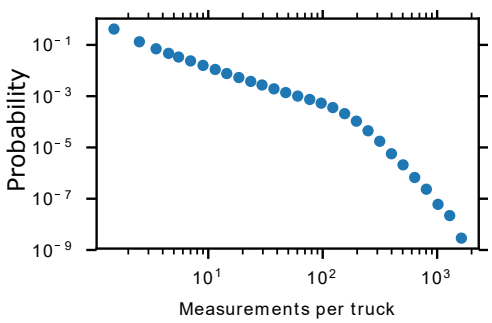
### 5.3.3 Co-driving network and node attributes

The co-driving network is an undirected weighted network $G = (V, E, w)$, where $V$ is the set of all trucks involved in a co-driving activity at least once. For a truck pair $(a, b) \in E$, the weight $w_{a,b}$ indicates the number of times the two trucks drove together. It should be greater than or equal to a certain threshold: $w_{a,b} \geq \Theta$. We furthermore consider four node attributes: (1) $country$, directly derived from the license plate; (2) $\tilde{v}$, the median truck speed; (3) $n_\ell$, the number of different locations where the truck was observed; and (4) $\ell_{\text{max}}$, the location where the truck was most frequently observed.

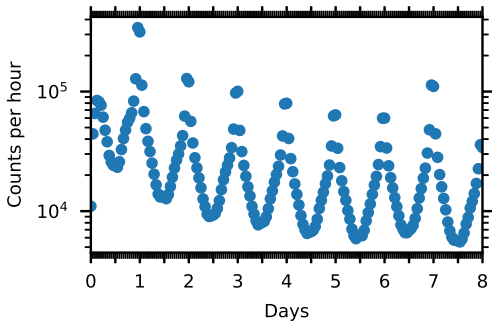
### 5.3.4 Two validation metrics

We validate our choice of $\Delta t_{\text{max}} = 8\,\text{s}$ by assessing whether two metrics from the raw truck measurement data differ when applied on two non-systematically ($w_{a,b} < 2$) co-driving truck pairs and two systematically ($w_{a,b} \geq 2$) co-driving truck pairs.

The first validation metric is $\Delta v$: the speed difference $|v_a - v_b|$ between two co-occurring trucks within $\Delta t_{\text{max}}$. We are inclined to assume that trucks that drive systematically together for longer distances would have a lower value of $\Delta v$ as their speed needs to be aligned. In Figure 5.3a, we observe that this is indeed the case. Here, the

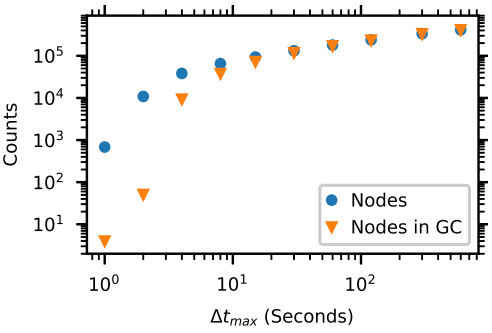



(a) Probability distribution of the number of measurements per truck.

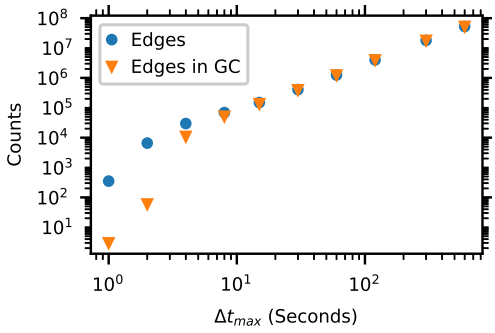



(b) Time interval that truck re-occur.

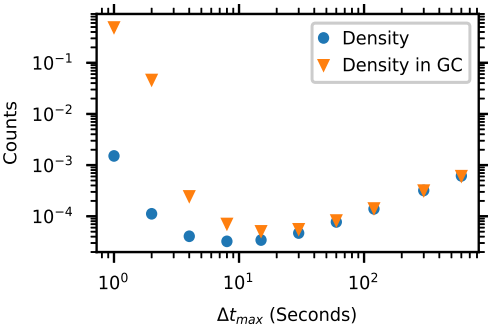
Figure 5.1: Summary statistics of the cargo truck data. (Note the logarithmic axes.)



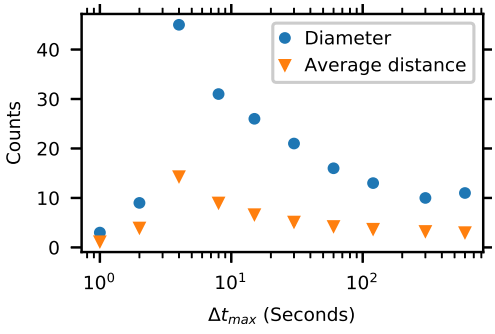

(a) Number of nodes in the network and its GC.




(b) Number of edges in the network and its GC.

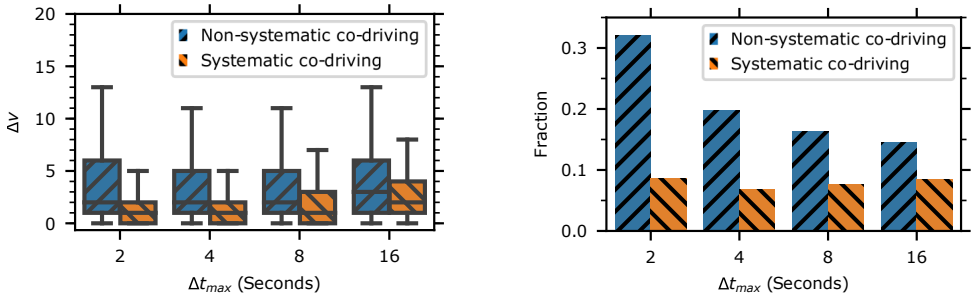



(c) Density of the network and its GC.




(d) Diameter and average shortest path length in the GC.

Figure 5.2: Statistics of the co-driving cargo truck network (for increasing $\Delta t_{\max}$).




(a) Distributions of the difference in speed $\Delta v$ (for increasing values of $\Delta t_{\max}$).

(b) The fraction of trucks driving on different lanes (for increasing values of $\Delta t_{\max}$).

Figure 5.3: Validation metrics for establishing systematic co-driving.

result is most evident for smaller values for $\Delta t_{\max}$, up to $8\,\mathrm{s}$. It hints that we selected the systematic co-driving events in a correct way.

The second validation metric is $h_a = h_b$, which means whether the considered pair of trucks are driving in the same lane. For a truck pair $(a, b)$ driving in the same lane it holds that $h_a = h_b$. In the case of systematic co-driving behavior, it is more likely that two trucks are in the same lane since they do not have to overtake each other to drive together. Figure 5.3b shows that, indeed, the fraction of trucks driving on a different lane ($h_a \neq h_b$) is more than two times higher for non-systematic co-driving than for systematic co-driving trucks. Thus, also this validation metric hints that we correctly selected the systematic co-driving events.

The two validation checks (see Figure 5.3) convince us that the derived co-driving network captures actual systematic co-driving behavior.

### 5.3.5 Regional co-driving network

Although trucks from various countries are observed in our data, we have additional information on Dutch trucks obtained from the Netherlands Vehicle Authority (RDW) (Dutch: RijksDienst voor het Wegverkeer). We use the additional information to construct a major contribution of our research, being a Dutch *regional co-driving network* which consists of trucks for which (1) the country was equal to the Netherlands (NL) (59% of the nodes) and (2) all systematic co-driving links between these trucks, having the following additional node attributes: (1) *city* where the truck is registered; (2) empty mass $m_{empty}$ of the truck; (3) maximum mass $m_{\max}$ of the truck; (4) *capacity* of the truck; (5) *company* that owns the truck; (6) registration date (*regdate*); and (7–10) $zip_{\{1,2,3,4\}}$ the zip code where the vehicle is registered with a higher number marking higher geographic precision. The regional co-driving network, together with the mentioned additional node attributes, are used in our research in (1) reducing traffic congestion and (2) optimizing fuel usage.

## 5.4 Chapter research methodology

Here we describe the techniques used to understand systematic co-driving behavior from a network perspective. We will start by outlining how assortativity can explain the driving forces in edge formation in Subsection 5.4.1, followed by the approach to detect communities within the co-driving network in Subsection 5.4.2.

### 5.4.1 Understanding co-driving behavior by assortativity

We will use assortativity to investigate what type of common node attributes explain the formation of links in the co-driving network. *Assortativity* is a measure of the preference of nodes in a network to connect with other nodes that are alike in some way [129], as explained in Subsection 1.2.1. The assortativity metric $r_a$ can be computed for each network's nominal and numerical node attribute $a$ using the definitions given in [127]. It should be noted that degree assortativity is the assortativity computed for the (numerical node attribute) degree, see Definition 8.

An assortativity value $r_a$ closer to $1$ indicates that nodes have more links to nodes with equal node attribute $a$. A value closer to $-1$ indicates disassortativity, meaning that nodes with different values for a node attribute $a$ are more likely to be connected. An assortativity of $0$ for an node attribute means no preferential attachment of edges between nodes based on the value of a node attribute $a$.

### 5.4.2 Understanding co-driving behavior by community structure

To better understand the co-driving network, we investigate the *community structure*, which can provide insights into the different groups of truck drivers. We use the well-known Louvain algorithm [19] to detect communities. It takes as input the structure of a weighted network and outputs an assignment of each node to a community. It furthermore has a *resolution parameter* $\gamma$ that predicts whether a more fine-grained or coarse-grained partitioning into communities should be found [105].

The Louvain algorithm uses heuristics to optimize the so-called *modularity* value $Q$, indicating the quality of the partitioning of the network into communities. A modularity value close to $1$ indicates that there are more edges *within* communities and fewer edges *between* communities. When adjusting the resolution parameter mentioned above, the value of modularity and the number of discovered communities $C$ change. At different resolutions, $\gamma$, similar values of $Q$ can be measured, each with a different number of communities $C$. This so-called modularity landscape *must* be explored to obtain the partitioning of the network into communities (and corresponding $\gamma$) that best explain the formation of groups in the underlying system [66].

We will propose to use the available node attribute information to explore these solutions automatically. Subsequently, we determine the assortativity for each node

attribute and average that per community. After that, we take the partitioning of the node attribute with the highest assortativity for each community. We take the average over all communities, obtaining the proposed metric of average maximal community assortativity $R = \frac{1}{C} \sum_c \max_a r_a^{G(c)}$.

In this equation, $C$ is the number of communities, $c$ is one of the communities (defined as the subset of nodes in this community), $a$ is a node attribute, $G(c)$ is the subgraph induced on the nodes in the community $c$ and $r_a^{G(c)}$ is the assortativity $a$ in subgraph $G(c)$. Based on the value of $R$ for different network partitions into communities as a result of varying the resolution parameter $\gamma$, we select the partition into communities for which $R$ is highest because that partition allows for the best explanation of the communities observed.

## 5.5 Analysis of co-driving behavior

In Subsection 5.5.1, we start by providing statistics of the co-driving network. The results of applying the two approaches to understanding the formation of links outlined in Section 5.4 are discussed in Subsection 5.5.2 and Subsection 5.5.3.

### 5.5.1 Network statistics

*Network metrics*, of which definitions can be found in Section 1.2, were computed using NetworkX [72], whereas *distance metrics* were computed using teexGraph [185]. The python-louvain package was used for community detection [7].

In Table 5.1, we list (1) basic network statistics for the full network and (2) the regional co-driving network of measured Dutch trucks. We note that the majority of activity is captured in the GC. The degree distribution for both networks (all trucks vs Dutch trucks only) is given in Figure 5.4, showing a power-law distribution, suggesting that the co-driving network is scale-free. It means that a few truck drivers drive with many other trucks, whereas the majority only do so with a relatively small number of others trucks. The weight distribution in Figure 5.4 shows that some co-driving trucks frequently drive together. The diameter of the GC (which is affected by distant outliers) is relatively high, with a value of $31$ and $28$ for the full and regional network, respectively. In contrast, the average shortest path length is higher than $6$, which is common in many real-world networks. However, with a value of $9$, the average shortest path length is still substantially lower than average shortest path lengths encountered in random networks with similar sizes [10]. The power-law exponent of the degree distribution is $3.6$. Together, the three metrics (diameter, average shortest path length, and power-law exponent) indicate that although the network has a very skewed degree distribution, nodes are not as close to each other as in other real-world networks.

Table 5.1: Statistics of the full and regional co-driving cargo truck networks (and their GC).

| Metric | Full network | Regional network |
|---|---|---|
| Number of nodes | $65,290$ | $35,706$ |
| Number of nodes (GC) | $37,858$ | $22,511$ |
| Number of edges | $68,958$ | $36,885$ |
| Number of edges (GC) | $51,730$ | $30,851$ |
| Density | $3.2 \times 10^{-5}$ | $5.8 \times 10^{-5}$ |
| Density (GC) | $7.2 \times 10^{-5}$ | $1.2 \times 10^{-4}$ |
| Diameter (GC) | $31$ | $28$ |
| Average shortest path length (GC) | $9$ | $9$ |
| Clustering coefficient | $0.06$ | $0.07$ |
| Power law exponent | $3.58$ | $3.61$ |

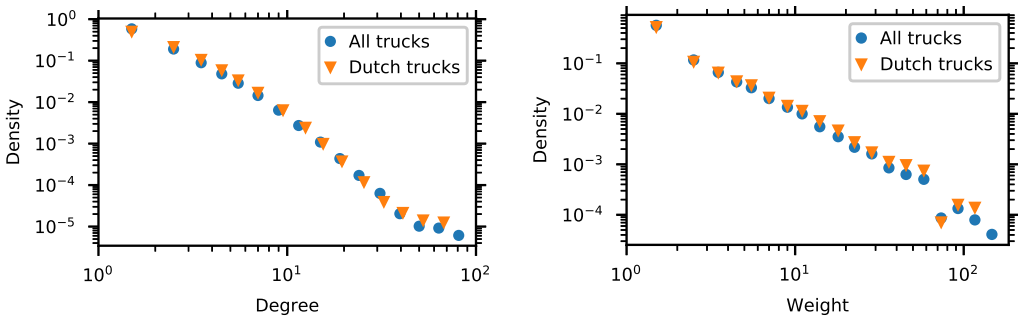



Figure 5.4: Degree (left) and weight (right) distribution of the (full and regional) co-driving cargo truck network.

### 5.5.2 Assortativity

The values reported in Table 5.2 were obtained by using the metric of *assortativity*, which was discussed in Subsection 5.4.1. The results indicate that actively co-driving trucks tend to be connected to other active co-driving trucks, as evidenced by the positive value for *degree assortativity*. The geographical information available about the trucks was found to be the most effective in explaining systematic co-driving behavior. Specifically, the *zip code* node attribute in the regional network showed substantially high assortativity metrics, and the *country attribute* in the full network had a value of $0.56$. These findings suggest that truck drivers from the same city or country are more likely to engage in systematic co-driving.

### 5.5.3 Average maximal community assortativity

The results of applying community detection to the GC of the entire network are shown in Figure 5.5. The number of communities and the modularity value are shown for increasing resolutions. A maximum value of $Q = 0.86$ is found for resolution $\gamma = 1$. This high value is the second evidence that our co-driving network is highly modular. We observe how there are several solutions with a similar modularity value but a very different number of communities.

To better understand these findings, we look at the average maximal community assortativity $R$ (see Subsection 5.4.2) shown in the bottom right of Figure 5.5. Although at $\gamma = 1$ the highest modularity is found, we see that for $\gamma = 2$ (as opposed to lower values of $\gamma$), the best community partitioning is obtained in terms of explainability using assortativity. For this value of the resolution, we find that 52 of the total 120 communities are best described using the *country* attribute. In contrast, the remaining attributes $\tilde{v}$, $n_\ell$, and $\ell_{\text{max}}$ explain 30, 29 and 9 communities respectively.

Table 5.2: Calculated assortativities of the full and regional truck co-driving network.

| Node attribute | Type | Full network | Regional network |
|---|---|---|---|
| degree | numeric | 0.12 | 0.12 |
| *country* | 17 categories | **0.56** | – |
| $\tilde{v}$ | numeric | 0.55 | 0.34 |
| $n_\ell$ | numeric | 0.45 | 0.40 |
| $\ell_{\max}$ | 17 categories | 0.25 | 0.21 |
| *city* | 1,319 categories | – | 0.33 |
| $m_{empty}$ | numeric | – | 0.30 |
| $m_{\max}$ | numeric | – | 0.35 |
| *capacity* | numeric | – | 0.32 |
| *company* | numeric | – | 0.29 |
| *regdate* | numeric | – | 0.13 |
| $zip_4$ | 1,975 categories | – | 0.32 |
| $zip_3$ | 718 categories | – | 0.33 |
| $zip_2$ | 90 categories | – | 0.35 |
| $zip_1$ | 9 categories | – | 0.41 |

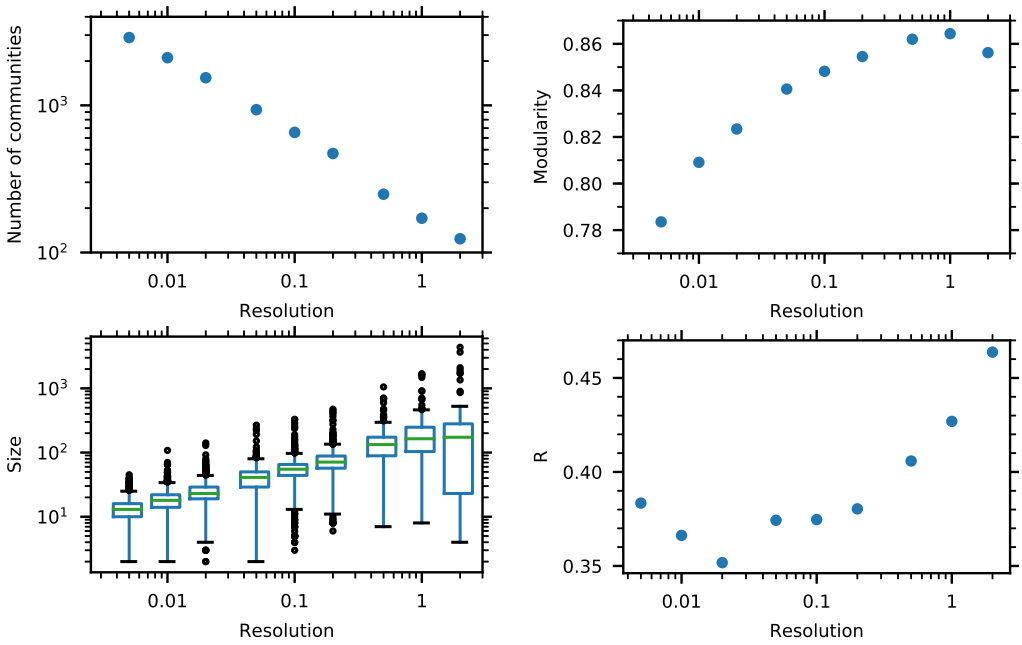



Figure 5.5: Properties of the communities (for various values of the resolution parameter). (Top left: Number of Communities; Top right: Modularity value $Q$; Bottom left: Average community size; Bottom right: Average maximal community assortativity $R$. Note the various logarithmic axes.)

## 5.6 Chapter conclusion

This chapter provides a detailed report on the extraction of truck measurement data for revealing its real-world properties and characteristics. Technically, we focus on a Giant Component, scale-free degree distribution, positive degree assortativity, and a highly modular community structure. The newly developed average maximal community assortativity metric is used to optimize the node attribute information to obtain a good partitioning into communities. Thereby we address Research question 4: "How can node attribute information be exploited to automatically create a good partitioning of a co-driving network into communities?" Our answer is that in the truck co-driving network as designed by us (see Section 5.3), we were able to establish that the highly modular community structure can be explained using different attributes' assortativity in each community, dominated by geographical features.

### Chapter outlook

Additional investigation into the relationship between the observed network characteristics and the domain is on our list of further research. Timestamps will be incorporated to investigate the co-driving network's dynamics, identifying which truck drivers initiate co-driving behavior and the conditions under which the behavior diffuses to other nodes.

Understanding the community structure of the truck co-driving network can lead to interventions to educate drivers on best practices. Moreover, truck drivers can save fuel and reduce traffic congestion by reduced aerodynamic drag when co-driving.

# 6

# Fair automated assessment of noncompliance in cargo ship networks

International cargo ships must follow a plethora of safety standards and international treaties [147]. Governmental inspectorates currently assess a ship's compliance with the help of a rule-based process using the color (white, gray, or black) of a ship's flag as a dominant factor. The flag's color is determined yearly by considering the fraction of noncompliant ships of that flag [145]. The usage of the flag's color can lead to confirmation bias and unfair inspections. Rather than using static ship characteristics, we wish to utilize information about the actions of the ship, i.e., its behavior. This brings us to the following research question.

**Research question 5:** *How can ship behavior be utilized to enable smart inspection of cargo ships?*

We propose an approach for smart inspection (cf. Definition 2), and aim to realize two crucial contributions. First, we would like to reduce confirmation bias by using a fair model. Second, we aim to extract relevant mobility patterns from a cargo ship network (see Definition 13), allowing us to derive meaningful behavioral features for ship classification. Our approach will improve fairness at the cost of a limited performance loss. Thereby, it will enhance *maritime safety and protection* through smarter inspection targeting. In a general sense, this work demonstrates how network science can use behavioral data for smart inspection.

The current chapter corresponds to the following publication:
G. J. de Bruin, A. Pereira Barata, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Fair automated assessment of non-compliance in cargo ship networks." *EPJ Data Science* 11, 13 (2022). DOI: 10.1140/epjds/s13688-022-00326-w

## 6.1 Smart cargo ship inspection

Maritime cargo transport is essential to global trade, often being the most cost-effective way to move goods from one place to another. It results in many ship movements worldwide; around 80% of world merchandise is carried by sea [190]. However, we mentioned in Chapter 1 that maritime transport has risks, such as (1) labor exploitation, (2) culpable ship accidents, and (3) environmental pollution. These risks need to be mitigated by shipowners. Port State Control (PSC) inspections are conducted when ships berth in a port to ensure mutual trust between countries that *all* ships adhere to the same international laws. There are two possible outcomes of an inspection; either the ship is found fully compliant, or there are particular noncompliances. These PSC inspections check for compliance with many regulations, including any deficiency that could lead to one of the aforementioned maritime risks. If severe enough, such deficiencies can lead to detention, meaning that the ship is not allowed to depart the port before the deficiencies are rectified, or to a ban meaning that the ship is not allowed to enter specific ports any longer. In this research, we aim to predict whether a ship will have a deficiency in port state control and thus is potentially noncompliant, which we consider equivalent to a ship posing a high risk.

In recent years, governments have established strict laws to mitigate the negative consequences of maritime transport. Members of the Paris Memorandum of Understanding (MoU)[1] introduced a so-called New Inspection Regime (NIR) [147]. Arguably the most significant innovation in the renewed memorandum is the introduction of a ship risk profile. It awards a score to each ship based on a weighted sum of six factors [147]. The six factors used in the risk profile for a given ship are [56] derived from (1) the type of a ship, (2) the age of a ship, (3) commercially issued safety certificates, (4) owning company's performance, (5) historical misconducts, and (6) the flag a ship is flying, or equivalently, the country of registration. Using the score, each ship is classified as low-risk or high-risk. Ships classified as low-risk should be inspected every three years, while ships classified as having a high-risk profile should be inspected every six months. With the ship risk profile, the NIR allows inspectorates to focus on noncompliant ships. It also leads to efficient use of the inspection capacity and budget, as every unnecessary port state control inspection costs the inspectorates on average around $\$1,000$ [98]. In [210], it was estimated that a noncompliant ship saves, on average, around $\$400,000$ on maintenance by not complying with regulations, whereas the loss of a ship can incur costs up to $\$67,000,000$. Shipowners with a low-risk profile can benefit by reducing inspection burden, saving precious turn-around time in the port.

From the six factors used in the current ship risk profile, the flag plays a vital role [36, 166]. The flag is considered black, gray, or white based on the detention ratio of the

[1]The following countries are part of the Paris MoU: all European Union coastal countries, Canada, Norway, Russia, and the United Kingdom.

country over a three-year rolling period [145]. Fleets from countries on the blacklist were significantly more often detained over a three-year period than fleets from countries on the whitelist. We mention three drawbacks in considering the flag for the ship risk profile.

1. There are ethical concerns. The use of the flag can be considered disparate treatment [57] because ships are intentionally treated differently based on membership of a privileged class, being the white flag.
2. There are opportunities for ships to change flags, opening up the possibility for noncompliant ships to "hide" under a white flag [37]. Although changing flags does not necessarily improve compliance, the NIR would grant such a ship a lower risk profile. In an ideal situation, merely changing an administrative property of a ship should keep the assessment of the risk associated with that ship the same.
3. Inspectors can use their discretion (possibly leading to subjectivity) to decide how thorough an inspection is.

Hence, ships flying a black flag could be subjected to stricter inspections, resulting in a higher probability of finding a noncompliant issue [20, 67]. This potential greater focus on ships flying a black flag may mean that these ships are inspected more often and stricter, contributing to a confirmation bias in historical inspection data [37]. The potential danger of inspectors' bias has been recognized, and great efforts are made to harmonize the training of inspectors, thereby making the overall inspectorate system consistent [56]. Nevertheless, complete global harmonization has yet to be achieved [67].

An option is to start ignoring a ship's flag altogether to reduce the confirmation bias mentioned earlier, thus providing what in the literature [75] is known as *equal opportunity*. However, correlations exist between the other characteristics of a ship and its target; thus, the classifier will indirectly learn to use the ship's flag, resulting in *inequality of outcomes*. Considering all drawbacks of using the flag in risk prediction, we argue that it might be better to get equal outcomes and therefore investigate how we can decorrelate the flag with respect to the outcome of the automated prediction of noncompliance. We do so by employing a so-called fair model [96] (see Definition 5), that can classify whether a ship is noncompliant but prevents (to a specified extent) correlation between its output and the ship's flag. Such a fair model reduces the confirmation bias and improves the overall fairness of the risk assessment.

## Our contribution

Rather than using potentially unfair and biased static ship characteristics, we prefer to consider the ships' actual behavior for noncompliance prediction, explicitly moving away from the six factors used in the ship risk profile. Ship behavior has been used to find anomalous ships [141], which may indicate noncompliance.

An example of ship behavior potentially characteristic of noncompliance is a ship sailing primarily on routes with much competition. Such routes may lead to reduced profit margins and a greater push for owners to cut shipping costs at the expense of safety.

While we have yet to determine the fares on specific routes, our proposed classifier will still consider relations between noncompliance and the sailed routes.

In the current study, we derive a cargo ship network from data containing notifications of ships calling a port. In the cargo ship network, nodes are ports, and edges are ships that travel between ports. By considering each port's structural function in the network, we extract mobility patterns for each ship. These mobility patterns are provided to the fair machine learning model, enabling automated assessment of the risk of ships based on their behavior. Altogether, we have devised an accurate, automated, interpretable and fair assessment of ship noncompliance based on ship behavior, providing an answer to Research question 5. The data used in our approach is available to all members of the European Maritime Safety Agency, allowing each of them to apply our approach.

The structure of this chapter is as follows. In Section 6.2, we provide related work on the ship risk profile and ship risk classification. Then, we explain the cargo shipping data used in this work in Section 6.3. Subsequently, we describe the research methodology in Section 6.4 We present the results of our proposed classifier in Section 6.5. A discussion of these results is provided in Section 6.6. Finally, conclusions are provided in Section 6.7.

## 6.2 Related work on ship risk profile

It is widely recognized that introducing the NIR, and thereby the ship risk profile has been beneficial to reducing the number of noncompliant ships [67, 166, 201, 202]. Nevertheless, some weaknesses have been identified [47, 48, 77, 176, 204–206]. We mention two of them, together with possible solutions that were provided. We then continue with discussing related work on the cargo ship network.

The first weakness in the existing ship risk profile, which assesses risks based on a weighted sum of six characteristic ship factors, is that the *weights are manually determined* [61]. In doing so, the model ignores any interactions between the factors. Here we remark that more complex models may consider more correlations, thereby improving performance [61, 207]. To this end, machine learning classifiers have been introduced that can automatically learn the weights and capture correlations between the factors. We provide two examples.

- A pipeline with a support vector machine and $k$-nearest neighbors have been used to find high-risk ships [61]. The support vector machine takes more complex (and non-linear) interactions into account and generalizes well, while $k$-nearest neighbors make the overall approach noise tolerant.
- A balanced random forest classifier has been used to predict ship detentions because only a tiny fraction of ships are detained [206].

The second weakness of the ship risk profile is that relatively static factors are used in risk assessment, meaning that the factors rarely change for a given ship. Indeed, many

datasets have been exploited that better reflect the current condition of a ship and hence will likely improve prediction. We mention four different datasets that have been used.

- Web scraping have been used to gather information from inspection reports [205].
- Company inspections have been used to enhance the ship risk profile [99].
- More historical information, such as times of changing flags and casualties in the last five years, have been proposed to add in the ship risk model [206].
- Information between different regimes should be more coherent, such that deficiencies and detentions in other regions can be used as well [99, 207].

The impact of the literature on our work is as follows. We read in the literature that it was strongly recommended to use additional data to come to a better prediction. We used port call data modeled as a cargo ship network. We mention the following four works on the cargo ship network, that have inspired us.

First, in 2010, the initial unveiling of a cargo ship network on a global scale was documented by Kaluza *et al.* [94]. According to their findings, the network had a *smaller diameter* (measuring $8$) than expected for a randomly constructed network of equivalent size. Additionally, they discovered that the *average distance* separating any two ports across the globe was just $2.5$.

Second, other researchers have found a diameter of only $7$ and an average distance of $3.3$ [113].

Third, the robustness of the cargo ship network has been studied by analyzing the transponder [151]. Different ship types were studied (oil tanker, container, dry bulk), and properties of these ship types have been reported for each sub-network derived from those ships. No measure of the distances in the network was reported, but a *density* (of $\sim 0.02$) similar to the first published cargo ship network was found.

Fourth, Van Veen (2020) analyzed the cargo ship network as derived from data of port calls [192]. Although the data was extracted only from journeys either departing or arriving at one of the Paris Memorandum of Understanding members, a diameter of $7$ was found and an average distance of $2.49$, similar to the reported values of other works.

In Section 6.3, we compare the properties of these networks to those of our cargo ship network. Ultimately, we *predict* noncompliance using a classifier with mobility patterns extracted from the cargo ship network (see Section 6.5). Our contribution is thus an approach that addresses the two weaknesses observed in the ship risk profile currently used by inspectorates: (1) manually adjusted weights and (2) relatively static factors.

## 6.3 Cargo shipping data

The chapter aims to classify ships' noncompliance using mobility data. The data used originates from two sources: (1) port calls (Subsection 6.3.1) and (2) inspections (Subsection 6.3.2). After collection, the port calls and inspections are merged (Subsection 6.3.3).

### 6.3.1 Port calls

The first data source, the port calls, contains notifications of cargo ships calling a port. The data contains only *calls to a port* participating in the Paris MoU and is accompanied by the following six pieces of information: (1) the International Maritime Organization (IMO) number — a unique identifier used in the maritime sector; (2) the port it calls to; (3) the date of arrival; (4) the duration that the ship is berthed; (5) the flag of the ship when it called; and (6) the ship risk profile (low, medium, high risk) computed when berthing. From this port call data, we reconstruct journeys that took place. A ship's journey goes from a departure port to an arrival port and has an associated travel time.

### 6.3.2 Inspections

The second data source, the inspections, provides information about ships with a deficiency. Also, we know whether such a deficiency has led to detention. Ships without deficiencies are assumed to be compliant because every ship should be inspected at least every three years at one of the ports participating in the Paris Memorandum of Understanding [56]. The inspection results are used as ground truth for our classifier. In Figure 6.1, we show the fraction of noncompliant ships that visit all countries. We observe that this fraction is very different across countries in Europe.

### 6.3.3 Merging port calls and inspections

Ships in these two datasets are linked using the IMO number. We select years occurring in data from both sources (2014–2018), resulting in over 3,000,000 calls from 28,416 cargo ships to a port in one of the thirty countries. Most of them, 97.3% (27,647 ships), did not change their flag during the years under consideration. Of these ships, the total number of ships with a white, gray, or black flag is 26,300, 672, and 675, respectively. Because only a tiny proportion of ships are flying a black or gray flag, we take them together and refer to the group as non-white flags. As mentioned before, ships can easily and quickly change their flag to either a so-called "Flag Of Convenience" (FOC) or a more trustworthy one with a better reputation [131]. In the data, 2.7% (1,347) of all ships changed their flag in 2014–2018. The distribution of flags over all countries is shown in Figure 6.2. We observe that most ships are registered in countries often identified as FOC, such as Panama and Liberia. Although difficult to observe, most ships are registered to Panama (2,904), Marshall islands (2,153), and Liberia (2,119), which are all known as typical FOC countries. In Figure 6.3, the fraction of noncompliant ships for each flag is shown. We observe that some black or gray flags are associated to a large fraction of noncompliant ships. The other way around, some of the white flag states have many noncompliant ships as well, such as the United States of America. Figures used in this section, can be downloaded at higher quality from our online repository [27].

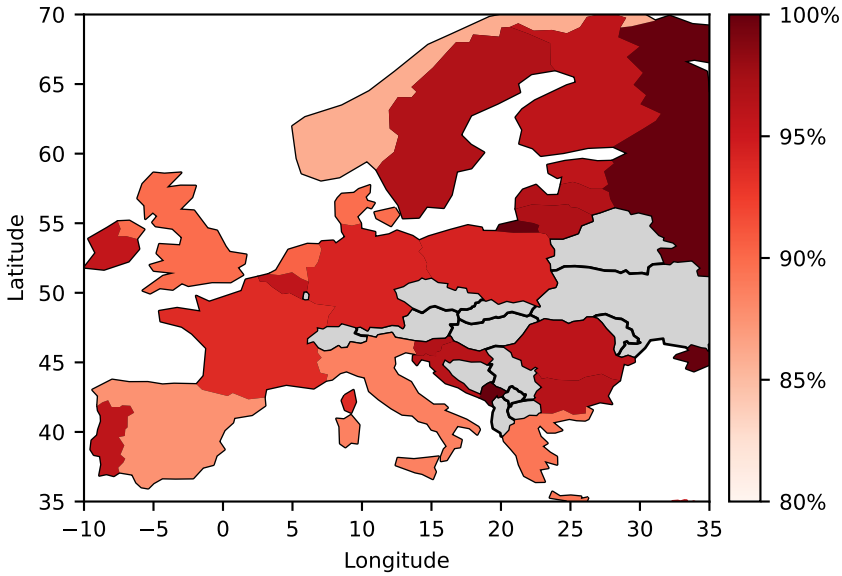



Figure 6.1: Fraction of ships being noncompliant per country. (Countries indicated in gray were not visited by a ship in the data.)

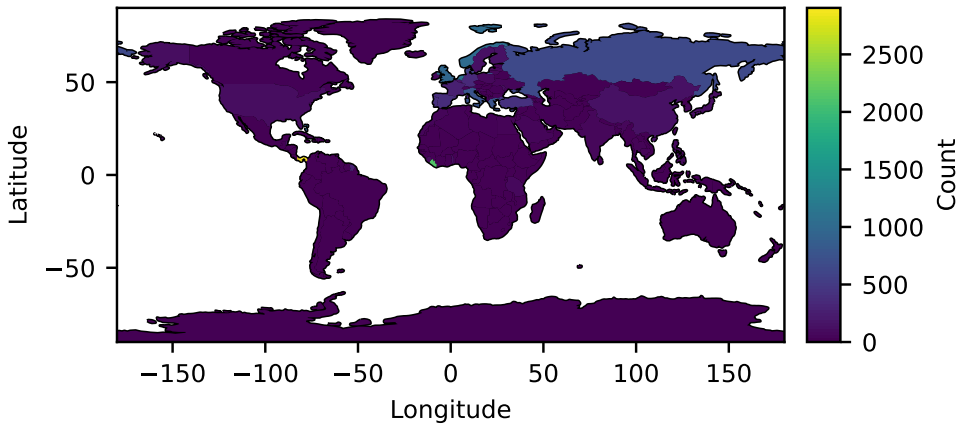



Figure 6.2: Number of ships registered to each country. (If multiple registrations for a single ship were observed, we use the most recent registration.)

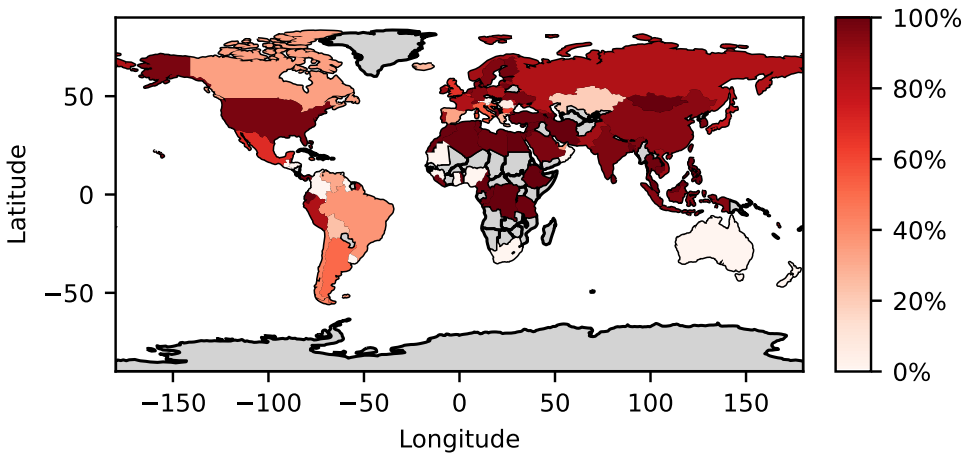



Figure 6.3: Fraction of ships for each flag state being noncompliant. (States without any ship registered to it, are indicated in gray.)

## 6.4 Chapter research methodology

We aim to create a machine learning classifier that performs a fair assessment of the risk for each ship. To this end, two feature types are input to the classifier; *network features* and *temporal features*.

In Subsection 6.4.1, we start by explaining the construction of the cargo ship network. We explain our approach to feature engineering, dealing with both the network and temporal features, in Subsection 6.4.2. Then, we discuss the classifier in machine learning in Subsection 6.4.3. We elucidate the fair model and explain the performance measures in Subsection 6.4.4. Finally, the fairness measures are explained in Subsection 6.4.5.

### 6.4.1 Cargo ship network

To obtain the structural importance of each port, we construct a cargo ship network. It is later used to characterize the behavior of ships. The edges of the directed weighted network are obtained by considering the journeys of all ships, linking a port to another port if at least one ship made a journey visiting those two ports immediately after each other. Edges are weighted according to how many journeys exist between the two ports. Hence, each node of the network is a port.

Below, we explain the structural properties of the cargo ship network in terms of their density, diameter, average distance, and clustering coefficient (for a definition of these elementary network measures, see Section 1.2). They help us understand whether our cargo ship network is, in fact, similar to earlier constructed networks of the same type. For each port, we obtain the following twelve structural importance measures:

- (1) in-degree; (2) out-degree; (3) degree;
- (4) in-strength; (5) out-strength; (6) strength;
- (7) closeness centrality and (8) weighted closeness centrality [60];
- (9) betweenness centrality and (10) weighted betweenness centrality [24, 59];
- (11) eigenvector centrality and (12) weighted eigenvector centrality [23].

These measures are used in Subsection 6.4.2 to engineer features that are provided to the machine learning classifier. We will now explain each of them.

- **Degree** of a node capture the number of routes (i.e., the number of edges connected to the node).
- **Strength** of a node capture the number of journeys connected to a port (i.e., the total weight of the edges connected to the node).
- **Closeness centrality** is equal to the reciprocal of the average shortest path distance from a node to all other nodes [60]. A more central node is closer to all other nodes and hence has a higher closeness centrality.
- **Betweenness centrality** is equal to the number of shortest paths between every pair of nodes that pass through to the node under consideration [59]. A node with high betweenness centrality is associated with playing an essential role in the network; disruption of this node will affect many shortest paths.
- **Eigenvector centrality** is determined using the eigendecomposition of the adjacency matrix [23]. High eigenvector values mean that the node is connected to many nodes with a high eigenvector centrality value.

With the latter three centrality measures, the aim is to capture a diverse set of measures for the structural role of a port in the cargo ship network.

The train set (used to learn the classifier) and the test set (used to estimate the classifier's performance) should be independent. To prevent the data used to *construct* the network is also used in testing, we work with separate hold-out data to construct the network. Hence, we assign every ship $i \in I$ to one of the two disjoint sets (here, $I$ denotes the set containing all ships). A 10% sample of all ships $I$ is then used for network construction ($I_{\text{network}}$), where the remaining ships ($I_{\text{classification}}$) are used in the classification part (later divided into train and test set by the cross-validation procedure, see Subsection 6.5.1).

## 6.4.2 Feature engineering

In $I_{\text{classification}}$, there are two different types of features that describe how ships behave: network features (see Subsection 6.4.2A) and temporal features (see Subsection 6.4.2B).

### 6.4.2A Network features

The network features aim to capture what type of ports a given ship visits. We obtain the network features in four steps.

**Step 1. Determination of structural importance of each port.** We characterize each ship's journey by the structural importance of the cargo ship network of both the departure and arrival ports. Only if the port is observed in the cargo ship network, the 12 structural importance measures (see Subsection 6.4.1) are determined. For each measure, we combine the value obtained from the departure port and the value obtained from the arrival port using the four arithmetic operations separately (sum, multiplication, absolute difference, and division). After this step, we have $12 \cdot 4 = 48$ values characterizing each journey.

**Step 2. Binning.** To capture the distribution of the values obtained for each journey, we make a histogram of all measures by splitting each of the 48 values obtained in the previous step into 10 equal-width bins. The edges of all these bins are learned from the journeys of $I_{\text{network}}$ to prevent information from leaking. After this step, we have $48 \cdot 10 = 480$ values for each journey.

**Step 3. Aggregation.** The model is ultimately provided with information about the individual ships' instances. Hence, we need to aggregate the information of each journey to a fixed set of values per ship. The 480 values, obtained from Step 2, can then be aggregated for each ship by summation of all journeys. After that, we normalize these values by dividing them by the total number of journeys. We use the total number of journeys as a separate feature and add it to the list. Normalization allows us to compare the distributions regardless of the number of journeys of a ship. In this way, we obtain $480 + 1 = 481$ features.

**Step 4. Encode the missingness.** In Step 1, we explained that the structural importance measures are only defined if the port was observed in the cargo ship network. The information that a port is missing in the network is informative for the classifier. Hence, we will encode this missingness, a common approach discussed in more detail in [138, 156]. We do so with two different features. The first feature equals the number of journeys where only one port was unobserved. The second feature equals the number of journeys where both ports were unobserved. In the end, we thus have $481 + 2 = 483$ network features.

#### 6.4.2B Temporal features

The temporal features are computed from the duration of a ship's journeys and port berths. Anomalous short or long ship berths or journeys may be indicative of noncompliance. For example, short berths may lead to rushing through safety procedures, while significantly longer berths may indicate problems with the port authorities. We first make a histogram of each ship's observed journey and port berth duration values to preserve the estimated distribution of the berth durations and travel timing during aggregation. The histogram is made by splitting each ship's berth and journey durations into 10 equal-width bins. The boundaries of the bins are learned from (1) the port calls and (2) the journeys occurring in $I_{\text{network}}$ to prevent information from leaking. In this way, $2 \cdot 10 = 20$ temporal features

are obtained. We sum all the values obtained for each ship of (1) the histogram of the berth duration and (2) the histogram of the journey duration and divide them by the total number of berths and journeys, respectively.

We have 483 network features and 20 temporal features, resulting in a total of 503 features describing each ship, represented by a vector $x_i$ for some ship $i$.

### 6.4.3 Fair random forest classifier

We employ a machine learning model to perform the automated assessment of noncompliance. The goal of the model is to learn for each ship $i \in I_{\text{classification}}$ from the feature vector $x_i \in X$ and target scalar $y_i \in Y$ a function $f\colon X \mapsto Z$ where $z_i \in Z$ is a score between $0$ and $1$. The positive instances, i.e., $y_i = 1$, indicate a noncompliant ship, and the negative instances a compliant ship. We may recall from Section 1.1 that in search of a particular type of fairness, we aim to reduce the classifier's dependency on sensitive features $s_i \in S$, where $s_i = 0$ marks a ship with a white flag (non-sensitive) and $s_i = 1$ otherwise.

We employ a *fair random forest classifier* [157], which is a modified random forest classifier. In brief, a random forest classifier works as follows. A bootstrapped training data sample is taken for every tree in the forest. Then, a decision tree is grown by recursively doing three steps:

1. Select a sample from all features available.
2. Optimize a criterion (commonly the information gain) calculated on each sampled feature.
3. Split the node into two child nodes based on the optimization outcome.

For more details of the working of a random forest classifier, we refer the reader to [76].

Like other tree learning classifiers, random forest classifiers have some beneficial properties. We mention two of them. The first property is that their robust performance has been *confirmed* in different domains, meaning that a minimum of tuning is needed [76]. The second property is that the criterion considered does not have to be differentiable, in contrast to many other classifiers, allowing to introduce the SCAFF criterion (see later on). Both properties together allow us to use a specifically designed criterion, called Splitting Criterion Area under the curve for Fairness (SCAFF) [157]. The criterion ensures that different labels are separated and the sensitive class remains mixed. We first give the definition and then explain the formulas.

$$\text{SCAFF}(Z, Y, S, \Theta) = (1 - \Theta) \cdot \text{AUC}_{\text{Y}}(Z, Y) - \Theta \cdot \text{AUC}_{\text{S}}(Z, S),$$

with $\text{AUC}_{\text{Y}}$ a value in the closed interval $[0, 1]$:

$$\text{AUC}_{\text{Y}}(Z, Y) = \frac{\sum_{i=1}^{y_+} \sum_{j=1}^{y_-} \sigma\left(Z_i, Z_j\right)}{y_+ \cdot y_-} \quad \text{with} \quad \sigma(Z_i, Z_j) = \begin{cases} 1, & \text{if } Z_i > Z_j \\ \frac{1}{2}, & \text{if } Z_i = Z_j \\ 0, & \text{otherwise} \end{cases},$$

where $y_+$ and $y_-$ mark the number of positive and negative instances. An $\text{AUC}_\text{Y}$ value of $0.5$ suggests random classification while $\text{AUC}_\text{Y} = 1$ indicates a perfect classifier. The $\text{AUC}_\text{S}$ considers the sensitive feature as the positive class. It is defined as follows:

$$\text{AUC}_\text{S}(Z, S) = \max\left(1 - \frac{\sum_{i=1}^{s_+}\sum_{j=1}^{s_-}\sigma\left(Z_i, Z_j\right)}{s_+ \cdot s_-}, \frac{\sum_{i=1}^{s_+}\sum_{j=1}^{s_-}\sigma\left(Z_i, Z_j\right)}{s_+ \cdot s_-}\right),$$

with $\sigma\left(Z_i, Z_j\right)$ defined exactly the same as for $\text{AUC}_\text{Y}$. The measure is closely related to strong demographic parity [93]. For $\text{AUC}_\text{S} = 0.5$, corresponding to a strong demographic parity of $0$, the split in the node is made regardless of the values of the sensitive features, meaning equality of outcome. A value of $\text{AUC}_\text{S} = 1$, corresponding to a strong demographic parity of $1$, is the worst score possible since, in that case, the classifier can predict the sensitive feature perfectly. The orthogonality parameter, $\Theta \in [0, 1]$, allows to balance the performance-fairness trade-off [96].

At a value of $\Theta = 0$, the fair random forest classifier optimizes solely for performance and does not consider any fairness. Hence, it corresponds, in that case, to the ordinary random forest classifier. At a value of $\Theta = 1$, the classifier optimizes fairness and neglects any performance. We refer the reader for more details on the fair random forest classifier to [157].

### 6.4.4 Performance measures

The classifier's performance can be determined by threshold-dependent and threshold-free metrics. Scores equal to or above the threshold $t \in [0, 1]$ are classified as positive ($\hat{y}_i = 1$), and values under the threshold are predicted as negative ($\hat{y}_i = 0$). Threshold-free metrics have the advantage that they do not require this explicit cut-off and instead consider the ranking imposed by the scores of the classifier. The three threshold-dependent performance metrics are (1) precision, (2) recall, and (3) the harmonic mean of those two, the $F_1$ score. The threshold-free performance metric used in this work is the $\text{AUC}_\text{Y}$ (see the previous section).

### 6.4.5 Fairness measures

Similar to the performance measures, fairness with respect to the sensitive group can also be quantified by two metrics: *threshold-dependent* and *threshold-free* metrics.

First, we report on the *threshold-dependent* metrics by (1) the precision and (2) the recall for the following two groups: (a) ships with a white flag and (b) ships with a non-white flag. A significant difference between these two groups indicates an unfair outcome of the model, which we aim to avoid.

Moreover, we report also on the *threshold-dependent* metrics by (3) demographic parity and (4) equalized odds [75]. These latter two measures consider the difference

in performance measures between the two groups, i.e., ships with a white flag and a non-white flag.

The *demographic* parity measure, denoted as $\epsilon_{\text{parity}}$, sets an accepted maximum on the absolute difference between the positive prediction rates of the two groups. It is mathematically represented as $\left|P\big(\hat{Y}=1|S=1\big)-P\big(\hat{Y}=1|S=0\big)\right| \leq \epsilon_{\text{parity}}$. Lower values of $\epsilon_{\text{parity}}$ signify more similar outcomes to the sensitive and non-sensitive groups, indicating fairer predictions.

The *equalized odds* metric, denoted as $\epsilon_{\text{odds}}$, imposes a maximum accepted difference on the equality of opportunity in a supervised learning setting. It is expressed as

$$\begin{aligned}\left|P\big(\hat{Y}=1|S=1,Y=0\big)-P(\hat{Y}=1|S=0,Y=0)\right| &\leq \epsilon_{\text{odds}},\\ \left|P\big(\hat{Y}=1|S=1,Y=1\big)-P(\hat{Y}=1|S=0,Y=1)\right| &\leq \epsilon_{\text{odds}}.\end{aligned}$$

Reduced values for $\epsilon_{\text{odds}}$ suggest greater equality of opportunity for the sensitive and non-sensitive groups, thus more fair predictions.

Finally, we have also reported on the *threshold-free* fairness measures (denoted by $\text{AUC}_S$), for which we refer to Subsection 6.4.3.

## 6.5 Results

The section starts with our experimental setup in Subsection 6.5.1. Then, we continue analyzing the cargo ship network in Subsection 6.5.2. In Subsection 6.5.3, we evaluate the baseline ship risk profile performance. Subsequently, we report on the performance of the non-fair random forest classifier in Subsection 6.5.4 and the fair random forest classifier (announced in Subsection 6.4.3 as our preferred choice) in Subsection 6.5.5. In Subsection 6.5.6, we report on the effects of the orthogonality parameter. Finally, in Subsection 6.5.7, we describe the effects of the threshold quantile in combination with the orthogonality parameter.

### 6.5.1 Experimental setup

In our experimental setup, we use five-fold nested cross-validation with stratified sampling [39]. The inner folds select the best parameter set for that specific outer fold. The considered parameters are all combinations of the selected values for the depth of each tree ($\{1,2,\ldots,10\}$) and the number of bins (10 or 2) used in discretization for continuous variables. Hence, there are $10\cdot 2=20$ candidate sets of parameters in each outer fold. The mean and standard deviation of the classifier's performance is evaluated on the five outer folds using the selected parameter set. We report the outcome of this cross-validation for 11 different values of the orthogonality parameter, $\Theta \in \{0, 0.1, 0.2, \ldots, 1\}$.

The code used in this research is publicly available [27]. It uses several open-source Python packages. Specifically, scikit-learn [149], SciPy [193], and Pandas [117] are used

for feature engineering and for measuring the performance of the baseline ship risk profile and the proposed classifier. The fair random forest is open source as well [152], making extensive use of the CVXpy package for optimizing SCAFF [2]. For analyzing the cargo ship network, we used the NetworkX package [72]. The C++ library teexGraph was used to determine the diameter of the network [185]. The packages used for visualization and all other dependencies and supportive software versions can be found at [27].

### 6.5.2 Cargo ship network

A quite "overwhelming" visualization of the cargo ship network obtained is shown in Figure 6.4. Still, we only show ports in Europe because we are interested in predicting the risk for ships that arrive in Europe. From the figure, we can learn the following four properties.

1. A GC connects virtually all ports.
2. Only a few ports have high strength, as indicated by the yellow color, of which (1) Puttgarden (Germany), (2) Rotterdam (Netherlands), and (3) Algeciras (Spain) have the highest strength.
3. Two different types of ports can be distinguished: (1) ports that are well-connected (e.g., ports in Germany, Netherlands, and Belgium), and (2) ports that are more in the network's periphery (e.g., Iceland and the Azores).
4. Some ports are connected by thick lines, indicating an edge with a high weight. The nodes connected by these edges are likely to have a high weighted betweenness centrality because the failure of such nodes would cause other shortest paths to run through edges with less weight.

In Table 6.1, we provide numeric information on sizes, relations, and distances. In the first column, we show our work's nine common properties of cargo ship networks. In the second column through the sixth column, we provide values for the properties of our network and four similar cargo networks observed in literature [94, 113, 151, 192]. We compare these properties to understand whether our 10% sample used to compute port features is representative. From Table 6.1, we see that although very different numbers of nodes and edges are reported in these works, the measures such as *density*, *diameter*, and *clustering coefficient* are similar. Hence, we may conclude that the constructed cargo ship network can extract mobility patterns for our ship compliance classifier in a sensible way.

### 6.5.3 Performance of the baseline ship risk profile

The confusion matrices for the baseline ship risk profile are shown separately for the white and non-white flags in Figure 6.5. Together with Table 6.2, where we show the *calculated performance* and *fairness measures*, they provide information on the performance of the baseline ship risk profile. We remark that low or medium risk ships are predicted as compliant.

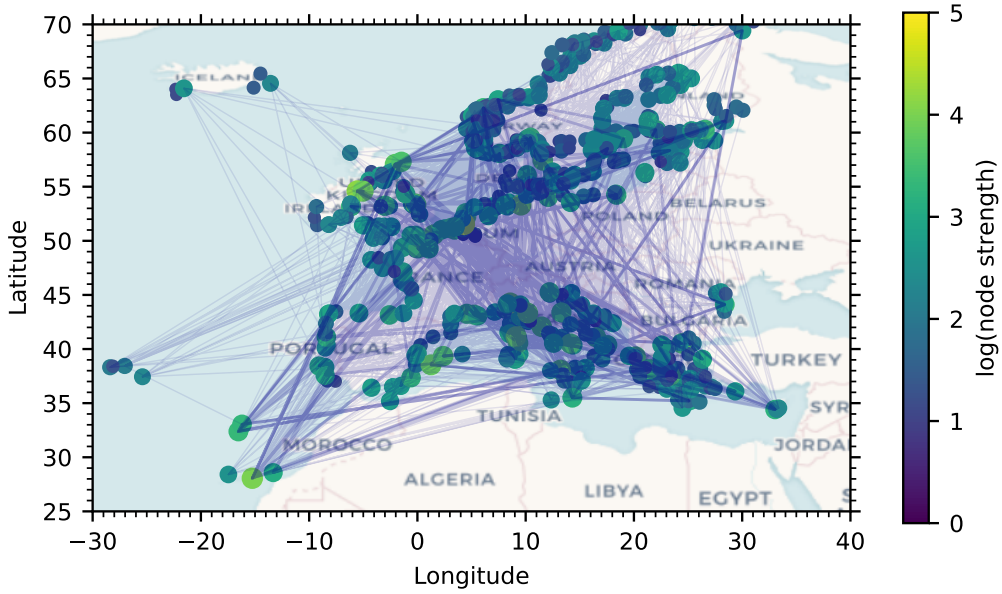



Figure 6.4: The considered cargo ship network. (Nodes are colored by their strength. Thicker edges mark busy routes. The figure is generated using OpenStreetMap data.)

Table 6.1: Summary statistics of considered cargo ship networks.

| Property | *This work* | [192] | [151] | [113] | [94] |
|---|---|---|---|---|---|
| Directed | Yes | Yes | No | No | No |
| Number of nodes | 1,459 | 728 | 1,488 | 439 | 951 |
| Number of nodes in GC | 1,445 | 726 | – | – | 935 |
| Number of routes | 28,653 | 18,142 | 17,135 | 2,331 | 36,328 |
| Number of routes in GC | 28,638 | 18,140 | – | – | – |
| Density in GC | 0.027 | 0.03 | 0.015 | 0.019 | 0.08 |
| Diameter in GC | 6 | 7 | – | 7 | 8 |
| Average distance in GC | 2.63 | 2.49 | 2.99 | 3.290 | 2.5 |
| Clustering coefficient in GC | 0.48 | 0.58 | 0.55 | 0.396 | 0.49 |

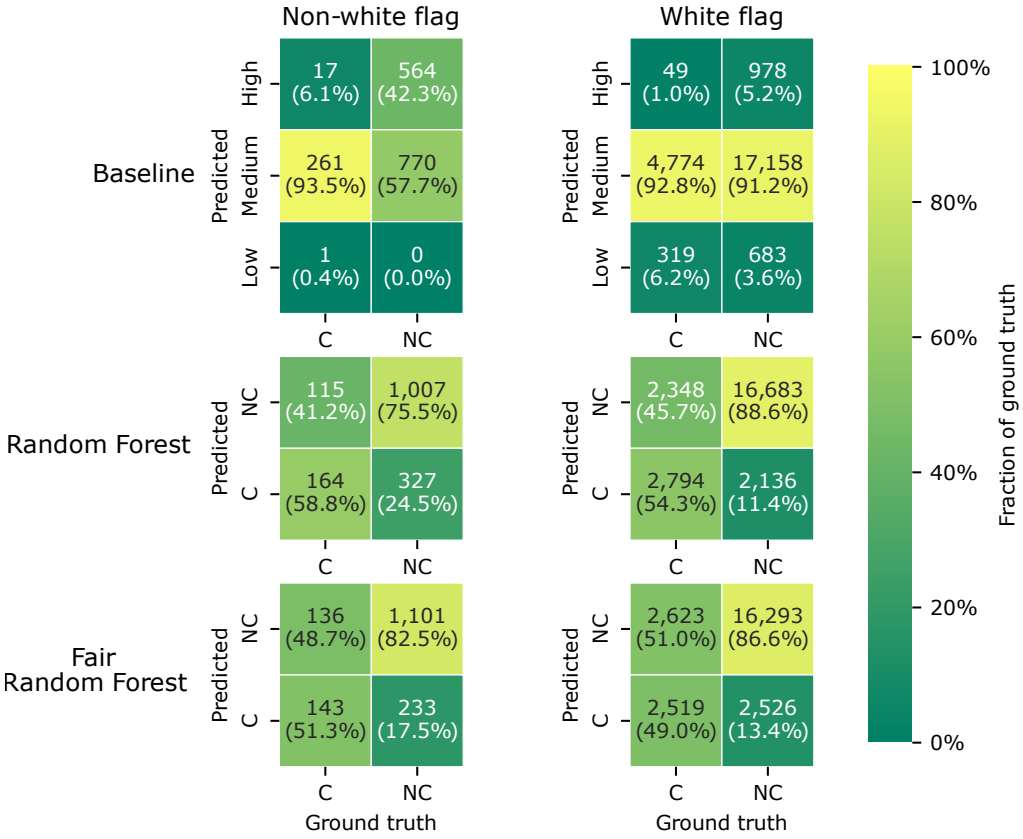



Figure 6.5: Confusion matrices (shown for both the white and non-white flagged ships). (Baseline model is the ship risk profile currently in use. C and NC mark Compliant and NonCompliant ships, respectively. The percentages [and color coding] are stratified based on the ground truth.)

Table 6.2: Performance (precision, recall, and $F_1$, and $\text{AUC}_\text{Y}$) and fairness (demographic parity and equalized odds, and $\text{AUC}_\text{S}$) measures for the different models.

| Measure | Baseline | Random forest | Fair random forest |
|---|---|---|---|
| precision (non-white) | 97.1% | 89.8% | 89.0% |
| precision (white) | 95.2% | 87.7% | 86.1% |
| recall (non-white) | 42.3% | 75.5% | 82.5% |
| recall (white) | 5.2% | 88.6% | 86.6% |
| $F_1$ (non-white) | 58.9% | 82.0% | 85.6% |
| $F_1$ (white) | 9.9% | 88.2% | 86.4% |
| $\epsilon_{\text{parity}}$ | 0.317 | 0.099 | 0.023 |
| $\epsilon_{\text{odds}}$ | 0.371 | 0.132 | 0.040 |
| $\text{AUC}_\text{Y}$ | $0.543 \pm 0.006$ | $0.814 \pm 0.004$ | $0.776 \pm 0.008$ |
| $\text{AUC}_\text{S}$ | $0.672 \pm 0.010$ | $0.627 \pm 0.014$ | $0.538 \pm 0.011$ |

Below, we make four observations from Figure 6.5 and Table 6.2.

First, we note that virtually no ship flying a non-white flag gets a low-risk profile (see left upper corner), indicating that the baseline model uses the flag to a large extent.

Second, most ships (90%) are classified as medium risk (see baseline predicted medium). Only a small fraction, $(261 + 4774)/(261 + 770 + 4774 + 17158) = 22\%$, is compliant.

Third, a smaller fraction, $(17 + 49)/(17 + 49 + 564 + 978) = 4\%$, is compliant from the ships with a high-risk profile. It results in high precision for the baseline model. However, the recall is relatively low as many ships with a medium risk profile are also noncompliant.

Fourth, unexpectedly, ships with a white flag with a low or medium risk profile are more noncompliant than ships with a non-white flag. It also results in a low value of the $\text{AUC}_\text{Y}$ value of only $0.543 \pm 0.006$ (see Table 6.2). Hence, we may conclude that using the data from 2014–2018, we cannot predict compliance with the baseline ship risk profile. It follows that the model is *quite unfair*. In particular, we observe a significant difference in the $F_1$ metric for the white and non-white group, resulting in high values for $\epsilon_{\text{parity}}$ and $\epsilon_{\text{odds}}$ (see Table 6.2). There is a strong correlation between the sensitive feature, i.e., the ship flag, and the scores of the model with $\text{AUC}_\text{S} = 0.672 \pm 0.010$ (see Table 6.2).

### 6.5.4 Performance of the random forest classifier

The confusion matrices of the random forest classifier are also shown in Figure 6.5. In Table 6.2 we report the performance and fairness metrics (column 3). Below, we make five observations. *First*, we observe that more ships are predicted correctly compared to the baseline model. *Second*, the recall is higher, meaning many actual positives are predicted. The table also shows decreased precision, indicating that many compliant ships are predicted as noncompliant. *Third*, the harmonic mean of the recall and precision, the $F_1$ measure, is higher than in the baseline model, indicating that the random forest classifier outperforms the baseline model. *Fourth*, the $\text{AUC}_\text{Y}$ measure, shows a high value of $0.814 \pm 0.004$, supporting also that the random forest classifier outperforms the baseline model. It implies that we accurately can assess the ship noncompliance in an automated fashion with a random forest classifier using behavioral data. *Fifth*, the confusion matrices (Figure 6.5) show that ships with a white flag are predicted to be noncompliant more often than ships with a non-white flag. The difference in frequency results in a higher recall for ships with a white flag.

Finally, we remark that the prediction by the random forest classifier is much more fair compared to the baseline model. In conclusion we remark that the random forest classifier does not use the flag as a feature, meaning that using only behavioral data thus makes the model more fair.

### 6.5.5 Performance of the fair random forest classifier

The confusion matrices of the fair random forest classifier are also shown in Figure 6.5. In Table 6.2 we report the performance and fairness metrics (column 4). Below we list our three observations. *First*, from the confusion matrices in Figure 6.5 and the performance and fairness metrics in Table 6.2, we observe that the fair random forest classifier has comparable true positive and true negative rates amongst ships flying a white and non-white flag, with only a small cost in predictive performance. *Second*, the $F_1$ performance measure drops only for the ships flying a white flag, so the difference between the two groups becomes minimal. *Third*, the demographic parity and equalized odds measures decrease when using a fair random forest classifier, suggesting that the classifier improved fairness.

### 6.5.6 The effect of the orthogonality parameter

Before drawing any conclusion, we show the effect of the orthogonality parameter ($\Theta$) in more detail (see Figure 6.6). Below we list our six observations. *First*, the top left figure (Figure 6.6A) shows that the $\text{AUC}_\text{Y}$ measure is only weakly influenced by a broad range of values for the orthogonality parameter, meaning that overall, we can reliably ensure equality of outcome while maintaining acceptable performance. *Second*, an orthogonality value of $0.7$ appears to give the best trade-off between performance and fairness in our work, with a performance of $\text{AUC}_\text{Y} = 0.776 \pm 0.008$ and fairness of $\text{AUC}_\text{S} = 0.538 \pm 0.011$. *Third*, the performance can be further improved (although slightly, to $\text{AUC}_\text{Y} = 0.814$), but only at decreased equality of outcome and vice versa.

Then we will closely investigate Figure 6.6B, where the two fairness measures decrease monotonically at increasing orthogonality values. *Fourth*, we make one observation that the extreme value of $\Theta = 1$, they are zero, but at this value, the predictive performance is also deficient, as can be observed in Figure 6.6A.

Subsequently, in Figure 6.6C and Figure 6.6D, we make two observations. *Fifth*, we observe that the precision and recall for ships flying a white and non-white flag have only minor differences for larger values of the orthogonality. The precision of the ships flying a non-white flag increases slightly at higher values of the orthogonality at the cost of precision for vessels with a white flag. *Sixth*, the threshold was set to $t = 0.34$ so that $P(Z \geq t)$ equals $P(Y = 1)$. This threshold is also used to calculate the confusion matrix shown in Figure 6.5.

In conclusion, we remark that the threshold $t$ is essential, as it determines how many ships are noncompliant. Higher threshold values result in fewer ships that are predicted as noncompliant. Therefore, we define the threshold quantile $Q_t$ so that $P(z \geq t)$ equals the threshold quantile.

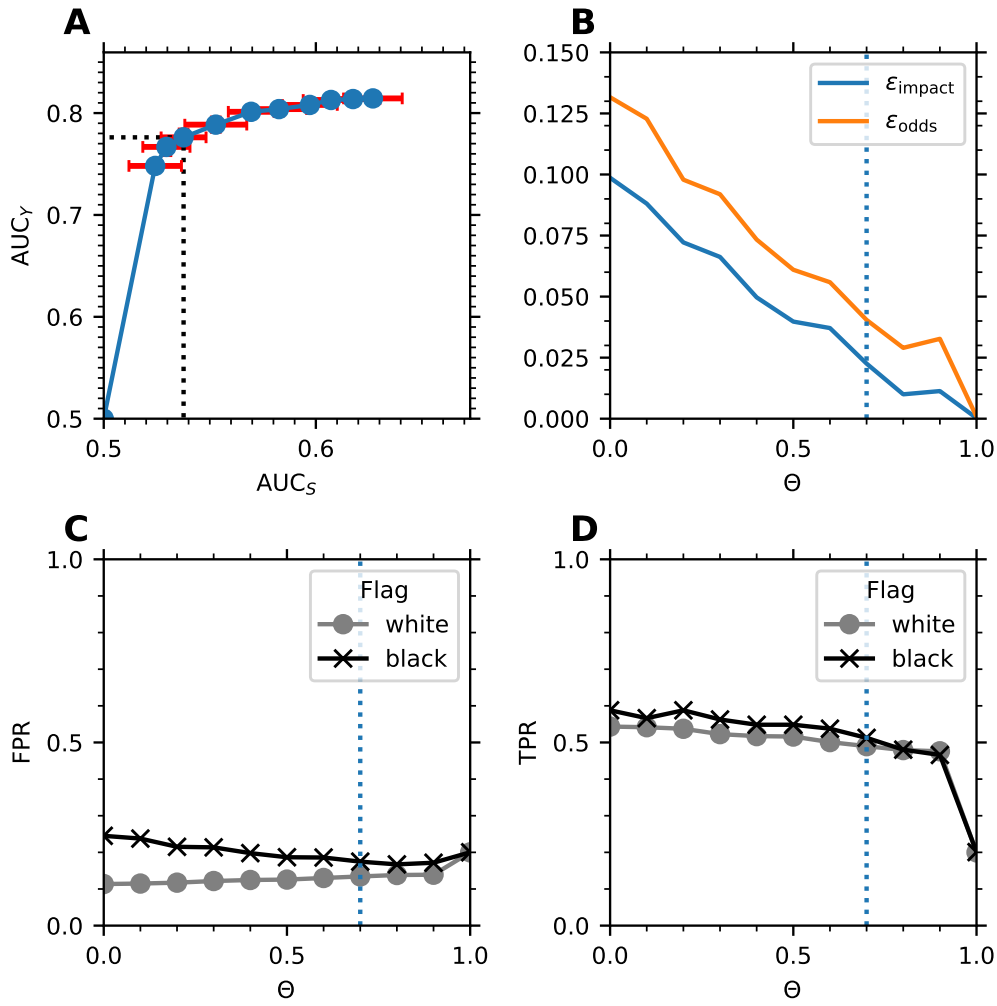



Figure 6.6: Performance and fairness of proposed ship selection classifier: (**A**) The performance of the fair random forest classifier for different values of the orthogonality. (**B**) The fairness performance is measured in demographic parity and equalized odds for different values of Θ. (**C**)–(**D**) The performance measured in precision (**C**) and recall (**D**) for different values of Θ, separated for ships flying a white and non-white flag.

### 6.5.7 The effect of the orthogonality and threshold quantile together

Finally, in Figure 6.7, we show the effect of the orthogonality and the threshold quantile on the selected threshold-dependent fairness measures. Below we list our three observations. *First*, we observe that high values of the orthogonality yield a fair prediction for all values of the threshold, even when the threshold quantile is set to a high value, such that most ships are predicted to be compliant. *Second*, for lower values of the orthogonality, we observe that the model's fairness is worst when the threshold quantile is near $0.5$. This result is expected (see below). *Third*, at other values of the threshold quantile, the performance for both groups is low, leading to a slight difference between the groups. Even at these "bad" choices for the orthogonality and threshold quantile, the values of the demographic parity measure and the equalized odds measure are still lower than observed for the baseline ship risk profile.

In conclusion, from these results, we may state that the fair random forest classifier *effectively reduces bias* towards a ship's flag for wide ranges of the used threshold and orthogonality. It answers Research question 5.

## 6.6 Discussion on limitations

This section discusses two limitations of our proposed classifier.

First, the ground truth might be biased toward the flag and the inspector's background [67]. The problem is that different inspectorates assess compliance differently for similar ships. The difference in assessment leads to inequality between ports and

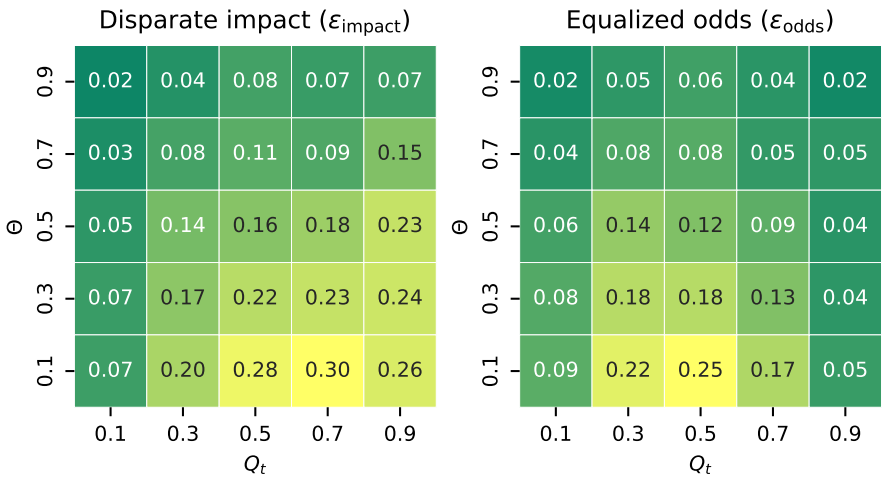



Figure 6.7: Fairness measures evaluated on the proposed classifier (as a function of the threshold quantile $Q_t$ and orthogonality $\Theta$).

so-called port-shopping. Port-shopping means that a noncompliant ship decides to go to another port solely because the inspection regime favors noncompliant vessels. In this way, the ship yields a lower risk profile. Port-shopping seriously influences our model since the ground truth data is unjustly positive for such noncompliant ships. The mission of the Paris MoU is to avoid this kind of competition between ports [147]. Hence, as a remedy, the inspection country could be added as a sensitive feature in future work, reducing the correlation between the inspectorate and the inspection outcome.

Second, we consider Goodhart's law, commonly formulated as: "When a measure becomes a target, it ceases to be a good measure" [181]. It applies to any ship risk model because ships are incentivized to get a *low-risk* profile. In the baseline ship risk model, a better risk profile could be achieved by *changing the administrative property* of the ship. In our fair random forest classifier endowed with orthogonality and threshold quantile setting, ships would need to change their behavior to get a better score, which is more complicated than merely changing administrative properties.

## 6.7 Chapter conclusions

The present research answers Research question 5: "How can ship behavior be utilized to enable smart inspection of cargo ships?" We devised an *accurate*, *automated*, *fair*, and *interpretable* assessment of ship risk, enabling smart inspection of cargo ships. This study has led to two conclusions.

**Conclusion 1**: We can offset the confirmation bias in historical inspection data using a fair random forest classifier. Experimental results indicated that the disparate impact and equalized odds measures improve significantly the assessment. This is regardless of chosen parameters, meaning that the constructed classifier works well.

**Conclusion 2**: The performance of our approach provided with behavioral data is $\text{AUC}_\text{Y} = 0.776 \pm 0.008$, which improves on the $\text{AUC}_\text{Y} = 0.543 \pm 0.006$ of the ship risk profile currently in use.

All in all, our *final conclusion* is that our work will support global efforts to minimize risks associated with maritime transport by conducting more targeted inspections. More generally, we have shown how ubiquitous mobility information can perform inspections to be better and more fair than so far. Finally, we believe that the devised approach may apply to inspection applications broader than port state control.

### Chapter outlook

Below we provide four directions of future research.

First, a natural continuation of this work is to (with the help of domain experts) determine (1) what behavior is often associated with high risk, and subsequently (2) how we can reduce riskful behavior.

Second, a direction for future work is to consider higher-order effects in the cargo ship network [170]. Building a higher-order network allows for a more accurate representation of the underlying complex system, which may enable more accurate network analysis results. It has been shown that relations up to the fifth order may be relevant in cargo shipping networks [170].

Third, we may investigate to what extent the temporal aspect of the network can be exploited to obtain a better, more accurate centrality measure that captures the true, time-aware structural importance of the ports [172].

Fourth, we may investigate to what extent the research under the third direction will result in an even better-performing classifier for the task at hand.

# 7 Conclusions

In this final chapter, we first answer the five research questions in Section 7.1. Subsequently, our answer to the problem statement is formulated in Section 7.2. Lastly, five future research directions (in addition to Section 6.7) are proposed in Section 7.3.

## 7.1 Answers to the research questions

We reiterate the research questions formulated in Chapter 1. Each research question is answered separately, along with references to relevant sections in which details can be found.

> **Research question 1:** *What is the relation between network structure and model performance in link prediction?*

In Chapter 2, we considered a large set of temporal, structurally diverse, real-world networks. We investigated the relationship between the structure of these networks and the model performance in link prediction for this set of networks. We found several structural network properties related to model performance in link prediction. Most notably, a negative correlation was discovered between network degree assortativity and link prediction performance. This negative correlation was also observed for real-world networks that had their degree assortativity artificially altered by means of a degree rewiring process. Our research showed that link prediction performance is generally higher in degree disassortative networks. In degree disassortative networks, the numerous low-degree nodes connect more frequently with hubs than with other low-degree nodes. For these low-degree nodes, the preferential attachment feature will provide higher scores for high-degree candidate node pairs. Hence, the supervised model can use this information to perform better (Finding 1).

In addition, regarding the temporal structure of networks, we distinguished between two classes of temporal networks, being temporal networks (1) containing only *persistent*

*relations* and (2) also containing *discrete events* (see Section 2.1). We found that model performance in link prediction improved significantly when in networks with discrete events, all events were explicitly taken into account. We coin this method "past event aggregation". It essentially is a method in which *all* information contained in both *persistent relations* and all *discrete events* is used (Finding 2).

Together, these two findings provide an answer to Research question 1.

**Research question 2:** *How can we obtain accurate estimates of the performance of link prediction models by using adequate splits into the train, validation, and test set?*

In Chapter 3, we described two dominant methods from the literature used to split network data in a train, validation, and test set for link prediction. We applied these two methods, called: the (1) random split and (2) temporal split, to six different temporal networks that have a considerable number of nodes and edges. We learned that the *random split* method provides (too) *optimistic* results. Therefore, the *temporal split* method should be used because we confirmed that it gives a *more realistic indication* of performance.

**Research question 3:** *How do network structure and vehicle attributes relate to co-driving behavior?*

In Chapter 4, we applied the link prediction approach to the truck co-driving network in an attempt to better understand the behavior of trucks and their drivers. Our research on the importance of features indicates that the network structure is better explained by co-driving behavior than by vehicle (node) characteristics. In particular, the *neighborhood features* that capture relevant information about the ego networks explained the observed co-driving behavior well.

**Research question 4:** *How can node attribute information be exploited to automatically create a good partitioning of a co-driving network into communities?*

In Chapter 5, we investigated the task of detecting communities of the truck co-driving network. The communities were detected by a modularity maximization algorithm, which has a resolution parameter. This parameter determines whether a more fine-grained or coarse-grained partition into communities is preferred. We proposed a method that considers node attributes to determine the best partitioning of the network into communities. In this method, a metric that we call average maximal community assortativity quantifies how well, on average, each community can be understood in terms of its node attributes. This metric was maximized to find the best choice for the resolution parameter. When applied to the truck co-driving network, results indicated that a good partitioning into communities was obtained by considering geographical aspects of the trucks as node attributes.

**Research question 5:** *How can ship behavior be utilized to enable smart inspection of cargo ships?*

The smart inspection entails the accurate, automated, fair, and interpretable assessment of (in our case) cargo ships. In Chapter 6, we proposed a machine learning model capable of predicting cargo ship noncompliance. We make use of (fair) random forests, because they allow humans to understand (1) what procedures were followed to make the model, (2) the inner workings of the model, and (3) how the model arrives at its predictions. The model's fairness was obtained using fair pre-trained models. The model decorrelates a ship's flag from the noncompliance prediction to reduce present bias in historical data and thereby prevent confirmation bias. The cargo ship network is constructed from behavioral data, which is less sensitive to manipulation than administrative information. Features derived from this cargo ship network served as input for the machine learning model. In summary, the entire approach led us to demonstrate how smart inspection should take place in the future.

## 7.2 Answer to the problem statement

After addressing the research questions, we now turn to the problem statement.

**Problem statement:** *How can network science methods leverage behavioral data for smart inspection of vehicles?*

The short answer to the problem statement is to be seen by applying the results of all five research questions. We summarize them below.

In answering Research question 1, we have shown that network science methods can generate useful features for a downstream machine learning task. This is *directly applicable* to the more fundamental link prediction task in networks, as seen in Chapter 2, and also *useful* in applied settings, for example, in identifying noncompliant ships (Chapter 6).

In answering Research question 2, we have shown that in link prediction, careful consideration must be given to splitting instances into an appropriate train, validation, and test set (Chapter 3).

Moreover, in answering Research question 3, we have explored other network science methods to better understand vehicle data, with a special focus on the relation between network structure, vehicle characteristics.

In answering Research question 4, we address the community structure (Chapters 4 and 5). The obtained results in these two chapters demonstrated that a network perspective on truck driving activities helps to uncover patterns that may ultimately be useful for promoting co-driving and reducing traffic congestion and fuel usage.

Finally, in answering Research question 5, we used network science tools to consider behavior as features in a machine learning model. By application of fair pre-trained models in Chapter 2, we achieved the desired smart inspection of vehicles.

## 7.3 Future research directions

The following are five directions (seen as addition to the four directions mentioned in Section 6.7) fruitful for future research.

1. **Argument:** Many current link prediction approaches have limitations in handling large and dynamic networks [101]. Applying dimensionality reduction before link prediction may improve scalability but could negatively impact interpretability.
   **Future research:** One straightforward direction is to produce interpretable techniques that scale well to larger networks. Many real-world networks are highly sparse, meaning the number of positive instances (pairs of nodes that will link) is very few compared to negative ones (pairs of nodes that do not link). Therefore, positive instances can be considered outliers, and thus outlier detection techniques may do well in link prediction, especially on large and dynamic networks.
2. **Argument:** We encountered limited availability of temporal network datasets.
   **Future research:** To advance link prediction, a *more diverse set* of temporal networks must be accessible to the public and not locked in private "silos" where they are accessible only by some [179]. To start, in Chapter 2, we presented a collection of 26 temporal networks.
3. **Assumption:** Incorporating features obtained from more sophisticated transport network models into smart inspection techniques may benefit prediction performance.
   **Future research:** Higher-order networks [170, 203] and evolutionary hypergraphs [212] have been proposed as more effective representations for capturing vehicle trajectories.
4. **Argument:** A natural progression of this work is to consider a more holistic approach toward inspection in the transport domain. Whereas in this work, we analyzed the cargo trucks and cargo ships separately, they are not independent in the real world. The containerization of the transport system facilitates smooth transfers between different modalities.
   **Future research:** A further study could assess the risk associated with the entire cargo journey.
5. **Argument:** In our work, we did not extensively consider the uncertainty in the network inferred from the available raw data. However, our data is likely partially incomplete, raising possible questions about to what extent the dataset is representative.
   **Future research:** A greater focus on measuring errors could shed more light on the difference between the data (i.e., what is measured) and the abstract, underlying network representation [150].

## General goal and general recommendation

Ultimately, our goal is to improve *cleanliness* and *safety* in the transport domain. The proposed approach to smart vehicle inspection is just one of the actions needed to arrive at transportation without any danger or unnecessary environmental pollution. A combined and continuous effort is needed from many professions (policymakers, scientists, inspectors, and of course, ultimately, the vehicle drivers themselves) to offset all negative transportation consequences. We expect that the work in this thesis will contribute to the ongoing shift toward the smart inspection of vehicles.

# References


[1] L. A. Adamic and E. Adar. „Friends and neighbors on the web." *Social Networks* 25.3 (2003), pages 211–230. DOI: 10.1016/S0378-8733(03)00009-1.

[2] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. „A rewriting system for convex optimization problems." *Journal of Control and Decision* 5.1 (2018), pages 42–60. DOI: 10.1080/23307706.2017.1397554.

[3] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. „Link prediction using supervised learning." In: *Proceedings of the 4th Workshop on Link Analysis, Counter-terrorism and Security*. 2006.

[4] M. Al Hasan and M. J. Zaki. „A survey of link prediction in social networks." In: *Social Network Data Analytics*. Springer, 2011, pages 243–275. DOI: 10.1007/978-1-4419-8462-3_9.

[5] R. Albert and A.-L. Barabási. „Topology of evolving networks: Local events and universality." *Physical Review Letters* 85 (2000), pages 5234–5237. DOI: 10.1103/PhysRevLett.85.5234.

[6] R. Albert and A.-L. Barabási. „Statistical mechanics of complex networks." *Reviews of Modern Physics* 74 (1 2002), pages 47–97. DOI: 10.1103/RevModPhys.74.47.

[7] T. Aynaud. *python-louvain: Louvain algorithm for community detection*. Github repository. Dec. 27, 2020.
https://github.com/taynaud/python-louvain (visited on Dec. 30, 2021).

[8] R. Baeza-Yates and J. Matthews. *Statement on principles for responsible algorithmic systems*. 2022.
https://www.acm.org/binaries/content/assets/public-policy/final-joint-ai-statement-update.pdf (visited on Dec. 23, 2022).

[9] A.-L. Barabási. „The origin of bursts and heavy tails in human dynamics." *Nature* 435 (2005), pages 207–211. DOI: 10.1038/nature03459.

[10] A.-L. Barabási. *Network science*. Cambridge University Press, 2016.
http://networksciencebook.com (visited on Dec. 28, 2023).

[11] A.-L. Barabási. *Love is all you need: Clauset's fruitless search for scale-free networks.* Mar. 6, 2018.
https://web.archive.org/web/20210726203453/https://www.barabasilab.com/post/love-is-all-you-need (visited on July 3, 2022).

[12] A.-L. Barabási and R. Albert. „Emergence of scaling in random networks." *Science* 286.5439 (1999), pages 509–512. DOI: 10.1126/science.286.5439.509.

[13] S. Barocas, M. Hardt, and A. Narayanan. *Fairness and machine learning: limitations and opportunities*. 2019.
http://www.fairmlbook.org.

[14] A. Barrat and C. Cattuto. „Temporal networks of face-to-face human interactions." In: *Temporal Networks*. Springer, 2013, pages 191–216. DOI: 10.1007/978-3-642-36461-7_10.

[15] B. Barzel and A.-L. Barabási. „Universality in network dynamics." *Nature Physics* 9 (2013), pages 673–681. DOI: 10.1038/nphys2741.

[16] M. G. H. Bell and Y. Lida. *Transportation network analysis*. John Wiley & Sons, Ltd, 1997. DOI: 10.1002/9781118903032.

[17] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[18] C. A. Bliss, M. R. Frank, C. M. Danforth, and P. S. Dodds. „An evolutionary algorithm approach to link prediction in dynamic social networks." *Journal of Computational Science* 5.5 (2014), pages 750–764. DOI: 10.1016/j.jocs.2014.01.003.

[19] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. „Fast unfolding of communities in large networks." *Journal of Statistical Mechanics: Theory and Experiment*, P10008 (2008). DOI: 10.1088/1742-5468/2008/10/P10008.

[20] M. Bloor, R. Datta, Y. Gilinskiy, and T. Horlick-Jones. „Unicorn among the cedars: On the possibility of effective "smart regulation" of the globalized shipping industry." *Social & Legal Studies* 15.4 (2006), pages 534–551. DOI: 10.1177/0964663906069546.

[21] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. „Complex networks: Structure and dynamics." *Physics Reports* 424 (4–5 2006), pages 175–308. DOI: 10.1016/j.physrep.2005.10.009.

[22] J. Bollen, B. Gonçalves, G. Ruan, and H. Mao. „Happiness is assortative in online social networks." *Artifical Life* 17 (3 2011), pages 237–251. DOI: 10.1162/artl_a_00034.

[23] P. Bonacich. „Power and centrality: a family of measures." *American Journal of Sociology* 92.5 (1987), pages 1170–1182. DOI: 10.1086/228631.

[24] U. Brandes. „A faster algorithm for betweenness centrality." *Journal of Mathematical Sociology* 25 (2 2001), pages 163–177. DOI: 10.1080/0022250X.2001.9990249.

[25] U. Brandes, P. Kenis, J. Lerner, and D. van Raaij. „Network analysis of collaboration structure in Wikipedia." In: *Proceedings of the 18th International Conference on World Wide Web (WWW)*. 2009, pages 731–740. DOI: 10.1145/1526709.1526808.

[26] A. D. Broido and A. Clauset. „Scale-free networks are rare." *Nature Communications* 10, 1017 (2019). DOI: 10.1038/s41467-019-08746-5.

[27] G. J. de Bruin. *Fair automated assessment of non-compliance in cargo ship networks*. Github repository. Nov. 25, 2021. DOI: 10.5281/zenodo.5727085.

[28] G. J. de Bruin. *Supervised temporal link prediction in large-scale real-world networks*. Github repository. Nov. 25, 2021. DOI: 10.5281/zenodo.8067342.

[29] G. J. de Bruin, A. Pereira Barata, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Fair automated assessment of non-compliance in cargo ship networks." *EPJ Data Science* 11, 13 (2022). DOI: 10.1140/epjds/s13688-022-00326-w.

[30] G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding behavioral patterns in truck co-driving networks." In: *Proceedings of the 7th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 813. Springer, 2018, pages 223–235. DOI: 10.1007/978-3-030-05414-4_18.

[31] G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Supervised temporal link prediction in large-scale real-world networks." *Social Network Analysis and Mining* 11, 80 (2021). DOI: 10.1007/s13278-021-00787-3.

[32] G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding dynamics of truck co-driving networks." In: *Proceedings of the 8th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 882. Springer, 2020, pages 140–151. DOI: 10.1007/978-3-030-36683-4_12.

[33] G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Experimental evaluation of train and test split strategies in link prediction." In: *Proceedings of the 9th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 994. Springer, 2021, pages 79–91. DOI: 10.1007/978-3-030-65351-4_7.

[34] E. Bütün, M. Kaya, and R. Alhajj. „A new topological metric for link prediction in directed, weighted and temporal networks." In: *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2016, pages 954–959. DOI: 10.1109/ASONAM.2016.7752355.

[35] E. Bütün, M. Kaya, and R. Alhajj. „Extension of neighbor-based link prediction methods for directed, weighted and temporal social networks." *Information Sciences* 463–464 (2018), pages 152–165. DOI: 10.1016/j.ins.2018.06.051.

[36] P. Cariou, M. Q. Meijia Jr, and F.-C. Wolff. „An econometric analysis of deficiencies noted in port state control inspections." *Maritime Policy & Management* 34.3 (2007), pages 243–258. DOI: 10.1080/03088830701343047.

[37] P. Cariou and F.-C. Wolff. „Do port state control inspections influence flag- and class-hopping phenomena in shipping?" *Journal of Transport Economics and Policy* 45.2 (2011), pages 155–177.

[38] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. „Time-varying graphs and dynamic networks." *International Journal of Parallel, Emergent and Distributed Systems* 27 (5 2012), pages 387–408. DOI: 10.1080/17445760.2012.668546.

[39] G. C. Cawley and N. L. C. Talbot. „On over-fitting in model selection and subsequent selection bias in performance evaluation." *The Journal of Machine Learning Research* 11 (2010), pages 2079–2107.

[40] H. Chen, X. Li, and Z. Huang. „Link prediction approach to collaborative filtering." In: *Proceedings of the 5th Joint Conference on Digital Libraries (JCDL)*. 2005, pages 141–142. DOI: 10.1145/1065385.1065415.

[41] T. Chen and C. Guestrin. „XGBoost: A scalable tree boosting system." In: *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD)*. 2016, pages 785–794. DOI: 10.1145/2939672.2939785.

[42] A. Criminisi, J. Shotton, and E. Konukoglu. „Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning." *Foundations and Trends in Computer Graphics and Vision* 7.2–3 (2012), pages 81–227. DOI: 10.1561/0600000035.

[43] B. R. da Cunha and S. Gonçalves. „Topology, robustness, and structural controllability of the Brazilian federal police criminal intelligence network." *Applied Network Science* 3, 36 (2018). DOI: 10.1007/s41109-018-0092-1.

[44] A. Cuttone, S. Lehmann, and M. C. González. „Understanding predictability and exploration in human mobility." *EPJ Data Science* 7, 2 (2018). DOI: 10.1140/epjds/s13688-017-0129-1.

[45] „Data in toezicht: Een pleidooi voor samenwerking." *Toezine* (Apr. 9, 2019). https://www.toezine.nl/data-in-toezicht-een-pleidooi-voor-samenwerking (visited on May 29, 2023).

[46] M. De Choudhury, H. Sundaram, A. John, and D. D. Seligmann. „Social synchrony: predicting mimicry of user actions in online social media." In: *Proceedings of the International Conference on Computational Science and Engineering (CSE)*. 2009, pages 151–158. DOI: 10.1109/CSE.2009.439.

[47] T. Degré. „The use of risk concept to characterize and select high risk vessels for ship inspections." *WMU Journal of Maritime Affairs* 6 (2007), pages 37–49. DOI: 10.1007/BF03195088.

[48] T. Degré. „From black-grey-white detention-based lists of flags to black-grey-white casualty-based lists of categories of vessels?" *Journal of Navigation* 61.3 (2008), pages 485–497. DOI: 10.1017/S0373463308004773.

[49] Y. Dhote, N. Mishra, and S. Sharma. „Survey and analysis of temporal link prediction in online social networks." In: *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2013, pages 1178–1183. DOI: 10.1109/ICACCI.2013.6637344.

[50] A. Divakaran and A. Mohan. „Temporal link prediction: A survey." *New Generation Computing* 38 (2020), pages 213–258. DOI: 10.1007/s00354-019-00065-z.

[51] P. S. Dodds, R. Muhamad, and D. J. Watts. „An experimental study of search in global social networks." *Science* 301 (5634 2003), pages 827–829. DOI: 10.1126/science.1081058.

[52] S. N. Dorogovtsev and J. F. F. Mendes. „Evolution of networks." *Advances in Physics* 51.4 (2002), pages 1079–1187. DOI: 10.1080/00018730110112519.

[53] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. „Preserving statistical validity in adaptive data analysis." In: *Proceedings of the 47th Annual Symposium on Theory of Computing (STOC)*. 2015, pages 117–126. DOI: 10.1145/2746539.2746580.

[54] J. E. van Engelen, H. D. Boekhout, and F. W. Takes. „Explainable and efficient link prediction in real-world network data." In: *Proceedings of the International Symposium on Intelligent Data Analysis (IDA)*. 2016, pages 295–307. DOI: 10.1007/978-3-319-46349-0_26.

[55] European Maritime Safety Agency. *Netherlands*. 2021. http://emsa.europa.eu/about/download/6820/2614/23.html (visited on Mar. 25, 2022).

[56] European Union. „Directive 2009/16/EC of the European Parliament and of the Council of 23 April 2009 on port State control." *Official Journal of the European Union* (Apr. 23, 2009). http://data.europa.eu/eli/dir/2009/16/oj (visited on Nov. 25, 2022).

[57] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. „Certifying and removing disparate impact." In: *Proceedings of the 21th International Conference on Knowledge Discovery and Data Mining (KDD)*. 2015, pages 259–268. DOI: 10.1145/2783258.2783311.

[58] S. Fortunato and D. Hric. „Community detection in networks: A user guide." *Physics Reports* 659 (2016), pages 1–44. DOI: 10.1016/j.physrep.2016.09.002.

[59] L. C. Freeman. „A set of measures of centrality based on betweenness." *Sociometry* 40.1 (1977), pages 35–41. DOI: 10.2307/3033543.

[60] L. C. Freeman. „Centrality in social networks conceptual clarification." *Social Networks* 1.3 (1978), pages 215–239.

[61] Z. Gao, G. Lu, M. Liu, and M. Cui. „A novel risk assessment system for port state control inspection." In: *Proceedings of the International Conference on Intelligence and Security Informatics (ISI)*. 2008, pages 242–244. DOI: 10.1109/ISI.2008.4565068.

[62] A. Ghasemian, H. Hosseinmardi, and A. Clauset. „Evaluating overfit and underfit in models of network community structure." *IEEE Transactions on Knowledge and Data Engineering* 32.9 (2019), pages 1722–1735. DOI: 10.1109/tkde.2019.2911585.

[63] A. Ghasemian, H. Hosseinmardi, A. Galstyan, E. M. Airoldi, and A. Clauset. „Stacking models for nearly optimal link prediction in complex networks." *Proceedings of the National Academy of Sciences* 117.38 (2019), pages 23393–23400. DOI: 10.1073/pnas.1914950117.

[64] M. Girvan and M. E. J. Newman. „Community structure in social and biological networks." *Proceedings of the National Academy of Sciences* 99 (2002), pages 7821–7826. DOI: 10.1073/pnas.122653799.

[65] V. Gómez, A. Kaltenbrunner, and V. López. „Statistical analysis of the social network and discussion threads in Slashdot." In: *Proceedings of the 17th International Conference on World Wide Web (WWW)*. 2008, pages 645–654. DOI: 10.1145/1367497.1367585.

[66] B. H. Good, Y.-A. de Montjoye, and A. Clauset. „Performance of modularity maximization in practical contexts." *Physical Review E* 81, 046106 (4 2010). DOI: 10.1103/PhysRevE.81.046106.

[67] A. Graziano, P. Cariou, F.-C. Wolff, M. Q. Mejia, and J.-U. Schröder-Hinrichs. „Port state control inspections in the European Union: Do inspector's number and background matter?" *Marine Policy* 88 (2018), pages 230–241. DOI: 10.1016/j.marpol.2017.11.031.

[68] J. Greene. „Amazon's cloud-computing outage on wednesday was triggered by effort to boost system's capacity." *Washington Post* (Nov. 28, 2020). https://www.washingtonpost.com/technology/2020/11/28/amazon-outage-explained/ (visited on Dec. 25, 2022).

[69] J. L. Gross, J. Yellen, and P. Zhang. *Handbook of graph theory*. 2nd edition. CRC Press, 2014.

[70] A. Grover and J. Leskovec. „node2vec: Scalable feature learning for networks." In: *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (KDD)*. 2016, pages 855–864. DOI: 10.1145/2939672.2939754.

[71] İ. Güneş, Ş. Gündüz-Öğüdücü, and Z. Çataltepe. „Link prediction using time series of neighborhood-based node similarity scores." *Data Mining and Knowledge Discovery* 30.1 (2016), pages 147–180. DOI: 10.1007/s10618-015-0407-0.

[72] A. Hagberg, P. Swart, and D. S Chult. *Exploring network structure, dynamics, and function using NetworkX*. U.S. Department of Energy, 2008. https://www.osti.gov/biblio/960616 (visited on Dec. 28, 2021).

[73] J. D. Hamilton. *Time series analysis*. Princeton University Press, 2020.

[74] W. L. Hamilton, R. Ying, and J. Leskovec. „Representation learning on graphs: Methods and applications." 2018. arXiv: 1709.05584 [cs.SI].

[75] M. Hardt, E. Price, and N. Srebro. „Equality of opportunity in supervised learning." In: *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS)*. Advances in Neural Information Processing Systems 29. 2016, pages 3315–3323.

[76] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. 2nd edition. Springer, 2009. DOI: 10.1007/b94608.

[77] C. Heij and S. Knapp. „Shipping inspections, detentions, and incidents: An empirical analysis of risk dimensions." *Maritime Policy & Management* 46.7 (2019), pages 866–883. DOI: 10.1080/03088839.2019.1647362.

[78] T. Hiraoka, N. Masuda, A. Li, and H.-H. Jo. „Modeling temporal networks with bursty activity patterns of nodes and links." *Physical Review Research* 2, 023073 (2020). DOI: 10.1103/PhysRevResearch.2.023073.

[79] T. Hogg and K. Lerman. „Social dynamics of Digg." *EPJ Data Science* 1, 5 (2012). DOI: 10.1140/epjds5.

[80] P. W. Holland and S. Leinhardt. „Transitivity in structural models of small groups." *Small Group Research* 2 (2 1971), pages 107–124. DOI: 10.1177/104649647100200201.

[81] P. Holme. „Modern temporal network theory: a colloquium." *The European Physical Journal B* 88, 234 (2015). DOI: 10.1140/epjb/e2015-60657-4.

[82] P. Holme. „Rare and everywhere: perspectives on scale-free networks." *Nature Communications* 10, 1016 (2019). DOI: 10.1038/s41467-019-09038-8.

[83] P. Holme and J. Saramäki. „Temporal networks." *Physics Reports* 519 (3 2012), pages 97–125. DOI: 10.1016/j.physrep.2012.03.001.

[84] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell. „What do we need to build explainable AI systems for the medical domain?" 2017. arXiv: 1712.09923 [cs.AI].

[85] „Homogeneous temporal activity patterns in a large online communication space." In: *Proceedings of the Workshop on Social Aspects of the Web (SAW)*. 2007. arXiv: 0708.1579 [cs.NI].

[86] „ILT verbetert datagedreven toezicht met datalab." *Toezine* (Sept. 26, 2017). https://www.toezine.nl/artikel/228/ilt-verbetert-datagedreven-toezicht-met-datalab/ (visited on Nov. 25, 2022).

[87] Inspectie Leefomgeving en Transport. *Jaarverslag ILT*. 2020. https://magazines.ilent.nl/jaarverslag-ilt/2020/01/index (visited on Mar. 24, 2022).

[88] Inspectie Leefomgeving en Transport. *ILT-brede risicoanalyse*. 2021. https://www.ilent.nl/documenten/rapporten/2021/09/21/ilt-brede-risicoanalyse-2021 (visited on Nov. 25, 2022).

[89] Inspectie Leefomgeving en Transport. *ILT-brede risicoanalyse*. 2022. https://www.ilent.nl/documenten/rapporten/2022/09/20/ilt-brede-risicoanalyse-2022 (visited on Nov. 25, 2022).

[90] Inspectie Leefomgeving en Transport. *Over de ILT*. 2022. https://www.ilent.nl/over-ilt (visited on Apr. 15, 2022).

[91] J. P. Ioannidis. „Why most published research findings are false." *PLoS Med* 2.8, 124 (2018). DOI: 10.1371/journal.pmed.0020124.

[92] M. Jacomy. „Epistemic clashes in network science: Mapping the tensions between idiographic and nomothetic subcultures." *Big data & Society* 7 (2 2020). DOI: 10.1177/2053951720949577.

[93] R. Jiang, A. Pacchiano, T. Stepleton, H. Jiang, and S. Chiappa. „Wasserstein fair classification." In: *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference (UAI)*. Proceedings of Machine Learning Research 115. 2020, pages 862–872.

[94] P. Kaluza, A. Kölzsch, M. T. Gastner, and B. Blasius. „The complex network of global cargo ship movements." *Journal of the Royal Society Interface* 7 (2010), pages 1093–1103. DOI: 10.1098/rsif.2009.0495.

[95] V. Kassarnig, E. Mones, A. Bjerre-Nielsen, P. Sapiezynski, D. Dreyer Lassen, and S. Lehmann. „Academic performance and behavioral patterns." *EPJ Data Science* 7, 10 (2018). DOI: 10.1140/epjds/s13688-018-0138-8.

[96] J. Kleinberg, S. Mullainathan, and M. Raghavan. „Inherent trade-offs in the fair determination of risk scores." 2016. arXiv: 1609.05807 [cs.LG].

[97] B. Klimt and Y. Yang. „The Enron corpus: A new dataset for email classification research." In: *Proceedings of the 15th European Conference on Machine Learning (ECML)*. Lecture Notes in Computer Science 3201. 2004, pages 217–226. DOI: 10.1007/978-3-540-30115-8_22.

[98] S. Knapp. „The econometrics of maritime safety." PhD thesis. Erasmus University Rotterdam, Jan. 25, 2007. http://hdl.handle.net/1765/7913 (visited on Dec. 28, 2021).

[99] S. Knapp and P. H. Franses. „Econometric analysis to differentiate effects of various ship safety inspections." *Marine Policy* 32.4 (2008), pages 653–662. DOI: j.marpol.2007.11.006.

[100] G. Kossinets and D. J. Watts. „Empirical analysis of an evolving social network." *Science* 311.5757 (2006), pages 88–90. DOI: 10.1126/science.1116869.

[101] A. Kumar, S. S. Singh, K. Singh, and B. Biswas. „Link prediction techniques, applications, and performance: A survey." *Physica A: Statistical Mechanics and its Applications* 553, 124289 (2020). DOI: 10.1016/j.physa.2020.124289.

[102] S. Kumar, W. L. Hamilton, J. Leskovec, and D. Jurafsky. „Community interaction and conflict on the web." In: *Proceedings of the World Wide Web Conference (WWW)*. 2018, pages 933–943. DOI: 10.1145/3178876.3186141.

[103] S. Kumar, F. Spezzano, V. S. Subrahmanian, and C. Faloutsos. „Edge weight prediction in weighted signed networks." In: *Proceedings of the 16th International Conference on Data Mining (ICDM)*. 2016, pages 221–230. DOI: 10.1109/ICDM.2016.175.

[104] J. Kunegis. „KONECT: The Koblenz network collection." In: *Proceedings of the 22nd International Conference on World Wide Web (WWW Companion)*. 2013, pages 1343–1350. DOI: 10.1145/2487788.2488173.

[105] R. Lambiotte, J.-C. Delvenne, and M. Barahona. „Random walks, Markov processes and the multiscale modular organization of complex networks." *IEEE Transactions on Network Science and Engineering* 1.2 (2014), pages 76–90. DOI: 10.1109/TNSE.2015.2391998.

[106] J. Leskovec, J. Kleinberg, and C. Faloutsos. „Graph evolution: densification and shrinking diameters." *ACM Transactions on Knowledge Discovery from Data* 1.1, 2 (2007). DOI: 10.1145/1217299.1217301.

[107] J. Leskovec and A. Krevl. *SNAP datasets: Stanford large network dataset collection*. June 2014.
http://snap.stanford.edu/data (visited on Dec. 28, 2021).

[108] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Majoney. „Statistical properties of community structure in large social and information networks." In: *Proceedings of the 17th International conference on World Wide Web (WWW)*. 2008, pages 695–704. DOI: 10.1145/1367497.1367591.

[109] M. Ley. „The DBLP computer science bibliography: Evolution, research issues, perspectives." In: *Proceedings of the 9th International Symposium on String Processing and Information Retrieval (SPIRE)*. Lecture Notes in Computer Science 2476. 2002, pages 1–10. DOI: 10.1007/3-540-45735-6_1.

[110] D. Liben-Nowell and J. Kleinberg. „The link-prediction problem for social networks." *Journal of the American Society for Information Science and Technology* 58.7 (2007), pages 1019–1031. DOI: 10.1002/asi.20591.

[111] R. Lichtenwalter and N. V. Chawla. „Link prediction: Fair and effective evaluation." In: *Proceedings of the 4th International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2012, pages 376–383. DOI: 10.1109/ASONAM.2012.68.

[112] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. „New perspectives and methods in link prediction." In: *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining (KDD)*. 2010, pages 243–252. DOI: 10.1145/1835804.1835837.

[113] C. Liu, J. Wang, and H. Zhang. „Spatial heterogeneity of ports in the global maritime network detected by weighted ego network analysis." *Maritime Policy & Management* 45.1 (2018), pages 89–104. DOI: 10.1080/03088839.2017.1345019.

[114] L. Lü and T. Zhou. „Link prediction in complex networks: A survey." *Physica A: Statistical Mechanics and its Applications* 390.6 (2011), pages 1150–1170. DOI: 10.1016/j.physa.2010.11.027.

[115] M. Marjan, N. Zaki, and E. A. Mohamed. „Link prediction in dynamic social networks: a literature review." In: *Proceedings of the 5th International Congress on Information Science and Technology (CIST)*. 2018, pages 200–207. DOI: 10.1109/CIST.2018.8596511.

[116] U. Mazureck and J. van Hattem. „Rewards for safe driving behavior: Influence on following distance and speed." *Transportation Research Record* 1980.1 (2006), pages 31–38. DOI: 10.1177/0361198106198000106.

[117] W. McKinney. „Data structures for statistical computing in Python." In: *Proceedings of the 9th Python in Science Conference (SciPy)*. 2010, pages 56–61.

[118] R. Michalski, S. Palus, and P. Kazienko. „Matching organizational structure and social network extracted from email communication." In: *Proceedings of the 14th International Conference on Business Information Systems (BIS)*. Lecture Notes in Business Information Processing 87. 2011. DOI: 10.1007/978-3-642-21863-7_17.

[119] S. Milgram. „The small world problem." *Psychology today* 2.1 (1967), pages 60–67.

[120] T. Miller. „Explanation in artificial intelligence: insights from the social sciences." *Aritificial Intelligence* 267 (2019), pages 1–38. DOI: 10.1016/j.artint.2018.07.007.

[121] Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. *Visie open overheid*. Sept. 2013. https://data.overheid.nl/sites/default/files/uploaded_files/visie-open-overheid kopie.pdf (visited on May 27, 2022).

[122] Ministerie van Justitie en Veiligheid. *Richtlijnen voor het toepassen van algoritmen door overheden en publieksvoorlichting over data-analyses.* Mar. 8, 2021. https://www.rijksoverheid.nl/documenten/richtlijnen/2021/09/24/richtlijnen-voor-het-toepassen-van-algoritmen-door-overheden-en-publieksvoorlichting-over-data-analyses (visited on May 28, 2022).

[123] C. Molnar. *Interpretable machine learning*. 2020. https://christophm.github.io/interpretable-ml-book (visited on Dec. 28, 2021).

[124] C. P. Muniz, R. Goldschmidt, and R. Choren. „Combining contextual, temporal and topological information for unsupervised link prediction in social networks." *Knowledge-Based Systems* 156 (2018), pages 129–137. DOI: 10.1016/j.knosys.2018.05.027.

[125] E. C. Mutlu, T. A. Oghaz, A. Rajabi, and I. Garibay. „Review on graph feature learning and feature extraction techniques for link prediction." *Machine Learning & Knowledge Extraction* 2.4 (2020), pages 672–704. DOI: 10.3390/make2040036.

[126] M. E. J. Newman. „Assortative mixing in networks.” *Physical Review Letters* 89, 208701 (2002). DOI: 10.1103/PhysRevLett.89.208701.

[127] M. E. J. Newman. „Mixing patterns in networks.” *Physical Review E* 67, 026126 (2 2003). DOI: 10.1103/PhysRevE.67.026126.

[128] M. E. J. Newman. „Modularity and community structure in networks.” *Proceedings of the National Academy of Sciences* 103.23 (2006), pages 8577–8582. DOI: 10.1073/pnas.0601602103.

[129] M. E. J. Newman. *Networks*. 2nd edition. Oxford University Press, 2018.

[130] M. E. J. Newman and J. Park. „Why social networks are different from other types of networks.” *Physical Review E* 68, 036122 (3 2003). DOI: 10.1103/PhysRevE.68.036122.

[131] NGO Shipbreaking Platform. *Flags of Convience*. 2019. https://shipbreakingplatform.org/issues-of-interest/focs/ (visited on Mar. 18, 2023).

[132] R. Noldus and P. van Mieghem. „Assortativity in complex networks.” 3 (4 2015), pages 507–542. DOI: 10.1093/comnet/cnv005.

[133] J. O'Madadhain, J. Hutchins, and P. Smyth. „Prediction and ranking algorithms for event-based network data.” *ACM SIGKDD Explorations Newsletter* 7.2 (2005), pages 23–30. DOI: 10.1145/1117454.1117458.

[134] A. Öczan and Ş. G. Öğüdücü. „Multivariate temporal link prediction in evolving social networks.” In: *Proceedings of the 14th International Conference on Computer and Information Science (ICIS)*. 2015, pages 185–190. DOI: 10.1109/ICIS.2015.7166591.

[135] A. Öczan and Ş. G. Öğüdücü. „Supervised temporal link prediction using time series of similarity measures.” In: *Proceedings of the 9th International Conference on Ubiquitous and Future Networks (ICUFN)*. 2017, pages 519–521. DOI: 10.1109/ICUFN.2017.7993838.

[136] OECD. *Regulatory enforcement and inspections, OECD best practice principles for regulatory policy*. 2014. DOI: 10.1787/23116013.

[137] OECD. *Data-driven, information-enabled regulatory delivery*. 2021. DOI: 10.1787/8f99ec8c-en.

[138] J. Olsen. *Data quality: The accuracy dimension*. 2003.

[139] T. Opsahl. „Triadic closure in two-mode networks: Redefining the global and local clustering coefficients.” *Social Networks* 35.2 (2013), pages 159–167. DOI: 10.1016/j.socnet.2011.07.001.

[140] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. „How many trees in a random forest?” In: *Proceedings of the 8th Machine Learning and Data Mining in Pattern Recognition (MLDM)*. Lecture Notes in Computer Science 7376. 2012, pages 154–168.

[141] G. Pallotta, M. Vespe, and K. Bryan. „Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction.” *Entropy* 15.6 (2013), pages 2218–2245. DOI: 10.3390/e15062218.

[142] L. Pappalardo, S. Rinzivillo, Z. Qu, D. Pedreschi, and F. Giannotti. „Understanding the patterns of car travel.” *The European Physical Journal Special Topics* 215.6 (2013), pages 61–73. DOI: 10.1140/epjst/e2013-01715-5.

[143] A. Paranjape, A. R. Benson, and J. Leskovec. „Motifs in temporal networks." In: *Proceedings of the 10th International Conference on Web Search and Data Mining (WSDM)*. 2017, pages 601–610. DOI: 10.1145/3018661.3018731.

[144] Pareshnah. *Zachary karate club social network*. Sept. 19, 2022. https://en.wikipedia.org/wiki/File:Zachary_karate_club_social_network.png (visited on Dec. 25, 2022).

[145] Paris Memorandum of Understanding. *Current flag performance list*. Memorandum Appendix. 2020. https://parismou.org/detentions-banning/white-grey-and-black-list (visited on Nov. 25, 2021).

[146] Paris Memorandum of Understanding. *Dealing with the pandemic*. Annual Report. 2020. https://www.parismou.org/sites/default/files/TBB_Jaarverslag Paris MoU 2020_totaal_HRlos.pdf (visited on Mar. 28, 2022).

[147] Paris Memorandum of Understanding. *Paris Memorandum of Understanding on Port State Control*. Oct. 2, 2020. https://parismou.org/sites/default/files/Paris MoU including 43rd amendment final.pdf (visited on Mar. 10, 2022).

[148] R. Pastor-Satorras and A. Vespignani. *Evolution and structure of the internet: a statistical physics approach*. Cambridge University Press, 2004.

[149] F. Pedregosa et al. „Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research* 12, 85 (2011), pages 2825–2830.

[150] L. Peel, T. P. Peixoto, and M. D. Domenico. „Statistical inference links data and theory in network science." *Nature Communications* 13, 6794 (2022). DOI: 10.1038/s41467-022-34267-9.

[151] P. Peng, S. Cheng, J. Chen, M. Liao, L. Wu, X. Liu, and F. Lu. „A fine-grained perspective on the robustness of global cargo ship transportation networks." *Journal of Geographical Sciences* 28 (2018), pages 881–889. DOI: 10.1007/s11442-018-1511-z.

[152] A. Pereira Barata. *Fair tree classifier using strong demographic parity*. Github repository. Nov. 16, 2021. https://github.com/pereirabarataap/fair_tree_classifier (visited on Dec. 30, 2021).

[153] A. Pereira Barata. „Reliable and fair machine learning for risk assessment." PhD thesis. Leiden University, Apr. 5, 2023. https://hdl.handle.net/1887/3590289 (visited on May 29, 2023).

[154] A. Pereira Barata, G. J. de Bruin, F. W. Takes, C. J. Veenman, and H. J. van den Herik. „Data-driven risk assessment in infrastructure networks." In: *ICT.open*. 2018. https://hdl.handle.net/1887/3283777 (visited on Apr. 29, 2022).

[155] A. Pereira Barata, G. J. de Bruin, F. W. Takes, C. J. Veenman, and H. J. van den Herik. „Finding anomalies in waste transportation data with supervised category models." In: *Proceedings of the 27th Belgian Dutch Conference on Machine Learning (BeNeLearn)*. Dec. 1, 2018. https://hdl.handle.net/1887/69186 (visited on Mar. 29, 2022).

[156] A. Pereira Barata, F. W. Takes, H. J. van den Herik, and C. J. Veenman. „Imputation methods outperform missing-indicator for data missing completely at random." In: *Proceedings of the 19th International Conference on Data Mining Workshops (ICDMW)*. 2019, pages 407–414. DOI: 10.1109/ICDMW.2019.00066.

[157] A. Pereira Barata, F. W. Takes, H. J. van den Herik, and C. J. Veenman. „Fair tree classifier using strong demographic parity." 2021. arXiv: 2110.09295 [cs.LG].

[158] B. Perozzi, R. Al-Rfou, and S. Skiena. „Deepwalk: online learning of social representations." In: *Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining (KDD)*. 2014, pages 701–710. DOI: 10.1145/2623330.2623732.

[159] N. Poor. „Mechanisms of an online public sphere: the website Slashdot." *Journal of Computer-Mediated Communication* 10, JCMC1028 (2 2005). DOI: 10.1111/j.1083-6101.2005.tb00241.x.

[160] Port of Rotterdam. *Facts & figures*. 2021. https://www.portofrotterdam.com/sites/default/files/2021-06/facts-and-figures-port-of-rotterdam.pdf (visited on Mar. 24, 2022).

[161] A. Potgieter, K. April, R. J. E. Cooke, and I. O. Osunmakinde. „Temporality in link prediction: understanding social complexity." *All Sprouts Content*, 195 (2008). https://aisel.aisnet.org/sprouts_all/195 (visited on Nov. 26, 2022).

[162] P. Probst, M. N. Wright, and A.-L. Boulesteix. „Hyperparameters and tuning strategies for random forest." *WIREs Data Mining and Knowledge Discovery* 9, e1301 (2019). DOI: 10.1002/widm.1301.

[163] U. Redmond and P. Cunningham. „A temporal network analysis reveals the unprofitability of arbitrage in the prosper marketplace." *Expert Systems with Applications* 40.9 (2013), pages 3715–3721. DOI: 10.1016/j.eswa.2012.12.077.

[164] Y. Ren, M. Ercsey-Ravasz, P. Wang, M. C. González, and Z. Toroczkai. „Predicting commuter flows in spatial networks using a radiation model based on temporal ranges." *Nature Communications* 5, 5347 (2014). DOI: 10.1038/ncomms6347.

[165] M. Richardson, R. Agrawal, and D. Pedro. „Trust management for the semantic web." In: *Proceedings of the 2nd International Semantic Web Conference (ISWC)*. Lecture Notes in Computer Science 2870. 2003. DOI: 10.1109/ICCEE.2009.241.

[166] E. Rodríguez and F. Piniella. „The new inspection regime of the Paris MoU on port state control: improvement of the system." *Journal of Maritime Research* 9.1 (2012), pages 9–16.

[167] R. Roelofs, J. Miller, M. Hardt, S. Fridovich-keil, L. Schmidt, and B. Recht. „A meta-analysis of overfitting in machine learning." In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*. Advances in Neural Information Processing Systems. 2019, 823, pages 9179–9189.

[168] M. Romero, J. Finke, C. Rocha, and L. Tobón. „Spectral evolution with approximated eigenvalue trajectories for link prediction." *Social Network Analysis and Mining* 10, 60 (2020). DOI: 10.1007/s13278-020-00674-3.

[169] H. Rosling. *Factfulness*. Sceptre, 2018.

[170] M. Saebi, J. Xu, L. M. Kaplan, B. Ribeiro, and N. V. Chawla. „Efficient modeling of higher-order dependencies in networks: From algorithm to application for anomaly detection." *EPJ Data Science* 9, 15 (2020). DOI: 10.1140/epjds/s13688-020-00233-y.

[171] P. Sapiezynski, A. Stopczynski, R. Gatej, and S. Lehmann. „Tracking human mobility using WiFi signals." *PloS ONE* 10.7, e0130824 (2015). DOI: 10.1371/journal.pone.0130824.

[172] I. Scholtes, N. Wider, and A. Garas. „Higher-order aggregate networks in the analysis of temporal networks: path structures and centralities." *The European Physical Journal B* 89, 61 (2016). DOI: 10.1140/epjb/e2016-60663-0.

[173] J. Scott. „Social network analysis." *Sociology* 22 (1 1988), pages 109–127. DOI: 10.1177/0038038588022001007.

[174] J. Scott. „Social network analysis: developments, advances, and prospects." *Social network analysis and mining* 1 (2011), pages 21–26. DOI: 10.1007/s13278-010-0012-6.

[175] V. Sekara, A. Stopczynski, and S. Lehmann. „Fundamental structures of dynamic social networks." *Proceedings of the National Academy of Sciences* 113.36 (2016), pages 9977–9982. DOI: 10.1073/pnas.1602803113.

[176] X. Q. Shuaian Wang Ran Yan. „Development of a non-parametric classifier: Effective identification, algorithm, and applications in port state control for maritime transportation." *Transportation Research Part B: Methodological* 128 (2019), pages 129–157. DOI: 10.1016/j.trb.2019.07.017.

[177] P. R. da Silva Soares and R. B. C. Prudencio. „Time series based link prediction." In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 2012, pages 1–7. DOI: 10.1109/IJCNN.2012.6252471.

[178] P. R. Soares and R. B. Prudêncio. „Proximity measures for link prediction based on temporal events." *Expert Systems with Applications* 40.16 (2013), pages 6652–6660. DOI: 10.1016/j.eswa.2013.06.016.

[179] A. Stopczynski, V. Sekara, P. Sapiezynski, A. Cuttone, M. M. Madsen, J. E. Larsen, and S. Lehmann. „Measuring large-scale social networks with high resolution." *PloS ONE* 9.4, e95978 (2014). DOI: 10.1371/journal.pone.0095978.

[180] M. Stopford. *Maritime economics*. Routledge, 2008.

[181] M. Strathern. „'Improving ratings': Audit in the British university system." *European Review* 5.3 (1997), pages 305–321.

[182] T. Strickx and J. Hartman. *Cloudflare outage on June 21, 2022*. June 21, 2022. https://blog.cloudflare.com/cloudflare-outage-on-june-21-2022/ (visited on Dec. 25, 2022).

[183] S. H. Strogatz. „Exploring complex networks." *Nature* 410 (2001), pages 268–276. DOI: 10.1038/35065725.

[184] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. 2nd edition. MIT Press, 2018.

[185] F. W. Takes and W. A. Kosters. „Determining the diameter of small world networks." In: *Proceedings of the 20th International Conference on Information and Knowledge Management (CIKM)*. 2011, pages 1191–1196. DOI: 10.1145/2063576.2063748.

[186] V. A. Traag, L. Waltman, and N. J. van Eck. „From Louvain to Leiden: Guaranteeing well-connected communities." *Scientific Reports* 9, 5233 (2019). DOI: 10.1038/s41598-019-41695-z.

[187] S. Tsugawa, S. Jeschke, and S. Shladovers. „A review of truck platooning projects for energy savings." *IEEE Transactions on Intelligent Vehicles* 1.1 (2016), pages 68–77. DOI: 10.1109/TIV.2016.2577499.

[188] S. Tsugawa and S. Kato. „Energy ITS: Another application of vehicular communications." *IEEE Communications Magazine* 48.11 (2010), pages 120–126. DOI: 10.1109/MCOM.2010.5621978.

[189] T. Tylenda, R. Angelova, and S. Bedathur. „Towards time-aware link prediction in evolving social networks." In: *Proceedings of the 3rd Workshop on Social Network Mining and Analysis (SNA-KDD)*. 2009, 9, pages 1–10. DOI: 10.1145/1731011.1731020.

[190] United Nations Conference on Trade and Development. *Review of maritime transport*. United Nations Publications, 2020. https://unctad.org/system/files/official-document/rmt2020_en.pdf (visited on Dec. 28, 2021).

[191] P. Van Mieghem, H. Wang, X. Ge, S. Tang, and F. A. Kuipers. „Influence of assortativity and degree-preserving rewiring on the spectra of networks." *European Physical Journal B* 76 (2010), pages 643–652. DOI: 10.1140/epjb/e2010-00219-x.

[192] N. van Veen. „The complex network of ship movements in Europe." Master's thesis. University of Amsterdam, July 2, 2020. https://www.gerritjandebruin.nl/attachments/nathalie.pdf (visited on Dec. 28, 2021).

[193] P. Virtanen et al. „Scipy 1.0: fundamental algorithms for scientific computing in Python." *Nature Methods* 17 (3 2020), pages 261–272. DOI: 10.1038/s41592-019-0686-2.

[194] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. „On the evolution of user interaction in Facebook." In: *Proceedings of the 2nd Workshop on Online Social Networks (WOSN)*. 2009, pages 37–42. DOI: 10.1145/1592665.1592675.

[195] I. Voitalov, P. van der Hoorn, R. van der Hofstad, and D. Kriuoukov. „Scale-free networks well done." *Physical Review Research* 1, 033034 (2019). DOI: 10.1103/PhysRevResearch.1.033034.

[196] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabási. „Human mobility, social ties, and link prediction." In: *Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining (KDD)*. 2011, pages 1100–1108. DOI: 10.1145/2020408.2020581.

[197] D. J. Watts and S. H. Strogatz. „Collective dynamics of 'small-world' networks." *Nature* 393 (1998), pages 440–442.

[198] Wetenschappelijke raad voor het regeringsbeleid. *Vertrouwen in burgers*. 88. May 22, 2012. https://www.wrr.nl/publicaties/rapporten/2012/05/22/vertrouwen-in-burgers (visited on May 27, 2022).

[199] Z. Whittaker. „A DNS outage just took down a large chunk of the internet." *TechCrunch* (July 22, 2021).
https://techcrunch.com/2021/07/22/a-dns-outage-just-took-down-a-good-chunk-of-the-internet/ (visited on Dec. 25, 2022).

[200] Wikileaks. *US democratic national committee leak*. July 22, 2016.
https://www.wikileaks.org/dnc-emails (visited on Dec. 28, 2021).

[201] Y. Xiao, G. Qi, M. Jin, K. F. Yuen, Z. Chen, and K. X. Li. „Efficiency of port state control inspection regimes: A comparative study." *Transport Policy* 106 (2021), pages 165–172. DOI: 10.1016/j.tranpol.2021.04.003.

[202] Y. Xiao, G. Wang, K.-C. Lin, G. Qi, and K. X. Li. „The effectiveness of the new inspection regime for port state control: Application of the Tokyo MoU." *Marine Policy* 115, 103857 (2020). DOI: 10.1016/j.marpol.2020.103857.

[203] J. Xu, T. L. Wickramarathne, and N. V. Chawla. „Representing higher-order dependencies in networks." *Science Advances* 2, e1600028 (5 2016). DOI: 10.1126/sciadv.1600028.

[204] R.-F. Xu, Q. Lu, K. X. Li, and H.-S. Zheng. „A risk assessment system for improving port state control inspection." In: *Proceedings of the 6th International Conference on Machine Learning and Cybernetics*. 2007, pages 818–823. DOI: 10.1109/ICMLC.2007.4370255.

[205] R. Xu, Q. Lu, K. Li, and W. Li. „Web mining for improving risk assessment in port state control inspection." In: *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*. 2007, pages 427–434. DOI: 10.1109/NLPKE.2007.4368066.

[206] R. Yan, S. Wang, and C. Peng. „An artificial intelligence model considering data imbalance for ship selection in port state control based on detention probabilities." *Journal of Computational Science* 48, 101257 (2021). DOI: 10.1016/j.jocs.2020.101257.

[207] R. Yan, S. Wang, and C. Peng. „Ship selection in port state control: Status and perspectives." *Maritime Policy & Management* 49 (4 2022), pages 600–615. DOI: 10.1080/03088839.2021.1889067.

[208] Y. Yang, N. V. Chawla, P. Basu, B. Prabhala, and T. La Porta. „Link prediction in human mobility networks." In: *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2013, pages 380–387. DOI: 10.1145/2492517.2492656.

[209] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla. „Evaluating link prediction methods." *Knowledge and Information Systems* 45.3 (2015), pages 751–782. DOI: 10.1007/s10115-014-0789-0.

[210] Z. Yang, Z. Yang, J. Yin, and Z. Qu. „A risk-based game model for rational inspections in port state control." *Transportation Research Part E: Logistics and Transportation Review* 118 (2018), pages 477–495. DOI: 10.1016/j.tre.2018.08.001.

[211] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich. „Local higher-order graph clustering." In: *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining (KDD)*. 2017, pages 555–564. DOI: 10.1145/3097983.3098069.

[212] P. Yu, Z. Wang, P. Wang, H. Yin, and J. Wang. „Dynamic evolution of shipping network based on hypergraph.” *Physica A: Statistical Mechanics and its Applications* 598, 127247 (2022). DOI: 10.1016/j.physa.2022.127247.

[213] W. W. Zachary. „An information flow model for conflict and fission in small groups.” *Journal of anthropological research* 33.4 (1977). DOI: 10.1086/jar.33.4.3629752.

[214] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi. „Fairness constraints: Mechanisms for fair classification.” In: *Proceedings of the 20th International Conference on Artifical Intelligence and Statistics*. Proceedings of Machine Learning Research 54. 2017, pages 962–970.

[215] H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun. „Influence maximization in dynamic social networks.” In: *Proceedings of the 13th International Conference on Data Mining (ICDM)*. 2013, pages 1313–1318. DOI: 10.1109/ICDM.2013.145.

# Summary

Inspectors are indispensable for monitoring essential regulations that protect the safe and clean transport of goods. However, finding all dangerous behavior with a limited number of inspectors and increasing personnel shortages is challenging. That is the reason inspectorates are looking for innovative methods to find dangerous behavior and improve compliance. We consider a data-driven approach to arrive at *smart* inspection. Smart inspection is performed when we assess compliance of vehicles in a (1) accurate, (2) automated, (3) fair, and (4) interpretable manner.

Models that assess vehicle compliance can be unintentionally biased against certain vehicle features. These vehicle features are divided into two classes, being *static* and *dynamic* features. Examples of static features are the type of vehicle, the size, the insurer, and the country of registration. A vehicle owner can change some of these features (such as insurer and country of registration) to influence an automated model without any actual reduction in the vehicle's dangerous behavior. Therefore, we choose to use dynamic features of the vehicles (such as the routes to be chosen), which say something about the *behavior* of a vehicle and its operator. We use *networks* to encode the vehicle's behavior, allowing us to model a particular part of the transport system as a whole. The *main problem* that the thesis aims to address is thus how network methods can leverage this behavioral data for the smart inspection of vehicles.

We start in Chapter 1 with establishing the context of smart vehicle inspection and the methods used to achieve it. As mentioned, we use behavioral data that we encode with the help of networks. Network science is a young multidisciplinary field of study in which much attention has been paid to the universal properties of networks. Many of those properties are also present in our temporal transport networks. A task often applied to temporal networks is link prediction, aiming at predicting *new* links between existing nodes in a temporal network. A temporal network is a network where the creation time

of edges is known. Link prediction is also a key aspect of our work; Chapters 2 to 4 relate to this task.

Previous research has shown a relationship between a network's structure and performance in a related task, being *missing* link prediction. Our work extended this line of research by applying it to the link prediction task in *temporal networks*. We are particularly interested in uncovering the relationship between network structure and model performance in link prediction.

In Chapter 2, we, therefore, analyze the link prediction task in 26 temporal networks. We do so using a machine-learned classification model fed with topological features. The model independently learns which pairs of nodes likely connect (and which do not). We mention four results obtained from experiments. *First*, we show that the performance of link prediction is higher when the temporal aspect is considered. *Second*, we find a relation between the *overall structure* of a network and the extent to which links can be predicted. In particular, the link prediction model performs well on networks exhibiting negative degree assortativity, i.e., networks wherein low-degree nodes primarily link to high-degree nodes (and vice versa). *Third*, we find that in a network with discrete events, we can improve link prediction performance further by adequately encoding discrete events. *Fourth*, we do not find any apparent performance differences between node-oriented and edge-oriented features except for networks from the information domain. Further research should reveal how this finding can be explained.

In machine learning on tabular data, it is common practice to validate and test model performance by applying the model to data that is *disjoint* and *independent* of the data used to train the model. However, independence cannot be guaranteed with relational data as they occur in networks. Specifically, it is a nontrivial task to estimate rather precisely the performance of link prediction models even when using adequate splits into train, validation, and test sets. In Chapter 3, we, therefore, compare two common approaches from the literature: (1) the *random* split, and (2) the *temporal* split. We compare the performances of these two approaches on the link prediction task and find that the random split gives overly optimistic results. The temporal split does give a more realistic indication of performances. Furthermore, our results prove robust for a wide selection of model parameters.

In the last three chapters, we explicitly focus on *smart vehicle inspection*. We start with *co-driving*, the activity where two trucks drive "together", i.e., pass by the same location and time. Investigating the co-driving behavior of trucks is important because it can positively impact the environment. As a case in point, co-driving may reduce aerodynamic drag and, therefore, may result in optimized fuel usage. We investigate how network structure and vehicle characteristics relate to co-driving behavior. As such, the main topic of Chapter 4 is the *truck co-driving network*. In this network, every node is a truck, and a link exists when two trucks are *systematically* co-driving. *Systematic co-driving* is when two trucks *frequently* drive together. Data for such a study were collected from 18,000,000

truck movements in the Netherlands. We have used insights gained by applying link prediction to this network to understand truck co-driving behavior. The model uses features that are categorized into (1) spatiotemporal, (2) topological, (3) node-, and (4) path-oriented features. We found that truck co-driving behavior is best encoded using *topological* features and, to a lesser extent, the *path-oriented* and *spatiotemporal* features. Our findings indicate that the dynamics of the co-driving network exhibit significant social network effects.

We also looked at its communities to better understand the truck co-driving network. A so-called community detection algorithm can use the structure of a network to arrive at a good partitioning into groups of densely connected nodes. In our specific case, however, we also have information on the truck (i.e., the network's nodes) that we use to arrive at a proper partitioning into communities. We investigated how node attributes can be exploited to automatically create a good partitioning of a co-driving network into communities.

In Chapter 5, we propose a new metric, *the average maximal community assortativity*, to better understand the structure of communities in a network using node attribute assortativity. More specifically, we propose to select solutions to the community detection problem that maximizes the average maximal community assortativity metric. A high assortativity for a particular feature then indicates a better community representation. In the case of the truck co-driving network, we observe that *geographical* node attributes especially characterize communities.

This thesis's final topic relates to smart vehicle inspection and network science. It concerns the question of how ship behavior can be utilized to enable smart inspection of cargo ships.

In Chapter 6, we provide such an approach to smart cargo ship inspection. We use a model that is *interpretable* and *fair*. The model cannot only use static administrative ship properties in its prediction but, in particular, utilizes features describing the ship's *behavior*. By incorporating ship behavior, meaningful characteristics can be derived and utilized as input for the model. It leads us to a smart risk assessment of cargo ships. Our approach allows inspectorates to trace specifically noncompliant cargo ships. Thereby, this chapter contributes to improved maritime safety and environmental protection.

In general, we demonstrate how network science and behavioral data can be utilized to arrive at a *smart inspection* of vehicles. With this explanation and interpretation of smart inspection, we are sure to have addressed the overall problem statement of the thesis.

# Samenvatting

Inspecteurs zijn onmisbaar voor toezicht op veilig en schoon goederenvervoer. Echter, het is moeilijk om al het gevaarlijk gedrag op te sporen met een beperkt aantal inspecteurs. De huidige personeelstekorten maken deze situatie nog actueler. Daarom zoeken inspectiediensten naar innovatieve methoden om gevaarlijk gedrag op te sporen en naleving te verbeteren. Wij onderzoeken een data-gedreven aanpak om te komen tot een *slimme* inspectie. Onder een slimme inspectie verstaan wij een (1) nauwkeurige, (2) geautomatiseerde, (3) onbevooroordeelde, en (4) verklaarbare manier om voertuigen te beoordelen op overtredingen.

Modellen, die gebruikt worden om voertuigen te beoordelen, kunnen onbedoeld vooringenomen zijn tegen bepaalde voertuigkenmerken. Deze kenmerken worden verdeeld in *statische* en *dynamische* kenmerken. Voorbeelden van statische kenmerken zijn voertuigtype, grootte, verzekeraar, en land van registratie. Sommige van deze kenmerken (zoals verzekeraar en land van registratie) kunnen door eigenaars veranderd worden om het model te beïnvloeden, zonder dat er daadwerkelijk vermindering plaatsvindt van ongewenst gedrag. Daarom gebruiken we dynamische kenmerken (zoals routes die genomen worden) die daadwerkelijk iets zeggen over het *gedrag* van een voertuig (en zijn exploitant) en moeilijk te vervalsen zijn. Om het voertuiggedrag te coderen gebruiken we *netwerken*. De *overkoepelende probleemstelling* van dit proefschrift is hoe netwerkmethoden deze gedragsgegevens kunnen benutten voor de slimme inspectie van voertuigen.

We starten Hoofdstuk 1 met het uiteenzetten van de context van slimme voertuig inspecties en de methoden waarmee dit bereikt wordt. Zoals genoemd, gebruiken we daarbij gedragsgegevens die we coderen met behulp van netwerken. De netwerkwetenschap is een jong en multidisciplinair onderzoeksgebied waar veel aandacht besteed wordt aan de universele eigenschappen van netwerken. Veel van deze eigenschappen zijn ook aanwezig in onze temporele transportnetwerken, d.w.z. netwerken waarbij van elke link bekend is wanneer deze ontstaan is. Een taak die vaak wordt uitgevoerd op temporele netwerken is de linkvoorspellingstaak, waarbij het doel is om *nieuwe* links te voorspellen tussen bestaande knopen. De linkvoorspellingstaak komt veel terug in ons werk, namelijk in Hoofdstukken 2 tot en met 4.

Uit eerder onderzoek is gebleken dat bij een verwante taak, het voorspellen van *missende* links, er een relatie bestaat tussen de structuur van het netwerk en de prestatie van de taak. In ons werk, hebben we deze onderzoekslijn uitgebreid naar de linkvoorspellingstaak op *temporele netwerken*. We zijn in het bijzonder geïnteresseerd om de relatie te begrijpen tussen (1) de netwerkstructuur en (2) de prestatie van de linkvoorspellingstaak.

In Hoofdstuk 2, analyseren wij de linkvoorspellingstaak op 26 temporele netwerken. We doen dit met behulp van een machinaal geleerd linkvoorspellingsmodel dat gevoed wordt met topologische kenmerken. Het model leert zelfstandig welke knopenparen waarschijnlijk verbinden (en welke niet). We noemen vier resultaten verkregen uit experimenten. *Ten eerste*, laten wij zien dat de prestatie van de linkvoorspellingstaak meer precies is wanneer er rekening gehouden wordt met het temporele aspect. *Ten tweede*, tonen wij aan dat er een relatie is tussen de *algehele structuur* van een netwerk en de mate waarin links voorspeld kunnen worden. In het bijzonder, valt op dat de prestatie van de linkvoorspellingstaak meer precies is in netwerken die een negatieve graadassortativiteit hebben, d.w.z. netwerken waarin lage-graad knopen voornamelijk linken met hoge-graad knopen (en vice versa). *Ten derde*, tonen we aan dat in netwerken met discrete gebeurtenissen, de prestatie van de linkvoorspellingstaak verhoogd kan worden bij het adequaat coderen van de discrete gebeurtenissen. *Ten vierde*, vinden wij geen duidelijke prestatieverschillen tussen knoop-georiënteerde en link-georiënteerde kenmerken. De enige uitzondering hierop lijkt te bestaan voor netwerken in het informatiedomein, maar verder onderzoek zal moeten uitwijzen hoe deze vondst kan worden verklaard.

Bij modellen die machinaal geleerd zijn op basis van tabelgegevens is het gebruikelijk dat de prestatie gevalideerd en getest wordt met gegevens die *apart gehouden zijn* en *onafhankelijk zijn* van de gegevens die gebruikt zijn om het model te leren. De onafhankelijkheid valt echter niet te waarborgen bij relationele gegevens zoals die geïmplementeerd zijn in netwerken. Meer in het bijzonder, is het niet-triviaal om schattingen te verkrijgen van de prestaties van linkvoorspellingsmodellen, zelfs niet wanneer gebruik gemaakt wordt van adequate splitsingen in *train*, *validatie*, en *test* sets. In Hoofdstuk 3 onderzoeken we daarom twee benaderingen om de splitsing uit te voeren: (1) de willekeurige splitsing en (2) de temporele splitsing. We vergelijken de prestaties van deze twee benaderingen

op de linkvoorspellingstaak en vinden dat de *willekeurige* splitsing te optimistische resultaten geeft. De *temporele* splitsing geeft een meer realistische indicatie van de prestaties en verder blijken onze resultaten robuust te zijn voor een brede selectie aan parameters.

In de laatste drie hoofdstukken, richten we ons expliciet op het doen van *slimme voertuiginspecties*. We starten met samenrijdgedrag, de activiteit waarbij twee vrachtwagens 'samen' rijden op dezelfde tijd en dezelfde plaats. Het onderzoek naar samenrijdgedrag van vrachtwagens is belangrijk omdat het een positieve bijdrage aan het klimaat kan leveren. Als voorbeeld noemen we dat samenrijden resulteert in verminderde luchtweerstand, wat dan kan resulteren in verminderd brandstofverbruik. We beginnen ons onderzoek met de vraag hoe netwerkstructuur en voertuigkenmerken relateren aan samenrijdgedrag. Als zodanig is het *samenrijden van vrachtwagens* het hoofdonderwerp in Hoofdstuk 4. In dat netwerk is elke knoop een vrachtwagen en bestaat er een link als twee vrachtwagens consequent samenrijden. *Systematisch* samenrijden gebeurt wanneer twee vrachtwagens frequent samenrijden. De gegevens voor deze studie zijn verzameld vanuit 18 miljoen vrachtwagenbewegingen in Nederland. We kunnen de inzichten van de linkvoorspellingstaak gebruiken om het gedrag van voertuigen te begrijpen. In dit model worden kenmerken gebruikt die we categoriseren in vier typen: (1) tijd-ruimtelijk, (2) topologisch, (3) knoop-, en (4) pad-georiënteerd. We hebben gevonden dat het samenrijdgedrag het best gecodeerd wordt met de topologische kenmerken en in mindere mate door de pad en tijd-ruimtelijke kenmerken. Onze bevindingen suggereren dat de dynamiek van het samenrijdnetwerk duidelijk sociale netwerk-effecten heeft.

Om het netwerk nog beter te begrijpen, kunnen we ook de *gemeenschappen* bestuderen die in het netwerk aanwezig zijn. Deze gemeenschappen worden gevonden met behulp van een zogenaamd gemeenschapsdetectie algoritme, die de structuur van het netwerk gebruikt om tot een goede partitionering in gemeenschappen te komen. Specifiek in ons geval, hebben we aanvullende kenmerken van de vrachtwagens beschikbaar, die we ook kunnen gebruiken om tot een adequate partitionering in gemeenschappen te komen. We onderzoeken hoe knoopkenmerken benut kunnen worden om automatisch tot een goede partitionering van het samenrijdnetwerk in gemeenschappen te komen.

In Hoofdstuk 5 hebben we een nieuwe maat voorgesteld, de *gemiddelde maximale gemeenschapsassortativiteit*, die helpt bij het begrijpen van de gemeenschapsstructuur in een netwerk door middel van de assortativiteit van knoopkenmerken. In het bijzonder stellen we voor om die splitsing in gemeenschappen te gebruiken waarbij de hoogste gemiddelde maximale gemeenschapsassortativiteit gevonden wordt. Een hogere assortativiteit voor een bepaald kenmerk geeft aan dat er hier een betere partitionering van die gemeenschap is gebruikt. In het geval van samenrijdende vrachtwagens, hebben we opgemerkt dat er vooral gemeenschappen bestaan die gekarakteriseerd worden door geografische kenmerken.

Het laatste onderwerp van deze thesis houdt verband met zowel *slimme voertuiginspecties* als *de netwerkwetenschap*. Het betreft de vraag hoe scheepsgedrag benut kan worden om slimme inspecties van vrachtschepen mogelijk te maken.

In Hoofdstuk 6 komen we tot een dergelijke aanpak voor het slim inspecteren van vrachtschepen. We maken gebruik van een *interpreteerbaar* en *onbevooroordeeld* machinaal geleerd model. Het model is niet alleen in staat om *statische* administratieve eigenschappen van schepen te gebruiken in de risicovoorspelling maar ook specifiek het *gedrag* van schepen. Het gebruik van gedragsgegevens stelt ons vervolgens in staat om betekenisvolle kenmerken van schepen af te leiden. Daarna kan het resultaat gebruikt worden om tot een *eerlijke* en *betere risico-inschatting* van schepen te komen. Dit betekent dat onze benadering inspectiediensten in staat stelt om specifiek de overtredende vrachtschepen op te sporen. Daarmee draagt dit hoofdstuk bij aan *maritieme veiligheid* en *milieubescherming*.

In meer algemene zin, hebben we laten zien hoe netwerkwetenschappen en gedragsgegevens benut kunnen worden om te komen tot een *slimme inspectie* van voertuigen. Daarmee voorzien we in een antwoord op de overkoepelende probleemstelling van dit proefschrift.

# Curriculum Vitae

Gerrit Jan de Bruin was born in Amersfoort, the Netherlands, on the 31st of October 1993. He completed high school at the gymnasium level with bilingual education at "Van Lodenstein College" in 2012. Gerrit Jan was also an active mystery shopper and IT worker during and after high school.

Afterward, he obtained the VNCI Topsector Chemie grant to study a B.Sc. in chemistry at Utrecht University. During his B.Sc., he got married to his wife, Mirjam. He graduated in 2015.

Gerrit Jan was particularly interested in the measurement aspect of chemistry. He therefore acquired his M.Sc. in Analytical Chemistry at both the Free University Amsterdam and the University of Amsterdam in 2017. His Master's thesis, titled "Efficient compliance monitoring", was supervised by Johan Westerhuis and Jasper van Vliet and conducted at the Human Environment and Transport Inspectorate.

After that, he was admitted to Leiden University, the Netherlands, to conduct the research presented in this doctoral thesis, funded by the Dutch Ministry of Infrastructure and Water Management. Frank Takes, Cor Veenman, and Jaap van den Herik supervised him.

At the moment, Gerrit Jan is a Scientist in Air Quality and Climate at TNO, an independent not-for-profit research organization, working towards reducing emissions of air pollutants and greenhouse gases.

# Publications

While working towards this thesis, the following contributions were made.

- G. J. de Bruin, A. Pereira Barata, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Fair automated assessment of non-compliance in cargo ship networks." *EPJ Data Science* 11, 13 (2022). DOI: 10.1140/epjds/s13688-022-00326-w
- G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Supervised temporal link prediction in large-scale real-world networks." *Social Network Analysis and Mining* 11, 80 (2021). DOI: 10.1007/s13278-021-00787-3
- G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Experimental evaluation of train and test split strategies in link prediction." In: *Proceedings of the 9th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 994. Springer, 2021, pages 79–91. DOI: 10.1007/978-3-030-65351-4_7
- G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding dynamics of truck co-driving networks." In: *Proceedings of the 8th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 882. Springer, 2020, pages 140–151. DOI: 10.1007/978-3-030-36683-4_12
- G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Understanding behavioral patterns in truck co-driving networks." In: *Proceedings of the 7th International Conference on Complex Networks and Their Applications*. Studies in Computational Intelligence 813. Springer, 2018, pages 223–235. DOI: 10.1007/978-3-030-05414-4_18
- A. Pereira Barata, G. J. de Bruin, F. W. Takes, C. J. Veenman, and H. J. van den Herik. „Finding anomalies in waste transportation data with supervised category models." In: *Proceedings of the 27th Belgian Dutch Conference on Machine Learning (BeNeLearn)*. Dec. 1, 2018.
  https://hdl.handle.net/1887/69186 (visited on Mar. 29, 2022)

- A. Pereira Barata, G. J. de Bruin, F. W. Takes, C. J. Veenman, and H. J. van den Herik. „Data-driven risk assessment in infrastructure networks.” In: *ICT.open*. 2018. https://hdl.handle.net/1887/3283777 (visited on Apr. 29, 2022)

# Acknowledgments

I could not have completed this thesis without the help of others.

I thank Frank Takes for bringing me into the science of networks and the art of writing scientific texts. I was a slow learner, and many rounds of comments and revisions were sent on Overleaf. Despite this, you always made an effort to take another look at a draft, although your responsibilities in LIACS expanded considerably during the years. Also, I have rarely seen anyone dealing so swiftly with difficult situations in the project like you.

I want to thank Cor Veenman for his critical thoughts on any writing I provided. It was always good when we ran into each other at unexpected places. Likely this still remains so, as it so happens that we remain colleagues.

Jaap van den Herik, you have a unique way of supervising your Ph.D. candidates. Thank you for your insightful comments on both writing and life as a researcher in general. Also, probably not known to the outsider, Jaap has a remarkable eye for detail. Every error was caught, whether it was an extra space, a misspelling, or an outdated link in a reference. In extension, I want to thank Tim van Tongeren, Jan van den Bos, Mattheus Wassenaar, and Jaap van den Herik for making this journey possible in the first place.

Jasper, thank you for this possibility. My endeavor at the Ministry would not have been started without you. We have been through quite some situations in the team, but you always remained enthusiastic, and your ideas seem to be endless.

António and Alexa, I owe you much. To start, I really enjoy the way you cook. Your dishes are always so delicious and flavorful. Also, to the confusion of other guests, meals do not have a clear start or end time at your place. You eat steak for lunch when you want. On a more serious note, António and I did our Ph.D. in a quite what one could call “anti-correlated” manner; when I no longer saw the project end well, you would

encourage me (and probably vice versa), helping each other through the endless journey a Ph.D. is.

Alexandra, Andrius, Anna, Anne, Carolina, Charles, Dominika, Furong, Hanjo, Hugo, Iris, Jan, Marios, Marie, Matthias, Max, Michiel, Mike, Lieuwe, Lise, Rachel, Rens, Riccardo, Roy, Sander, Solomiia, Thomas, Quinten, Xue, Yuchen, and many more, thank you for the good times at LIACS. Because of the growing tendency of LIACS, it was hard to keep track of all people I met, but I sure liked the cultural diversity in our group. I thank each of you for all the moments we had together. Thomas, for the open doors of "your" office when I was finalizing this thesis.

Arjan, Bernard, Christoph, Corline, Dirk Jan, Gerben, Harm, Jacques, Jasper, Jelmar, Lucy, Karima, Kees, Maarten, Mara, Margje, Marius, Noortje, Paul, Piet, Robert, Shpat, Solomiia, Stephanie, Tjitse, Tony, and Victor, thanks to you as well for the moments together. The company of each of you, also during Corona times, was pleasant. I liked your efforts to make the Netherlands a more pleasant place to be.

Heleen, Koen, Michael, and Nathalie, thanks for you being part of my research. I hope you are doing well.

My colleagues from TNO, thank you for your patience and the opportunity to be distracted while putting the last hand at my thesis. I like your can-do mentality and your commitment to make the world a little better.

I want to thank my korfball team for their support throughout my Ph.D. and for providing some physical exercise. Thanks also for the many dinners and evenings we shared.

Daniël, Deliana, Hannah, and Johan, you belong to the best friends I have. We connect at many levels, and without the support and kind words, this journey to the Ph.D. was way more challenging. In extension, I want to thank all the other people I count as my friends.

Last but one, I want to thank my family (+ in-law). Regardless of how annoyed I was with my difficulties, you always supported me. I especially want to thank my parents, who always supported my curiosity.

Last, Mirjam. I owe you so much. Too many times, I struggled hard, especially mentally. You supported me in thinking positively. Your love has always been so much for me, without conditions, and that was what I needed. Without you, this would not have happened.

# SIKS Dissertation Series

2011 01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models

02 Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems

04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference

05 Bas van der Raadt (VU), Enterprise Architecture Coming of Age — Increasing the Performance of an Emerging Discipline.

06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage

07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction

08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues

09 Tim de Jong (OU), Contextualised Mobile Media for Learning

10 Bart Bogaert (UvT), Cloud Content Contention

11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective

12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining

13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling

14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets

15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval
16 Maarten Schadd (UM), Selective Search in Games of Different Complexity
17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness
18 Mark Ponsen (UM), Strategic Decision-Making in complex games
19 Ellen Rusman (OU), The Mind's Eye on Personal Profiles
20 Qing Gu (VU), Guiding Service-oriented Software Engineering — A view-based approach
21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
22 Junte Zhang (UVA), System Evaluation of Archival Description and Access
23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans on Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics
26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication — Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns
28 Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
29 Faisal Kamiran (TUE), Discrimination-aware Classification
30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
33 Tom van der Weide (UU), Arguing to Motivate Decisions
34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training
36 Erik van der Spek (UU), Experiments in serious game design: A cognitive approach

37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization
39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games
40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development
41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control
42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution
43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge
44 Boris Reuderink (UT), Robust Brain-computer Interfaces
45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection
46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
47 Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression
48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent
49 Andreea Niculescu (UT), Conversational Interfaces for Task-oriented Spoken Dialogues: Design Aspects Influencing Interaction Quality

2012 01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda
02 Muhammad Umair (VU), Adaptivity, Emotion, and Rationality in Human and Ambient Agent Models
03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories
04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications
05 Marijn Plomp (UU), Maturing Interorganisational Information Systems
06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks
07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories
09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-aware Service Platforms
10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment

11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
13 Suleman Shahid (UvT), Fun and Face: Exploring Non-verbal Expressions of Emotion during Playful Interactions
14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.
16 Fiemke Both (VU), Helping People by Understanding Them — Ambient Agents Supporting Task Execution and Depression Treatment
17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
18 Eltjo Poort (VU), Improving Solution Architecting Practices
19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval
22 Thijs Vis (UvT), Intelligence, Politie en Veiligheidsdienst: Verenigbare Grootheden?
23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
25 Silja Eckartz (UT), Managing the Business Case Development in Inter-organizational IT Projects: A Methodology and its Application
26 Emile de Maat (UVA), Making Sense of Legal Text
27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
28 Nancy Pascall (UvT), Engendering Technology Empowering Women
29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
30 Alina Pommeranz (TUD), Designing Human-centered Systems for Reflective Decision Making
31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
32 Wietske Visser (TUD), Qualitative Multi-criteria Preference Representation and Reasoning

33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
34 Pavol Jancura (RUN), Evolutionary Analysis in PPI Networks and Applications
35 Evert Haasdijk (VU), Never Too Old to Learn — On-line Evolution of Controllers in Swarm- and Modular Robotics
36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation
38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms
39 Hassan Fatemi (UT), Risk-aware Design of Value and Coordination Networks
40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia
41 Sebastian Kelle (OU), Game Design Patterns for Learning
42 Dominique Verpoorten (OU), Reflection Amplifiers in Self-regulated Learning
43 *Withdrawn*
44 Anna Tordai (VU), On Combining Alignment Techniques
45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions
46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation
47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior
48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data
49 Michael Kaisers (UM), Learning against Learning — Evolutionary Dynamics of Reinforcement Learning Algorithms in Strategic Interactions
50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering
51 Jeroen de Jong (TUD), Heuristics in Dynamic Scheduling; a Practical Framework with a Case Study in Elevator Dispatching

---

2013 01 Viorel Milea (EUR), News Analytics for Financial Decision Support
02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics
04 Chetan Yadati (TUD), Coordinating Autonomous Planning and Scheduling
05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns
06 Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
07 Giel van Lankveld (UvT), Quantifying Individual Player Differences

08 Robbert-Jan Merk (VU), Making Enemies: Cognitive Modeling for Opponent Agents in Fighter Pilot Simulators
09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications
10 Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.
11 Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services
12 Marian Razavian (VU), Knowledge-driven Migration to Services
13 Mohammad Safiri (UT), Service Tailoring: User-centric Creation of Integrated IT-based Homecare Services to Support Independent Living of Elderly
14 Jafar Tanha (UVA), Ensemble Approaches to Semi-supervised Learning Learning
15 Daniel Hennes (UM), Multiagent Learning — Dynamic Games and Applications
16 Eric Kok (UU), Exploring the Practical Benefits of Argumentation in Multi-agent Deliberation
17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid
18 Jeroen Janssens (UvT), Outlier Selection and One-class Classification
19 Renze Steenhuizen (TUD), Coordinated Multi-agent Planning and Scheduling
20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval
21 Sander Wubben (UvT), Text-to-text Generation by Monolingual Machine Translation
22 Tom Claassen (RUN), Causal Discovery and Logic
23 Patricio de Alencar Silva (UvT), Value Activity Monitoring
24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning
25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A New Way of Representing and Implementing Clinical Guidelines in a Decision Support System
26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning
27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance
28 Frans van der Sluis (UT), When Complexity Cecomes Interesting: An Inquiry into the Information eXperience
29 Iwan de Kok (UT), Listening Heads

30 Joyce Nakatumba (TUE), Resource-aware Business Process Management: Analysis and Support
31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications
32 Kamakshi Rajagopal (OUN), Networking For Learning; The Role of Networking in a Lifelong Learner's Professional Development
33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere
34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search
35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction
36 Than Lam Hoang (TUe), Pattern Mining in Data Streams
37 Dirk Börner (OUN), Ambient Learning Displays
38 Eelco den Heijer (VU), Autonomous Evolutionary Art
39 Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems
40 Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games
41 Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning
42 Léon Planken (TUD), Algorithms for Simple Temporal Reasoning
43 Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

---

2014 01 Nicola Barile (UU), Studies in Learning Monotone Models from Data
02 Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method
03 Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions
04 Hanna Jochmann-Mannak (UT), Websites for Children: Search Strategies and Interface Design — Three Studies on Children's Search Performance and Evaluation
05 Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability
06 Damian Tamburri (VU), Supporting Networked Software Development
07 Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior
08 Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints
09 Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language
10 Ivan Salvador Razo Zapata (VU), Service Value Networks
11 Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support

12 Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control
13 Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains
14 Yangyang Shi (TUD), Language Models With Meta-information
15 Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare
16 Krystyna Milian (VU), Supporting Trial Recruitment and Design by Automatically Interpreting Eligibility Criteria
17 Kathrin Dentler (VU), Computing Healthcare Quality Indicators Automatically: Secondary Use of Patient Data and Semantic Interoperability
18 Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations
19 Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support
20 Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link
21 Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments
22 Marieke Peeters (UU), Personalized Educational Games — Developing Agent-supported Scenario-based Training
23 Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era
24 Davide Ceolin (VU), Trusting Semi-structured Web Data
25 Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction
26 Tim Baarslag (TUD), What to Bid and When to Stop
27 Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
28 Anna Chmielowiec (VU), Decentralized $k$-Clique Matching
29 Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software
30 Peter de Cock (UvT), Anticipating Criminal Behaviour
31 Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support
32 Naser Ayat (UvA), On Entity Resolution in Probabilistic Data
33 Tesfa Tegegne (RUN), Service Discovery in eHealth
34 Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems
35 Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach

36 Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models
37 Maral Dadvar (UT), Experts and Machines United Against Cyberbullying
38 Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing
39 Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital
40 Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education
41 Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text
42 Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models
43 Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments
44 Paulien Meesters (UvT), Intelligent Blauw. Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden
45 Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach
46 Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity
47 Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

---

2015 01 Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response
02 Faiza Bukhsh (UvT), Smart Auditing: Innovative Compliance Checking in Customs Controls
03 Twan van Laarhoven (RUN), Machine learning for Network Data
04 Howard Spoelstra (OUN), Collaborations in Open Learning Environments
05 Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding
06 Farideh Heidari (TUD), Business Process Quality Computation — Computing Non-functional Requirements to Improve Business Processes
07 Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis
08 Jie Jiang (TUD), Organizational Compliance: An Agent-based Model for Designing and Evaluating Organizational Interactions
09 Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems
10 Henry Hermans (OUN), OpenU: Design of an Integrated System to Support Lifelong Learning
11 Yongming Luo (TUE), Designing Algorithms for Big Graph Datasets: A Study of Computing Bisimulation and Joins
12 Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks

13 Giuseppe Procaccianti (VU), Energy-efficient Software
14 Bart van Straalen (UT), A Cognitive Approach to Modeling Bad News Conversations
15 Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation
16 Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-robot Teamwork
17 André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18 Holger Pirk (CWI), Waste Not, Want Not! — Managing Relational Data in Asymmetric Memories
19 Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners
20 Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
21 Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
22 Zhemin Zhu (UT), Co-occurrence Rate Networks
23 Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
24 Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
25 Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
26 Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
27 Sándor Héman (CWI), Updating compressed colomn stores
28 Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
29 Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-player and Two-player Domains
30 Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning
31 Yakup Koç (TUD), On the robustness of Power Grids
32 Jerome Gard (LEI), Corporate Venture Management in SMEs
33 Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources
34 Victor de Graaf (UT), Gesocial Recommender Systems
35 Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing Medication Reviews through Decision Support: Prescribing a Better Pill to Swallow

03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VU), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust Scheduling in an Uncertain Environment
07 Jeroen de Man (VU), Measuring and Modeling Negative Emotions for Virtual Training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UVA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-agent Systems
13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa — An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and Data Model Independent Approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media

28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation — A Study on Epidemic Prediction and Control
29 Nicolas Höning (TUD), Peak Reduction in Decentralised Electricity Systems — Markets and Prices for Flexible Planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web Content Monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, Exercises in Learning from Just One Example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The Design of Nonverbal Interaction Behavior Optimized for Robot-specific Morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action — a Conceptual and Computational Inquiry
38 Andrea Minuto (UT), Materials that Matter — Smart Materials Meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance
42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (LEI), Real-time foresight — Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

2017 01 Jan-Jaap Oerlemans (LEI), Investigating Cybercrime
02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UVA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VU), Detecting Interesting Differences: Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling Patterns of Time and Emotion in Twitter #anticipointment
12 Sander Leemans (TUE), Robust Process Mining with Guarantees
13 Gijs Huisman (UT), Social Touch Technology — Extending the Reach of Social Touch through Haptic Technology
14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction
16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (LEI), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UVA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VU), Logics for causal inference under Uncertainty
23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with Applications to Multimorbidity Analysis and Literature Search

26 Merel Jung (UT), Socially Intelligent Robots that Understand and Respond to Human Touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VU), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (UvT), From IT-Business Strategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT
30 Wilma Latuny (UvT), The Power of Facial Expressions
31 Ben Ruijl (LEI), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VU), Interpreting Natural Science Spreadsheets
36 Yuanhao Guo (LEI), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting Properties of the Human Auditory System and Compressive Sensing Methods to Increase Noise Robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support for Applications in Human-aware Support Systems
41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42 Elena Sokolova (RUN), Causal discovery from Mixed and Missing Data with Applications on ADHD datasets
43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
44 Garm Lucassen (UU), Understanding User Stories — Computational Linguistics in Agile Requirements Engineering
45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement
46 Jan Schneider (OU), Sensor-based Learning Support
47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration
48 Angel Suarez (OU), Collaborative Inquiry-based Learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations
02 Felix Mannhardt (TUE), Multi-perspective Process Mining

03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-aware Applications, and Behavior Prediction
04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
07 Jieting Luo (UU), A Formal Account of Opportunism in Multi-agent Systems
08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems
09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10 Julienka Mollee (VUA), Moving Forward: Supporting Physical Activity Behavior Change through Intelligent Technology
11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
12 Xixi Lu (TUE), Using Behavioral Context in Process Mining
13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker Discovery in Heart Failure
16 Jaebok Kim (UT), Automatic Recognition of Engagement and Emotion in a Group of Children
17 Jianpeng Zhang (TUE), On Graph Sample Clustering
18 Henriette Nakad (LEI), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VUA), Emergent Relational Schemas for RDF
20 Manxia Liu (RUN), Time and Bayesian Networks
21 Aad Slootmaker (OUN), EMERGO: A Generic Platform for Authoring and Playing Scenario-based Serious Games
22 Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-made: Motivational Messages for Behavior Change Technology
27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis

28 Christian Willemse (UT), Social Touch Technologies: How They Feel and how They Make You Feel
29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech
30 Wouter Beek, The "K" in "Semantic Web" Stands for "Knowledge": Scaling Semantics to the Web

2019 01 Rob van Eijk (LEI), Web Privacy Measurement in Real-time Bidding Systems. A Graph-based Approach to RTB System Classification
02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding Stable Causal Structures from Clinical Data
05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for Energy Efficiency in Software Systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VU), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN), Extracting Actionable Information from Microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering Technically Augmented Human Collective Intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

22 Martin van den Berg (VU), Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement — Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic Factors in (Automatic) Image Description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OUN), The Gamification Design Process Applied to (Massive) Open Online Courses
28 Esther Kuindersma (LEI), Cleared for Take-off: Game-based Learning to Prepare Airline Pilots for Critical Situations
29 Daniel Formolo (VU), Using Virtual Agents for Simulation and Training of Social Skills in Safety-critical Circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OUN), Gamification with Digital Badges in Learning Programming
36 Kevin Ackermans (OUN), Designing Video-enhanced Rubrics to Master Complex Skills
37 Jian Fang (TUD), Database Acceleration on FPGAs
38 Akos Kadar (OUN), Learning Visually Grounded and Multilingual Representations

---

2020 01 Armon Toubman (LEI), Calculated Moves: Generating Air Combat Behaviour
02 Marcos de Paula Bueno (LEI), Unraveling Temporal Processes using Probabilistic Graphical Models
03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
04 Maarten van Gompel (RUN), Context as Linguistic Bridges
05 Yulong Pei (TUE), On Local and Global Structure Mining
06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation — An Approach and Tool Support

07 Wim van der Vegt (OUN), Towards a Software Architecture for Reusable Game Components
08 Ali Mirsoleimani (LEI), Structured Parallel Programming for Monte Carlo Tree Search
09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
10 Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining
11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data Augmentation Methods for Long-Tail Entity Recognition Models
12 Ward van Breda (VU), Predictive Modeling in E-mental Health: Exploring Applicability in Personalised Depression Treatment
13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
15 Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling
17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
20 Albert Hankel (VU), Embedding Green ICT Maturity in Organisations
21 Karine da Silva Miras de Araujo (VU), Where is the Robot? Life as it Could be
22 Maryam Masoud Khamis (RUN), Understanding Complex Systems Implementation through a Modeling Approach: The Case of E-government in Zanzibar
23 Rianne Conijn (UT), The Keys to Writing: A Writing Analytics Approach to Studying Writing Processes using Keystroke Logging
24 Lenin da Nobrega Medeiros (VUA/RUN), How are you Feeling, Human? Towards Emotionally Supportive Chatbots
25 Xin Du (TUE), The Uncertainty in Exceptional Model Mining
26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
27 Ekaterina Muravyeva (TUD), Personal Data and Informed consent in an Educational Context
28 Bibeg Limbu (TUD), Multimodal Interaction for Deliberate Practice: Training Complex Skills with Augmented Reality
29 Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference

30 Bob Zadok Blok (LEI), Creatief, Creatiever, Creatiefst
31 Gongjin Lan (VU), Learning better — From Baby to Better
32 Jason Rhuggenaath (TUE), Revenue Management in Online Markets: Pricing and Online Advertising
33 Rick Gilsing (TUE), Supporting Service-dominant Business Model Evaluation in the Context of Business Model Innovation
34 Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
35 Siamak Farshidi (UU), Multi-Criteria Decision-making in Software Production

---

2021 01 Francisco Xavier Dos Santos Fonseca (TUD), Location-based Games for Social Interaction in Public Space
02 Rijk Mercuur (TUD), Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models
03 Seyyed Hadi Hashemi (UVA), Modeling Users Interacting with Smart Devices
04 Ioana Jivet (OU), The Dashboard that Loved Me: Designing Adaptive Learning Analytics for Self-regulated Learning
05 Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems
06 Daniel Davison (UT), "Hey robot, what do you think?" How Children Learn with a Social Robot
07 Armel Lefebvre (UU), Research Data Management for Open Science
08 Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking
09 Cristina Zaga (UT), The Design of Robothings. Non-anthropomorphic and Non-verbal Robots to Promote Children's Collaboration through Play
10 Quinten Meertens (UvA), Misclassification Bias in Statistical Learning
11 Anne van Rossum (LEI), Nonparametric Bayesian Methods in Robotic Vision
12 Lei Pi (LEI), External Knowledge Absorption in Chinese SMEs
13 Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning
14 Negin Samaeemofrad (LEI), Business Incubators: The Impact of their Support
15 Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-configurations through the Business Services Paradigm
16 Esam A. H. Ghaleb (UM), Bimodal Emotion Recognition from Audio-visual cues
17 Dario Dotti (UM), Human Behavior Understanding from Motion and Bodily Cues using Deep Neural Networks

18 Remi Wieten (UU), Bridging the Gap between Informal Sense-Making Tools and Formal Systems — Facilitating the Construction of Bayesian Networks and Argumentation Frameworks
19 Roberto Verdecchia (VU), Architectural Technical Debt: Identification and Management
20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-sided Exposure Bias in Recommender Systems
21 Pedro Thiago Timbó Holanda (CWI), Progressive Indexes
22 Sihang Qiu (TUD), Conversational Crowdsourcing
23 Hugo Manuel Proença (LEI), Robust Rules for Prediction and Description
24 Kaijie Zhu (TUE), On Efficient Temporal Subgraph Query Processing
25 Eoin Martino Grua (VUA), The Future of E-health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-health Mobile Applications
26 Benno Kruit (CWI & VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables
27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You
28 Christoph Selig (LEI), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

2022 01 Judith van Stegeren (UT), Flavor text generation for role-playing video games
02 Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey
03 Ali el Hassouni (VUA), A Model a Day Keeps the Doctor Away: Reinforcement Learning for Personalized Healthcare
04 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework
05 Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization
06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding
07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics
08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design
09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach
10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines
11 Mirjam de Haas (UT), Staying Engaged in Child-robot Interaction, a Quantitative Approach to Studying Preschoolers' Engagement with Robots and Tasks during Second-language Tutoring

12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases
13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge
14 Michiel Overeem (UU), Evolution of Low-code Platforms
15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining
16 Pieter Gijsbers (TU/e), Systems for AutoML Research
17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification
18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation
19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation
20 Fakhra Jabeen (VUA), Dark Side of the Digital Media — Computational Analysis of Negative Human Behaviors on Social Media
21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments
22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its Potential to Foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations
23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents
24 Samaneh Heidari (UU), Agents with Social Norms and Values — A Framework for Agent Based Social Simulations with Social Norms and Personal Values
25 Anna L.D. Latour (LU), Optimal Decision-making under Constraints and Uncertainty
26 Anne Dirkson (LU), Knowledge Discovery from Patient Forums: Gaining Novel Medical Insights from Patient Experiences
27 Christos Athanasiadis (UM), Emotion-aware Cross-modal Domain Adaptation in Video Sequences
28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems
29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality
30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays
31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management
32 Cezara Pastrav (UU), Social Simulation for Socio-ecological Systems
33 Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management

34 Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change
35 Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

2023 01 Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions
02 Mariana Rachel Dias da Silva (TiU), Grounded or In Flight? What Our Bodies Can Tell Us about the Whereabouts of Our Thoughts
03 Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations
04 Gineke Wiggers (LEI), The Relevance of Impact: Bibliometric-enhanced Legal Information Retrieval
05 Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-scale Real-valued Optimization, Including Real-World Medical Applications
06 António Pereira Barata (LEI), Reliable and Fair Machine Learning for Risk Assessment
07 Tianjin Huang (TU/e), The Roles of Adversarial Examples on Trustworthiness of Deep Learning
08 Lu Yin (TU/e), Knowledge Elicitation using Psychometric Learning
09 Xu Wang (VUA), Scientific Dataset Recommendation with Semantic Techniques
10 Dennis J.N.J. Soemers (UM), Learning State-action Features for General Game Playing
11 Fawad Taj (VUA), Towards Motivating Machines: Computational Modeling of the Mechanism of Actions for Effective Digital Health Behavior Change Applications
12 Tessel Bogaard (VUA), Using Metadata to Understand Search Behavior in Digital Libraries
13 Injy Sarhan (UU), Open Information Extraction for Knowledge Representation
14 Selma Čaušević (TUD), Energy resilience through self-organization
15 Alvaro Henrique Chaim Correia (TU/e), Insights on Learning Tractable Probabilistic Graphical Models
16 Peter Blomsma (TiU), Building Embodied Conversational Agents: Observations on Human Nonverbal Behaviour as a Resource for the Development of Artificial Characters
17 Meike Nauta (UT), Explainable AI and Interpretable Computer Vision —- From Oversight to Insight
18 Gustavo Penha (TUD), Designing and Diagnosing Models for Conversational Search and Recommendation

19 George Aalbers (TiU), Digital Traces of the Mind: Using Smartphones to Capture Signals of Well-being in Individuals
20 Arkadiy Dushatskiy (TUD), Expensive Optimization with Model-based Evolutionary Algorithms applied to Medical Image Segmentation using Deep Learning
21 Gerrit Jan de Bruin (LEI), Network Analysis Methods for Smart Inspection in the Transport Domain