

Tilburg University

Data analytics in action

Schouten, Gerard; Arena, Giuseppe; Leeuwen, Frederique van; Heck, Petra; Mulder, Joris; Aalbers, Rick; Leenders, Roger; Böing-Messing, Florian

Published in:

Data science for entrepreneurship

DOI:

[10.1007/978-3-031-19554-9_10](https://doi.org/10.1007/978-3-031-19554-9_10)

Publication date:

2023

Document Version

Peer reviewed version

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Schouten, G., Arena, G., Leeuwen, F. V., Heck, P., Mulder, J., Aalbers, R., Leenders, R., & Böing-Messing, F. (2023). Data analytics in action. In W. Liebrechts, W-J. van den Heuvel, & A. van den Born (Eds.), *Data science for entrepreneurship: Principles and methods for data engineering, analytics, entrepreneurship, and the society* (pp. 205-233). Springer. https://doi.org/10.1007/978-3-031-19554-9_10

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Data Analytics in Action

*Gerard Schouten, Giuseppe Arena,
Frederique van Leeuwen, Petra Heck,
Joris Mulder, Rick Aalbers,
Roger Leenders,
and Florian Böing-Messing*

Contents

- 10.1 Introduction – 207**
- 10.2 BagsID: AI-Powered Software System to Reidentify Baggage – 208**
 - 10.2.1 Business Proposition – 209
 - 10.2.2 System Overview – 210
 - 10.2.3 AI Engine – 212
 - 10.2.4 Software Engineering Aspects – 214

Gerard Schouten and Petra Heck contributed the *BagsID* case study in ► Sect. 10.2. Giuseppe Arena, Joris Mulder, Rick Aalbers, and Roger Leenders contributed the *Understanding employee communication with longitudinal social network analysis of email flows* case study in ► Sect. 10.3. Frederique van Leeuwen contributed the *Using vehicle sensor data for Pay-How-You-Drive insurance* case study in ► Sect. 10.4. Florian Böing-Messing provided feedback on the case studies and helped shape the chapter.

Shared first authorship: The first authors of the case studies (Gerard Schouten, Giuseppe Arena, and Frederique C. A. van Leeuwen) are co-first authors of this chapter.

10.3	Understanding Employee Communication with Longitudinal Social Network Analysis of Email Flows – 216
10.3.1	Digital Innovation Communication Networks – 217
10.3.2	The Relational Event Modeling Framework – 218
10.4	Using Vehicle Sensor Data for Pay-How-You-Drive Insurance – 223
10.4.1	Time Series – 224
10.4.2	Driving Behavior Analysis – 226
	References – 231

Learning Objectives

- Understand the characteristics of reidentification deep learning and how this technique can be applied to promptly identify mishandled luggage at airports.
- Understand information sharing dynamics among employees of an organization by means of longitudinal social network analysis.
- Understand what controller area network bus technology is and what the possibilities are with respect to driving behavior analysis.

10.1 Introduction

In this chapter, we present three case studies that cover a broad spectrum of problems and methods in the area of data analytics. We begin with the BagsID case study in ► Sect. 10.2, which is carried out in collaboration with Vanderlande, PTTTRNS.ai, and Eindhoven Airport. The case study illustrates how computer vision and reidentification deep learning can be applied to reidentify mishandled luggage at airports. The approach uses Re-ID neural networks that can be trained to predict the degree of similarity between individual objects (pieces of luggage in this case) rather than categorizing objects. The BagsID case study emphasizes that getting robust AI-powered software systems into production is quite different from building proof-of-concept AI prototypes.

The second case study in ► Sect. 10.3 analyzes the effect of a business intervention strategy on the employees of a multinational service company. More specifically, a European branch of the company implemented multiple interventions aimed at stimulating its employees to open their minds to innovation. The efficacy of these interventions can be assessed by investigating how they shape communication and discussions about innovation between the employees. To this end, the case study analyzes email communication between employees using longitudinal social network analysis.

The third case study in ► Sect. 10.4 considers how vehicle sensor data can be used for insurance purposes. Through the standardization of the controller area network bus technology in modern cars, a large amount of sensor data is generated every day. This enables insurance industries to obtain more reliable and direct characterizations of driving styles for their Pay-How-You-Drive models. If used wisely, accidents can be prevented instead of restituted. This is beneficial for both the customers and the insurance industry.

10.2 BagsID: AI-Powered Software System to Reidentify Baggage

BagsID¹ is a Dutch company that aims at improving baggage handling systems worldwide by using the bag itself as an ID. At the core of their technology stack, they employ computer vision, powered by deep learning. The company is currently moving towards initial deployment to showcase its potential, in close collaboration with three organizations. These organizations are (1) Vanderlande, the global market leader for logistic process automation at airports; (2) PTTRNS.ai, a software company that specializes in developing and integrating artificial intelligence (AI) solutions to accelerate digital innovation; and (3) Eindhoven Airport. A joint project is set up at Eindhoven Airport to prove the proposition that baggage can be identified with state-of-the-art vision AI. A scale-up of the system to other European airports, and in a later stage to airports worldwide, is foreseen. This case study describes one possible application of the BagsID reidentification system: that of mishandled baggage. To illustrate this application, we begin this case study with a short user story:

► Example

March 4, 2020: Just after midnight, Jane lands at Tromsø Airport with the last flight from Oslo Gardermoen. A few hours ago, she departed from Amsterdam Schiphol. After descending from the aircraft staircase and a short walk outdoors on the slippery platform, she enters the terminal. The arrival hall is divided into two public spaces. The first area is dominated by a conveyor belt to pick up luggage, and the other area hosts a few offices of car rental companies and holds the exit doors as well as a few uncomfortable seats. As in most airports in northern Europe, the hall is decorated with huge posters showing local wildlife and snowy winterscapes with northern lights skies. The conveyor belt runs already, and soon the first suitcases appear. One by one, the passengers of SAS flight SK4438 pick up their bags and leave the hall facing the freezing cold. After 20 min, the conveyor belt stops and all fellow travelers are gone. Jane's suitcase did not appear. She is all alone at the completely deserted airport. ◀

This is no fantasy. Regular travelers could easily feel the unease of the situation sketched above. Being the last person at the airport's conveyor belt and slowly realizing that your bag is not coming is a traveler's nightmare. Better baggage handling is not just about keeping passengers happy. Claims due to lost or mishandled luggage cost airlines around the world 2.4 billion US dollars in 2018 (Air Transport IT, 2019). Over the past few years, most airlines have introduced a baggage track and trace at key points in the journey—check-in, loading onto the aircraft, transfers, and arrival—in response to IATA's Resolution 753 (IATA, 2020). Now, most bags are tracked from start to finish. Despite these efforts, the number of mishandled bags rose to 24.8 million in 2018, a figure that translates to 5.7 bags per 1000

1 ► <https://bagsid.com>.

passengers (Air Transport IT, 2019). Of all mishandled bags in 2018, 77% is delayed, about 17% is seriously damaged or pilfered, and 5% is stolen. Transferring bags from one aircraft to another, or one airline to another, is a major cause for delays of flights as well as late delivery of luggage.

This case study shows work in progress. It illustrates how an initial business idea is translated into a software-based AI solution. The BagsID case is beyond schoolbook AI. It clearly demonstrates that machine learning algorithms cannot be applied “just like that” to practical cases. We argue that a componentized extendable architecture, an iterative planning approach, and a solid software engineering process for AI embodiment are all needed for successfully building professional and maintainable AI-powered software solutions.

10.2.1 Business Proposition

The current handling of baggage depends on stickers and paper tags, which are wrapped around handles of suitcases, trolleys, or other luggage items (ski boxes, bike bags, etc.) at check-in. These stickers and tags are labeled with a printed barcode and a three-letter abbreviation of the destination airport.² The barcode is uniquely coupled to the traveler. At depots where mishandled baggage is gathered, a human-centric exception handling process—i.e., people scanning the tags with line-of-sight barcode readers³ and initiating logistic actions—is in place to identify the bags and resend them to their legitimate owners, either to the destination airport or to their home address. Serious problems with the current track and trace functionality arise when these tags have become unreadable or are even detached from the luggage. Relying on physically attached labels makes the current system inherently vulnerable.

In recent years, vision AI has drastically improved (Krizhevsky et al., 2012; LeCun et al., 2015; Howard et al., 2017; Canziani et al., 2017). With state-of-the-art deep learning, using convolutional neural networks (CNNs) as a backbone, a reliable machinery can be built to detect and identify objects in images. The ubiquitous use of face recognition, from unlocking your smartphone to crowd security management, is probably the best known example of this progress (Ye et al., 2020). So, why not apply this technology to reidentify baggage? In this way, the bag itself can become an ID. It removes the abovementioned bottleneck, that is, the problem of ripped-off tags. This business opportunity of suitcase fingerprinting has the potential to further improve efficiency and reduce the chances of a bag being mishandled. It saves not only money for airlines but also agony and discomfort for travelers. It is to be expected that the magic number of 5.7 mishandled bags per 1000 items can significantly be lowered by implementing this idea.

2 Each airport in the world is characterized with a unique three-letter combination.

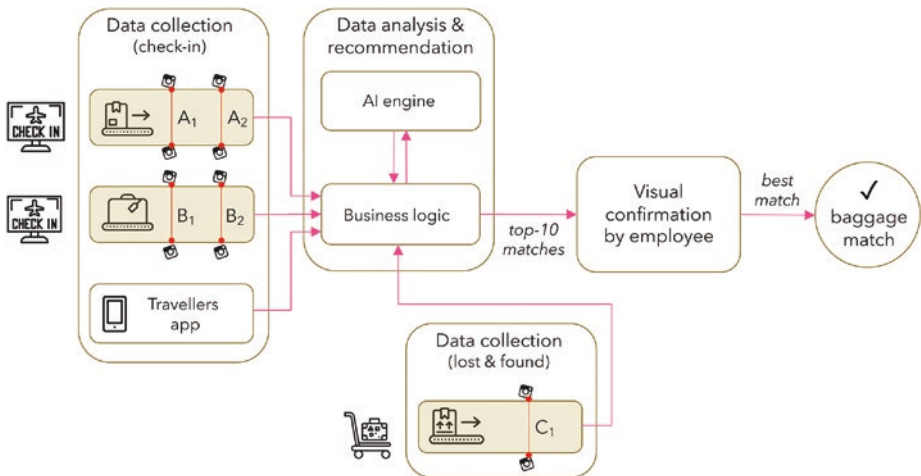
3 RF-ID tags and near-field RF-ID scanners could solve some of the issues, but this solution is too expensive.

10.2.2 System Overview

The task of the BagsID system is simple: Find for each mishandled luggage item its legitimate owner as fast as possible. The technical concept to do this is object fingerprinting, that is, find for a mishandled item the matching or corresponding item at check-in. A simplified overview is depicted in ■ Fig. 10.1. This system is divided into three major building blocks: (1) data collection at check-in, (2) data collection of lost and found bags, and (3) data analysis and recommendation. Note that there is a human in the loop for the final visual inspection. An airport or airline baggage handler is still needed to narrow down the top 10 matches of the system to an accepted best match.

Data Collection and Storage The hardware setup of the system consists of multiple camera-equipped conveyor belts. Directly after either self check-in or desk check-in, the luggage that is put on the belt is registered with two multi-camera systems that are placed just after one another. The registration establishes a link between captured suitcase images and a boarding pass. The data of this imaging system is optionally enriched with luggage information that can be entered via a traveler's app (see below). Each multi-camera system photographs passing luggage from different viewpoints. In the simplified schematic system overview of ■ Fig. 10.1, two camera-equipped conveyor belts are envisioned at check-in, and only one camera-equipped conveyor belt is present to handle lost-and-found luggage. The multi-camera systems are indicated with A_1 and A_2 for conveyor belt A, B_1 and B_2 for conveyor belt B, and C_1 for conveyor belt C. When a lost or mishandled bag with an unknown destination (because of an unreadable or ripped-off tag) is found, it will be scanned with the multi-camera system C_1 . These query images are compared with images in the gallery set that were previously captured at the check-in.

10



■ Fig. 10.1 System overview of the baggage reidentification system. Happy flow: for a lost bag, the system will retrieve the top 10 matching luggage items, based on images which were taken at check-in. A final visual check is done by an airport or airline employee. (Author's own)

Business Logic and AI Engine The data analysis building block consists of two modules. The AI engine is responsible for finding, for each mishandled bag, the best K matches from the data collected at check-in. It will be discussed in more detail in the next section. The rule-based business logic module will be connected to the airport flight schedule system and takes into account various logical time-related constraints and statistics (e.g., performance monitoring). For instance, flights might be delayed or cancelled. The task of the business logic module is fourfold: (1) narrow down the search possibilities for the AI engine up front, that is, establish the gallery set; (2) filter out matches that are logically not possible; (3) monitor performance of the AI engine; and (4) inform airport personnel what can best be done with a positively reidentified bag. Can it still be boarded at the intended airplane in time? If not, what are the best options to send it to the final destination?

Traveler's App The system also comes with a user-friendly smartphone app for travelers. It is an extension of the onboarding process and will be developed in the second phase of the project. The idea of the traveler's app is to enrich the image information that is captured at the airport's check-in. Once travelers have registered for this app, they can create and maintain a list of personal luggage items. For each bag or suitcase, they can specify values for a number of characteristic attributes, like luggage type (suitcase, trolley, backpack, guitar case, ski box, etc.), brand, color, presence of a lock, numbers of wheels, hardcover or soft side, and presence of damage marks such as scratches. These attributes correspond to the IATA baggage ID chart.⁴ This information helps to identify unique luggage items. The app is optional, that is, the system should also work if this information is not, or only partly, available.

► Example

October 28, 2021: Jane attaches the printed tags to her red old suitcase and puts it on the conveyor belt at the luggage drop-off. The coronavirus pandemic is over, and she looks forward to a short autumn break in the Mediterranean. Transavia flight HV6607 to Faro is about to leave in an hour from Eindhoven Airport. The advantage of regional airports is that the waiting time is limited. After a cappuccino, Jane buys a magazine and walks to the gate. She looks out of the window and recognizes her suitcase on one of those special airport vehicles. The red suitcase is loaded to the waiting plane. What Jane did not know was that her suitcase fell from the conveyor belt and that the loosely attached tag was ripped off. An airport employee picked up the untagged suitcase and brought it to the lost-and-found depot. Luckily, Jane—as a frequent flyer—had registered her luggage item with the BagsID traveler's app. The BagsID system was able to show ten possible matches within 30 s based on the photos taken and the earlier registered suitcase details (such as the scratch near the handle). The best match linked the red suitcase to Jane. The airport employee confirmed this best match, and 15 min later, Jane's suitcase enters the waiting airplane that was prepared to leave to Faro in 25 min. One year ago, it was unthinkable to deliver a lost-and-found suitcase to the right airplane within such a short time frame. ◀

4 ► <https://www.iata.org/en/publications/store/baggage-id/>

10.2.3 AI Engine

CNNs are a known solution for categorizing images. These feed-forward neural networks are inspired by human vision. They can abstract from viewpoint and illumination variations and are able to capture the very essentials of objects that are present in images. However, *category-level* object classification—where two images are considered similar as long as they belong to the same semantic class of objects—is not sufficient for a *search-by-example* image application. Search by example requires a more fine-grained distinction between objects that belong to the same category (Wang et al., 2014). As a simplified and intuitive example, for classification, a “Red Samsonite Omni Spinner” (hardcover suitcase), “Green Travelpro Maxlite 5” (soft-side suitcase), “Black Karrimor Ridge 32” (outdoor backpack), “Delsey Luggage Helium Aero Blue” (hardcover trolley), and “Black Briggs & Riley Baseline Vista Print” (soft-side trolley) are all luggage items. For luggage reidentification (Re-ID) on the other hand, if the query image is characterized by the phrase “red hardcover 4-wheeled suitcase,” it is essential to rank the “Red Samsonite Omni Spinner” higher than the other gallery items. Stated more formally, the objective of the Re-ID AI engine is as follows: Given a query baggage item of interest, determine the K best recommendations from the luggage gallery set. The hypothesis is that the ranking of top- K matches contains the bag (captured by a different camera in another place at a distinct time) that corresponds to the query image.

10

Re-ID Neural Network Architecture The neural network architecture of the Re-ID AI engine that is able to generate a suitcase fingerprint is shown in Fig. 10.2. It consists of two parts: an encoder module and a reidentification module. This architecture is state of the art for reidentification learning or search-by-example problems (Ye et al., 2020; Wang et al., 2014). The encoder part can be seen as a feature engineering process. Captured images—i.e., “low-level” raw pixel data—are processed with

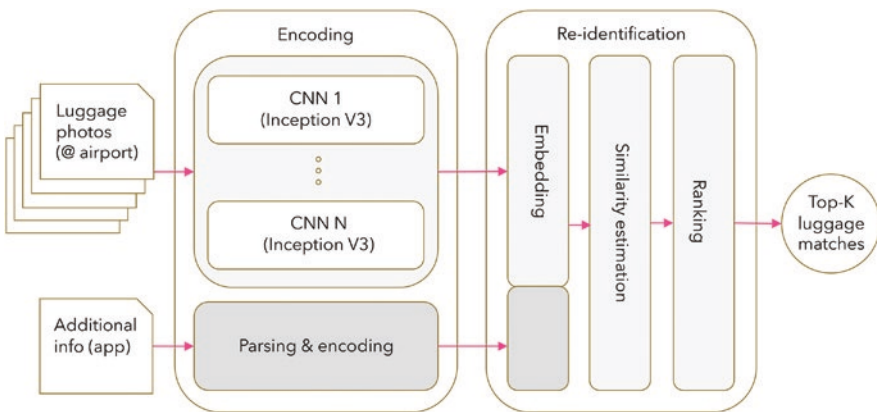


Fig. 10.2 Architecture of the AI engine. A top- K ranking is derived from N parallel CNN pipelines and app data. The app information (dark gray blocks) will be added in a later phase of the project. (Author’s own)

Google’s Inception V3 CNN framework resulting in a number of feature maps or “high-level” encodings that are more suitable for visual tasks. Inception V3 is a widely used image recognition model. The model is the culmination of many ideas developed by multiple researchers over the years; it is made up of various building blocks, including convolutions, average pooling, max pooling, concatenation, dropouts, and fully connected layers. It is based on the original paper of Szegedy et al. (2015). In the proposed architecture, each camera viewpoint is coupled to a separate Inception V3 CNN encoding pipeline.

In the next step, these visual encodings are concatenated and combined with the information that travelers provide via the app to a so-called embedding layer. An embedding is a mapping of high-dimensional data, such as images (pixel data), to a vector. This vector is a relatively low-dimensional space that summarizes the relevant information in the data into a meaningful representation. Ideally, an embedding captures some of the semantics of the input by placing similar inputs close together in the embedding space. The embedding layer flattens, reduces, and normalizes the output of all CNNs (as well as the coded app information) to a fixed-size vector. In terms of the neural network, an embedding is just a hidden layer and is learned with backpropagation during the training process.

In practice, embeddings are often used to make recommendations or to rank possible match candidates, that is, to find nearest neighbors in the embedding space. To do this, a distance metric is needed. Several options are available for this, such as the standard euclidean distance, Manhattan distance, or a cosine similarity distance metric.⁵ A retrieved top 10 ranking list can then be obtained by sorting the calculated query-to-gallery similarity.

Training the Network: Triplets, Hinge Loss, Semiautomatic Labeling The standard CNN approach in supervised learning is to estimate a function $f(\cdot)$ that maps the entire set of input images as best as possible to probabilities for given category labels. This is done in the training phase by changing the weights of the CNN (usually a few million) in such a way that a so-called loss function is minimized. Usually, this is a cross-entropy loss or mean squared error between the CNN predictions and the actual labels. For Re-ID or ranking with deep learning, however, two accommodations are needed that go beyond this standard recipe (Wang et al., 2014; Hermans et al., 2017). First of all, it is common practice to train the network with triplets of input images. A triplet $t_i = (p_i, p_i^+, p_i^-)$ contains a query image p_i , a positive image p_i^+ , and a negative image p_i^- , where the positive image is more similar to the query image than the negative image (Wang et al., 2014).

Secondly, the loss function associated with ranking and triplets is a so-called hinge loss. It is defined as

$$l(p_i, p_i^+, p_i^-) = \max\left\{0, \Omega + D(g(p_i), g(p_i^+)) - D(g(p_i), g(p_i^-))\right\} \quad (10.1)$$

⁵ The latter is the dot product between the two normalized embeddings and ranges from -1 , most dissimilar, to $+1$, most similar.

where the function $g(\cdot)$ represents the embedding and D is a distance metric: in our case, the dot product between two normalized embeddings. As explained by Schroff et al. (2015), the hinge loss tries to bring the query image and the positive image close together in the embedding space and at the same time as far away as possible from the embedding of the negative image. As long as $D(g(p_i), g(p_i^-))$ is larger than $\Omega + D(g(p_i), g(p_i^+))$, there will be no gain for the algorithm to condense the query and positive image any further. The learning process boils down to finding the best embedding $g(\cdot)$ for generating fine-grained baggage sensitivity, that is, enabling similarity ranking. Note that the distance metric itself is given up front and not modified by the training process.

For the triplets, a semiautomatic labeling process will be bootstrapped, where a priori similarity information is exploited. This semiautomatic labeling process is extended with random triplet sampling from the large image database of stored luggage photographs combined with human labeling. A labeling service like the CloudFactory platform⁶ or the Amazon Mechanical Turk crowdsourcing marketplace⁷ can be used to obtain these human labels. Crowdsourcing is a good way to break down a manual, time-consuming task—such as labeling thousands of images—into smaller, more manageable “microtasks” to be completed by distributed workers over the Internet. Traditionally, tasks like this have been accomplished by hiring a large temporary workforce, which is time consuming, expensive, and difficult to scale or to undo. These platforms offer APIs to upload your data, to carefully describe the requested task, and to ask for specific skill levels.

10

Inference with the Network: Reidentify a Mishandled Bag Once the AI engine is trained, it can be used in inference mode, that is, in operation. The fixed weights of the model produce an embedding for a mishandled luggage item. This embedding will be compared with other embeddings of luggage items that are in the gallery. The business logic provides filters and other constraints for items that will be put in the gallery. Based on the chosen distance metric, a top- K ranking will be made of the most similar embeddings.

10.2.4 Software Engineering Aspects

The BagsID system includes several software components that interact with the AI engine: data collection software, business logic software, user interfaces, traveler’s app, etc. For the AI engine to be successful in a production environment with multiple airports and multiple camera systems per airport, it is of utmost importance that its deployment strategy is carefully designed and integrated with the deployment strategy for the other software components. This integrated approach is often referred to as MLOps.⁸

6 ► <https://www.cloudfactory.com>

7 ► <https://www.mturk.com>

8 MLOps is a practice for collaboration and communication between data scientists and operations professionals to help manage the production machine learning life cycle; see ► <https://en.wikipedia.org/wiki/MLOps>

The development of software systems with AI components has several intricacies (Heck, 2019) that also apply to the BagsID system. Some issues and questions that have to be addressed when designing the system are the following:

1. Collecting high-quality data is crucial for the success of the model training. That is why the BagsID system makes use of custom-built industry-grade camera systems (hardware and software). This ensures a constant image quality and robust recording from the luggage belts.
2. After the deployment of the trained AI model, it needs to be monitored for performance (so-called online testing and logging (Heck, 2020)) because with new images coming in, the generalization capability of the model may drop and periodical retraining might be needed.
3. When the system will be scaled up (i.e., will be installed at other airports), a multi-site deployment strategy will be needed. In particular, issues that have to be discussed and settled are when and how to introduce new models in the live systems and how to version both models and data.
4. Huge amounts of data will be collected. It needs to be decided how and where to store this (on premise or in the cloud?), for how long to keep the data, which privacy laws are applicable, how to be compliant with local legislation, if airports are willing to share data for better models, etc.
5. There needs to be a scalable way of serving the model to the BagsID system for inference purposes. The model needs to be decoupled from the rest of the system such that it can be more easily updated to new algorithms or new versions. It might be necessary to have multiple versions of the model running simultaneously, for example, for different countries. State-of-the-art software engineering practices will be used for this. It is planned to deploy the AI model(s) as a REST API with Docker containers in a cloud environment.

The project started with a data collection/data preparation phase (Rollins, 2015). For this, a first camera system is set up at Eindhoven Airport where real baggage is recorded. As said, the camera system is custom-built and also contains software to preprocess the recorded images. The collected images are used to train the AI model. The training is done using Jupyter Notebooks with Python and TensorFlow 2.0 in an AWS cloud environment. Amazon SageMaker is used to support the training process and deploy the trained models for testing purposes. Next to improving the model, time is spent on preparing the deployment phase of the project for both the AI engine and other software components.

Discussion and Conclusion

Reidentification learning is a challenging and fast-growing field within the computer vision community. Most Re-ID applications deal with human faces, persons, or vehicles. In this case study, it is applied to another use case: reidentification of luggage. A system based on this technology enables airports and airlines to provide more reliable information on the whereabouts of baggage at each step in the journey. The main message of this case study is that in practice AI innovations typically (1) combine or concatenate multiple known AI concepts in a specific setting that is new and

never tried before and (2) have a mixed design of known (well-established) algorithms (such as CNNs) and principles (such as embeddings) complemented with unique business rules that have to be derived from case-specific requirements. Like the wheels in a clockwork, hardware, software, and AI components should be synchronized and fit together smoothly. The design and implementation of these components should be balanced and tuned carefully. We would like to emphasize that taking an AI model into production and maintaining it demand a serious effort and might be as complicated as designing the model itself. We also see in practice that for various reasons—e.g., safety, security, accountability, and trust—AI-powered solutions often need a human in the loop.

10.3 Understanding Employee Communication with Longitudinal Social Network Analysis of Email Flows

Innovation is the spice of life for organizations and is generally seen as a requirement for long-term survival and attaining and sustaining above-average performance. Yet, innovation can be hard to accomplish.

In this case study, we consider the innovation struggle of a European branch of a multinational service company (referred to in the case study as STRATSERV). Innovation typically requires a company's employees to change the way they do their work, either by doing different things (such as providing a new service or engaging in new procedures) or by doing things differently (such as using new technology to do the work more efficiently). This means that, especially in service organizations, innovation can hardly be successful without the willingness of employees to change (the way they do) their work. This realization stimulated STRATSERV's management to attempt to open the minds of their employees to innovation. Hence, they organized various events where employees could suggest innovative ways of working, offered prizes for the best ideas, and provided resources to employees to explore their ideas further. In sum, the approach was to first open the minds of employees to the idea of innovation, stimulate the employees to come up with innovative suggestions, and then build on that joint openness to the innovation in order to implement new services and new procedures. Of course, this assumes that the minds of the STRATSERV employees would respond favorably and long-lasting to the company's innovative wishes.

Although the STRATSERV management believed in this approach, they also realized that they needed a way to test whether their approach was working. Did their efforts indeed create an innovation mindset in the heads of their employees and did that mindset last? Moreover, they wondered if all employees responded alike or whether the competitions, gatherings, newsletters, challenges, and other activities organized by the company's task force only affected certain employees but not others.

In this situation, it makes little sense to send out a survey to the employees, asking them whether they were thinking about innovation regularly. This would likely trigger socially acceptable answers and could not provide the detailed

insight into the effect of the activities that the company was looking for. In addition, surveys are poorly suited to monitor how employees respond over time, including repeated surveys. The company reached out for help to an external team of researchers. Below, we will show part of the analysis that was performed.

10.3.1 Digital Innovation Communication Networks

When employees discuss innovation, an innovation communication network emerges within the company. The structure and pervasiveness of this network are key indicators whether STRATSERV's approach is working. In addition, innovative activity is essentially a network activity (Aalbers & Leenders, 2016; Kratzer & Leenders, 2004; Leenders et al., 2003). Innovation is, by necessity, a collaborative effort. Existing knowledge and ideas merge into new combinations, and as formerly separated knowledge comes together, new knowledge emerges. Although the imagery of the lone inventor profoundly developing is appealing, it is an image rarely found in modern times. Innovation is a "team sport," where individuals work together in teams, teams work together in projects, organizations work together in alliances, and countries work together in international technology agendas. In fact, even the mythical lone inventor probably rarely operated in splendid isolation anyway, since it is likely that much of the inventor's inspiration came from interaction with other people or organizations, the financial resources may have been granted by banks or friends, the actual development of the product often involved the help of factories, and customers had to become involved in order to test the product for feasibility. No matter which (great) innovation one would look at, it is bound to be couched in network interaction of some sort (Leenders, 2016). In sum, an ideal approach to see if innovation was catching on as a core topic and activity inside STRATSERV was to measure how the innovation communication network developed.

Networks can be measured in a number of ways. The most common approach is to administer surveys to ask who communicates with whom. Alternatively, one could observe the interactions of employees throughout their working activities. These methods do not work in our case, since we wanted to follow the interactions of employees in real time for a full year. Alternative tools such as using video to see who interacts with whom or collecting data from proximity badges would not provide information on whether the conversation included innovation as a topic. Hence, the choice was made to analyze the email interaction between the employees over the course of a year.

Digital communication, in particular email, has become one of the most important means of communication in organizations. As email leaves digital traces about senders, receivers, and timing, these rich network data contain high-resolution information to understand how communication structures change when working teams reach deadlines, to understand new employee integration processes (and how these are affected by cultural differences and team compositions), or to under-

stand how ideas spread through a network of employees (and how this is affected by the actors' hierarchical positions, for example). Besides the academic/theoretical interest, these insights are also useful from a practical point of view as they can be used to optimize communication structures in deadline situations, they can be used to optimize the integration processes of new employees, and they can be used to reach all employees regarding certain working topics as fast as possible.

In this case study, we show one approach that can be used to study and understand how networks evolve over time, in real time, and how this knowledge can be leveraged in practice.

10.3.2 The Relational Event Modeling Framework

Description of the Data Our analysis focuses on the innovation communication networks in a European branch of STRATSERV. After developing and implementing procedures to ensure employee privacy and informed consent was received from the parties involved, we used text mining techniques to score the email messages on whether the exchanged text addressed innovation-related topics. The empirical data in this case study consist of a time-ordered sequence of $M = 1340$ email messages that were exchanged between 153 employees over the course of a year. An example of the data is given in ■ Table 10.1 where each row represents the 3-tuple (t_m, s_m, r_m) with, respectively, the time, the sender, and the receiver of the m th email in the sequence of emails $E = \{(t_1, s_1, r_1), \dots, (t_M, s_M, r_M)\}$.

We assume that email interaction is regulated and driven by factors that can depend either on workers' characteristics (e.g., one's status or outgoingness), on the dyadic characteristics of sender and receiver (e.g., hierarchy differences, co-location), on the history of workers' past interactions (e.g., the exchange of email that occurred in the past), or on the workers' location in the social structure (e.g., interaction with joint colleagues, norms of reciprocity). In particular, we will focus on modeling whether and how this email stream depends on working in the same building, the difference in hierarchy level between sender and receiver of the email, the tenure of the sender, the tendency of sender and receiver to continue to exchange email messages among each other (i.e., persistence or inertia), and the norms of

■ Table 10.1 Example of longitudinal network of emails

Time	Sender	Receiver
03 Jan 2010 08:21:33	Marco	Jane
03 Jan 2010 08:43:09	Jane	Marco
∅	∅	∅
31 Dec 2010 18:39:22	Paul	Jane

Compiled by authors

reciprocity between employees. Moreover, we allow a possible memory effect where recent email activity may have a relatively large effect on the future activity between actors.

The Model The novel modeling framework that is well suited to analyze time-to-event sequence data in networks is the so-called *relational event model* (REM) (Butts, 2008; Mulder & Leenders, 2019; Leenders et al., 2016). This framework aims to model the rate at which specific directed interaction (i.e., a given email being sent) between two actors (here: employees) occurs; in other words, we model the *emailing rate* among any pair of employees. In social network terms, such a pair is called a *dyad*. Within this framework, each email message constitutes a *relational event* characterized by the *sender* (s), who initiates the action (i.e., who sends the email); the *receiver* (r), to whom the action is targeted (i.e., who receives the email); and *time* (t), the exact time point at which the relational event occurs. At each time point in the sequence, 153 potential senders can send an email to 152 potential receivers (excluding email messages people send to themselves), which means that at any point in time $153 \times 152 = 23,256$ email dyads can potentially occur. The aim of the analysis is to model who sends an email message to whom at what point in time over the course of 1 year. Mathematically, the joint probability to model the whole sequence of emails is similar to the well-known event history model or survival model (Lawless, 2003; Cox, 1972).

In the REM, we model the rate at which an email is sent from a given sender to a given receiver at a given point in time as a loglinear model that (apart from the exponent that occurs in the equation) resembles the well-known linear regression structure. The model then takes into account every possible sender, every possible receiver, and every possible point in time, for the entire observation period. One of our substantive interests in this study is whether the emailing rates of employees depend only (or mainly) on the recent email interactions of the employees or whether they also take into account email exchanges that happened longer ago. This is important for STRATSERV, as it shows how long the effects of interventions last. If it turns out that employees mainly respond to innovation-related messages they received recently, and much less to messages received or exchanged longer ago, this is a sign that employees apparently need to be “reminded” of innovation constantly and that it has not become a routine part of their conversations.

In particular, we will investigate this for inertia and reciprocity (see ■ Table 10.2). In order to accomplish this, both the inertia and reciprocity variables are calculated according to two different event history lengths. For both variables, we include in the model a short-run version where only past events that occurred *until 30 days* before the time of the email are included (*recent past*) and a long-run version that includes the past events that occurred *more than 30 days* before the email was sent (*less recent past*) (cf. Quintane et al., 2013). A complete description of the variables used in our analysis can be found in ■ Table 10.2.

Model Comparison We estimate two models: in *Model 1*, all the variables in ■ Table 10.2 are embedded in the loglinear predictor; in *Model 2*, only the short-run and long-run versions of inertia and reciprocity are included. Via this model com-

Table 10.2 Predictor variables and their interpretations

Predictor variable	Description
ShortInertia	The number of messages a potential sender sent to a potential receiver in the last 30 days
LongInertia	The number of messages a potential sender sent to a potential receiver more than 30 days ago
ShortReciprocity	The number of messages a potential sender received from a potential receiver in the last 30 days
LongReciprocity	The number of messages a potential sender received from a potential receiver more than 30 days ago
SameBuilding	A binary variable which indicates whether potential sender and potential receiver work in the same building (1) or not (0)
DiffHierarchy	The hierarchical difference between the sender and receiver on a scale from 1 to 9
LogSenderTenure	The number of years a potential sender works in the organization on a log scale
Compiled by authors	

10

parison, we can learn whether a simpler model without exogenous effects may be enough for a good fit for the data. Considering the specification of Model 1, the email rate (λ) at time t_m for the dyad (sender, receiver) = (Marco, Jane) is

$$\lambda(t_m, \text{Marco, Jane}) = \exp\{\beta_{\text{Intercept}} + \beta_{\text{ShortInertia}} \text{ShortInertia}(t_m, \text{Marco, Jane}) + \beta_{\text{LongInertia}} \text{LongInertia}(t_m, \text{Marco, Jane}) + \beta_{\text{ShortReciprocity}} \text{ShortReciprocity}(t_m, \text{Marco, Jane}) + \beta_{\text{LongReciprocity}} \text{LongReciprocity}(t_m, \text{Marco, Jane}) + \beta_{\text{SameBuilding}} \text{SameBuilding}(\text{Marco, Jane}) + \beta_{\text{DiffHierarchy}} \text{DiffHierarchy}(\text{Marco, Jane}) + \beta_{\text{LogSenderTenure}} \text{LogSenderTenure}(\text{Marco})\} \quad (10.2)$$

where $\beta = (\beta_{\text{Intercept}}, \beta_{\text{ShortInertia}}, \beta_{\text{LongInertia}}, \beta_{\text{ShortReciprocity}}, \beta_{\text{LongReciprocity}}, \beta_{\text{SameBuilding}}, \beta_{\text{DiffHierarchy}}, \beta_{\text{LogSenderTenure}})$ is the vector of effects describing the impact of the variables on the rate of occurrence of an email being sent from a sender to a receiver. Positive effects (negative effects) imply that as the variable increases in value, it increases (decreases) the email rate. As regards Model 2, the rate of an email sent from Marco to Jane at time t_m becomes

$$\lambda(t_m, \text{Marco, Jane}) = \exp\{\beta_{\text{Intercept}} + \beta_{\text{ShortInertia}} \text{ShortInertia}(t_m, \text{Marco, Jane}) + \beta_{\text{LongInertia}} \text{LongInertia}(t_m, \text{Marco, Jane}) + \beta_{\text{ShortReciprocity}} \text{ShortReciprocity}(t_m, \text{Marco, Jane}) + \beta_{\text{LongReciprocity}} \text{LongReciprocity}(t_m, \text{Marco, Jane})\}. \quad (10.3)$$

The results of both models can be found in **Table 10.3**. Model 1 seems to be better supported by the data since the BIC and AIC for Model 1 are lower than for

Model 2. In addition to this, the email rate is mainly affected by recent email history, that is, by the short-run effects of inertia and reciprocity. Although the effect of long-run inertia (LongInertia) is statistically significant, the effects of long-run inertia and long-run reciprocity (LongReciprocity) are negligibly small and hence barely affect the email rate. The results of Model 2 (which only includes inertia and reciprocity) show that these effects are stable and unaffected by the other variables. In other words, the employees tend to repeat their recent behavior and mainly respond to innovation-related messages received in the recent past, while innovation messages that were sent or received more than 30 days ago seem to no longer affect emailing behavior today. In other words, employees appear to discuss innovation because it is what they recently discussed, not because it is something that is on their minds in the long run. This is a sign that STRATSERV has not been able to make innovation an integral part of their employees' mindset.

From Model 1, we see that employees send emails at lower rates to other employees who are lower in the organizational hierarchy than they are themselves and send their email messages at higher rates to those who have higher hierarchy levels than they have themselves ($\hat{\beta}_{\text{DiffHierarchy}} = -0.3003$). In other words, email messages about innovation are more readily sent up the organizational hierarchy than down. This is consistent with the idea that the STRATSERV employees are willing to inform their superior about potential innovation but are less likely to put their ideas into action themselves by discussing it with those lower in the chain of command. Conversely, employees who enjoy higher hierarchical positions are more popular targets for such email messages than are those who occupy low status positions in the organization. Again, innovation discussion is directed up the chain, but much less to the lower levels.

Except for DiffHierarchy, all other variables in Model 1 have positive effects on the emailing rates. For instance, the email rate of a sender to a receiver who works in the same building (SameBuilding = 1) is around two and a half times ($\exp\{\hat{\beta}_{\text{SameBuilding}}\} = 2.679$) higher than the email rate from that same sender to a colleague working in a different building, holding constant all the other variables. This is an important finding, as it suggests that physical boundaries (i.e., working in a different building) also appear to function as communication boundaries: STRATSERV employees more intensely discuss innovation-related topics with those whom they routinely meet at the coffee machine, and much less with those they do not run into that often.

We also observe that the rate at which employees send innovation-related email increases with the time they have been at the organization. Conversely, newcomers and juniors turn out less active in communicating about innovation than are the seniors of the firm, which makes sense.

Discussion and Conclusion

The relative importance of the different effects can be used to improve and optimize information sharing. For example, as there is a large positive (negative) effect of interaction when employees work in the same (in different) buildings, interaction may be greatly improved by setting up interventions in the organizations that stimu-

Table 10.3 Model 1 and Model 2: maximum likelihood estimates, standard errors, z-values, p-values, AIC, and BIC

Variable	Model 1					Model 2						
	$\hat{\beta}$	se($\hat{\beta}$)	z-value	p-value	$\hat{\beta}$	se($\hat{\beta}$)	z-value	p-value	$\hat{\beta}$	se($\hat{\beta}$)	z-value	p-value
Intercept	-11.6322	0.0862	-34.914	0.000	-9.2559	0.0249	-371.323	0.000				
ShortInertia	0.0831	0.0005	151.582	0.000	0.0869	0.0005	181.294	0.000				
LongInertia	0.0058	0.0005	10.871	0.000	0.0065	0.0005	14.025	0.000				
ShortReciprocity	0.0484	0.0104	4.628	0.000	0.0345	0.0101	3.406	0.0006				
LongReciprocity	-0.0070	0.0170	-0.409	0.682	-0.0094	0.0162	-0.579	0.563				
SameBuilding	0.9854	0.0401	24.591	0.000								
DiffHierarchy	-0.3003	0.0096	-31.307	0.000								
LogSenderTenure	0.9234	0.0378	24.413	0.000								
AIC	16,004.33					16,981.15						
BIC	16,045.93					17,007.15						

Compiled by authors

late discussions across employees in different buildings. In addition, it is important to know for managers that STRATSERV's employees are less likely to share innovation-related communication with colleagues they are not co-located with. Although this can partly be addressed by strategically placing employees in their various locations, it is also important for managers to realize where communication may flow more easily and where it is likely to be hampered.

Furthermore, STRATSERV learns from this analysis that a temporary silence in innovation-related activity tends to remove the topic from the active attention of its employees. This could potentially be addressed by organizing activities around innovation, but it also signals that the current activities have not been successful in making innovation part of the normal conversation of STRATSERV's employees. This may be a reason to reevaluate the effectiveness of the current strategy while, at the same time, taking into account that it may take a long time to establish an innovation mindset.

Thanks to the relational event model, we are able to understand which factors play a role in employee interaction. Specifically, the observed differences in intensities and signs of the relative effects showed that certain characteristics can impact the email rate to different degrees and in different directions. Using targeted interventions, these insights can be used to reach more employees in a shorter amount of time. For further reading on relational event models, we refer interested readers to Leenders et al. (2016), Schecter et al. (2017), and Pilny et al. (2016).

10.4 Using Vehicle Sensor Data for Pay-How-You-Drive Insurance

The emergence and growth of connected technologies and big data are changing the face of all industries. An example of an industry which is expected to avail tremendous benefits from the relevant data generated by the billions of connected devices is the insurance industry. One of the most popular cases of big data adoption within the insurance industry is the Pay-How-You-Drive (PHYD) paradigm (Carfora et al., 2019). This means that instead of calculating insurance premiums based on only demographic characteristics, personal driving characteristics—either exposure or behavioral—are also incorporated in the insurance models (Tselentis et al., 2016).

In order to understand people's driving behavior, data is gathered about, for example, the driver's speeding and braking behavior. State-of-the-art research about modeling human driving behavior is mostly based on GPS data (Grengs et al., 2008), including variables such as the GPS location, traveled distance, and coarse-grained speed profile. However, nowadays, the standardization of the controller area network (CAN) bus technology and the increase of the electronic control units (ECUs) in modern cars offer a large availability of sensor data, enabling a more reliable and direct characterization of driving styles (Fugiglando et al., 2017). Considering the car as a human body, the CAN bus is the nervous system enabling communication between the different body parts (ECUs). Modern cars

may have up to 70 ECUs, such as the cruise control, audio systems, and engine control unit. Hence, the ability to connect the different ECUs and sensors in a vehicle through CAN bus technology enables the gathering of valuable information about, for example, the state of the vehicle and the driving behavior of the driver.

Despite the useful data provided by the numerous sensors in modern cars, the interpretation of data is cumbersome due to the different implementations of the CAN messaging system (de Hoog et al., 2019). Whereas the CAN protocol is standardized, the actual implementation differs for every manufacturer and even for every car model. So, in order to obtain the useful information, CAN bus traffic has to be analyzed and reverse engineered for every car type separately (Huybrechts et al., 2017). As this is a very time-consuming task, the use of CAN bus data to model driving behavior for PHYD insurance is barely adopted so far (Fugiglando et al., 2018).

With the flexible CAN solutions established by *Beijer Automotive B.V.*,⁹ one is able to access the complex vehicle sensor data hidden in cars. This enables the analysis of an enormous amount of informative data about not only the drivers (e.g., speed, brake, steering position, wheel speed, odometer, left/right direction indicator), but also their surroundings (e.g., fog/hazard lights, wipers, ambient air temperature). Although this overload of data may be promising concerning the reliability of driving-style characterization, it remains a complex concept influenced by a burdensome number of factors and possible interpretations of the driver response (Martinez et al., 2017). In other words, due to many (external) conditions affecting the driving behavior, it is difficult to understand what factors exactly caused a certain driving behavior. Did the driver brake suddenly because of an unexpected event caused by another driver or because he or she was distracted by his or her phone? This and many other questions could arise while analyzing all the variables. What can actually be learned from all these variables and should they be analyzed separately or simultaneously?

10.4.1 Time Series

Before continuing with discussing some interesting applications, a bit more should be mentioned about the data. As the measurements from the CAN bus are collected at uniformly spaced time instants, the gathered data can be considered as a *time series*:

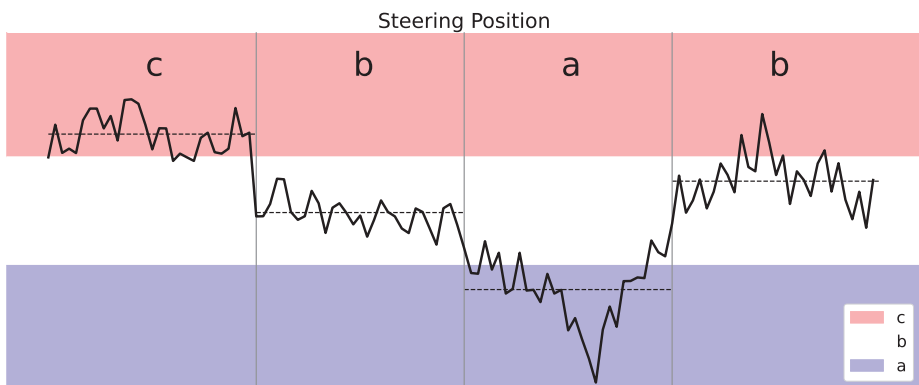
A time series $T = \{t_1, \dots, t_n\}$ is an ordered sequence of n real-valued numbers, often measured at fixed time intervals.

9 ► <https://www.beijer.com/en/>

The series can be univariate as described above or multivariate when several series simultaneously span multiple dimensions within the same time range (Esling & Agon, 2012). As all the data from the sensors and ECUs in the car are measured at the same time, they can be considered as multivariate time series.

While winning data from the CAN bus is already challenging, the actual problem begins when one wants to decode the gathered data. As mentioned before, this is due to the many different implementations of the CAN messaging system. However, imagine that you possess the information to make the right translation and thus that you can transform the raw data into long time series representing the variables of interest. Even when one is able to arrive at this stage, understanding the actual driving behavior remains challenging. This is due to the volume of the data; almost every 10 ms, a signal is sent through the CAN messaging system. Consequently, one ride of ± 1.5 h results in time series including over half a million data points. Hence, efficient algorithms are needed in order to analyze this data.

There are many different methods to analyze time series data, summarized by Esling and Agon (2012). As the obtained data is high-dimensional, algorithms directly applied to the raw time series would be computationally too expensive. To reduce the data dimensionality, one can use representation techniques. A widely used method for this is called Symbolic Aggregate approXimation (SAX) introduced by Lin et al. (2007). The method consists of two stages. First, the time series is converted into a piecewise aggregate approximation of a predefined number of segments. Afterwards, the average value of each segment is transformed into a symbol according to a set of break points. As a result, the time series is transformed into a *string* consisting of, for example, 3 symbols (see ■ Fig. 10.3). With string compression algorithms such as GrammarViz (Senin et al., 2018), grammar rules (e.g., *bba* in *acbbaacbbba*) can be inferred from the newly created string. These rules represent repeating patterns (*motifs*) in the time series. In a similar way, also anomalous patterns (*discords*) can be detected.



■ Fig. 10.3 SAX is used to transform a time series into a sequence of letters (a *string*). This figure illustrates a time series of 130 data points which is converted into a string *cbab* of 4 letters (i.e., segments). (Author's own)

Although dimensionality reduction techniques may increase the efficiency of time series data mining tasks, the downside is that details may be overlooked. In cases where those details play an important role, analysis can be better done on the raw time series. Depending on the application, the right technique should be chosen. Examples of applications in which motif or discord discovery could be of interest and situations in which dimensionality reduction techniques are not favorable are discussed in the coming sections.

10.4.2 Driving Behavior Analysis

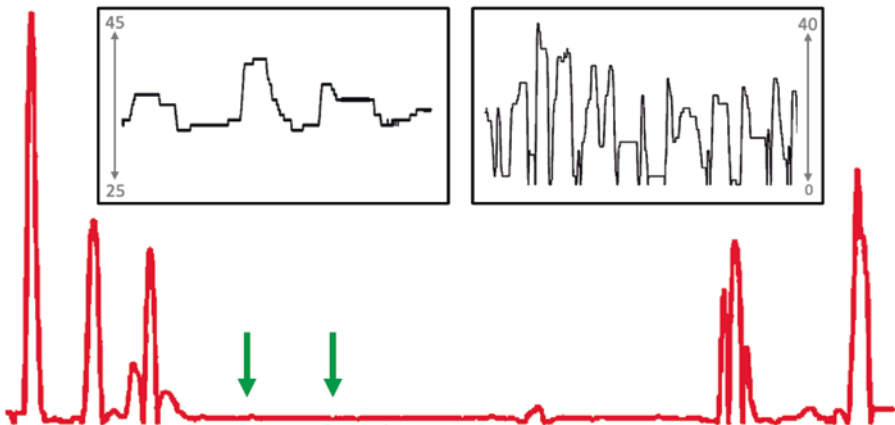
One insurance company in the Netherlands calculates its premiums based on their customers' driving behavior. For this task, they use four variables: speed, curves, brake, and acceleration. Although this provides insight into the driving style of a client, it is still very general. What exactly defines *safe* (or dangerous) driving behavior? Safety is a vague concept and could become more tangible when it is known what the patterns actually represent. In other words, only using those four variables does not include anything about the context of the ride. When more variables are included, maybe the cause of certain behavior can be detected and thus *safety* can be based on those events rather than on a variable like speed.

One of the main contributing factors to the road safety problem is an *inattentive driving style*, often caused by distracting activities (Meiring & Myburgh, 2015). Potential distracting activities may include attention to a person, object, or event outside the car, eating or drinking, talking, texting, and distracting weather conditions. Note that an inattentive driving style differs from an aggressive driving style due to its instantaneous and sporadic nature. Aggressive driving can be often observed as a pattern of misbehavior over a longer period of time (Meiring & Myburgh, 2015). The main challenge is how to use the gathered variables to detect such inattentive driving behavior. This contradiction serves well as an example for how the corresponding time series data should be analyzed: the detection of aggressive driving behavior may ask for motif discovery, while discords are of higher interest for the detection of inattentive driving behavior due to its anomalous nature.

Phone Usage Lately, especially the use of mobile phones is considered to be a threat to the safety on the road. Motivated by the impact on the overall safety, governments have enacted regulations that prohibit mobile phone usage while driving. But how can it be controlled? Is it possible to detect people being distracted in the car by using their mobile phone? Although previous methods promise to be effective in detecting the use of mobile phones while driving, they are dependent on either camera systems or on radars (Leem et al., 2017). As these attributes do not belong to the standard car equipment, they were specifically installed for the controlled experiment setup. The data from the CAN bus, however, is accessible in every car and could also be obtained from uncontrolled environments. Below, two variables from the CAN bus are described which could help to detect (or get insights into) phone usage or, more generally, driver inattentiveness.

Steering position. Beijer Automotive B.V. conducted an experiment in which they let people drive the same route twice: the first time without any instructions and the second time with the instruction to read a text message which was sent to them. The time interval in which the message was read is indicated by the green arrows in ■ Fig. 10.4. This figure shows the steering wheel position on the y-axis versus the time on the x-axis. A high peak corresponds to turning to the right or left.¹⁰ Although at first sight no difference was visible in the two different rides, when zoomed in, the difference came to light. Everyone who drives in a straight line moves the steering wheel lightly, resulting in a pattern similar to the left black curve in ■ Fig. 10.4. When distracted—in this case by reading the text message—people move the steering wheel more heavily, as shown in the right black curve in ■ Fig. 10.4.

As the motion is very detailed, it may not be advantageous to use dimensional-reduction techniques. On the other hand, when many rides need to be analyzed, it would become computationally too expensive to analyze the entire time series. Nonetheless, during this experiment, it became clear that one can use *one variable* (i.e., univariate time series) to get insights into the driving behavior of the driver. Although it was easy to identify the different patterns during this experiment, it becomes more challenging when no knowledge exists about the exact time slot in which a phone is used. When much data is generated in uncontrolled environments, it could be therefore useful to include more variables. In this way, the context can be used to understand a certain steering wheel action. Moreover, other variables like the brake may increase the accuracy of detecting people using their phone.



■ Fig. 10.4 The red curve shows the steering position during the route which was driven twice. The green arrows indicate a time interval, and the two black boxes show the steering position during that time interval for the two different rides. The left black curve represents a normal steering behavior when the driver did not receive a text message, and the right black curve represents the interval when the driver was distracted by his or her phone while driving. (Author's own)

¹⁰ The peaks for both steering actions are positive as the signal is unsigned.

Wheel speed. Another way to detect potential inattentive driving behavior is by analyzing the wheel speeds. At the top of **Fig. 10.5**, the speeds of the four wheels (front left, front right, rear left, and rear right) are visualized. While these variables separately may not seem informative, they include valuable information when analyzed simultaneously. The dark blue curve shows the difference in speed between the front and rear wheels (front–rear: $(FL + FR) - (RL + RR)$), and the other blue curve shows the difference in speed between the left and right wheels (left–right: $(FL + RL) - (FR + RR)$). The latter includes similar information as the steering position (gray curve): every time the steering wheel is moved to, for example, the right, the difference of the wheel speeds between left and right increases. While turns to the left or right are clearly visible through the big peaks, more detailed actions are also captured by the difference in the wheel speeds and can be used to detect anomalous driving patterns.

Not only the steering behavior of the driver is captured in the wheel speeds, but they also include additional information. In **Fig. 10.5**, there are two anomalies visible in between the two black vertical lines halfway in the blue curves. When considering the separate wheel speeds, this is unexpected as the driver drove a continuous speed during that moment. Moreover, the steering position during that time period indicates that no steering action was performed. What these anomalies could represent is discussed in the next section.

10

Road Conditions Another important aspect of road safety is the monitoring of the road conditions (Meiring & Myburgh, 2015). Fazeen et al. (2012) demonstrated in their paper that by using mobile smartphones one is able to evaluate overall road conditions, including bumps, potholes, and rough, uneven, and smooth road. By

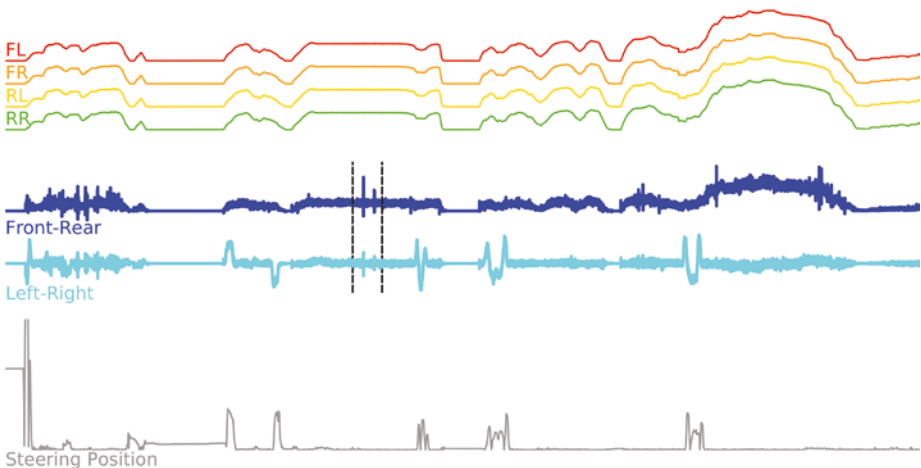



Fig. 10.5 At the top of this figure, the speeds of the four wheels are visualized. The blue curves show the differences in speed between the front and rear wheels (dark blue) and the left and right wheels (light blue). The gray curve represents the steering position. All curves share the same x-axis, which represents the time. The two black vertical lines highlight a time interval including two anomalies. (Author's own)

using the mobile phone's accelerometer, subtle or extreme vibrations were recorded inside the vehicle. Combining these accelerometer readings with GPS coordinates enabled them to make an accurate (85.6%) road condition mapping. However, to achieve accurate measurements, the location and orientation of five phones inside the car needed configuration.

By analyzing the wheel speeds, one is also able to detect road anomalies. Peaks as highlighted between the two black vertical lines in  Fig. 10.5 could indicate such anomalies. When exposed to a bump or pothole in the road, the speed of one wheel changes significantly compared to the other wheels. This leads to an anomaly in the differences between the wheel speeds. How accurate the detection of road anomalies via wheel speeds is has been hardly researched yet and is an interesting topic for future research. Nonetheless, with signals every 10 ms, small vibrations caused by either driving behavior or road conditions could be captured. Moreover, with many cars on the road, an enormous amount of data can be gathered and analyzed on a daily basis.¹¹ This enables a more reliable detection of road anomalies.

External Factors As all journeys differ considerably, the driver gets exposed every single journey to different external factors. Although variables like the wheel speed or steering position may include useful information, it still may be hard to detect anomalies in uncontrolled environments. One of the main advantages of using CAN bus data is that it includes informative data not only about the drivers, but also about their surroundings. Sensors like fog lights, hazard lights, wipers, and temperature provide insights into the climate, and other sensors like the direction indicator, the brake, and the throttle may include information about certain events. If, for example, the driver brakes heavily after driving 120 km/h, it may be more interesting to analyze the steering wheel position prior to this event than when someone is driving 30 km/h and uses the left direction indicator to turn to the left. Likewise, stormy days may elicit other driving responses than calm and sunny days, and so forth. By utilizing the information included in the overload of data retrieved from the CAN bus, one is able to understand the context of the driving scene and external conditions. This enables a more reliable and direct characterization of the driving behavior (Fugiglando et al., 2017). Note that in this case, there is no longer only dependency of one variable on its past values, but there is also some dependency between the other variables that has to be captured. Hence, techniques are needed which not only do have to deal with abnormal values or subsequences in each time series separately, but are also able to detect the relationships among the variables (Li et al., 2017).

Discussion and Conclusion

Whereas most car insurance companies quantify accident risk based on either demographic characteristics or GPS data, CAN bus data is expected to better characterize human driving behavior and thus accident risk (Fugiglando et al., 2018). Before

11 An example of a platform which brings together CAN bus data of many cars is *Vetuda* (► <https://www.vetuda.com/en/>). Not only road conditions can be analyzed, but it also provides information for applications such as incident, weather, and traffic management.

driving profiles of customers can be determined, many experiments should be conducted. By matching patterns in uncontrolled environments with the ground truth from controlled experiments, one is enabled to characterize *inattentive* and *aggressive* driving behavior. It is important to note that in uncontrolled environments, only rough proxies of inattentive driving behavior can be detected. Due to the lack of labels (ground truth), it is hard to determine the exact cause of anomalous driving patterns. Sometimes, people chose for an unsafe driving environment themselves (e.g., by using their phone), but also external factors such as other drivers can play a role in the decisions made by the driver. However, by focusing on steering actions such as corrections and unstable steering positions as depicted in ■ Fig. 10.4, it is possible to get a general overview of the driving behavior of customers.

Using this rich information not only is interesting for calculating the premiums of car insurance customers, but may also help insurance companies to understand the exact circumstances of accidents. Are there certain scenarios or places which cause many drivers to be distracted? Such information could be used to warn their customers and influence them to drive more safely. Ultimately, this could even lead to a shift in the core of their business model: a shift from restitution to prevention. Customers may also benefit from this new business model. With a reduction in restitution costs through prevention, discounts on premiums can be offered to those who drive safely. Using this information to adopt the Pay-How-You-Drive paradigm can be beneficial for the customers as they can now directly impact their paid premium. The safer you drive, the less you pay, and maybe even more importantly, the less we all pay.

10

Conclusion

In this chapter, we presented three case studies showing data analytics in action. The case studies considered diverse problems and provided an insight into the data analytical toolkit that is available to solve these problems. Of course, the data analytical toolkit is vast and there are many tools that we did not cover in this chapter. Nevertheless, the case studies illustrated how powerful modern data analysis techniques are for answering intricate questions that would otherwise remain open. We also emphasized that these techniques require careful adaptation to the problem at hand in order to deliver the desired results. However, if this adaptation is done right, data analytics can provide deep insights and produce practical outcomes that are highly valuable for businesses and consumers.

Discussion Points

1. AI education should be enriched with practical cases.
2. The inclusion of specific behavioral patterns in the dynamic social network analysis improves the understanding of the information flow between employees and helps refining business strategies.

3. In uncontrolled environments, only rough proxies of, for example, inattentive driving behavior can be detected. Due to the lack of labels (ground truth), it is hard to determine the exact cause of anomalous driving patterns. This should be taken into consideration when driving profiles are determined.

Take-Home Messages

- It takes a serious engineering effort to get an AI-powered software system into production. This is quite different from building AI demonstrators.
- It is an illusion to believe that a business intervention strategy affects all employees equally. Analyzing the communication between employees can help the management understand how, where, and for how long interventions carry an effect. Cutting-edge developments in longitudinal social network analysis can help target interventions more effectively and assess policy effectiveness realistically and in real time.
- By analyzing the enormous amount of informative data from CAN bus technology, human driving behavior—and thus accident risk—can be better characterized.

Acknowledgements Gerard Schouten and Petra Heck thank Erik van Breusegem of PTTRNS.ai for providing and reviewing the BagsID case study. We are more than happy that we can use this illustrative deep learning and software engineering case for educational purposes. We also thank Jesse Berger, graduate student at PTTRNS.ai, for his assistance and valuable support in digging up substantial business and technical information.

References

- Aalbers, R. H. L., Dolfsma, W., & Leenders, R. T. A. J. (2016). Vertical and horizontal cross-ties: Benefits of cross-hierarchy and cross-unit ties for innovative projects. *Journal of Product Innovation Management*, 33(2), 141–153. <https://doi.org/10.1111/jpim.12287>
- Air Transport IT. (2019). Insights, online. Retrieved March 2020, from <https://www.sita.aero/resources/type/surveys-reports/air-transport-it-insights-2019>
- Butts, C. T. (2008). A relational event framework for social action. *Sociological Methodology*, 38(1), 155–200. <https://doi.org/10.1111/j.1467-9531.2008.00203.x>
- Canziani, A., Culurciello, E., & Paszke, A. (2017). *An analysis of deep neural network models for practical applications*. arXiv:1605.07678.
- Carfora, M. F., Martinelli, F., Mercaldo, F., Nardone, V., Orlando, A., Santone, A., & Vaglini, G. (2019). A “pay-how-you-drive” car insurance approach through cluster analysis. *Soft Computing*, 23(9), 2863–2875.
- Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society Series B (Methodological)*, 34(2), 187–220. www.jstor.org/stable/2985181
- de Hoog, J., Castermans, N., Mercelis, S., & Hellinckx, P. (2019, November). Online reverse engineering of CAN data. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 776–785). Springer.

- Esling, P., & Agon, C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1), 1–34.
- Fazeen, M., Gozick, B., Dantu, R., Bhukhiya, M., & González, M. C. (2012). Safe driving using mobile phones. *IEEE Transactions on Intelligent Transportation Systems*, 13(3), 1462–1468.
- Fugiglando, U., Massaro, E., Santi, P., Milardo, S., Abida, K., Stahlmann, R., Netter, F., & Ratti, C. (2018). Driving behavior analysis through CAN bus data in an uncontrolled environment. *IEEE Transactions on Intelligent Transportation Systems*, 20(2), 737–748.
- Fugiglando, U., Santi, P., Milardo, S., Abida, K., & Ratti, C. (2017, October). Characterizing the “Driver DNA” through CAN bus data analysis. In *Proceedings of the 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services* (pp. 37–41).
- Grengs, J., Wang, X., & Kostyniuk, L. (2008). Using GPS data to understand driving behavior. *Journal of Urban Technology*, 15(2), 33–53.
- Heck, P. (2019). Software engineering for machine learning applications, online. Retrieved from <https://fontysblogt.nl/software-engineering-for-machine-learning-applications/>
- Heck, P. (2020). Testing machine learning applications, online. Retrieved from <https://fontysblogt.nl/testing-machine-learning-applications/>
- Hermans, A., Beyer, L., & Leibe, B. (2017). *In defense of the triplet loss for person re-identification*. arXiv:1703.07737.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. arXiv:1704.04861.
- Huybrechts, T., Vanommeslaeghe, Y., Blontrock, D., Van Barel, G., & Hellinckx, P. (2017, November). Automatic reverse engineering of CAN bus data using machine learning techniques. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 751–761). Springer.
- IATA. (2020). Baggage tracking, online. Retrieved March 2020, from <https://www.iata.org/en/programs/ops-infra/baggage/baggage-tracking>
- Kratzer, J., Leenders, R. T. A. J., & Van Engelen, J. M. L. (2004). Managing creative team performance in virtual environments: an empirical study in 44 R&D teams. *Technovation*, 26(1), 42–49. <https://doi.org/10.1016/j.technovation.2004.07.016>
- Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS* (pp. 1106–1114).
- Lawless, J. F. (2003). *Statistical models and methods for lifetime data*. John Wiley & Sons. <https://doi.org/10.1002/9781118033005>. Print ISBN: 9780471372158. Online ISBN: 9781118033005.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444.
- Leem, S. K., Khan, F., & Cho, S. H. (2017). Vital sign monitoring and mobile phone usage detection using IR-UWB radar for intended use in car crash prevention. *Sensors*, 17(6), 1240.
- Leenders, R. T. A. J., & Dolfsma, W. A. (2016). Social networks for innovation and new product development. *Journal of Product Innovation Management*, 33(2), 123–131. <https://doi.org/10.1111/jpim.12292>
- Leenders, R. T. A. J., Contractor, N. S., & DeChurch, L. A. (2016). Once upon a time: Understanding team processes as relational event networks. *Organizational Psychology Review*, 6(1), 92–115. <https://doi.org/10.1177/2041386615578312>
- Leenders, R. T. A. J., Van Engelen, J. M. L., & Kratzer, J. (2003). Virtuality, communication, and new product team creativity: A social network perspective. *Journal of Engineering and Technology Management*, 20(1–2), 69–92. [https://doi.org/10.1016/S0923-4748\(03\)00005-5](https://doi.org/10.1016/S0923-4748(03)00005-5)
- Li, J., Pedrycz, W., & Jamal, I. (2017). Multivariate time series anomaly detection: A framework of Hidden Markov Models. *Applied Soft Computing*, 60, 229–240.
- Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2), 107–144.
- Martinez, C. M., Heucke, M., Wang, F. Y., Gao, B., & Cao, D. (2017). Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 19(3), 666–676.
- Meiring, G. A. M., & Myburgh, H. C. (2015). A review of intelligent driving style analysis systems and related artificial intelligence algorithms. *Sensors*, 15(12), 30653–30682.

- Mulder, J., & Leenders, R. Th. A. J. (2019). Modeling the evolution of interaction behavior in social networks: A dynamic relational event approach for real-time analysis. *Chaos, Solitons & Fractals*, 119, 73–85. ISSN: 0960-0779. <https://doi.org/10.1016/j.chaos.2018.11.027>.
- Pilny, A., Schechter, A., Poole, M. S., & Contractor, N. (2016). An illustration of the relational event model to analyze group interaction processes. *Group Dynamics*, 20(3), 181–195. <https://doi.org/10.1037/gdn0000042>
- Quintane, E., Pattison, P. E., Robins, G. L., & Mol, J. M. (2013). Short- and long-term stability in organizational networks: Temporal structures of project teams. *Social Networks*, 35(4), 528–540. ISSN: 03788733. <https://doi.org/10.1016/j.socnet.2013.07.001>
- Rollins, J. (2015). Online. Retrieved from <https://www.ibmbigdatahub.com/blog/why-we-need-methodology-data-science>
- Schechter, A., Pilny, A., Leung, A., Poole, M. S., & Contractor, N. (2017). Step by step: Capturing the dynamics of work team process through relational event sequences. *Journal of Organizational Behavior*, 1–19. <https://doi.org/10.1002/job.2247>
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). *FaceNet: A unified embedding for face recognition and clustering*. arXiv:1503.03832.
- Senin, P., Lin, J., Wang, X., Oates, T., Gandhi, S., Boedihardjo, A. P., Chen, C., & Frankenstein, S. (2018). Grammarviz 3.0: Interactive discovery of variable-length time series patterns. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(1), 1–28.
- Szegedy, C., Vanhoucke, V., Ioffe, S., & Shlens, J. (2015). *Rethinking the inception architecture for computer vision*. arXiv:1512.00567.
- Tselentis, D. I., Yannis, G., & Vlahogianni, E. I. (2016). Innovative insurance schemes: Pay as/how you drive. *Transportation Research Procedia*, 14, 362–371.
- Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., & Wu, Y. (2014). *Learning fine-grained image similarity with deep ranking*. arXiv:1404.4661.
- Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., & Hoi, S. (2020). *Deep learning for person re-identification: A survey and outlook*. arXiv:2001.04193.