Data-driven Disease Surveillance

Thesis for the Degree of Doctor of Natural Sciences

submitted by

Moritz Kulessa

Knowledge Engineering Group
Department of Computer Science
Technische Universität Darmstadt

Date of submission: 02/11/2021
Date of the oral examination: 17/12/2021

1^{st} Reviewer: Prof. Dr. Carsten Binnig
2^{nd} Reviewer: Prof. Dr. Johannes Fürnkranz
3^{rd} Reviewer: Prof. Dr. Myra Spiliopoulou

Kulessa, Moritz : Data-driven Disease Surveillance
Darmstadt, Technische Universität Darmstadt
Year of publication of the dissertation on TUprints: 2022
Date of oral examination: 17.12.2021

Erklärungen laut Promotionsordnung

§ 8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Datum und Unterschrift

Abstract

The recent and still ongoing pandemic of *SARS-CoV-2* has shown that an infectious disease outbreak can have serious consequences on public health and economy. In this situation, public health officials constantly aim to control and reduce the number of infections in order to avoid overburdening health care system. Besides minimizing personal contact through political measures, a fundamental approach to contain the spread of diseases is to isolate infected individuals. The effectiveness of the latter approach strongly depends on a timely detection of the outbreak as the tracking of individuals can quickly become infeasible when the number of cases increases. Hence, a key factor in the containment of an infectious disease is the early detection of a potential larger outbreak, commonly known as *outbreak detection*.

For this purpose, epidemiologists rely on a variety of statistical surveillance methods in order to maintain an overview of the current situation of infections by either monitoring confirmed cases or cases with early symptoms. Mainly based on statistical hypothesis testing, these methods automatically raise an alarm if an unexpected increase in the number of infections is observed. The practical usefulness of such methods highly depends on the trade-off between the ability to detect outbreaks and the chances of raising a false alarm. However, this hypothesis-based approach to disease surveillance has several limitations. On the one hand, it is a hand-crafted approach which requires domain knowledge to set up the statistical methods, especially if early symptoms are monitored. On the other hand, outbreaks of emerging infectious diseases with different symptom patterns are likely to be missed by such a surveillance system.

In this thesis, we focus on data-driven disease surveillance and address these challenges in the following ways. To support epidemiologists in the process of defining reliable disease patterns for monitoring cases with early symptoms, we present a novel approach to discover such patterns in historic data. With respect to supervised learning, we propose a fusion classifier which can combine the output of multiple statistical methods using the univariate time series of infection counts as the only source of information. In addition, we develop algorithms based on unsupervised learning which frame the task of outbreak detection as a general anomaly detection task. This even includes the surveillance of emerging infectious diseases. Therefore, we contribute a novel framework and propose a new approach based on sum-product networks to monitor multiple disease patterns simultaneously. Our results show that data-driven approaches are ideal to assist epidemiologists by processing large amounts of data that cannot fully be understood and analyzed by humans. Most significantly, the incorporation of additional information into the surveillance through machine learning techniques shows reliable and promising results.

Kurzfassung

Die jüngste und immer noch andauernde Pandemie von *SARS-CoV-2* hat gezeigt, dass ein Ausbruch einer Infektionskrankheit schwerwiegende Folgen für die Bevölkerung und die Wirtschaft haben kann. In dieser Situation sind die Gesundheitsämter ständig bemüht, die Zahl der Infektionen zu kontrollieren und zu reduzieren, um eine Überlastung des Gesundheitssystems zu vermeiden. Neben der Minimierung des persönlichen Kontakts durch politische Maßnahmen ist ein wesentlicher Ansatz zur Eindämmung der Ausbreitung von Krankheiten die Identifizierung von Infizierten. Die Effektivität des letztgenannten Ansatzes hängt stark von einer rechtzeitigen Erkennung des Ausbruchs ab, da die Verfolgung von Einzelpersonen schnell undurchführbar werden kann, wenn die Zahl der Fälle zunimmt. Daher ist ein Schlüsselfaktor bei der Eindämmung einer Infektionskrankheit die frühzeitige Erkennung eines potenziellen größeren Ausbruchs, allgemein bekannt als *Ausbruchserkennung*.

Zu diesem Zweck stützen sich Epidemiologen auf eine Vielzahl statistischer Überwachungsmethoden. Sie ermöglichen es einen Überblick über das aktuelle Infektionsgeschehen zu erhalten, indem sie entweder bestätigte Fälle oder Fälle mit frühen Symptomen von Infektionskrankheiten überwachen. Diese Methoden, die hauptsächlich auf statistischen Hypothesentests beruhen, lösen automatisch einen Alarm aus, wenn ein unerwarteter Anstieg der Zahl der Infektionen beobachtet wird. Ihr praktischer Nutzen hängt in einem hohen Maß von der Abwägung zwischen der Fähigkeit Ausbrüche zu erkennen und der Wahrscheinlichkeit eines Fehlalarms ab. Dieser hypothesengestützte Ansatz der Krankheitsüberwachung hat jedoch mehrere Nachteile. Zum einen handelt es sich um einen sehr händisch geprägten Ansatz, der Fachwissen zur Einrichtung der statistischen Methoden erfordert, insbesondere wenn frühe Symptome überwacht werden. Zum anderen werden Ausbrüche neu auftretender Infektionskrankheiten mit unterschiedlichen Symptommustern von einem solchen Überwachungssystem wahrscheinlich übersehen.

Um diese Herausforderungen zu bewältigen, konzentrieren wir uns in dieser Arbeit auf die datengesteuerte Überwachung von Krankheiten. Um Epidemiologen bei der Definition zuverlässiger Krankheitsmuster für die Überwachung von Fällen mit frühen Symptomen zu unterstützen, präsentieren wir einen neuartigen Ansatz, mit dem solche Muster in historischen Daten erfasst werden. Im Hinblick auf das überwachte Lernen stellen wir einen Fusionsklassifikator vor, der die Ergebnisse mehrerer statistischer Methoden kombinieren kann, wobei die univariate Zeitreihe der Infektionszahlen als einzige Informationsquelle dient. Darüber hinaus entwickeln wir Algorithmen auf der Grundlage des unüberwachten Lernens, die die Aufgabe der Erkennung von Krankheitsausbrüchen als ein allgemeines Problem der Anomalieerkennung auffassen. Dies schließt auch die Überwachung neu auftretender Infektionskrankheiten ein. Hierfür stellen wir einen neuartiges Framework zur Verfügung und präsentieren

einen neuen Ansatz auf der Grundlage von Sum-Product Networks, mit dem mehrere Krankheitsmuster gleichzeitigen überwacht werden können. Unsere Ergebnisse zeigen, dass datengesteuerte Ansätze ideal sind, um Epidemiologen bei der Verarbeitung großer Datenmengen zu unterstützen, die von Menschen nicht vollständig verstanden und analysiert werden können. Vor allem die Einbeziehung zusätzlicher Informationen in den Überwachungsprozess durch maschinelle Lernverfahren zeigt zuverlässige und vielversprechende Ergebnisse.

Acknowledgements

First of all, I express my deepest thanks to my supervisor Johannes Fürnkranz and my colleague Eneldo Loza Mencía for the guidance and support during the last three years. Particularly, I am very grateful to Johannes for continuing to mentor me even though he changed the university and to Eneldo for the very close collaboration among many different topics. Without them, this work would not have been possible.

I am very grateful to Carsten Binnig and Kristian Kersting, who had supervised my master's thesis and thus put me on the right track for my future research. I am also thankful for all of my co-authors, in particular Michael Rapp, Alejandro Molina, and Benjamin Hilprecht for the work they have put into the projects.

A warm thanks goes to the project partners of the ESEG (Erkennung und Steuerung epidemiologischer Gefahrenlagen) project. More precisely, the *Health Protection Authority of Frankfurt*, the *Hesse State Health Office and Centre for Health Protection*, the *Hesse Ministry of Social Affairs and Integration*, the *Robert Koch-Institut*, the *Epias GmbH* and the *Sana Klinikum Offenbach GmbH* who provided insight and expertise that greatly assisted the research. Especially, I thank Theresa Kocher, Alexander Ulrich, Birte Wagner, Madlen Schranz, Sonia Boender and Linus Grabenhenrich from the *Robert Koch-Institut* for their valuable feedback and the fruitful discussions.

Finally, I am forever thankful to my family and friends for supporting me during this important period of my life.

Contents

1

Introduction

The rise of infectious diseases can be traced back to the beginning of agriculture around 11,000 years ago. On the one hand, humans began to organize their life in larger and more dense populations while, on the other hand, they also came into much closer and more frequent contact with animals (Wolfe et al., 2007). This fostered the evolution of pathogens, mostly caused by animals (60%–80%), which then could spread among the human population (Morens and Fauci, 2013).

From a general point of view, the progression of an infectious disease in an individual can be categorized into three stages as visualized in Figure 1.1. First, the individual is exposed to the pathogen either through direct contact (e.g., droplet infection) or indirect contact (e.g., contaminated food). If not immediately detected and contained by the immune system, the pathogen begins to reproduce in its new host. After the pathogen has multiplied to a certain level, the infected individual starts to show symptoms. Depending on the impact on the body and the ability of the immune system to fight the disease, the infection can also turn into a severe illness or even death. The success and thus the persistence of infectious diseases in the human population is caused by continuously transmitting the disease. In particular, during the time period from the infection until showing first symptoms (also referred to as *incubation time*) the infection usually remains unknown while the disease may already be passed to other individuals.

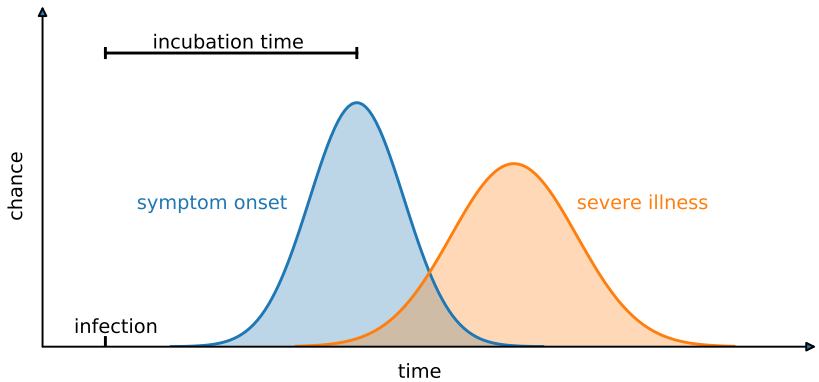


Figure 1.1: Stages of the progression of an infection (slightly adapted from Henning (2004)).

Throughout history, major outbreaks of infectious diseases have caused millions of deaths. Among the most well-known outbreaks is the *Great Influenza Pandemic* which took place between the years 1918 and 1920 and has killed approximately 40 million people worldwide (Barro et al., 2020). A distinctive feature of this pandemic was a high mortality rate among young and healthy people. This emphasizes the potential risk of commutable diseases independent of age or pre-existing medical conditions. Nowadays, the threat of severe outbreaks even increased because of growing international traffic due to tourism and trade. Especially, the recently emerged and still ongoing pandemic of *SARS-CoV-2* has shown that the situation can quickly turn into a global health threat with devastating consequences.

Epidemiologists typically classify the risk level of infectious diseases as *endemic*, *epidemic*, or *pandemic* based on the reproduction rate and the affected geographical area (Grennan, 2019). More precisely, during an endemic the disease spreads with a constant rate and is fixed to a particular region. The situation is under control but can turn into an epidemic when the number of infections increases unexpectedly. If the disease also spreads across international borders, it is termed a pandemic. While in an epidemic the spread of the disease still can be prevented, a pandemic is considered to be out of control with cases appearing all over the world. In this situation, the main goal is to reach herd immunity either through vaccination or by recovered infected individuals. However, mutations of the pathogen, mortality, and the long process of developing vaccines and immunizing the population can have a significant impact on reaching this goal.

1.1 Outbreak Detection

To ensure public health, it is crucial to control the number of cases of infectious diseases with high reproduction and mortality rate. This can be accomplished either by political measures (e.g., minimize social interactions) or by isolating infected individuals. For the latter approach, a timely detection of outbreaks is important since the tracking of individuals can quickly become infeasible when the number of infections increases. Ideally, outbreaks of novel emerging infectious diseases should be contained before they can spread globally and manifest in the world (Wolfe et al., 2007).

Two approaches to outbreak detection are distinguished in the literature. The traditional approach, to which we refer as *traditional surveillance*, is to monitor the number of confirmed infections. This confirmation often requires laboratory testing which can take up to several days until results are available, introducing a significant delay in the process of outbreak detection. A more recent approach is *syndromic surveillance* which aims to monitor the cases with early symptoms of an infectious disease. As it can be seen in Figure 1.1, the onset of symptoms allows a much more timely detection

of the infection. In contrast to traditional surveillance, cases do not need to be confirmed and thus can be obtained from many different data sources. For example, by tracking over-the-counter sales of specific pharmaceuticals or by observing the number of patients arriving at an emergency department with a particular medical condition (Buckeridge, 2007). Even though it allows a more timely detection, the signal contains much more noise since early symptoms are usually shared among many different diseases.

1.1.1 The Statistical Perspective

At the lowest level of traditional as well as syndromic surveillance, the tracked number of infections can be seen as a time series of count data. An example of such a time series is depicted in Figure 1.2. The main goal of disease surveillance is to identify any kind of changes as early as possible which indicate a shift from an endemic to an epidemic situation.

For this purpose, epidemiologists mainly rely on statistical surveillance methods. More precisely, for each disease and each geographical region under surveillance a particular statistical method is used to model the number of infections over time. Based on statistical hypothesis testing, these methods raise an alarm if an unexpected increase of cases is observed. Each of these alarms are then reported to public health officials which trigger a further investigation of the situation (Fricker, 2014).

Ideally, such methods are completely automated while still being able to be applied on a wide spectrum of different infections and disease patterns (Noufaily et al., 2019). However, if not chosen wisely or configured properly, they may also raise many false alarms which can overwhelm epidemiologists. In particular for large surveillance systems, where many time series for different diseases and locations are monitored simultaneously, the false alarm rate is a major concern and therefore highly determines the practical usefulness of an outbreak detection method (Shmueli and Burkom, 2010).

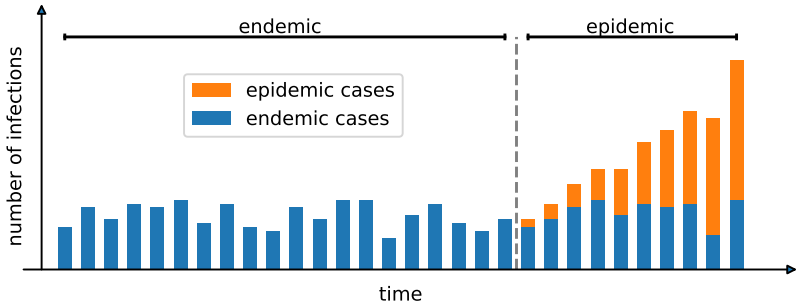


Figure 1.2: Exemplary time series of the number of cases for a particular infectious disease over time.

1.1.2 A Machine Learning Perspective

The configuration of hypothesis-driven approaches for disease surveillance can be a time-consuming and laborious process. Even though data availability has increased over recent decades, critical surveillance systems mainly remain on manually collected and coded data (Bansal et al., 2016). Especially, the definition of disease patterns for monitoring cases with early symptoms is usually based solely on expert knowledge of epidemiologists (Mandl et al., 2004).

An interesting avenue to improve disease surveillance is the use of machine learning. Compared to the hypothesis-driven approach, machine learning models are self taught and aim to improve their predictive performance based on historic data (Bi et al., 2019). More precisely, instead of formulating and evaluating a specific hypothesis, it can be seen as a search through the space of possible hypotheses in order to generalize from data (Witten and Frank, 2005). Rather than replacing statistical methods, the main idea is to complement current surveillance systems with data-driven models. This includes the design of algorithms which can support epidemiologists on the basis of data-driven suggestions, such as tools for a user-guided generation and comparison of disease patterns (Hopkins et al., 2017).

The main benefit of data-driven algorithms is to process large amounts of data that cannot fully be understood and analyzed by humans. On the one hand, it allows to include complex data sources into the surveillance process with which common statistical methods would struggle (Bi et al., 2019). In particular, novel data sources, such as web search data and social media updates, enable to put a much larger population under surveillance (Althouse et al., 2015). On the other hand, it allows to augment current hypothesis-driven approaches with additional information which may be difficult to incorporate. For example, epidemiologically relevant information (e.g., weather data, holidays, seasonality) can be integrated into the monitoring process by learning a model which combines the additional information with the output of the statistical method. As a result, more accurate and reliable notifications can be obtained (Hay et al., 2013).

Among major concerns for public health is also the emergence of new infectious diseases (Jones et al., 2008; Lombardo and Buckeridge, 2012). While traditional and syndromic surveillance only focus on the monitoring known infectious diseases, outbreak detection can also be treated as a general anomaly detection problem. In particular, an alarm can be triggered if the distribution of incoming data changes in an unforeseen and unexpected way. Combined with interpretable explanations, data-driven surveillance can be an important tool to increase situational awareness about emerging diseases (Simonsen et al., 2016). For example, instead of monitoring a particular disease pattern in an emergency department, complete patient information can be analyzed to detect an increase of cases among any kind of clinical picture.

1.2 Contributions and Structure of Work

Reliable and early outbreak detection is a key aspect in the containment of infectious diseases. In this thesis, we concentrate on improving disease surveillance based on data-driven models. Our proposed approaches are designed to support epidemiologists with respect to traditional surveillance, syndromic surveillance and even the surveillance of emerging diseases.

The structure of this work is described in the following:

Chapter 2: This thesis will continue with a brief introduction into statistical theory. In particular, we focus on probability theory to model count data and statistical hypothesis testing which is required to understand most algorithms developed in the area of syndromic surveillance.

Chapter 3: Thereafter, basic concepts of machine learning and anomaly detection are explained, including a short description of machine learning algorithms which have been used throughout this thesis.

Chapter 4: We then proceed by providing a broad overview of disease surveillance. This includes a formal definition of traditional and syndromic surveillance, including concepts of data acquisition and evaluation. Afterwards, common statistical surveillance methods are introduced and the relation to the area of machine learning is discussed.

Chapter 5: The definition of disease patterns for syndromic surveillance is often challenging, as early symptoms are usually shared among many diseases and a particular disease can have several clinical pictures in the early phase of an infection. In our first contribution, we present an approach which can support epidemiologists to define such patterns. The general idea is to identify indicators in a health-related data source which correlate with the reported number of infections in the respective geographic region. This contribution is published in:

- M. Rapp, M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Correlation-based discovery of disease patterns for syndromic surveillance. *Frontiers in Big Data*, 4, 2022

Chapter 6: The practical usefulness of statistical surveillance methods highly depends on the trade-off between the detection rate of outbreaks and the chances of raising a false alarm. Our second contribution aims to improve these statistical methods by fusing several of these based on a machine learning technique which is known as *stacking*. In addition, for comparison and evaluation, a new measure is introduced which captures the performance of an outbreak detection method with respect to a low rate of false alarms more precisely than previous works. These contributions are published in:

- M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Improving outbreak detection with stacking of statistical surveillance methods. In *Workshop Proceedings: Epidemiology Meets Data Mining and Knowledge Discovery (held in conjunction with ACM SIGKDD 2019)*, 2019b

- M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Improving the fusion of outbreak detection methods with supervised learning. In *Proceedings of the 16th International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 55–66, 2020

Chapter 7: Most research in syndromic surveillance mainly focuses on the monitoring of specific, known diseases, putting the focus on the definition of the disease pattern under surveillance. Until now, only little effort has been devoted to what we call *non-specific* syndromic surveillance that is the use of all available data for detecting any kind of outbreak. This even includes the monitoring of emerging infectious diseases. As our third contribution in this thesis, we revisit published approaches for non-specific syndromic surveillance and present a set of simple statistical modeling techniques which can serve as baselines for more elaborate machine learning approaches. In addition, a unified framework based on global and local modeling techniques is proposed. These contributions are published in:

- M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Revisiting non-specific syndromic surveillance. In *Proceedings of 19th International Symposium on Intelligent Data Analysis*, pages 128–140, 2021a

- M. Kulessa, E. Loza Mencía, and J. Fürnkranz. A unifying framework and comparative evaluation of statistical and machine learning approaches to non-specific syndromic surveillance. *Computers (Special Issue: Artificial Intelligence for Health)*, 10(3):32, 2021b

Chapter 8: Our work on non-specific syndromic surveillance had shown that our proposed statistical baselines already achieve very competitive results and often outperform more elaborate algorithms. For our last contribution, we enhanced the concept of the baselines by modeling the joint probability distribution of syndromic count data with sum-product networks, a generative machine learning algorithm. Such a model is able to capture correlations in the monitored data and even allows to consider environmental factors during the monitoring process which might have an influence on the number of infections. In contrast to the conventional use of sum-product networks, we present a new approach to detect anomalies by evaluating p-values on the learned model. These contributions are published in:

- M. Kulessa, B. Wittelsbach, E. Loza Mencía, and J. Fürnkranz. Sum-product networks for early outbreak detection of emerging diseases. In

Proceedings of the 19th International Conference on Artificial Intelligence in Medicine, pages 61–71, 2021c. *(nominated for best student paper award)*

Chapter 9: This chapter summarizes the findings and results of this work. Based on that directions for future work are given.

2

Background on Statistics

To understand most of the algorithms developed in disease surveillance, a fundamental understanding of statistical theory is required which is reviewed in this chapter. In particular, we focus on probability theory (cf. Section 2.1) and how it can be used to model count data (cf. Section 2.2). Given such a statistical model, hypotheses can be tested (cf. Section 2.3) in order to validate certain questions of interest.

2.1 Probability Theory

We start with a brief review of probability theory which is mainly based on the book of Shao (2003). It serves mainly the purpose of clarifying our terminology and notation, and can be safely skipped by readers familiar with elementary statistics.

Probability theory aims to model the likelihood of outcomes of a random experiment in terms of probabilities. We speak of a *random experiment*, when the outcome of the experiment is uncertain. For example, rolling a dice can be considered as such an experiment since it is unknown which exact value the dice will take before it is thrown. The set of all possible outcomes of a random experiment is referred to as the *sample space* which is denoted by Ω. With respect to our example of rolling a dice, the sample space can defined as follows:

$$\Omega = \Big\{ \boxdot, \boxdot, \boxdot, \boxdot, \boxdot, \boxdot \Big\} \tag{2.1}$$

In case of rolling two dice at the same time, Ω would contain a total of 36 elements.

On the sample space Ω, an event of interest $A \subseteq \Omega$ can be specified. Considering our example, we could be interested in the event $\{\boxdot, \boxdot, \boxdot\}$ which represents that the outcome of the rolled dice is odd. A set of events $\mathcal{A}$ is referred to as an *event space* or *σ-algebra* if the following conditions hold:

1. $\Omega \in \mathcal{A}$
2. If $A \in \mathcal{A}$ then $A^C \in \mathcal{A}$
3. If $A_i \in \mathcal{A}$, $i = 1, 2, \ldots$ then $\bigcup A_i \in \mathcal{A}$

Note that the set of events $\mathcal{A}$ is closed under the complement as well as closed under countable unions. In case that the sample space is continuous (e.g., $\Omega = \mathbb{R}$) the σ-algebra is also called *Borel* σ-algebra. The pair $(\Omega, \mathcal{A})$ is called a *measurable space*.

With respect to rolling a single dice, the largest event space can be constructed by taking the power set $\mathcal{A} = \mathcal{P}(\Omega)$ which is the set of all subsets of Ω. Conversely, if we are only interested in the event {⚀, ⚂, ⚅}, the minimal event space would contain the following events:

$$\mathcal{A} = \left\{ \emptyset, \{⚀, ⚂, ⚅\}, \{⚁, ⚃, ⚄\}, \{⚀, ⚁, ⚂, ⚃, ⚄, ⚅\} \right\} \tag{2.2}$$

Given an event space $\mathcal{A}$, a function $P : \mathcal{A} \to [0, 1]$ is called a *probability measure* if the following conditions hold:

1. $P(\emptyset) = 0$
2. $P(\Omega) = 1$
3. If $A_1 \cap A_2 = \emptyset$ then $P(A_1 \cup A_2) = P(A_1) + P(A_2)$ where $A_1, A_2 \in \mathcal{A}$

More precisely, a probability is assigned to each event $A \in \mathcal{A}$. In terms of a continuous sample space, such as $\Omega = \mathbb{R}$, probabilities are assigned to all intervals $A \in \mathcal{A}$. Considering our dice example, the mapping of the probability function for the event space specified in Equation 2.2 over the sample space specified in Equation 2.1 could be:

$$P(\emptyset) = 0\% \qquad P\Big(\{⚀, ⚂, ⚅\}\Big) = 50\%$$
$$P(\Omega) = 100\% \qquad P\Big(\{⚁, ⚃, ⚄\}\Big) = 50\%$$

2.1.1 Random Variables and Distributions

From the example above we can observe that the use of the measurable space $(\Omega, \mathcal{A})$ can be inconvenient as always the complete set $A \in \mathcal{A}$ needs to be specified. Especially, this can be an obstacle when the sample space Ω becomes more complex. For simplification, the measurable space $(\Omega, \mathcal{A})$ is often mapped onto a simpler measurable space $(\Lambda, \mathcal{G})$ using a function $g : \Omega \to \Lambda$. Such a function is called a *measurable function* and has the inverse image

$$g^{-1}(B) = \{g \in B\} = \{\omega \in \Omega : g(\omega) \in B\}, \qquad B \subset \Lambda.$$

In particular, if $\Lambda = \mathbb{R}$ and $\mathcal{G}$ is a *Borel* σ-algebra then this kind of function is called a *random variable* which is commonly denoted as $X : \Omega \to \mathbb{R}$. The use of a random variable is quite convenient when we are only interested in some of the subsets in the sample space Ω. Moreover, the mapped space consists of numbers which are easier to handle than sets. With respect to our dice example in which we are interested in whether the outcome of the dice is odd, we could define the following random variable:

$$X(\omega) = \begin{cases} 1, & \text{if } \omega \in \{\boxdot, \boxdot, \boxdot\} \\ 0, & \text{otherwise} \end{cases}$$

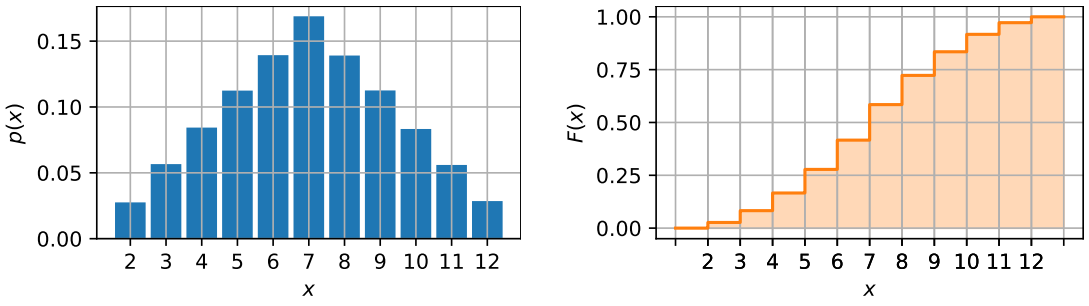
As it can be seen, the random variable X returns 1 is the outcome of the dice is odd and otherwise 0. Instead of writing $P(\{\boxdot, \boxdot, \boxdot\})$, we can now define $P(X^{-1}(1))$ as $P(X = 1)$ in order to obtain the probability of rolling an odd number with the dice. In particular, $P \circ X^{-1}$ is called the *distribution* of X.

Discrete random variables. A random variable X is considered to be discrete, if the image of X is countable. With respect to the aforementioned example, the image of the random variable only contains two values $\{0, 1\}$. To each specific value of the image a particular probability can be assigned which is often denoted as the *probability mass function*:

$$p(x) = P(X = x) = P(X^{-1}(x))$$

If an order among the discrete values $x_1 < x_2 < \ldots$ in the image of the random variable X exists, the *cumulative distribution function* can be denoted as:

$$F(x) = \begin{cases} \sum_{i=1}^{n} p(x_i), & \text{if } x_n < x < x_{n+1} \\ 0, & \text{if } -\infty < x < x_1 \end{cases}$$



(a) Probability distribution.

(b) Cumulative probability distribution.

Figure 2.1: Probability and stepwise cumulative distribution for a discrete random variable which represents the sum of two rolled dice.

As an example, assume our random experiment consists of rolling two dice at the same time and our random variable X maps on the space $\{2, 3, \ldots, 12\}$, which is the sum of the two rolled dice. An exemplary distribution of X is depicted in Figure 2.1a, whereas the respective cumulative distribution is shown in Figure 2.1b.

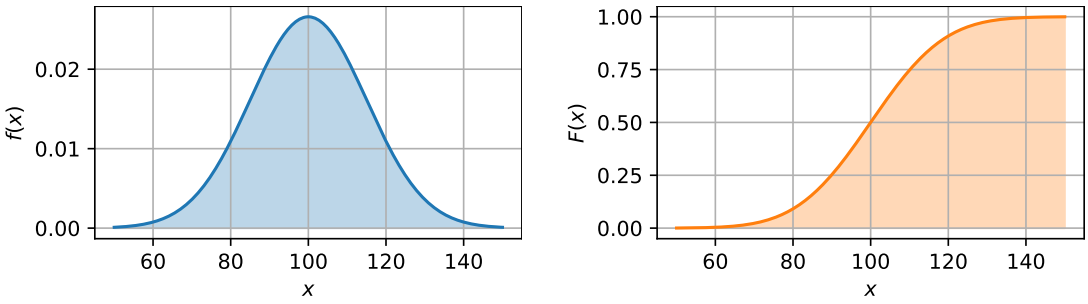
Continuous random variables. In contrast, a random variable X is considered to be continuous, if the image of X is uncountable infinite. For this kind of random variable the cumulative distribution function is defined as $F(x) = P(X \leq x)$. As there are infinite many values in the image of X, the probability of observing a specific value is zero. On such a random variable probabilities can only be assigned to intervals of the image. Nevertheless, to still obtain a point of reference for the likelihood for a particular value x the *probability density function* is used which is the deviation of the cumulative distribution:

$$f(x) = \frac{\partial F(x)}{\partial x}$$

Vice versa the cumulative distribution function can be defined with the use of the probability density function:

$$F(x) = \int_{-\infty}^{x} f(u)du$$

For example, let us assume that the continuous random variable X maps on the intelligence quotient. An exemplary probability density function $f(x)$ is depicted in Figure 2.2a, whereas the respective cumulative distribution is shown in Figure 2.2b.



(a) Probability distribution. **(b)** Cumulative probability distribution.

Figure 2.2: Distribution of the probability density and cumulative probability distribution of a continuous random variable which represents the intelligence quotient.

2.1.2 Computation of p-values

Once the probability distribution for a random variable X is known, we have an intuition about which outcomes are likely to occur. More precisely, we know what to expect for future trials of the random experiment which allows us to reason about upcoming events. In this thesis, we are interested in whether a new observed value x of random variable X is likely to be generated by the given probability distribution. Therefore, we can compute the following statistics for the observed value x based on the respective cumulative distribution function $F(x)$:

- The left-sided p-value represents the probability of observing smaller values which are at least as extreme as the observed value x and can be obtained by computing the probability:

$$p_{\text{left-sided}} = P(X \geq x)$$

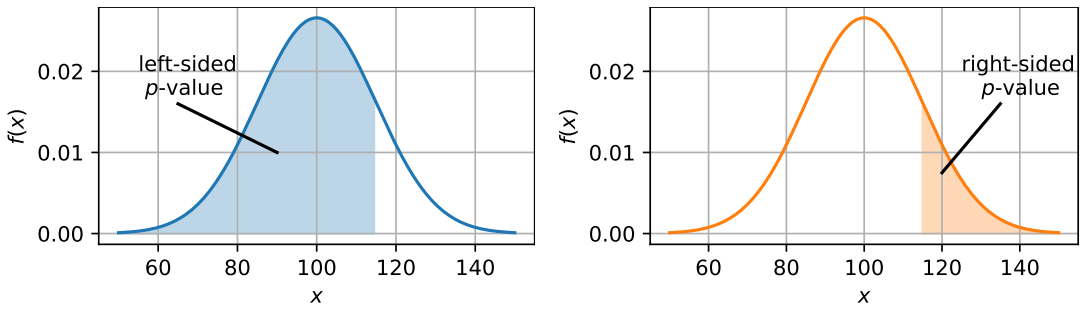
- The right-sided p-value represents the probability of observing higher values which are at least as extreme as the observed value x and is computed as:

$$p_{\text{right-sided}} = P(X \leq x)$$

- The two-sided p-value represents the probability of observing values which are at least more extreme as the observed value x. It can be obtained by computing the probability:

$$p_{\text{two-sided}} = 2 \cdot \min\{P(X \geq x), P(X \leq x)\}$$

For example, let us assume that a continuous random variable for the intelligence quotient is distributed according to the continuous probability distribution which is



(a) Left-sided p-value.

(b) Right-sided p-value.

Figure 2.3: Covered area of exemplary p-values on a continuous probability distribution for x equal to 115.

depicted in Figure 2.2a. If x is equal to 115, the probability for the left-sided p-value of observing an equal or lower intelligence quotient than 115 is close to 84% while the right-sided p-value would obtain a probability close to 16%. For the two-sided p-value we would obtain a probability close to 32%. The respective areas over which the p-values are computed is depicted in Figure 2.3.

2.2 Modeling Count Data with Probability Distributions

A fundamental concept of this thesis is the modeling of *count data*. From a theoretical perspective, a count can take values of $\mathbb{N}_0$ and can be interpreted as the number of occurrences of a specific event in a given time interval. For example, one could count the number of influenza cases reported in a single day. We refer to count data as a collection of such counts over time, each measured on a different time interval. Figure 2.4 visualizes this concept where the x-axis represents the time and the blue dots denote the events. The number of dots in each time interval visualized by the orange lines represents the respective count.

In terms of probability theory, a count can be seen as the outcome of a random experiment. For this setting, the random variable is discrete since the space $\mathbb{N}_0$ is countable (cf. Section 2.1.1). Thus, it is possible to model count data with a probability distribution. However, for most scenarios in the real world the true underlying probability distribution is unknown. Usually the probability distribution is estimated based on a finite random sample $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ of the image of random variable X.

In this thesis, we mainly rely on parametric probability distributions which have been widely used in disease surveillance before (cf. Section 4.4). A parametric probability distribution P_θ is solely defined by its parameters θ and makes basic assumptions about how the underlying data is distributed. For our application scenario, these parametric probability distributions model basic properties of count data. The parameters θ are usually obtained by computing the maximum likelihood estimate over the given random sample $\mathcal{X}$.

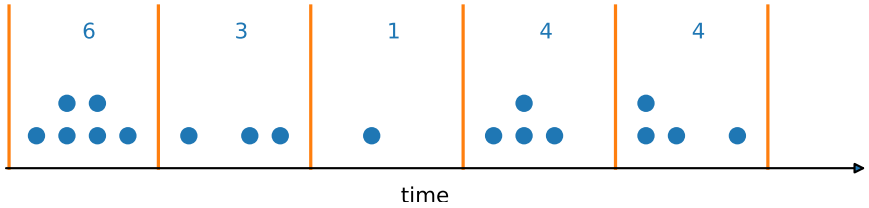


Figure 2.4: Exemplary collection of count data where the blue dots denotes the events of random experiment over time and the orange lines represent the time intervals in which the events are counted.

Table 2.1: Overview of the described parametric probability distributions.

name	type	distribution	
Poisson	discrete	$p(x) = \lambda^x e^{-\lambda}/x!$	$x \in \mathbb{N}_0$ $\lambda > 0$
negative binomial	discrete	$\binom{x-1}{r-1} p^r (1-p)^{x-r}$	$x \in \mathbb{N}_0$ $p \in [0,1]$ $r > 0$
Gaussian	continuous	$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2}$	$x \in \mathbb{R}$ $\mu \in \mathbb{R}$ $\sigma > 0$

An overview of the characteristics of the following described probability distributions is provided in Table 2.1. To fit the parameters of these distributions via maximum likelihood estimation, we rely on the empirical mean μ and the empirical standard variance σ^2 which can be computed over the sample $\mathcal{X}$ as:

$$\mu = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} x_i \qquad\qquad \sigma^2 = \frac{1}{|\mathcal{X}| - 1} \sum_{i=1}^{|\mathcal{X}|} (x_i - \mu)^2$$

2.2.1 Poisson Distribution

A natural choice to model count data is the Poisson distribution. It can be defined with the following probability mass function:

$$p(x) = \lambda^x e^{-\lambda}/x! \qquad\qquad \begin{array}{l} x \in \mathbb{N}_0 \\ \lambda > 0 \end{array}$$

This distribution assumes that the underlying events are independent of each other and occur with a constant probability over time. The main characteristic of the Poisson distribution is that the mean and the variance of modeled data is always the same which is also known as the *equidispersion criterion* (Hilbe, 2011). The maximum likelihood estimation for the parameter λ is the empirical mean μ.

2.2.2 Negative Binomial Distribution

In fact, the assumptions of the Poisson distribution are hardly met in most of the application scenarios in the real world (Hilbe, 2011). The most common problem

when modeling count data is *overdispersion* which occurs when the variance of the data is greater than the mean. One of the reasons which can lead to overdispersed data is *occurrence dependence* which violates the assumption of independence among the events (Winkelmann, 2008). In particular, an observed event increases the probability of observing an additional event over a certain time period. A good example for occurrence dependence appears in count data of infectious diseases because infected people are likely to transmit the disease to other people which increases the chance to observe new cases.

The negative binomial distribution offers a wider range of variability and flexibility to model count data than the Poisson distribution. In particular, the parameters of the negative binomial distribution allow to adjust the variance of the model for the count data which enables to also adapt to overdispersion. The negative binomial distribution is a discrete probability distribution which can be defined with the following probability mass function:

$$p(x) = \binom{x-1}{r-1} p^r (1-p)^{x-r} \qquad \begin{array}{l} x \in \mathbb{N}_0 \\ r > 0 \\ p \in [0,1] \end{array}$$

The standard way to fit the parameters of the negative binomial distribution is to use the maximum likelihood estimate:

$$r = \frac{\mu^2}{\sigma^2 - \mu} \qquad\qquad p = \frac{r}{r+\mu}$$

2.2.3 Gaussian Distribution

Even though count data consists of distinct values, it is quite common that statisticians interpret the data as continuous (Hilbe, 2011). We include the Gaussian distribution in the analysis of this thesis since it is a common approach to model count data in disease surveillance methods (cf. Section 4.4). The Gaussian distribution can be described by its probability density function:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2} \qquad \begin{array}{l} x \in \mathbb{R} \\ \mu \in \mathbb{R} \\ \sigma > 0 \end{array}$$

Strictly speaking, the Gaussian distribution is not suitable to represent count data since it also models negative values. However, it can serve as a reference point when comparing algorithms in an evaluation. The maximum likelihood estimation for the parameters is the empirical mean μ and variance σ^2 itself.

2.3 Statistical Hypothesis Testing

Statistical hypothesis testing is a formal approach to verify a statement about data of which only a sample is obtained. Since the true underlying probability distribution of the data is unknown, the statistical relationships in the sample can be interpreted either as they have occurred by chance or they reflect a relationship in the data (Jhangiani et al., 2019). Therefore, the statement of concern is usually formulated as two opposing hypothesis:

Null hypothesis (H_0): The statement which is believed to be true.

Alternative hypothesis (H_1): The logical counterpart to the null hypothesis.

The outcome of the statistical hypothesis test is either accepting the null hypotheses or rejecting it, in which case the alternative hypothesis is considered to be correct. In order to decide between the hypothesis, a *test statistic* T is computed over the sample under the assumption that the null hypothesis is true. Depending on the used statistical test and its assumptions at hand, the test statistic is distributed according to a known probability distribution $T \sim P_{H_0}$ representing the likelihood of accepting the null hypothesis (cf. Section 2.1.1). For decision making, the sample space of this probability distribution is divided into two complement regions: (1) The region of acceptance R_0 and (2) the region of rejection R_1, also referred to as the *critical* region. If the computed test statistic falls into R_0, the null hypothesis is accepted (Lehmann and Romano, 2005). When a statistical test is performed, two kinds of errors can be made:

Type I error: Falsely accepting the null hypotheses when it is actually false.

Type II error: Falsely rejecting the null hypothesis when it is actually true.

Ideally, the probability for both types of errors is kept at a minimum but this is not possible as they directly influence each other. However, depending on the context of the statistical test, the minimization of the probability for one type of error might be more important than the other one. In order to control the trade-off between both errors, a significance level α is specified which assigns a probability to incorrectly rejecting the null hypothesis. Given the probability distribution P_{H_0} of the test statistic, the critical region R_1 need to be chosen such that the following equation holds (Lehmann and Romano, 2005):

$$P_{H_0}(R_1) \leq \alpha$$

The choice for the significance level α is often arbitrary since there is not a precise limit to the probability of type I error that can be tolerated. To avoid this, the specification of the significance level α can be omitted by directly reporting the p-value of the test

(cf. Section 2.1.2). The p-value represents the smallest significance level at which the null hypothesis is rejected. In particular, Amrhein et al. (2019) argue that p-values are superior to pre-specified significance levels since the strict binary decision of significance can lead to wrong interpretations. In contrast, the p-value contains information about the extent that the null hypothesis is contradicted.

2.3.1 Test Statistics

Given the formal approach to statistical hypothesis testing described in the previous section, we only need to formulate the hypotheses and choose a suitable test statistic. A variety of statistical hypothesis tests exist each tailored to answer certain types of hypotheses. In the following, we review a few of these test statistics based on the formulated hypotheses which are made in this work.

Out-of-distribution test. In disease surveillance it is common to use probability distributions to model count data as described in Section 2.2. Under the assumption that the fitted statistical model represents the count data correctly, a new observed count can be checked whether it is likely to be generated by this probability distribution. This task can be interpreted as a statistical hypothesis test for which we can formulate the following hypotheses:

- Null hypothesis H_0: The count has been generated by the given probability distribution.
- Alternative hypothesis H_1: The count has *not* been generated by the given probability distribution.

Note that in contrast to conventional statistical hypothesis testing, we assume that the fitted statistical model represents the true underlying probability distribution of the count data. Therefore, it can be directly used as the probability distribution P_{H_0} of the test statistic on which the p-value for the observed count can be computed.

Table 2.2: Contingency table.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{\dots}$	$\mathbf{x}_k$
$\mathbf{y}_1$	$o_{1,1}$	$o_{1,2}$	$\dots$	$o_{1,k}$
$\mathbf{y}_2$	$o_{2,1}$	$o_{2,2}$	$\dots$	$o_{2,k}$
$\mathbf{\dots}$	$\dots$	$\dots$	$\dots$	$\dots$
$\mathbf{y}_m$	$o_{m,1}$	$o_{m,2}$	$\dots$	$o_{m,k}$

Table 2.3: Example for contingency table.

	vaccine	**placebo**	$\sum$
infection	3	56	59
no infection	397	344	741
$\sum$	400	400	800

Pearson's chi-square independence test. The chi-square test of independence by Pearson (1900) is used to examine the relationship between two discrete random variables, X with outcomes $\{x_1, x_2, \ldots, x_k\}$ and Y with outcomes $\{y_1, y_2, \ldots, y_m\}$. The co-occurrences of these outcomes can be summarized in a contingency table by counting their respective frequencies in the sample. Such a table is shown in Table 2.2 where $o_{i,j}$ represents the number of times that x_i and y_j appeared together. The hypotheses for this test can be formulated as follows:

- Null Hypothesis H_0: The random variables X and Y are independent of each other.
- Alternative Hypothesis H_1: Correlation among the random variables X and Y exists.

For example, suppose a study has been conducted to evaluate the efficacy of a vaccine. In this study, 3 out of 400 of the vaccinated participants have been infected with the respective disease whereas in the control group, who only obtained a placebo, 56 out of 400 participants contracted the disease. The contingency table for this example is displayed in Table 2.3. By performing the above mentioned statistical hypothesis test, the effectiveness of the vaccine can be evaluated.

Based on the expected frequencies of each cell in the contingency table

$$e_{i,j} = \frac{(\sum_{r=1}^{m} o_{r,j})(\sum_{s=1}^{k} o_{i,s})}{(\sum_{r=1}^{m} \sum_{s=1}^{k} o_{r,s})}$$

the test statistic for the chi-square independence test is distributed according to a chi-square distribution with $(m-1)(k-1)$ degrees of freedom and can be defined as (McHugh, 2013):

$$t = \sum_{i=0}^{m} \sum_{j=0}^{k} \frac{(o_{i,j} - e_{i,j})^2}{e_{i,j}} \qquad t \sim \mathcal{X}^2_{(m-1)(k-1)}$$

Fisher's exact test. The Fisher's exact test by Fisher (1934) is used for the same hypothesis as for the chi-square independence test. However, the drawback of the chi-square test of independence is that the test statistic maps the discrete number of co-occurrences onto an approximated continuous distribution. To obtain precise results for this test statistic usually a large sample is required.

The Fisher's exact test overcome this problem by using a discrete probability distribution for the test statistic and, therefore, it is well-suited for small sample sizes (Kim,

2017). In its original form, it can only be applied on a 2×2 contingency table. Under the given null hypothesis, the Fisher's exact test statistic is distributed according to a hyper-geometrical distribution and can be defined as:

$$t = o_{1,1} \qquad t \sim Hyp(K, n, N) \qquad \begin{aligned} K &= o_{1,1} + o_{1,2} \\ n &= o_{1,1} + o_{2,1} \\ N &= o_{1,1} + o_{1,2} + o_{2,1} + o_{2,2} \end{aligned}$$

2.3.2 Testing Multiple Statistical Hypotheses

In practice, it is often required to perform a set of statistical hypothesis tests at the same time. For example, a set of out-of-distribution tests could be used to monitor multiple infectious diseases at once. In this case, the overall null hypothesis is usually that none of the underlying tests is significant. However, since the p-value of each underlying test is distributed uniformly between 0% and 100% under the assumption that the respective null hypothesis is true, the likelihood of observing at least one extreme p-value in the whole set increases with the number of performed tests. For example, if ten independent statistical hypothesis tests with a significance level of 5% are performed, the chance of observing at least one significant result for one of the tests is $1 - (1 - 0.05)^{10} \approx 40\%$ even though the null hypothesis might be true. Hence, in the case that the multiplicity is disregarded the chance of obtaining type I errors increases from 5% to 40%. This effect is also known as the *multiple-testing problem* and need to be taken into account when performing multiple comparisons (Lehmann and Romano, 2005).

Based on this problem, several methods have been proposed to aggregate p-values which ideally provide a composite p-value for the overall null hypothesis. These methods can be mainly separated into two groups: (1) Methods which assume independence among the underlying tests and; (2) Methods which can take dependence among the tests into account. In the following, a few of these methods for each group are reviewed which are relevant for this thesis.

Aggregation of independent hypothesis tests. If the underlying tests are independent of each other, the problem of aggregating k independent p-values $p_1, \ldots, p_k$ can be formulated as a statistical hypothesis test. Following Heard and Rubin-Delanchy (2018), the hypotheses for this test can be formulated as:

- Null hypothesis H_0: The p-values are independently uniformly distributed $p_i \sim \text{Uniform}[0, 1]$ for $(1, \ldots, k)$.
- Alternative hypothesis H_1: The p-values might follow a unknown distribution $p_i \sim f_i$ for $(1, \ldots, k)$.

For a detailed analysis of test statistics for this hypothesis, we refer to Heard and Rubin-Delanchy (2018). In the following, we only consider test statistics which are relevant for this thesis:

Tippett's method: This test statistic is based on the minimal p-value which is closely related to the *Bonferroni correction*. It favors the most extreme observation among the underlying tests and, therefore, sensitive to small p-values. Under the given null hypothesis, the test statistic is distributed according to a Beta distribution and can be defined as followed:

$$t = min(p_1, \ldots, p_k) \qquad\qquad t \sim Beta(1, k)$$

Stouffer's method: Rather than combining the p-values directly, the Stouffer's method first transforms the p-values into z-scores. Under the given null hypothesis, the test statistic is distributed according to a Gaussian distribution and can be defined as followed where ϕ^{-1} is the inverse of the cumulative standard Gaussian distribution:

$$t = \sum_{i=1}^{k} \phi^{-1}(p_i) \qquad\qquad t \sim N(0, k)$$

Fisher's method: Probably the most well-known test statistic to combine several p-values is the Fisher's method. Under the given null hypothesis, the test statistic is distributed according to a chi-square distribution and can be defined as followed:

$$t = -2 \sum_{i=1}^{k} log(p_i) \qquad\qquad t \sim \mathcal{X}^2_{2k}$$

Aggregation of dependent hypothesis tests. If correlation among the test statistics of the underlying hypothesis tests exists and the dependency structure is known, it can be used reduce the chance of type I errors. As an extreme example consider to perform the exact same hypothesis test simultaneously. In this case, the chance for obtaining type I errors would be equal to the specified significance level α of the test. Another example relates to the area of neuro-imaging, in which cluster-extent based thresholding techniques are used to identify areas of brain activity (Lindquist and Mejia, 2015). In this case, it is assumed that the statistical hypothesis tests are spatially correlated.

However, often the dependency structure between the underlying tests is unknown. Without any prior information about the correlations, the framework proposed by Leek and Storey (2008) can be used, which captures the dependencies in the data

before the statistical tests are conducted. This information is then used to adjust the data in a way so that the actually performed statistical tests are independent of each other. In contrast, Vovk and Wang (2020) focus on simple aggregation functions and recommend to use the *harmonic mean* in case of substantial dependence among the merging p-values and even suggest to use the *geometric* or the *arithmetic mean* if the dependence is really strong. In this thesis we consider the weighted versions of these three merging functions:

$$\bar{p}_{average} = \sum_{i}^{k} w_i p_i \qquad \bar{p}_{geometric} = \prod_{i}^{k} p_i^{w_i} \qquad \bar{p}_{harmonic} = \sum_{i}^{k} w_i \Big/ \sum_{i}^{k} \frac{w_i}{p_i}$$

3

Background on Machine Learning

Machine learning is one of the main components of this thesis. In this chapter, we will first discuss the basic concept of machine learning which includes learning scenarios (cf. Section 3.1), model selection (cf. Section 3.2) and evaluation (cf. Section 3.3). Next, we will briefly introduce a set of supervised learning algorithms (cf. Section 3.4) which have been used throughout this thesis and sum-product networks (cf. Section 3.5) which are used in Chapter 8. Finally, we take a closer look at the area of anomaly detection in Section 3.6 and stream mining in Section 3.7.

3.1 Learning Scenarios

Similar to how a human learns from experience, a machine can learn from data by analyzing and extracting patterns. In these terms data can be described as a finite sample $\mathcal{D}$ of a set of explanatory random variables $X_1, \ldots, X_m$ whose underlying probability distribution is unknown. Depending on the learning scenario at hand, machine learning algorithms are designed to capture certain relationships of interest in the data. In this thesis, we focus on the following learning scenarios:

Supervised learning: For this learning scenario an additional target random variable Y is given whose value need to be predicted solely based on the values of the random variables $X_1, \ldots, X_m$. The data for this task can be represented as $\mathcal{D} = \{(x_i, y_i) \mid x_i \in X_1 \times \ldots \times X_m \wedge y_i \in Y\}$ where $i = 1 \ldots n$ denotes the index of a particular instance. Based on this data set a supervised machine learning algorithm learns a mapping $f : X_1, \ldots, X_m \to Y$ which can assign to a given instance a particular value. In case Y is a discrete random variable, this task is known as *classification* while for a continuous random variable it is named *regression*.

Unsupervised learning: Instead of only capturing relationships to one particular target random variable, unsupervised machine learning algorithms aim to extract general patterns among the random variables $X_1, \ldots, X_m$. In this case no target

random variable is given for which reason the finite sample can be described as the data set $\mathcal{D} = \{x_i \mid x_i \in X_1 \times \ldots \times X_m\}$ where $i = 1 \ldots n$ and n is number of instances. For example, *clustering* algorithms separate the instances of this data set into distinct groups based on their similarity while in the area of *generative machine learning*, algorithms learn the underlying joint probability distribution $P(X_1, \ldots, X_m)$ of the data set.

The main strength of machine learning is that the extracted relationships of interest not only cover the given data set but may also apply to future observations of the random variables $X_1, \ldots, X_m$.

3.2 Model Selection

The most straightforward approach to machine learning is to extract all relationships of interest which perfectly describe the learning task on the given data set $\mathcal{D}$. However, learning an exact mapping of the data is not ideal since the data set usually contains noise and the extracted patterns should also generalize to newly obtained data. In particular, we can distinguish between two types of error sources when learning a model:

Bias: On the one hand, an error can be introduced by learning a model which is too simple to adequately capture the relationships of interest in the data set $\mathcal{D}$. This error can be reduced by increasing the complexity of the model. Note that the used machine learning algorithm might be limited in their model complexity due to its design.

Variance: On the other hand, an error can be caused due to overly complex models. Keep in mind that the data set $\mathcal{D}$ is only a finite sample which possibly contains relationship due to the variance in the sampling. Capturing these chance patterns can have an impact on the performance.

The interaction between these errors is commonly termed as the *bias-variance trade-off* and is visualized in Figure 3.1. In particular, if the error of the bias surpasses the error of the variance, we speak of *underfitting* while vice-versa it is named *overfitting*. Ideally, the learned model neither underfit nor overfit by finding a compromise between these errors (Japkowicz and Shah, 2011). A common approach to avoid overfitting is a technique called *regularization* (Bishop, 2006). In order to avoid to fit the model on the noise in the data set $\mathcal{D}$, a regularization term is used during the learning process which penalizes overly complex models. This usually results in a more general and interpretable model. In case we have to decide between two models which perform equally well, the simpler should be preferred over the complex according to the principle of *Occam's Razor* (Blumer et al., 1987).

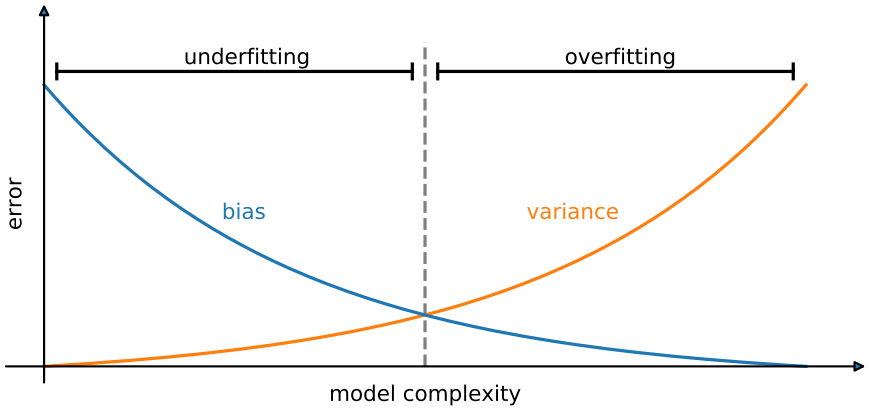


Figure 3.1: Visualization of the bias-variance trade-off.

3.3 Evaluation

Even though overfitting can be avoided with regularization, it is often not known to which extent regularization should be used. To find a model with good performance on new data an evaluation is performed. Therefore, the data set $\mathcal{D}$ is split into three parts, a train set D_{train}, a validation set D_{val}, and a test set D_{test}. The idea is to exclude instances from the learning process which then serve to measure the performance of the created model on newly observed data.

First, the optimal parameter configuration for the model is determined by learning models with different parameter settings on the train set D_{train} and evaluating them on the validation set D_{val}. However, this only gives us the best performing model with respect to the validation set D_{val}. In order to obtain the actual performance of the model on new data, the best parameter configuration is used to learn a model on the data $D_{train} \cup D_{val}$ which is then evaluated on the test set D_{test}. In particular, the results obtained on the test set D_{test} allows us to reason about the performance and compare different machine learning algorithms which can be used throughout this optimization process.

However, the distribution of instances across the three sets, which is normally chosen at random, has an influence on the evaluation. For example in a classification task, it is possible that the validation set $\mathcal{D}_{val}$ mainly contains instances for class A while in the test set $\mathcal{D}_{test}$ class B is the most common. Such an imbalance has an impact on the learning and, therefore, also on the evaluation and the comparison of the algorithms. To reduce this influence *cross validation* can be used which performs multiple evaluations on the data (Murphy, 2012).

3.4 Supervised Learning Algorithms

In the following, we introduce a set of supervised machine learning algorithms which have been used throughout this thesis. These algorithms are described only superficially since they are not a main component of this work. For a detailed description we refer to the corresponding references.

3.4.1 Linear Regression

In linear regression, it is assumed that the mapping $f : X_1, \ldots, X_m \rightarrow Y$ between the explanatory and the target random variables can be explained by a linear function:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_m X_m$$

where $\beta_0, \ldots, \beta_m$ are coefficients. Given a data set $\mathcal{D}$, these coefficients can be learned by minimizing the mean squared error. Due to its simplicity and interpretability, it is also commonly used in statistics. However, in the real world the assumption of a linear relationship is often not fulfilled. For a gentle introduction to linear regression, we refer to Montgomery et al. (2001).

3.4.2 K-Nearest Neighbour

Instead of learning a model, the idea of the k-nearest neighbour algorithm is to predict the target random variable solely based on the most similar instances in the given data set $\mathcal{D}$. Therefore, the distance between the instance to be classified and each instance in the data set $\mathcal{D}$ is computed using a common distance measure such as the *Euclidean distance*. Based on the value of the target random variable of the k closest instances, a value can be predicted. For example, in regression the average is taken whereas in classification the majority class will be predicted. The major drawback of this approach is the computational complexity since for each prediction the complete data set need to be scanned for the most similar instances. For further information about the algorithm, we refer to Jiang et al. (2007).

3.4.3 Random Forest

Introduced by Breiman (2001), random forest is an ensemble of randomized decision trees. Each tree has been learned over a *bootstrap sample* which is obtained by uniformly sampling instances of the given data set $\mathcal{D}$ with replacement (Breiman, 1996).

Starting from the root node, inner nodes of a random tree are constructed recursively by distributing the training instances according to splits which maximize a particular splitting criterion among a random subset of random variables. In case that no further tests can be created, a leaf will be constructed in which information about the assigned instances will be collected. Due to its state-of-the-art performance it is still frequently used in the literature.

3.5 Sum-Product Networks

A sum-product network (SPN) is a generative machine learning algorithm and belongs to the family of probabilistic graphical models which model the joint probability distribution $P(X_1, \ldots, X_m)$ of a given data set $\mathcal{D}$. Hence, it can be categorized as an unsupervised learning algorithm (cf. Section 3.1). In contrast to other probabilistic graphical models, such as Bayesian networks (Pearl, 1988), the main advantage of SPNs is that they can efficiently compute *exact inference* for a large class of distributions (Poon and Domingos, 2011).

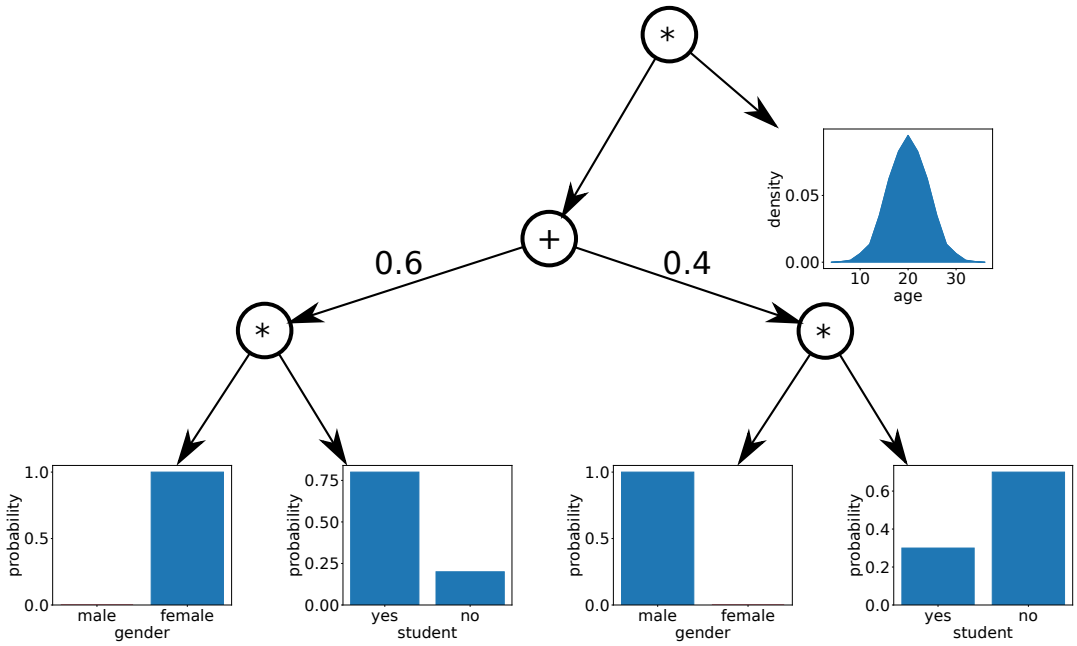


Figure 3.2: Exemplary SPN learned over the random variables *gender*, *student* and *age*.

3.5.1 Structure

The structure of an SPN is a rooted directed acyclic graph of sum, product and leaf nodes. In this graph, the *scope* of a particular node is defined as the set of random variables appearing in the subgraph below that node. Formally, sum nodes provide a weighted mixture of distributions by combining nodes which share the same scope, whereas product nodes represent the factorization over independent distributions by combining nodes defined over disjunct scopes. Finally, each leaf node contains a univariate distribution $P(X)$ for a particular random variable $X \in \{X_1, \ldots, X_m\}$ (cf. Section 2.1.1). Given this formulation a tractable multivariate distribution can be created by simply combining univariate distributions with sum and product nodes (Gens and Pedro, 2013).

As an example, Figure 3.2 displays an SPN which has been learned over three random variables: (1) the *gender*, (2) the status of being a *student* and (3) the *age*. From the structure of the SPN, we can observe that the distribution of *age* is independent of the other two random variables. In contrast, the sum node splits the population into two sub-populations, one representing females which cover 60% and males which cover 40% of the total population. Within these sub-populations we can further observe that females are more likely to be a student than males.

3.5.2 Learning

Similar to learning a neural neural network, the structure of an SPN can be specified in advance. In this case only the weights of the sum nodes and the distributions in the leaves of the SPN are learned using gradient descent (Poon and Domingos, 2011). However, this approach also requires domain knowledge to create the structure which is often not available or hard to acquire (Molina et al., 2018).

A more sophisticated approach is to learn the structure of the SPN from the data. The general approach to structure learning is a top-down procedure in which the data is recursively sliced into smaller chunks until leaf nodes can be created. Starting with the whole data set $\mathcal{D}$, the process of construction can be defined by three operations:

Leaf creation: If the assigned data slice only contains a single random variable, a leaf node will be created in which the random variable will be captured by a univariate distribution.

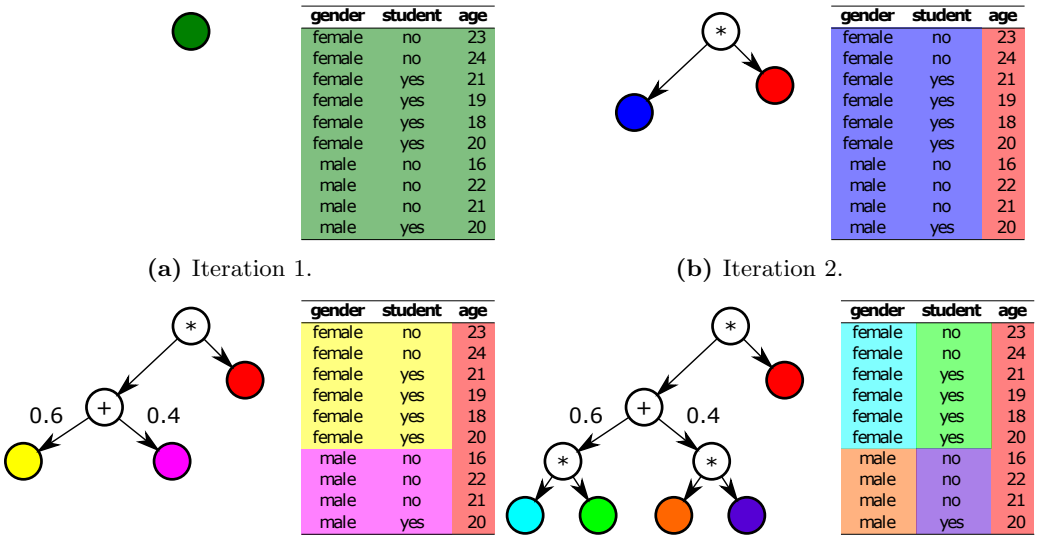
Decomposition: In case the data is multivariate, the next step is to check for independence between the assigned random variables. If independence between subsets of random variables can be detected, a product node is created which split the respective random variables.

Conditioning: Lastly, in case no independence between the random variables can be detected, clustering on the assigned data is performed. The resulting clusters are then split by a sum node where the weights represent proportion of instances assigned to the respective cluster.

The structure learning can mainly be controlled by two parameters: (1) A threshold for the independence test and (2) an early stopping criterion for the growing of the SPN. For example, if the given data slice contains less instances than a pre-specified threshold, the creation a product node is forced which generates univariate data slices. Algorithms for structure learning mainly differ in the independence test, the clustering algorithm and the distributions used in the leaves. For example, Molina et al. (2017) propose a structure learning algorithm with which count data can be modeled based on Poisson distributions. For comprehensibility, we have visualized an example of how an SPN can be constructed in Figure 3.3 based on the before introduced example. The data available at a specific node during construction is highlighted with the respective color.

Iteration 1: In the first step of the construction, the whole dataset is analyzed (cf. green color in Figure 3.3a). Since the data is not univariate, decomposition is performed which results in a product node which separates *age* from *gender* and *student*, as it can be seen in Figure 3.3b.

Iteration 2: Continuing the construction process with the *blue* data slice in Figure 3.3b, neither the data is univariate nor independence among the random variables can be detected. Hence, clustering is performed and a sum node is created.



(a) Iteration 1. **(b)** Iteration 2.

(c) Iteration 3. **(d)** Iteration 4.

Figure 3.3: Exemplary construction of an SPN.

Iteration 3: The random variables in the *yellow* and the *pink* data slice depicted in Figure 3.3c are both independent of each other. Hence, for both data slices a product node is created.

Iteration 4: Finally, the whole data set has been split up in univariate data slices which can be seen in Figure 3.3d. To complete the construction process, leaf nodes are created which capture the distribution of the respective data slices.

3.5.3 Inference

To answer a probabilistic query $P(X_1 = x_1, \ldots, X_m = x_m)$, the evidence is forwarded to the leaves of the SPN. For each univariate distribution in the leaves a probability is computed for the condition of the respective random variable. In a bottom-up procedure the probabilities are merged by multiplication in case of product nodes and by a weighted average in case of sum nodes. The probability obtained at the root node represents the result for the query. If the query only contains conditions for a subset of the random variables, leaf nodes of irrelevant random variables can be marginalized by returning a probability of 100%. For example, Figure 3.4 visualizes how the probability for query $P(student = yes)$ can be computed on our before introduced example.

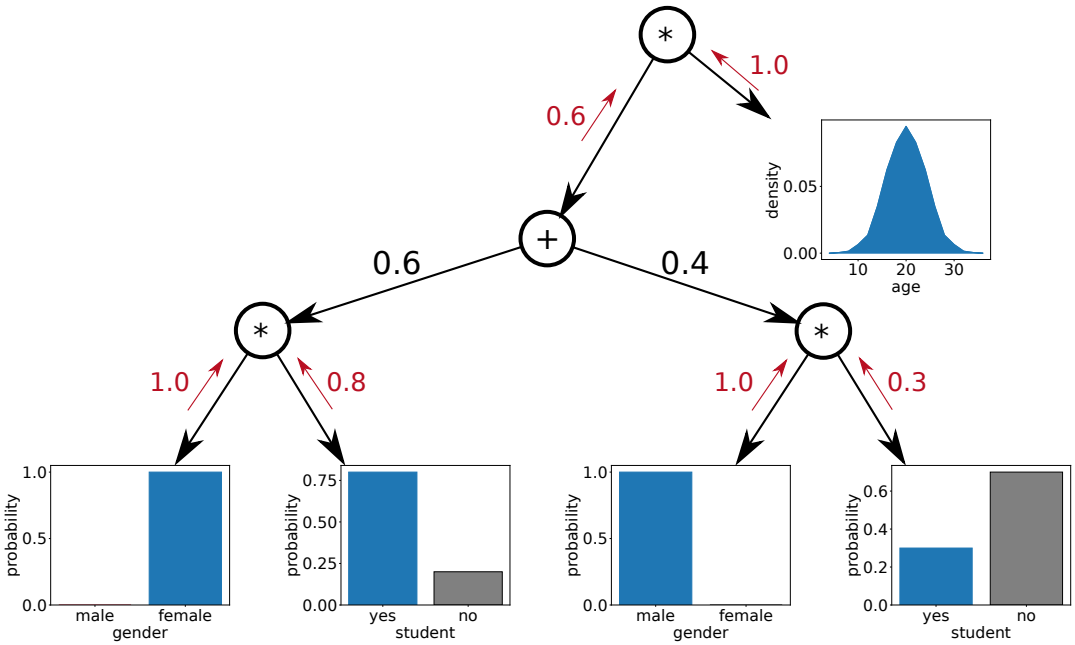


Figure 3.4: Computing the probability $P(student = yes)$.

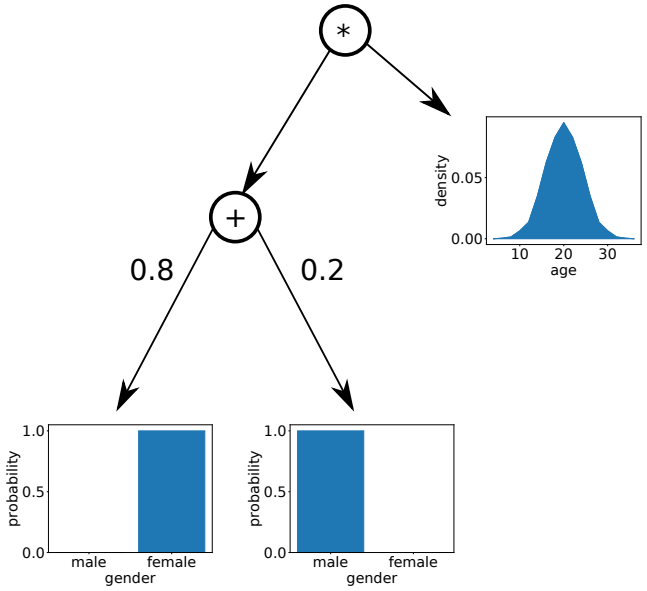


Figure 3.5: Conditional joint probability distribution $P(gender, age|student = yes)$.

3.5.4 Conditioning

The structure of an SPN can be adapted to represent the conditional joint probability distribution $P(\mathcal{X} \setminus X_i \mid X_i = x_i)$. Therefore, the probability of the condition $X_i = x_i$ is evaluated in the leaves and the resulting probability is propagated bottom up, performing multiplications at product and weighted sums at sum nodes. Whenever a sum node is passed, the weight for a child node is updated by multiplying it with the up-coming probability. In a final step, the weights for each sum node are normalized and the leaves for attribute X_i are removed from the SPN. Conditioning on multiple variables is done accordingly. For example, Figure 3.5 displays the conditional joint probability distribution $P(gender, age|student = yes)$ of the SPN shown in Figure 3.2

3.6 Anomaly Detection

Anomaly detection is a sub-discipline of machine learning with the main objective of finding patterns in data that do not conform to expected behavior (Chandola et al., 2009). Based on a finite sample of a set of random variables $X_1, \ldots, X_m$, the goal is to obtain a theoretical understanding of the data which allows to separate normal from anomalous instances. To that extent we distinguish between the following learning scenarios:

Supervised anomaly detection: Given a labeled data set, in which each instance belongs to either the normal or the anomalous class, the task of anomaly detection can be seen as a binary classification problem. Based on this setting, a classifier $f : X_1, \ldots, X_m \rightarrow Y$ can be learned where $Y = \{normal, anomalous\}$. A major drawback of this approach is that the data set need to cover all types of anomalies to ensure that the classifier can properly identify these.

Unsupervised anomaly detection: In contrast, unsupervised anomaly detection approaches do not require labeled data. Under the assumption that anomalies are rare, algorithms for this setting learn patterns with which most of the data can be explained. Instances which do not conform to these general patterns are considered to be anomalous. In particular, the specification of anomalies is omitted compared to the supervised setting. Generative machine learning algorithms are frequently used for this type of scenario.

In many real world scenarios labeled data is often not available or difficult to acquire for which reason the common approach to anomaly detection is the unsupervised one.

After the learning scenario has been chosen, one need to specify which kind of anomalies are of interest. Following Chandola et al. (2009) the types of anomalies can be classified in the following three categories:

Point anomaly: A single instance which is anomalous compared to the rest of the data set. This type of anomaly is also termed *global anomaly*. For example, observing a patient with an age over 100 years in an emergency department.

Contextual anomaly: A single instance which is only anomalous in a specific context but not otherwise. This kind is also named *conditional anomalies*. For example, observing an influenza case during the summer.

Collective anomaly: A group of instances with similar properties which are different with respect to the entire data set. For example, a group of patients with a particular clinical picture in an emergency department which is otherwise not so common.

3.7 Data Stream Mining

Data stream mining is concerned with a constant and possibly infinite stream of instances which need to be processed and analyzed in order to extract knowledge. In contrast to the finite data set $\mathcal{D}$ described in Section 3.1, instances arrive over time and are associated with particular time stamps which imposes a temporal order. Major challenges in stream mining are memory management and processing time

since instances can arrive at a high rate. Therefore, algorithms developed in this area usually aim to approximate solutions for the given learning task in order to use less time and memory (Bifet et al., 2018).

According to Gaber et al. (2005), algorithms for stream mining can be categorized into the following two groups:

Data-based techniques: The general idea of this kind of techniques is to map the incoming data onto a smaller representation which is easier to process. Accordingly, algorithms aim to summarize the whole data stream (e.g., synopsis data structures) or only analyze a subset of the incoming data (e.g., sampling techniques).

Task-based techniques: Algorithms of this kind aim to adopt existing techniques or invent new ones in order to achieve time and space efficient solutions. This includes approximation algorithms which have been specifically designed for computationally difficult tasks and sliding window approaches which only focuses on the most recent data of the stream.

4

Background on Disease Surveillance

This chapter covers the fundamentals of disease surveillance. First, a formal definition of traditional and syndromic surveillance is given in Section 4.1 and 4.2. Thereafter, possible data sources for monitoring diseases are discussed (cf. Section 4.3) and a set of statistical surveillance methods which are relevant for this thesis are reviewed (cf. Section 4.4). Next, the evaluation of these methods is addressed (cf. Section 4.5) and a brief introduction into multivariate surveillance is given (cf. Section 4.6). Finally, an overview is given in Section 4.7 about how machine learning algorithms can be applied for disease surveillance.

4.1 Traditional Surveillance

The main objective of disease surveillance is to monitor the presence of an infectious disease over time and to conduct an investigation by epidemiologists if an unexpected high number of infections is observed (Jackson et al., 2007). The traditional approach, also referred to as *diagnosis-based surveillance* (Henning, 2004), is to take only the number of confirmed infections into account which have been reported to local health departments. However, this verification often requires laboratory testing to confirm the presence of the respective pathogen which can take up to several days until results are available. In addition, delays in the reporting can have an significant impact on the early detection of outbreaks.

4.2 Syndromic Surveillance

Rather than tracking the confirmed cases, syndromic surveillance focuses on early indicators of a disease to allow a more timely detection of outbreaks (Shmueli and Burkom, 2010). In the context of syndromic surveillance, such indicators are usually encapsulated as syndromes:

Definition 1 (Syndrome (Fricker, 2014)) *A syndrome is a set of symptoms or conditions that occur together and suggest the presence of a certain disease or an increased chance of developing the disease.*

Notably, this definition differs slightly from the original meaning of a syndrome, which is only described by a set of symptoms, to also include the monitoring of nonclinical data sources (Henning, 2004). For example, the sales of a specific pharmaceutical product against flu could be used for the detection of influenza outbreaks but cannot be described as a symptom directly. In general, syndromic surveillance can be defined as:

Definition 2 (Syndromic surveillance (Buehler et al., 2008)) *Syndromic surveillance is an investigational approach where health department staff, assisted by automated data acquisition and generation of statistical alerts, monitor disease indicators in real-time or near real-time to detect outbreaks of diseases earlier than would otherwise be possible with conventional reporting of confirmed cases.*

The general approach to syndromic surveillance is to first decide on a disease under surveillance and based on that a syndrome is specified which needs to be monitored. Consequently, most surveillance approaches are tailored to specific diseases and their unique characteristics, such as influenza (Hiller et al., 2013), pneumonia (Hope et al., 2008), or norovirus (Edge et al., 2006). Several of these approaches are often bundled into one surveillance system to monitor multiple diseases simultaneously (e.g., Ansaldi et al., 2008; Heffernan et al., 2004; Ising et al., 2006; Lober et al., 2003; Wu et al., 2008).

The above described approach has a strong focus on the definition of syndromes which can be challenging since symptoms are often shared by different diseases and a particular disease can have different disease patterns in the early phase of an infection. Moreover, it is a handcrafted approach and only allows to monitor known infectious diseases.

In contrast, only few algorithms have been proposed which aim to identify outbreaks without specifying a particular disease in advance. For example, Reis et al. (Reis and Mandl, 2003; Reis et al., 2003) monitor the total number of patient visits in an emergency department rather than particular syndromes. However, a high number of patient visits can be caused by various reasons, making the resulting signal of the syndromic surveillance method noisy and unreliable. Better results in this setting can be obtained with the help of machine learning techniques as we will see in chapters 7 and 8.

Table 4.1: Exemplary data sources for syndromic surveillance.

clinical data source	alternative data sources
emergency department visits	school or work absenteeism
emergency hotline calls	pharmacy sales
insurance claims	internet-based searches
laboratory results	animal illnesses or deaths
...	...

4.3 Data Sources

The first step in syndromic surveillance is to choose a suitable data source that can be used to track infected individuals. In general, the presence of an infectious disease outbreak can only be determined through the actions of infected people. The infection remains unknown, if an infected person does not contact any service that allows to collect information about the case. According to Henning (2004), health-related data sources can be separated into two categories (examples are listed in Table 4.1):

Clinical data sources: Sources which provide reliable measurements of symptoms. For example, confirmed diagnosis by clinical experts.

Alternative data sources: Sources which indirectly measure the presence of a disease. For example, internet-based health inquiries.

Figure 4.1 exemplary visualizes how the number of infections can be tracked. For this purpose, let us consider a constant stream of patients arriving at an emergency department (cf. Section 3.7) which can be classified as a clinical data source. First, the patients are grouped together according to pre-specified time slots. Next, based on the assigned diagnoses, the patients are identified which match the syndrome definition (cf. patients highlighted in blue). Finally, a time series is constructed by counting the number of infections in each group.

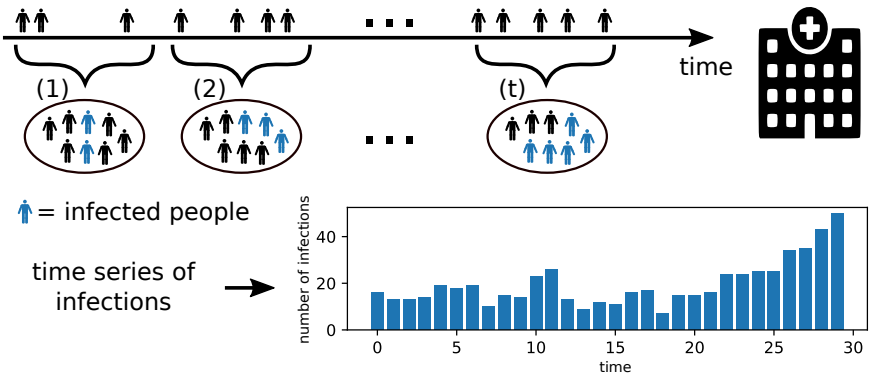


Figure 4.1: Exemplary generation of the time series of infections from a data source.

A specific characteristic of syndromic data is *seasonality* which is also known as also known as *cyclic drift* in machine learning area (Webb et al., 2016). It is a special form of *concept drift* in which the target concept changes over time with respect to a fixed time frame. For example, Hughes et al. (2014) and Dirmyer (2018) show that the cold weather in winter has an influence on the symptoms of the people arriving in emergency departments. Furthermore, Johnson et al. (2014) capture seasonal patterns in emergency department data due to respiratory illnesses. This kind of drift is predictable, and syndromic surveillance algorithms can take advantage of it.

4.4 Statistical Surveillance Methods

A broad variety of disease surveillance methods have been studied in previous work. Most of these approaches are based on statistics and are designed to monitor a single time series of count data. In this thesis, we focus on statistical methods which require comparably little historic data on their own. Such methods are universally applicable and serve as *drop-in* approaches for surveillance systems. They only rely on the detection of a local increase in incidents without the need to capture effects like seasonality and trend (Hutwagner et al., 2003).

Let us denote with $(c(0), c(1), \ldots, c(n)) \in \mathbb{N}^n$ the time series of infection counts for a particular disease (e.g., as shown in Figure 4.1). The methods rely on a sliding window approach which uses the previous m counts as reference values for fitting a particular parametric distribution (cf. Section 2.2). The sliding mean $\mu(t)$ and variance $\sigma^2(t)$ can be computed over these m reference values as follows:

$$\mu(t) = \frac{1}{m}\sum_{i=0}^{m-1} c(t-i) \qquad \sigma^2(t) = \frac{1}{m}\sum_{i=0}^{m-1}(c(t-i)-\mu(t))^2$$

On the fitted distributions, a statistical significance test is performed (cf. out-of-distribution test in Section 2.3.1) in order to identify suspicious spikes of counts. For the purpose of outbreak detection, we rely on one tailed-tests for the statistical algorithms in order to only capture the observation of unusual high number of infections. Therefore, the right-sided p-value for $c(t)$ is computed on the fitted distribution (cf. Section 2.1.2). The following algorithms have been considered in this thesis:

EARS C1 The C1 variant of the *Early Aberration Reporting System* (EARS) (Fricker et al., 2008; Hutwagner et al., 2003) relies on the assumption of a Gaussian distribution. The significance of the current observation c_t with the C1 method is computed as in the following:

$$c_t \overset{\text{C1}}{\sim} N(\mu(t), \sigma^2(t))$$

EARS C2 is a variation of C1, which adds a gap of two time points between the reference values and the current observed count c_t:

$$c_t \overset{\text{C2}}{\sim} N(\mu(t-2),\ \sigma^2(t-2))$$

EARS C3 combines the result of the C2 method over a period of three previous observations. For convenience of notation, the incidence counts c_t for the C3 method are transformed according to the statistics so that it fits to the normal distribution:

$$\left[\frac{c_t - \mu(t-2)}{\sqrt{\sigma^2(t-2)}} - \sum_{i=1}^{2} \max(0, \frac{c_{t-i} - \mu(t-2-i)}{\sqrt{\sigma^2(t-2-i)}} - 1)\right] \overset{\text{C3}}{\sim} N(0,1)$$

Despite the inaccurate assumption of the Gaussian distribution for low counts, the EARS variants are often included in comparative studies due to its simplicity and still serves as competitive baseline (Bédubourg and Le Strat, 2017; Fricker et al., 2008; Hutwagner et al., 2005).

RKI method. In contrast to the family of C-algorithms, the RKI algorithm assumes a Poisson distribution for count data with a low mean. In Salmon et al. (2016) it is implemented as follows:

$$c_t \overset{\text{RKI}}{\sim} \begin{cases} Poisson(\lfloor \mu(t) \rfloor + 1), & \text{if } \mu(t) \leq 20 \\ N(\mu(t),\ \sigma^2(t)), & \text{otherwise} \end{cases}$$

Bayes method. The Bayes algorithm, as implemented by Salmon et al. (2016), relies on the assumption of a negative binomial distribution:

$$c_t \overset{\text{Bayes}}{\sim} NB(m \cdot \mu(t) + \frac{1}{2},\ \frac{m}{m+1})$$

4.5 Evaluation

The evaluation of disease surveillance methods is usually difficult due to the lack of labeled data. In particular, for some scenarios of infectious disease outbreaks, such as the intentional release of *Bacillus anthracis*, none or only very few outbreaks happened in the past. Even though this data is sometimes available, it is rarely fully accessible to the public (Lotze et al., 2007). In addition, a precise norm for the labeling of outbreaks does not exist, making it difficult to obtain standardized data sets on which algorithms can be evaluated. According to Buckeridge et al. (2005), the evaluation data used for disease surveillance can be described by three categories:

Wholly authentic: Data that capture a real infectious disease outbreak which has been analyzed and labeled by epidemiologists. Although these data are genuine, quantitative analysis often cannot be performed due to limited number and variety of outbreaks and due to inconsistent labeling between different experts.

Wholly simulated: Fully synthetic data that allows to precisely control the underlying environment and the characteristics of outbreaks. However, it is difficult to generate realistic data and, therefore, it has only limited validity.

Simulated outbreaks in authentic data: This type of data uses real data as a basis into which synthetic outbreaks are injected. It is superior to wholly simulated but the problem with the validity remains since the simulated outbreaks may not represent real outbreaks.

Most of the proposed algorithms in the literature are evaluated using wholly simulated or simulated outbreaks superimposed on real data. This allows a detailed analysis of the performance of the proposed algorithm in a controlled setting.

To measure the performance of outbreak detection methods, it is common to use the *activity monitor operating characteristic (AMOC)* (Fawcett and Provost, 1999). AMOC can be seen as an adaptation of the *receiver operating characteristic* (Spackman, 1989) in which the true positive rate is replaced by the *detection delay*, i.e., the number of time points until an outbreak has been first detected by the algorithm. In case the algorithm does not raise an alarm during the period of an outbreak, the detection delay is equal to the length of the outbreak. Figure 4.2 shows an exemplary AMOC-curve.

Moreover, for disease surveillance we are interested in a very low false alarm rate for the algorithms and therefore only consider the partial area under AMOC-curve (cf. blue highlighted area in Figure 4.2). Note that contrary to conventional AUC

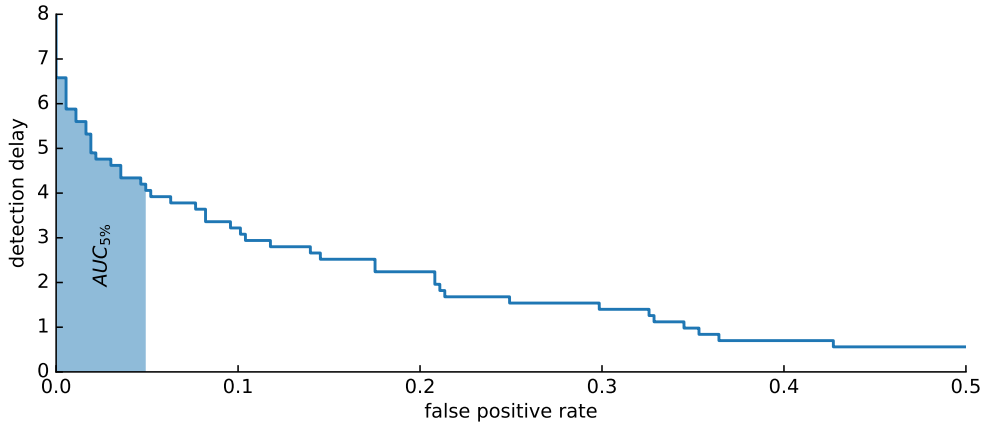


Figure 4.2: Exemplary visualization of a AMOC-curve. The blue highlighted area represents the partial area under curve for a false alarm rate up to 5%.

values in this case lower values represent better results. Since one data stream does normally not contain enough outbreaks to draw conclusions, the evaluation is usually performed on a set of data streams. To obtain a final score for the set, the average over the results of each data stream is taken.

4.6 Multivariate Surveillance

The surveillance of multiple indicators or multiple data sources simultaneously facilitates the detection of outbreaks (Roure et al., 2007). This area is known as *multivariate disease surveillance* for which we could identify the following categories:

Spatial surveillance: Simultaneously monitoring of disease counts at different locations with possible spatial correlations (e.g., Heffernan et al., 2004; Lober et al., 2003). For example, monitoring syndromes in different emergency departments simultaneously which are located in the same district.

Multiple syndrome surveillance: Simultaneous surveillance of multiple syndromes for the same disease. In particular, for syndromic surveillance based on emergency department data, it has been shown that the use of more data can improve the ability of detecting outbreaks. For example, Reis and Mandl (2004) show that the surveillance of chief complaints and diagnostic codes together in an emergency department yield better results than alone. Moreover, Held et al. (2005) simultaneously monitor a syndrome with respect to particular groups of patients which differ in their demographic characteristics.

Multiple data source surveillance: Monitoring of different types of data sources at the same time (e.g., Faverjon et al., 2016; Kulldorff et al., 2007). For example, monitoring over-counter sales in pharmacies and emergency department visits simultaneously.

4.7 Relation to Machine Learning

Seen from a machine learning perspective, syndromic data are a constant stream of instances (cf. Section 3.7). To detect changes in the data stream, which might indicate an outbreak, the instances are usually grouped together according to fixed time intervals as it is done for modeling count data (cf. Section 2.2 and 4.3). For example, all patients which arrive at an emergency department on a specific day are grouped together as a set of instances. Hence, the stream can be represented as a time series of sets of instances. The goal of disease surveillance is to detect any major changes for the last observed set in the stream which might indicate an outbreak of

an infectious disease. As stated in Section 4.2, one can either pre-process the data in order to extract only the information pertinent to the definition of a specific syndrome or monitor all available data for unusual distributional changes.

Generally speaking, the main objective of disease surveillance can be described as anomaly detection (cf. Section 3.6). In particular, the focus is put on patterns which indicate an increasing number of infections over time which can be described by collective and sequential anomaly detection at the same time. Directly applying point anomaly detection, which aims to identify single instances as outliers, such as encountering a patient over a hundred years old in an emergency department, is not of interest for disease surveillance (Wong et al., 2002). However, by forming a univariate time series of counts for a particular syndrome, as it is done in syndromic surveillance, the problem can be reduced to point anomaly detection. Most approaches to disease surveillance can be categorized as statistical anomaly detection techniques (e.g., EARS (Hutwagner et al., 2003), Farrington (Noufaily et al., 2013), and many more).

In contrast, the area of *emerging pattern mining* (Dong and Li, 1999) directly relates to the problem of disease surveillance. It aims to discover item sets whose support increases significantly from one data set to the other. Similarly, *contrast set mining* (Bay and Pazzani, 2001) aims to find conjunctions of attributes and values that differ meaningfully in their distributions across data sets. Such techniques can be used to compare the last observed set of instances to the previous sets of instances in order to detect significant changes in the frequencies of any group of instances. Both approaches have also been viewed as instantiations of a general framework for supervised descriptive rule learning (Novak et al., 2009). The framework is a generalization of *subgroup discovery* (Wrobel, 1997), where labels (e.g., with respect to a concrete syndrome) are assumed to be available.

Due to the lack of labeled data, most algorithms for disease surveillance are unsupervised. Apart from unsupervised anomaly detectors, generative machine learning algorithms can also be used for disease surveillance, such as sum-product networks (cf. Section 3.5) or *Bayesian networks* (Jensen, 1996). This type of algorithm allows to capture the underlying probability distribution of the data source. Anomalies can be detected by comparing the expected distribution of the data with the distribution of the current observed set of instances. In this way, disease surveillance can also be seen from the perspective of *exceptional model mining* (Duivesteijn et al., 2016). Therefore, it can be formulated as the identification of a subset of instances in which a model of the current set of instances differs substantially from the models for previous sets of instances.

In general, the output of an anomaly detector for disease surveillance should be seen as a signal that an outbreak may be occurring which triggers a further investigation of the situation by public health officials (Fricker, 2014). To avoid unnecessary and costly interventions, the signal ideally includes information about the reason of the

detected anomaly allowing the epidemiologist to quickly judge the importance of the alarm. Therefore, disease surveillance could also benefit from the area of interpretable machine learning, focusing on approaches which can provide explanations to their predictions (Molnar et al., 2020).

5

Correlation-based Discovery of Disease Patterns for Syndromic Surveillance

Starting with this chapter, we introduce novel algorithms to improve disease surveillance. One of the major challenges in syndromic surveillance is the definition of disease patterns (cf. Section 4.2). They highly depend on the infectious disease and the health-related data source under surveillance. Since early symptoms are usually shared among many diseases and because a particular disease can have several clinical pictures at early stages of an infection, it is difficult to obtain reliable syndromes. For this reason, the definition of disease patterns is usually based solely on expert knowledge of epidemiologists, a time-consuming and laborious process (Mandl et al., 2004). This motivates the demand for tools that allow for a user-guided generation and comparison of syndrome definitions. To be useful in practice, such tools should be flexible enough to be applied to different types of data (Hopkins et al., 2017).

In this chapter, we present a data-driven approach that aims at supporting epidemiologists in the process of identifying disease patterns for infectious diseases. It discovers syndrome definitions from health-related data sources, based on their correlation to the reported number of infections in the respective geographical area. First, we introduce a formal definition of this correlation-based discovery task. Afterwards, we present an algorithm for the automatic extraction of disease patterns that uses techniques from the field of inductive rule learning. To provide insight into the data, the syndromes it discovers may be suggested to epidemiologists, who can adjust the input or the parameters of the algorithm to interactively refine the syndromes according to their domain knowledge. To better understand the capabilities and shortcomings of the proposed method, we evaluate its ability to reconstruct randomly generated disease patterns with varying characteristics. Furthermore, we apply our approach to emergency department data to learn disease patterns for Influenza, Norovirus and SARS-CoV-2. To assess the quality of the obtained patterns, we discuss the indicators they are based on and relate them to the number of infections according to publicly available reports, as well as handcrafted syndrome definitions.

5.1 Preliminaries

In the following, we formalize the problem that we address in this chapter, including a definition of relevant notation and an overview of related work.

5.1.1 Problem Definition

We are concerned with the deduction of patterns from a health-related data source $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathcal{X}$. It incorporates information about individual *instances* $\boldsymbol{x}_n \in X$ from a population $\mathcal{X}$, which are represented in terms of a finite set of predefined *attributes* $A = \{a_1, \ldots, a_K\}$. An instance $\boldsymbol{x} = (x_1, \ldots, x_K)$, e.g., representing a patient that has received treatment in an emergency department, assigns discrete or numerical values x_k to the k-th attribute a_k. For example, discrete attributes can be used to specify a patients' gender, whereas numerical attributes are suitable to encode continuous values, such as body temperature, blood pressure or the like.

The values for individual attributes may also be missing, e.g., because some medical tests have not been carried out as part of an emergency treatment. In addition, each instance in a data source is subject to a mapping $h : \mathbb{N}_+ \to \mathbb{N}_+$. It associates the n-th instance with a corresponding period in time, identified by a timestamp $t = h(n)$. Instances that correspond to the same interval, e.g., to the same week, are assigned the same timestamp $t : 1 \leq t \leq T$.

For each timestamp t, the instances in a data source may be associated with, a corresponding *target variable* $y_t \in \boldsymbol{y}$ to be provided as part of a secondary data source $\boldsymbol{y} = (y_1, \ldots, y_T) \in \mathcal{Y}$. The target space $\mathcal{Y}$ corresponds to the number of infections that may occur within consecutive periods of time. Consequently, a particular target variable $y_t \in \mathbb{N}_+$ specifies how many cases related to a particular infectious disease have been reported for the t-th time interval.

The learning task, which we address in this work, requires to find an interpretable model $f : \mathcal{X} \to \mathcal{Y}$. Given a set of instances $X \subset \mathcal{X}$ that are mapped to corresponding time intervals via a function h, it provides an estimate $\hat{\boldsymbol{y}} = f(X, h) = (\hat{y}_1, \ldots, \hat{y}_T) \in \mathcal{Y}$ of the number of infections per time interval. The selection of instances and the number of reported cases, which are provided for the training of such model, must neither originate from the same source, nor comprise information about identical subgroups of the population. As a consequence, the estimates of a model are not obliged to reflect the provided target variables in terms of their absolute values. Instead, we are interested in capturing the correlation between indicators that may be derived from the training instances and the number of infections that have arised during the considered timespan. To assess the quality of a model, we compare the estimates it provides to the target variables with respect to a suitable correlation coefficient, such

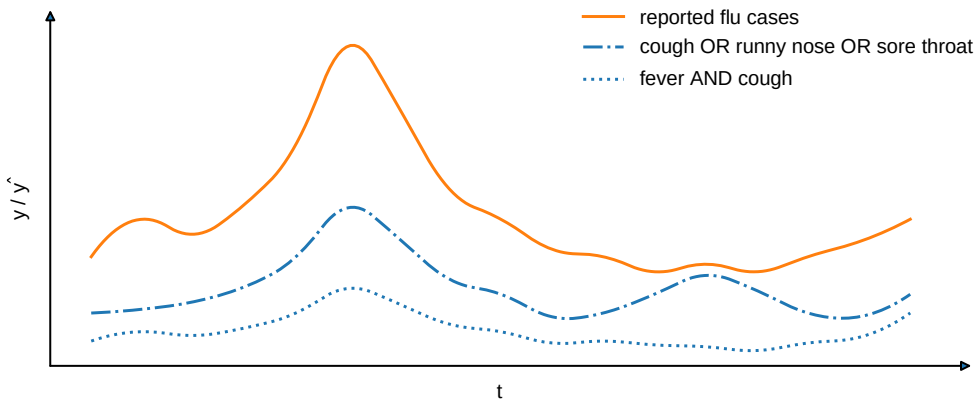


Figure 5.1: Exemplary comparison of two syndrome definitions (blue lines) with reported cases (orange line). The Pearson's correlation for "fever AND cough" is 0.98 and for "cough OR runny nose OR sore throat" is 0.88.

as *Spearman's* ρ, *Kendall's* τ, or *Pearson's* correlation. For example, one could align patient data of an medical office with local reported flu cases. Figure 5.1 exemplary visualizes two syndrome definitions which are obtained by counting the number of patients per timestamp which fulfill a particular clinical picture. The syndrome "fever AND cough" covers less cases but it has an higher Pearson's correlation coefficient than the other syndrome (0.98 compared to 0.88).

5.1.2 Related Work

Disease patterns for syndromic surveillance are usually defined according to the knowledge of domain experts. This requires a manual examination of the available health-related data to identify indicators that may be related to a particular disease at hand. For example, Edge et al. (2006) and Muchaal et al. (2015) analyze information about the sales of pharmaceuticals to reason about the spread of Norovirus infections, based on their effectiveness against gastrointestinal symptoms. Similarly, the data that is gathered in emergency departments may also serve as a basis for the definition of disease patterns. In this case, definitions are usually based on the symptoms of individual patients and the diagnoses made by the medical staff. For example, Ivanov et al. (2002) and Suyama et al. (2003) rely on standardized codes for the *International Classification of Diseases* (ICD) (Trott, 1977). Boender et al. (2021) additionally use chief complaints of the patients at the emergency departments. The majority of syndrome definitions are targeted at common infectious diseases, such as gastrointestinal infections, influenza-like illnesses or respiratory diseases (e.g., Boender et al. (2021); Bouchouar et al. (2021); Heffernan et al. (2004); Suyama et al. (2003)). However, they are also used to detect other health-related epidemics, e.g., increased usage of psychoactive substances (Nolan et al., 2017).

The deduction of indicators from unstructured data, such as textual reports of complaints or diagnoses, is particularly challenging. To be able to deal with such data, text documents are often represented in terms of keywords they consist of. For example, Lall et al. (2017) use syndromes that apply to the keywords contained in medical reports. Similarly, Heffernan et al. (2004) use a list of exclusive keywords to reduce the chance of misclassifications, Bouchouar et al. (2021) utilize regular expressions to extract symptoms from texts and Ivanov et al. (2002) use a classifier system that takes textual data as an input to assign syndromes to individual patients. In order to train a classifier, the latter approach requires labeled training data that must manually be created by experts.

The analysis of textual data is even more profound in approaches to syndromic surveillance that are based on web data. For example, Velardi et al. (2014) analyze Twitter messages to capture indicators for the spread of influenza-like illnesses. Starting with a handcrafted set of medical conditions that are related to the respective disease, they learn a language model that aims to identify closely related terms based on clustering.

The problem of learning syndrome definitions in a data-driven way, without relying on expert knowledge, has for example been addressed by Kalimeri et al. (2019). The authors of this work propose an unsupervised, probabilistic framework based on matrix factorization. Their goal is to identify patterns of symptoms in structured data that has been obtained from participatory systems. Given a set of 19 symptoms, e.g., fever or vomiting, they construct a matrix that incorporates information about the occurrences of individual symptoms over time. Ultimately, syndromes can be generated from this matrix by extracting latent features that correspond to linear combinations of groups of symptoms.

Another method that relies on structured data is proposed by Goldstein et al. (2011). It is aimed at capturing the likelihood of syndromes for a particular infectious disease. The authors propose to use expectation maximization and deconvolution to identify syndromes, which are highly correlated with the occurrences of symptoms that have been reported in regular time intervals. However, their approach does only allow to evaluate and compare disease patterns that have been specified in advance. Even though the aforementioned algorithms deal with structured data that is less cumbersome to handle than unstructured inputs, they have only be applied to small and pre-selected sets of features.

The problem of learning from assignments of target variables to sets of instances, rather than individual instances, is known as *multiple instance learning* (Dietterich et al., 1997). Chevaleyre and Zucker (2001) tackle such task by adapting the quality criterion used by the well-known rule learning method RIPPER. To be able to deduce classification rules from sets of instances, Bjerring and Frank (2011) incorporate the separate-and-conquer rule induction technique into a tree learner. Both approaches

are limited to the assignment of a binary signal to a bag of instances and are not intended to cope with multiple instance regression tasks (Ray and Page, 2001). The mapping of numeric values to bags of instances, as in the syndrome definition learning task at hand, is a much less explored problem in the literature. We are not aware of any existing work that approaches this kind of problem with the goal to obtain rule-based models.

5.2 Learning of Syndrome Definitions

In the following, we propose an algorithm for the automatic induction of syndrome definitions, based on the indicators that can be constructed from a health-related data source. Each indicator c_m, which is included in such a model, refers to a certain attribute that is present in the data. It compares the values, which individual instances assign to this particular attribute, to a constant using relational operators, such as $=$ if the attribute is discrete, or $\leq$ and $>$ if it is numerical. By definition, if an indicator is concerned with an attribute for which an instance's value is missing, the indicator is not satisfied.

We strive for a combination of different indicators via logical AND ($\wedge$) and OR ($\vee$) operators. The model that is eventually produced is given in *disjunctive normal form*, i.e., as a disjunction of conjunctions. Such a logical expression $r = r_1 \vee \cdots \vee r_L$ with $r_l = c_{l,1} \wedge \cdots \wedge c_{l,M}$ evaluates to $r(\boldsymbol{x}_n) = 1$ (*true*) or $r(\boldsymbol{x}_n) = 0$ (*false*), depending on whether it is satisfied by a given instance $\boldsymbol{x}_n$ or not. If the context is clear, we abbreviate $c_{l,i}$ with c_i. The number of infected cases, which are estimated by a logical expression r for individual time intervals t, calculate as

$$\hat{\boldsymbol{y}} = r(X) = \left(\sum\nolimits_{\boldsymbol{x}_n \in X} [\![h(n) = t]\!] r(\boldsymbol{x}_n) \right)_{1 \leq t \leq T}, \tag{5.1}$$

where $[\![p]\!] = 1$ if the predicate p is true, and 0 otherwise.

The representation of syndromes introduced above is closely related to sets of conjunctive rules r_l as commonly used in *inductive rule learning* which is an established and well-researched area of machine learning (see, e.g., (Fürnkranz et al., 2012) for an overview on the topic). Consequently, we rely on commonly used techniques from this particular field of research to learn the definitions of syndromes. We use a sequential algorithm that starts with an empty hypothesis to which new conjunctions of indicators $r_1, \ldots, r_L$ are added step by step.

Given a data source that incorporates many features, the number of possible combinations of indicators can be very large. For this reason, we rely on *top-down hill climbing* to search for suitable combinations. With such an approach, conjunctions of indicators that can potentially be added to a model are constructed greedily.

At first, single indicators are taken into account individually. They are evaluated relative to the existing model and the one that promises the highest improvement in quality is ultimately selected. Afterwards, it is iteratively refined by evaluating the combinations that possibly result from a conjunction of already chosen indicators with an additional one. The search continues to add more indicators, resulting in more restricted patterns that apply to fewer instances, as long as an improvement of the model's quality can be achieved.

Optionally, the maximum number of indicators per conjunction M can be limited via a parameter. If $M = 1$, the algorithm is restricted to learn disjunctions of indicators. Furthermore, we enforce a *minimum support* $s \in \mathbb{R}$ with $0 < s < 1$, which specifies the number of instances $N \cdot s$ a conjunction of indicators must apply to. Once it has decided for a conjunction of indicators to be included in the model, the algorithm attempts to learn another conjunction to deal with instances that have not yet been adequately addressed by the model. The training procedure terminates as soon as it is unable to find a new pattern that improves upon the quality of the model. In addition, an upper bound can be imposed on the number of disjunctions L by the user.

The search for suitable indicators and combinations thereof is guided by a target function to be optimized at each training iteration. It assesses the quality that results from adding an additional conjunction of indicators to an existing model in terms of a numeric score. We denote the estimates that are provided by a model after the l-th iteration as $\hat{\boldsymbol{y}}^{(l)}$. When adding a conjunction of indicators r_l to an existing model, the estimates of the modified model can be computed incrementally as

$$\hat{\boldsymbol{y}}^{(l)} = r^{(l)}(X) = r^{(l-1)}(X) + r_l(X). \tag{5.2}$$

We assess the quality of a model's estimates in terms of the absolute *Pearson's correlation coefficient*. At a particular training iteration, it can be computed in a single pass over the target time series $\boldsymbol{y}$ and the current estimates $\hat{\boldsymbol{y}}^{(l)}$ according to the formula

$$m_P\left(\boldsymbol{y}, \hat{\boldsymbol{y}}^{(l)}\right) := \left| \frac{T \sum_t^T y_t \hat{y}_t \sum_t^T y_t \sum_t^T \hat{y}_t}{\sqrt{T \sum_t^T y_t^2 - \left(\sum_t^T y_t\right)^2} \sqrt{T \sum_t^T \hat{y}_t^2 - \left(\sum_t^T \hat{y}_t\right)^2}} \right|. \tag{5.3}$$

If the score that is computed for a potential modification according to the target function m_P is greater than the quality of the current model, it is considered an improvement. Among all possible modifications that are considered during a particular training iteration, the one with the greatest score is preferred.

5.3 Evaluation

To evaluate the previously proposed learning approach, we have implemented the methodology introduced above by making use of the publicly available source code of the BOOMER rule learning algorithm (Rapp et al., 2020). In adherance to the principles of reproducible research, our implementation can be accessed online[i]. A major goal of the empirical study, which is outlined in the following, is to investigate whether the proposed methodology is able to deduce patterns from health-related data that correlate with the number of infections supplied via a secondary data source. For our experiments, we relied on routinely collected and fully anonymized data from 12 German emergency departments which capture information about patients that have consulted these institutions between January 2017 and April 2021.

In a first step, we conducted a series of experiments using synthetic syndrome definitions. The objective was to validate the algorithm and to better understand its capabilities and limitations when it comes to the reconstruction of known disease patterns in a controlled environment. On the one hand, we considered synthetic syndromes with varying characteristics and complexity. On the other hand, we investigated the impact that the temporal granularity of the available data has on the learning approach. As elaborated below, the health-related data used in this chapter are available on a daily basis. By using synthetic syndromes, we were able to validate the algorithm's behavior when dealing with a broader or more fine-grained granularity as well. The use of synthetic syndromes also allows to investigate the ability of the proposed approach independently of the negative effects of artifacts that may be present in real data. This includes delays of reports, inaccuracies in the reported dates or instances that are present in one data source, but not in the other. For example, cases may have been reported in one of the considered districts, but have not been treated in one of the emergency departments included in our dataset. Vice versa, it is also possible that cases have been treated at one of the considered departments but have not been reported to the public agencies.

Such artifacts almost certainly play a role in our second experiment, where we tried to discover patterns that correlate with the publicly reported cases. We selected cases from the notifiable diseases of *Influenza* and *Norovirus*, which have extensively been studied in existing work (e.g., Heffernan et al., 2004; Kalimeri et al., 2019; Muchaal et al., 2015), as well as of the recently emerged *SARS-CoV-2*, which has for example been analyzed by Bouchouar et al. (2021). To evaluate whether the algorithm is able to identify meaningful indicators that are related to these particular diseases, we provide a detailed discussion of the discovered syndromes and compare them to manually defined disease patterns.

[i] `https://github.com/mrapp-ke/SyndromeLearner`

Table 5.1: Attributes included in the emergency department data.

name	type	#values	missing values in %
① diagnosis			
MTS presentation	discrete	57	0.01
MTS indicator	discrete	179	5.10
ICD code	discrete	5901	65.45
ICD code (short)	discrete	1509	65.45
② demographic information			
gender	discrete	3	0.00
age	discrete	21	0.00
③ vital parameters			
blood pressure systolic	numeric	-	57.19
blood pressure diastolic	numeric	-	57.22
temperature	numeric	-	59.31
respiration rate	numeric	-	59.55
pulse frequency	numeric	-	91.91
oxygen saturation	numeric	-	57.18
④ contextual information			
isolation	discrete	11	1.81
transport	discrete	6	59.74
disposition	discrete	13	90.56

5.3.1 Experimental Setup

Health-related data. As shown in Table 5.1, we have extracted 15 attributes from the emergency department data. Each of the available attributes corresponds to one out of four categories. The first category, *diagnosis*, includes an initial assessment in terms of the *Manchester Triage System* (MTS) (Gräff et al., 2014). It is obtained for each patient upon arrival at an emergency department. Besides, this first category also comprises an ICD (Trott, 1977) code that represents a physician's assessment. In addition to the full ICD code, we also consider a more general variant that consists of the leading character and the first two digits (e.g., *U07* instead of *U07.1*). Features that belong to second category, *demographic information*, indicate the gender and age of patients, whereas *vital parameters* correspond to measurement data, such as blood pressure or pulse frequency, that may have been registered by medical staff. Features of the last category, *contextual information*, may provide information about why a patient was possibly quarantined (*isolation*), the means of transport used to get to the emergency department (*transport*) and the status when exiting the department (*disposition*).

In contrast to existing work on the detection of disease patterns (e.g., Goldstein et al., 2011; Kalimeri et al., 2019), we have not applied any pre-processing techniques to

the health-related data, such as a manual selection of symptoms that are known to be related to an infectious disease. As a consequence, the data contains a lot of noise, e.g., diagnoses related to injuries, and many missing values (cf. Table 5.1). In accordance with the findings of Hartnett et al. (2020), we observed a reduced number of emergency department visits during the first weeks of the SARS-CoV-2 pandemic. However, preliminary experiments suggested that this anomaly has no effect on the operation of our algorithm. To obtain a single dataset, we have merged the data from the considered emergency departments. It consists of approximately 1,900,000 instances. Each of the instances corresponds to a particular week (i.e., around 8,500 instances per week). Additional information about the emergency data used in this chapter is provided by Boender et al. (2021), who used a slightly different subset of the data set to evaluate their handcrafted syndrome definitions.

Number of infections. The number of cases corresponding to the infectious diseases Influenza, Norovirus and SARS-CoV-2 have been retrieved from the *SurvStat*[ii] platform provided by the *Robert Koch-Institut*. To match the temporal information in the health-related dataset, we have aggregated the weekly reported numbers for German districts ("Landkreise" and "Stadtkreise") where the considered emergency departments are located.

Parameter setting. For all experiments that are discussed in the following, we have set the minimum support to $s = 0.0001$. With respect to the approximately 1,900,000 instances contained in the training dataset, this means that each conjunction of indicators considered by the algorithm must apply to at least 190 patients. In preliminary experiments we have found this setting to produce reasonable results, while keeping the training time at an acceptable level (typically under one minute). In addition, we have limited the maximum number of disjunctions in a model to $L = 50$. However, the algorithm usually terminates before this number is reached.

5.3.2 Reconstruction of Synthetic Syndromes

In our first experiment, we validated the ability of our algorithm to discover disease patterns under the assumption that the reported cases are actually present in the data. For this purpose, we defined synthetic syndromes with varying characteristics from the emergency department data. For each syndrome, we determined the number of instances they apply to over time. The goal of the algorithm was to reconstruct the original syndrome definitions, exclusively based on the correlation with the corresponding number of cases. For this experiment, we focused on syndromes that use ICD codes and MTS representations, since these indicators are most commonly used

[ii] `https://survstat.rki.de`

in related work (e.g., Boender et al., 2021; Ivanov et al., 2002; Suyama et al., 2003). We have not used short versions of the ICD codes due to their overlap with the full codes. The following three different types of synthetic syndromes were considered:

1. Conjunctions of indicators (AND):

$$r = c_1 \wedge \ldots \wedge c_M, \quad \text{where } M \in \{2, 3\}$$

2. Disjunctions of indicators (OR):

$$r_1 \vee \ldots \vee r_L, \text{ where } r_l = c \text{ and } L \in [2, 9]$$

3. Disjunctions of conjunctions (AND-OR):

$$r_1 \vee \ldots \vee r_L, \text{ where } r_l = c_1 \wedge c_2 \text{ and } L \in [2, 5]$$

For each syndrome type, we generated 100 artificial definitions by randomly selecting indicators that are present in the data, such that each indicator and each conjunction of indicators applies to at least 200 patients. This ensures that the syndromes that are ultimately generated apply to this particular number of patients at minimum. In addition, we have considered three temporal granularities to determine the number of cases different syndromes apply to. Experiments have been conducted with counts that are available on a daily, weekly or monthly basis. To quantify to which extent our approach is able to reconstruct the original syndrome definitions, we compute the percentage of correctly identified patterns, i.e., syndromes that use the exact same indicators, referred to as the *reconstruction rate*. A visualization of the experimental results is given in Figure 5.2.

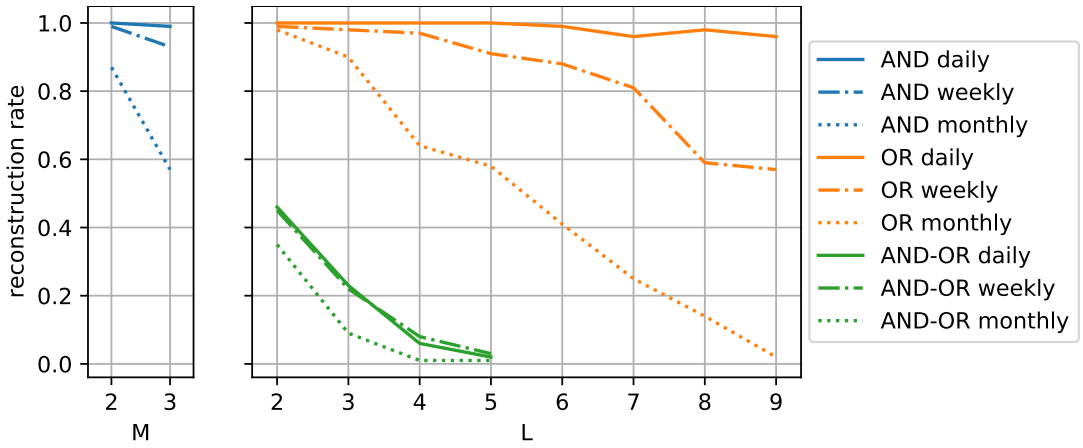


Figure 5.2: Percentage of successfully reconstructed syndrome definitions of different types for varying complexities of the predefined syndromes.

Generally, we can observe that the algorithm's ability to capture the predefined disease patterns benefits from a more fine-grained granularity of the available data (e.g., daily instead of weekly reported numbers). This meets our expectations, as a greater temporal resolution results in more specific patterns of covered cases, given a particular syndrome. As a result, it is easier to identify the indicators that allow to replicate a certain disease pattern and separate them from unrelated ones. In particular, syndromes that are exclusively based on disjunctions (OR) or conjunctions (AND), regardless of their complexity, can reliably be captured when supplied with daily numbers. When dealing with a broader temporal granularity, the uniqueness of disease patterns vanishes and they become more likely to interfere with the numbers resulting from similar syndromes.

Regarding the different types of predefined syndromes, it can be seen that their reconstruction becomes more difficult as their complexity increases. Especially when dealing with syndromes that include both, disjunctions and conjunctions (AND-OR), the reconstruction rate mostly depends on the number of indicators, whereas the temporal resolution plays a less important role. This shows the limitations of a greedy hill climbing strategy when it comes to the reconstruction of complex patterns. To overcome these shortcomings, approaches for the re-examination of previously induced rules, such as pruning techniques, could be considered. It is also possible to extend the search space that is explored by the training algorithm, e.g., by conducting a beam search, where several promising solutions are explored instead of focusing on a single one at each step. However, if the patterns, which have been found by the algorithm, only slightly differ from the predefined syndromes (e.g., by omitting or including infrequent ICD codes). While we did not evaluate this in depth, we believe they could still comprise useful information, e.g., by providing alternative, but nearly equivalent, descriptions of the syndrome.

5.3.3 Discovery of Syndrome Definitions from Real-World Data

In our second experiment, we used the proposed algorithm to obtain syndrome definitions for the infectious diseases Influenza, Norovirus and SARS-CoV-2. In the literature, the quality of syndromes is either evaluated by experts (e.g., Bouchouar et al., 2021; Heffernan et al., 2004; Ivanov et al., 2002; Lall et al., 2017) or by measuring the correlation with reported infections, reported deaths or expert definitions (e.g., Edge et al., 2006; Kalimeri et al., 2019; Muchaal et al., 2015; Nolan et al., 2017; Suyama et al., 2003; Velardi et al., 2014). We follow the latter approach by reporting the Pearson's correlation coefficient of the automatically discovered disease patterns with the publicly reported number of infections supplied for training, as well as syndromes that have been handcrafted by ourselves. In addition, we provide a detailed discussion of the indicators included in our models.

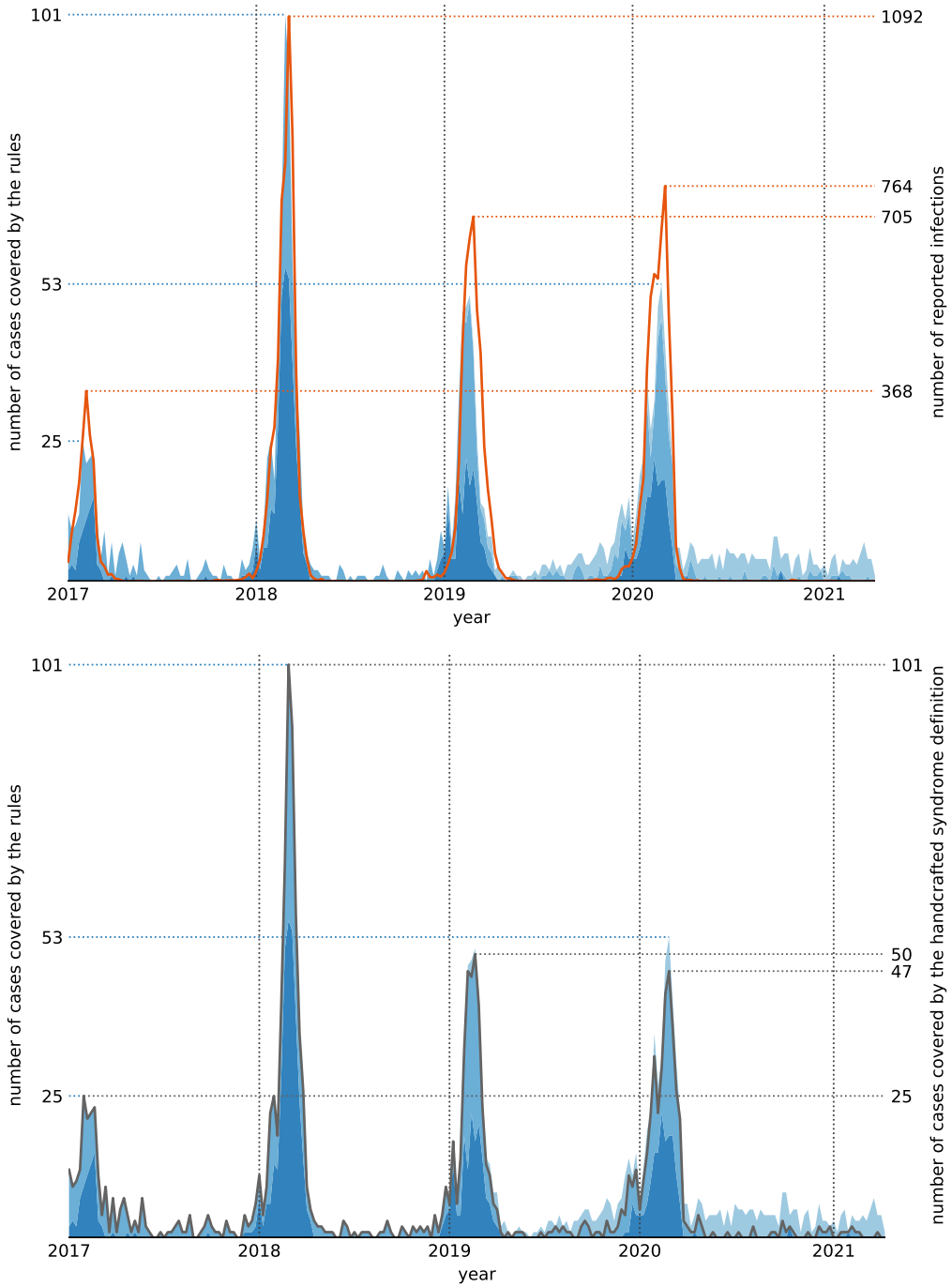


Figure 5.3: Number of cases satisfying the discovered syndrome definition for Influenza visualized as an stacked plot (blue). Rules of the discovered syndrome definition found during early iterations are colored darker. We compared the syndrome definition to the actual cases (top, orange) and the handcrafted syndrome (bottom, black).

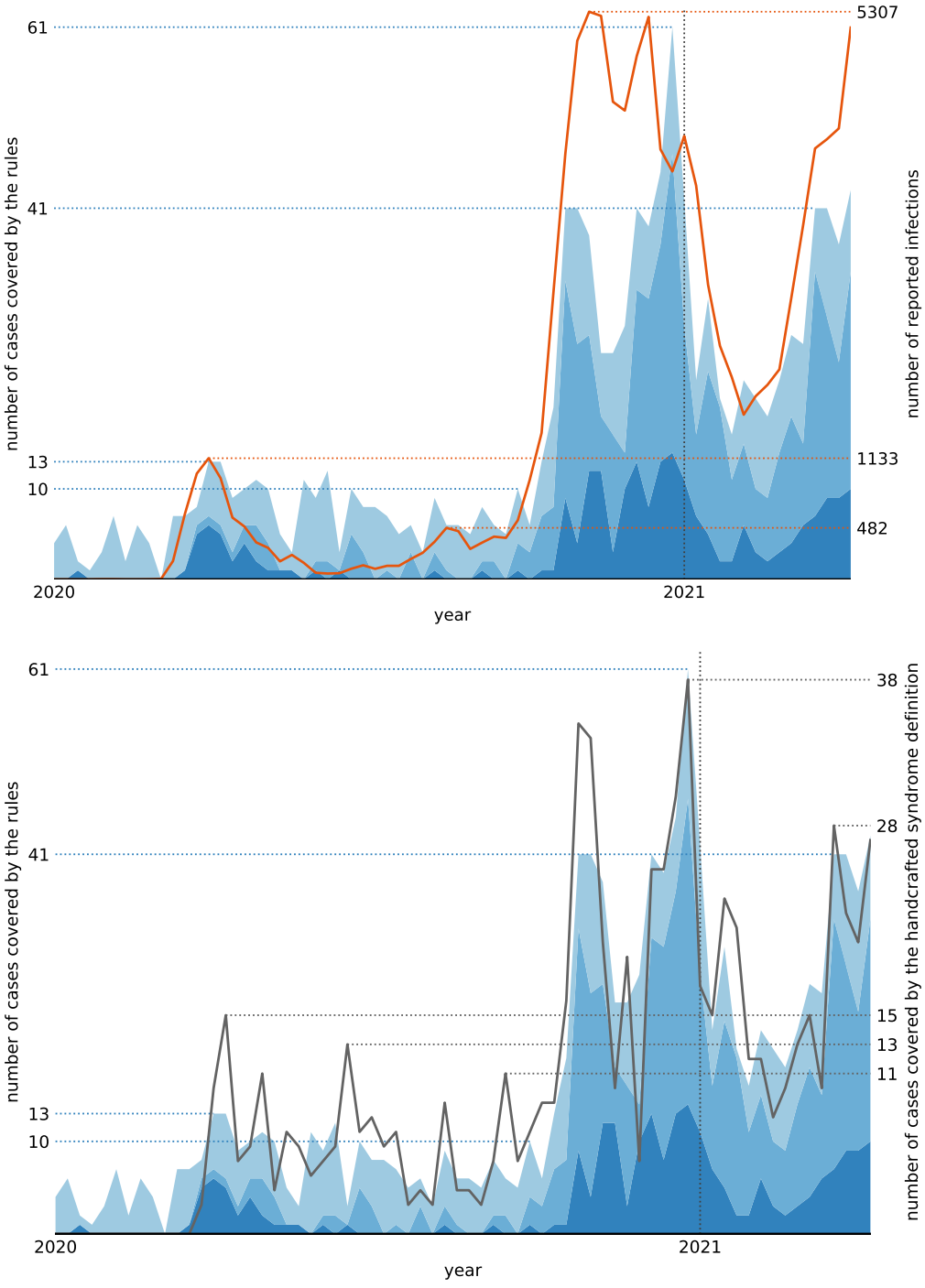


Figure 5.4: Number of cases satisfying the discovered syndrome definition for SARS-CoV-2 visualized as an stacked plot (blue). Rules of the discovered syndrome definition found during early iterations are colored darker. We compared the syndrome definition to the actual cases (top, orange) and the handcrafted syndrome (bottom, black).

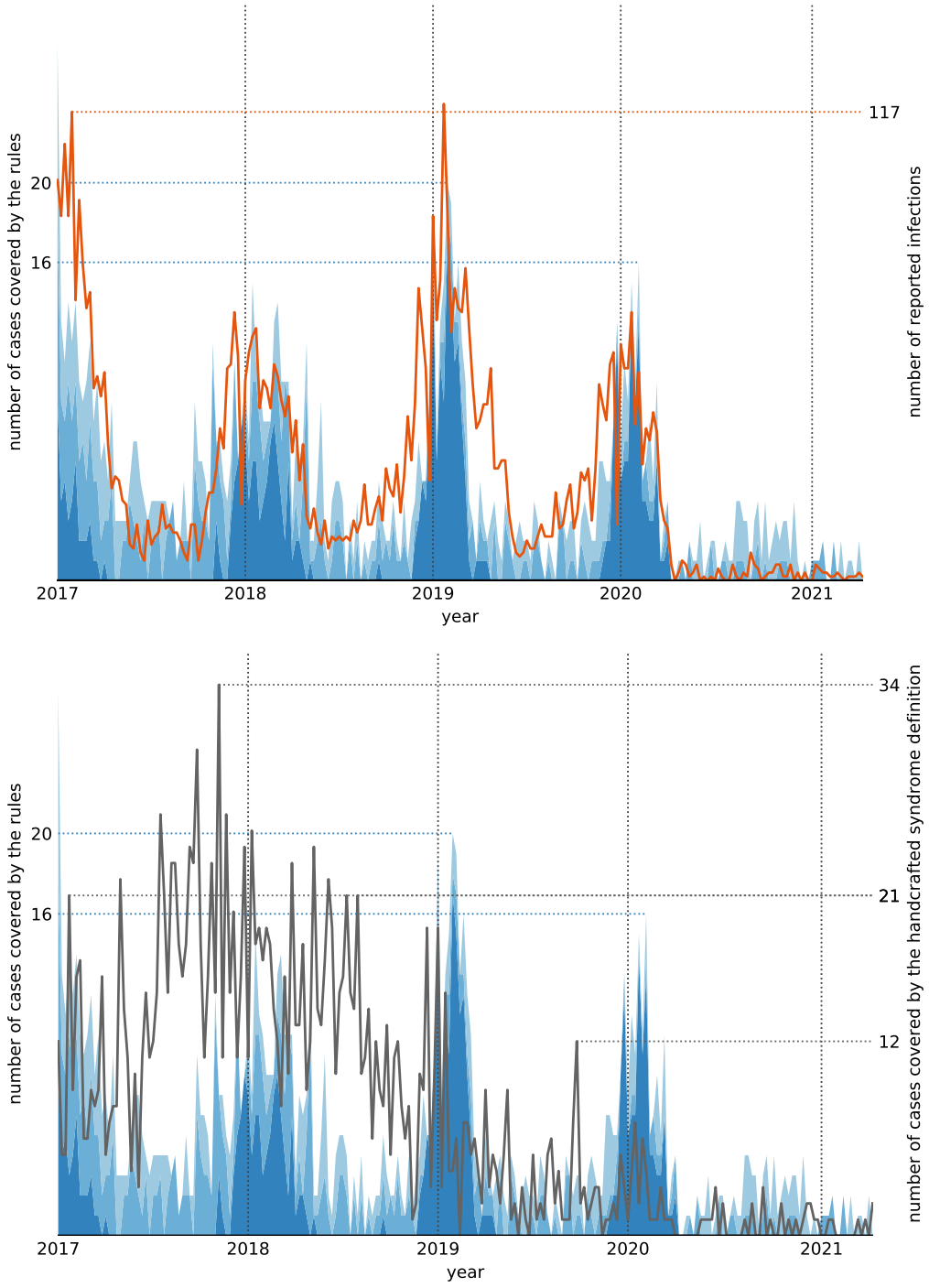


Figure 5.5: Number of cases satisfying the discovered syndrome definition for Norovirus visualized as an stacked plot (blue). Rules of the discovered syndrome definition found during early iterations are colored darker. We compared the syndrome definition to the actual cases (top, orange) and the handcrafted syndrome (bottom, black).

Table 5.2: Pearson's correlation coefficient between cases identified by automatically learned syndromes on different feature categories and actually reported cases as well as cases that match the handcrafted syndrome definitions.

disease	feature categories				reported cases	handcrafted syndromes
	①	②	③	④		
Influenza	✓				0.9354	0.9917
	✓	✓			0.9357	0.9796
	✓		✓		0.9480	0.9768
	✓			✓	0.9366	0.9948
	✓	✓	✓	✓	0.9493	0.9800
SARS-CoV-2	✓				0.9399	0.9473
	✓	✓			0.9454	0.9219
	✓		✓		0.9528	0.8689
	✓			✓	0.9464	0.9506
	✓	✓	✓	✓	0.9528	0.8689
Norovirus	✓				0.7669	0.2761
	✓	✓			0.7669	0.2761
	✓		✓		0.7303	0.1470
	✓			✓	0.7167	0.1608
	✓	✓	✓	✓	0.7242	0.1672

Inspired by the expert syndrome definitions for Influenza and SARS-CoV-2 used by Boender et al. (2021), we created a set of similar, but much simpler, definitions solely based on ICD codes. They incorporate the ICD codes that correspond to suspected or confirmed cases of a particular disease, i.e., *J10* (Influenza due to identified seasonal influenza virus) or *J11* (Influenza, virus not identified) for Influenza, *A08* (viral and other specified intestinal infections) for Norovirus and *U07.1* (COVID-19, virus identified) or *U07.2* (COVID-19, virus not identified) for SARS-CoV-2. We have found the number of cases these ICD codes apply to be very similar to those matched by the aforementioned expert definitions.

For each of the considered diseases, we trained several models using different sets of features. First of all, for a fair comparison with the handcrafted syndromes, we provided our algorithm with the features that belong to the first category in Table 5.1, i.e., ICD codes and MTS representations. A visualization of the number of infections that correspond to the disease patterns that have been discovered with respect to these features is shown for each disease in figures 5.3, 5.4, and 5.5. Each one of them includes a comparison with the reported number of infections supplied for training and the number of cases our handcrafted syndromes apply to, respectively. In the case of Influenza and SARS-CoV-2, all of these numbers are strongly correlated. In the first case, one can clearly observe an increase of infections during the first months of each year. In the latter case, the different peaks of SARS-CoV-2 infections according to the publicly reported numbers are replicated by both, the handcrafted syndromes and

the automatically learned patterns. The correlation between syndromes and reported numbers is less strong with respect to Norovirus. However, compared to the handcrafted syndromes, the automatically discovered patterns appear to better capture the seasonal outbreaks of this particular disease. In general, the numbers that correspond to the syndrome definitions are much lower than the reported numbers, as only a small fraction of detected cases have actually been treated in emergency departments.

In addition to ICD codes and MTS representations, we have also conducted experiments, where we provided the algorithm with one additional set of features, as well as with all features available. To validate whether the availability of additional features comes with an advantage for an accurate reproduction of the infected cases, we rely on the Pearson's correlation coefficients that result from different feature selections in Table 5.2. For all experiments, we report the correlation of the autonomously learned syndromes with both, the number of reported cases used for training and the cases captured by the handcrafted syndromes. In the case of Influenza and SARS-CoV-2, the inclusion of vital parameters introduces a minor advantage for matching the reported numbers. Understandably, the use of additional features typically reduces correlation with the handcrafted syndromes, as they do not make use of these features. In the case of Norovirus, the Pearson's correlation does not benefit from the availability of vital parameters. Regardless of any specific disease, this does also apply to the contextual and demographic information. We consider the absence of demographic indicators as positive, as none of the diseases appears to be specific to gender or age.

5.3.4 Discussion of Discovered Syndrome Definitions

As the use of ICD codes and MTS representations is sufficient in most cases to match the reported number of infections, we mostly focus on models that have been trained with respect to these features in the following discussion. A selection of exemplary syndromes that have been learned by our algorithm is also shown in Table 5.3.

Influenza. The indicators that have been selected by our algorithm for modeling the number of Influenza cases include the ICD codes *J10* and *J11* that are also included in our handcrafted definition. These indicators have been selected during the first iterations of the algorithm and therefore are considered more important than the subsequent ones. As indicated by using different shades of blue in Figure 5.3, patterns found during early iterations (dark blue) mostly focus on the strongly pronounced seasonal peaks. Indicators that have been selected at later iterations (lighter blue) are more likely to match irrelevant cases and hence are often unrelated to the respective disease. In the case of Influenza, this includes clearly irrelevant ICD codes, such as *Z96.0* (presence of urogenital implants) or *S53.1* (dislocation of elbow, unspecified) as fourth and fifth indicator, but also codes that may be related to Influenza-like illnesses,

Table 5.3: Exemplary automatically induced syndrome definitions.

① **Influenza**
J10 ∨ J11 ∨ "new confusion condition" ∨ Z96.0 ∨ ...

① **SARS-CoV-2**
(J12 ∧ "breathing problems") ∨ U07.1 ∨ "pain in lower abdomen" ∨ ...

① **Norovirus**
J21.0 ∨ D40 ∨ (J34 ∧ "recent problem")

① ③ **Influenza**
J10
∨ (J11 ∧ diastolic ≤ 92.5 ∧ systolic ≤ 156.5 ∧ temperature > 38.5)
∨ (temperature ≤ 40.5 ∧ diastolic ≤ 108.5 ∧
systolic ≤ 162 ∧ 187.5 ≤ heart rate ≤ 207.5)
∨ ...

① ② ③ ④ **Influenza**
J10
∨ (J11 ∧ diastolic ≤ 92.5 ∧ systolic ≤ 156.5 ∧ temperature > 38.5)
∨ (temperature ≤ 40.5 ∧ diastolic ≤ 110 ∧ systolic ≤ 162 ∧
187.5 ≤ heart rate ≤ 212.5 ∧ no isolation ∧ patient sent home)
∨ ...

D40	=	Neoplasm of uncertain/unknown behaviour of male genital organs
J10	=	Influenza due to identified seasonal influenza virus
J11	=	Influenza, virus not identified
J12	=	Viral pneumonia, not elsewhere classified
J21.0	=	Acute bronchiolitis due to respiratory syncytial virus
J34	=	Other disorders of nose and nasal sinuses
U07.1	=	COVID-19, virus identified
Z96.0	=	Presence of urogenital implants

such as *J18.8* (other pneumonia) or *J34.2* (deviated nasal septum) at positions 10 and 15. When the algorithm has access to vital parameters, the indicator *J11* is combined with information about blood pressure and body temperature as follows:

$$\begin{aligned} &\text{J11} \wedge \text{blood pressure diastolic} \leq 92.5 \\ &\quad \wedge \text{blood pressure systolic} \leq 156.5 \\ &\quad \wedge \text{temperature} > 38.5 \end{aligned}$$

Due to the lack of domain knowledge, we are not qualified to decide whether such a pattern is in fact characteristic of Influenza. This shows the demand for experts, who are indispensable for the evaluation of machine-learned models.

SARS-CoV-2. When used to learn patterns for SARS-CoV-2, our algorithm considers the MTS presentation "breathing problem", as well as the ICD codes *J12* (viral pneumonia) and *U07.1* (COVID-19, virus identified), as most relevant. The latter of these ICD codes is also included in the handcrafted syndrome definition. Besides clearly irrelevant indicators, it further selects the ICD code *J34.2* (deviated nasal septum) at a later stage of training that may be related to this particular illness. When provided with vital parameters, the algorithm decides to use the ICD code *J12* in combination with data about a patients' blood pressure and temperature in its most relevant pattern:

$$\begin{aligned} \text{J12} \wedge\ & 81.5 < \text{blood pressure systolic} \leq 149.5 \\ \wedge\ & \text{blood pressure diastolic} \leq 77.5 \\ \wedge\ & \text{temperature} > 36.5 \end{aligned}$$

Norovirus. When it comes to modeling the infections with Norovirus, the algorithm fails to identify any ICD codes that are related to this particular illness, such as the ones included in our manual definition or codes related to symptoms like diarrhea. Instead, it uses indicators like *J21.0* (acute bronchiolitis due to respiratory syncytial virus) or *J34* (other disorders of nose and nasal sinuses) in combination with other indicators to match the reported numbers. This is most probably due to the similar seasonality of Norovirus and Influenza-like illnesses. This illustrates another difficulty one may encounter when pursuing a data-driven approach to syndromic surveillance. If high numbers of infections with respect to multiple diseases occur during a similar timespan, the algorithm is not able to distinguish between indicators that relate to different types of infections. In such case it is necessary to provide additional knowledge to the learning algorithm, as it is unable to grasp the semantics of individual features on its own. In particular, this motivates the need for an interactive learning approach, where a human expert interacts with the computer in order to guide the construction of models. For example, by prohibiting the use of certain indicators or features that have been identified to be irrelevant to the problem at hand.

5.4 Discussion and Limitations

Our experimental evaluation using both, synthetic and real-world data, provided several insights into the problem domain addressed in this chapter. First of all, we were able to demonstrate that a correlation-based learning approach for the extraction of disease patterns is indeed capable of identifying meaningful indicators that are closely related to a particular disease under surveillance. In particular, the learned definitions showed a similar fit to the real distributions as handcrafted expert definitions (figures 5.3, 5.4, and 5.5). Also, the experiments with synthetic syndrome definitions

showed a good reconstruction rate, and the discovered real-world syndrome definitions contained plausible features.

Nevertheless, the frequent inclusion of unrelated indicators revealed some challenges and limitations of such an approach. Most of them relate to the fact that the training procedure has only limited access to the target information associated with each patient. In contrast to fully labeled data, where information about each patient's medical conditions are available, the learning method is restricted to broad information about a large group of individuals. In addition, the use of temporally aggregated data, depending on its granularity, introduces ambiguity into the learning process. As a result of these constraints, several solutions that satisfy the evaluation criterion to be optimized by the learner exist, even though many of them are undesirable from the perspective of domain experts. This is evident from the fact that the tested algorithm, regardless of the disease and the features used for training, was always able to find strongly correlated patterns, despite the use of unrelated indicators.

As another source of problems, we identified the noise, potential inconsistencies and missing pieces of information that may be encountered when dealing with unprocessed and unfiltered real-world data. The consequences become most obvious when taking a look at the results with respect to Norovirus, where the algorithm failed to detect meaningful syndrome descriptions due to the overlap to other, more frequent, diseases with a similar seasonality and more pronounced patterns in the reported numbers.

So far, we were only interested in the identification of patterns that match the target variables as accurate as possible. However, the goal of machine learning approaches usually is to obtain predictions for unseen data. To be able to generalize well beyond the provided training data, this requires models to be resistant against noise and demands for techniques that effectively prevent overfitting. The incorporation of such techniques into our learning approach may improve its ability to find useful patterns despite the noise and ambiguities that are present in the data.

For example, successful rule learning algorithms often come with pruning techniques that aim at removing problematic clauses from rules after they have been learned. This requires to split up the training data into multiple partitions in order to be able to obtain unbiased estimates of a rule's quality, independent of the data used for its induction. By splitting up the time series data, the quality of indicators that are taken into account for the construction of syndromes could more reliably be assessed in terms of multiple, independent estimates determined on different portions of the data. Despite such technical solutions, we believe that the active participation of domain experts is indispensable for the success of machine-guided syndromic surveillance. An interactive learning approach, where the syndromes that are discovered by an algorithm are suggested to epidemiologists and feedback is fed back into the system, may prevent the inclusion of undesired patterns and would most likely help to increase the acceptance of machine learning methods among healthcare professionals.

Furthermore, we consider the use of the Pearson's correlation coefficient as a limitation of our approach. When modeling the outbreak of a disease, it is especially important to properly reflect the points in time that correspond to high numbers of infections. Other correlation measures, like weighted variants of the Pearson's correlation coefficient, may provide advantages in this regard. We expect this aspect to be particularly relevant when modeling rather infrequent diseases with generally low incidences.

Another problem are possible discrepancies between the data obtained from the emergency departments and the data that incorporates information about the number of infections, e.g., resulting from reporting delays. To circumvent potential issues that may result from such inconsistencies, approaches that have specifically been designed for measuring the similarity between temporal sequences, like dynamic time warping (Keogh and Pazzani, 2001), could be used in the future. They allow for certain static, and even dynamic, displacements of the sequences to compare.

5.5 Conclusion

In this chapter, we have presented a novel approach for the automatic induction of syndrome definitions from health-related data sources. As it aims at finding patterns that correlate with the reported numbers of infections, as provided by publicly available data sources, there is no need for labeled training data. This reduces the burdens imposed on domain experts, who otherwise must manually create labeled data in a laborious and time consuming process. Although the proposed algorithm is able to identify meaningful indicators, due to artifacts in the data and technical limitations, we have found that autonomously created syndromes are likely to include indicators that are unrelated to the disease under surveillance. As a result, the knowledge of experts is still indispensable for the evaluation and supervision of such a machine learning method. Nevertheless, our investigation shows the potential of data-driven approaches to syndromic surveillance, due to their ability to process large amounts of data that cannot fully be understood and analyzed by humans.

In the future, we plan to investigate technical improvements to our algorithm that may help to prevent overfitting and allow for a more extensive, yet computationally efficient, exploration of promising combinations of indicators. In addition, valuable insights can possibly be obtained by applying our approach to different types of health-related data sources, as well as by the investigation of different correlation measures that can potentially be used to guide the search for meaningful syndromes.

6

Stacking of Statistical Surveillance Methods

In the previous chapter, we focused on the definition of syndromes to track the number of infections for a particular disease. In order to detect an outbreak in such a time series, the number of infections are usually monitored with statistical surveillance methods. As outlined in Chapter 4, the practical usefulness of these methods highly depends on the reliability of their output. As our first step towards improving the detection of outbreaks, we propose an approach to combine the output of multiple statistical surveillance methods using a machine learning technique named stacking (Wolpert, 1992). Therefore, we set our focus on traditional surveillance (cf. Section 4.1) and use the univariate time series of infection counts as the only source of information.

Prior work on improving disease surveillance mainly focuses on forecasting the number of infections for a disease (e.g., Chakraborty et al. (2014); Farrow et al. (2017)). A comparably lower amount of research has been devoted to improving statistical algorithms to raise alarms. Jafarpour et al. (2015) used *Baysian networks* to identify the determinants for detection performance to find appropriate algorithm configurations for outbreak detection methods. In particular, classification algorithms and voting schemes have been used for the fusion of outbreak detection methods on univariate time series (Jafarpour et al., 2013; Texier et al., 2019) as well as on multi-stream time series (Burkom et al., 2011; Lau et al., 2008; Mnatsakanyan et al., 2009). However, the examined approaches only rely on the binary output (*alarm* or *no alarm*) of the underlying statistical methods for the fusion which limits the information about a particular observation. Prior research in the area of machine learning has shown that more precise information of the underlying models improves the overall performance of the fusion (Ting and Witten, 1999).

In this chapter, we propose an approach for the fusion of outbreak detection methods which uses the p-values of the underlying statistical methods. Moreover, one can also incorporate different information for the outbreak detection (e.g., weather data, holidays, or statistics about the data) by augmenting the data with additional attributes. In addition, the way outbreaks are labeled in the data also has a major influence on

the learnability of outbreak detectors. Thus, we propose adaptions for the labeling of outbreaks in order to maximize the detection rate of machine learning algorithms.

Furthermore, in line with Kleinman and Abrams (2006), we propose a method which uses the p-values of the statistical methods in order to evaluate their performance. In particular, we propose a variant of Receiver Operating Characteristic (ROC) curves, which shows the false alarm rate on the x-axis and the detection rate—in contrast to the true positive rate—on the y-axis. By using the area under a *partial* ROC curve (Ma et al., 2013), we obtain a measure for the performance of an algorithm that satisfies a given constraint on the false alarm rate (e.g., less than 1% false alarms). This criterion serves as the main measure for our evaluations and enables us to analyze the trade-off between the false alarm rate and the detection rate of outbreak detection methods precisely.

6.1 Statistical Algorithms for Disease Surveillance

The key idea of our approach is to learn to combine predictions of commonly used statistical outbreak detection methods with a trainable machine learning algorithm. Thus, we first need to generate a series of aligned prediction vectors, each consisting of one entry for each method. This sequence can then be used for training the machine learning model.

We have chosen the EARS methods, the Bayes method and the RKI method. For detailed description of these methods, we refer to Section 4.4. They all have in common that they require comparably little historic data on their own, which allows us to train the machine learning method on longer sequences. Moreover, such methods are universally applicable and serve as drop-in approaches for surveillance systems since they only rely on the detection of a local increase in incidents without the need to capture effects like seasonality and trend.

6.2 Fusion Methods

The combination of information from several sources in order to obtain a unified picture is known as *fusion* (Khaleghi et al., 2013). *Classifier fusion* is a special case which combines the outputs of multiple classifiers in order to improve classification performance. In our context, the statistical algorithms for disease surveillance can be seen as classifiers, each classifying the current observation into the classes *alarm* or *no alarm*.

A straightforward way for combining the predictions of multiple outbreak detection methods is to simply vote and follow the majority prediction. A more sophisticated approach consists of training a classifier that uses the predictions of the detection methods as input, and is trained on the desired output, a technique that is known in machine learning as *stacking* (Wolpert, 1992).

Recent work in the area of outbreak detection and fusion has focused on fusing the information obtained by simultaneously monitoring multiple time series for a particular disease. Lau et al. (2008) have shown that the performance of statistical algorithms can already be improved by combining them with simple voting schemes. Mnatsakanyan et al. (2009) could further improve the performance using Bayesian networks and including further information about the patients (e.g., age) as additional attributes. Moreover, Burkom et al. (2011) have used a hierarchy of Bayesian networks in order to incorporate additional information about health surveillance data and environmental sensors.

Only little research has been devoted to improving the performance of statistical algorithms on univariate time series. In particular, Texier et al. (2019) have used the machine learning technique *hierarchical mixture of experts* (Jordan and Jacobs, 1994) to combine the output of the methods from EARS. However, the authors note that all algorithms rely on the assumption of a Gaussian distribution, which limits their diversity. In contrast, Jafarpour et al. (2013) have used a variety of classification algorithms (*logistic regression*, *CART* and *Baysian Networks*) for the fusion of outbreak detection methods. As underlying statistical algorithms they have used the Cumulative Sum (CUSUM), two Exponential Weighted Moving Average algorithms, the EARS methods (C1,C2,C3) and the Farrington algorithm (Noufaily et al., 2013). In general, the results of Texier et al. (2019) and Jafarpour et al. (2013) indicate that machine learning improves the ability to detect outbreaks while simple voting schemes (e.g., weighted voting and majority vote) did not perform well. Moreover, the algorithms have not been evaluated with respect to data which include seasonality and trend.

6.3 Fusion with Augmented Stacking

Prior work only focused on the fusion of the binary outputs (*alarm* or *no alarm*) of the underlying statistical methods, which limits the available information about a particular observation. In this chapter, we show that the availability of additional information can further improve the performance of the fusion classifier. Therefore, we first propose to use p-values of the statistical methods for the fusion in order to include information about the certainty of an alarm, and then show how to add additional external information to the learning process of the machine learning algorithm. Finally, we investigate different variants for labeling outbreaks.

6.3.1 Fusion with p-values

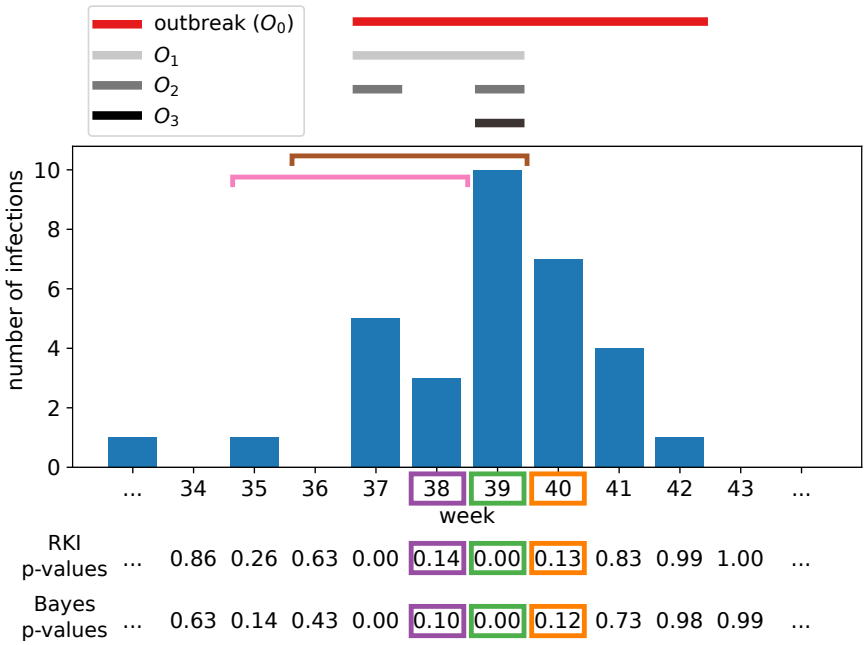
Given base estimators $g_1(x), \ldots, g_K(x)$, a *fusion combiner* $h(g_1(x), \ldots, g_K(x))$ is a function that combines the predictions of the base functions. In the simple case of binary voting, i.e., $g_i(x) \in \{0, 1\}$, the combiner $h(x) = \frac{1}{K} \sum_i g_i(x)$ with a threshold of 0.5 would model the majority rule. In *stacking* the function $h : X^K \rightarrow Y$ is learned by training a machine learning classifier on a set of previous observations $(g_1(x_1), \ldots, g_K(x_1)), \ldots, (g_1(x_n), \ldots, g_K(x_n))$ –derived from applying g_i on x_t– with associated targets $y_1, \ldots, y_n \in Y$. We refer to this as the training set in contrast to the evaluation set, which contains new, unseen observations. In outbreak detection, the instances x_t correspond to the points in the time series $C = (c_0, c_1, \ldots, c_n) \in \mathbb{N}^n$ of infection counts c_t and $y_t \in \{0, 1\}$ denotes the labeling of a time point as belonging to an outbreak (1) or not (0).

Previous approaches (Jafarpour et al., 2013; Texier et al., 2019) used the binary alarms ($\{0,1\}$) of base outbreak detectors. In this chapter instead, we propose to base our stacking model on the p-values, i.e., $g_i(x) \in [0, 1]$, provided by the underlying statistical approaches (cf. Section 6.1). In fact, the p-values can directly be seen as the certainty of currently observing an outbreak, enabling the learning algorithm to make use of the base estimations in a much more fine grained way. This information is otherwise lost when using binary alarms, which are indeed obtained by just applying a fixed threshold on the computed p-values. In addition to the circumvented difficulty of tuning such threshold, previous studies on stacking have shown empirically that using the raw predictions can improve over the discretized option (Ting and Witten, 1999).

Figure 6.1 visualizes an example on how the data for the learning algorithm is created by using the p-values of the statistical algorithms Bayes and RKI. The columns RKI_t and Bayes_t represent the computed p-values for the current observation while the other columns (mean_t, RKI_{t-1} and Bayes_{t-1}) represent additional information explained in the following section.

6.3.2 Additional Features

The use of a trainable fusion method allows us to include additional information which can help to decide whether a given alarm should be raised or not. As additional features, we propose to include the *mean* of the counts over the last m time points (the same number of time points as used by the statistical methods), which can give us evidence about the reliability of a particular outcome. For example, the assumption of a Gaussian distribution for a low mean of count data (≤ 20) is known to be imprecise. Therefore, a learning algorithm might induce in this scenario that the p-values of



index week t	augmented features			p-values		target
		prev. p-values				
	$\mathbf{mean}_t$	$\mathbf{RKI}_{t-1}$	$\mathbf{Bayes}_{t-1}$	$\mathbf{RKI}_t$	$\mathbf{Bayes}_t$	$\mathbf{outbreak}_t$
...	...	...	...	...	...	...
34	1.00	0.59	0.63	0.86	0.63	no
35	0.50	0.86	0.63	0.26	0.14	no
36	0.50	0.26	0.14	0.63	0.43	no
37	0.50	0.63	0.43	0.00	0.00	yes
38	1.50	0.00	0.00	0.14	0.10	yes
39	2.25	0.14	0.10	0.00	0.00	yes
40	4.50	0.00	0.00	0.13	0.12	yes
41	6.25	0.13	0.12	0.83	0.73	yes
42	6.00	0.83	0.73	0.99	0.98	yes
43	5.50	0.99	0.98	1.00	0.99	no
...	...	...	...	...	...	...

Figure 6.1: Example for the creation of training data for the learning algorithm including the statistical algorithms Bayes and RKI with a window size of one ($w = 1$) and the mean over the previous four counts ($m = 4$) as features. On the top, the time series for a particular disease is visualized representing the number of cases of infections over time. The computed p-values of the statistical algorithms (underneath) and the label indicating an outbreak for each observation (above) are placed at the respective time index t. Using this information the data instances can be created as shown on the bottom table: Each particular time point is represented by one training instance, labeled according to the original targets O_0.

the EARS methods C1, C2 and C3 may not be trustworthy. Moreover, under the assumption that a time series is stationary an unusual high mean can also be a good indicator to detect an outbreak, especially in the case that an outbreak arises slowly over time. The column mean_t in Figure 6.1 illustrates how the mean over the last four observed counts ($m = 4$) is added as an additional feature.

Finally, we also include the output of the statistical methods for previous time points in a window of a user-defined size w as additional features. For the example in Figure 6.1, we have used a window size of one ($w = 1$) which includes the previous output of both statistical algorithms.

6.3.3 Modelling the Output Labels for Learning

A major challenge for machine learning algorithms is that the duration of an outbreak period is not clearly defined (Shmueli and Burkom, 2010). A simple strategy—which we refer to as O_0—is to label all time points positive as long as cases for the particular epidemic are reported (e.g., time points prior to the peak of an outbreak and a few time points after the peak). In this case, the goal of the learning algorithm is to predict most time points in an ongoing epidemic as positive, regardless of their time stamp. Indeed, our early results indicate that the predictor learns to recognize the fading-out of an outbreak (e.g., weeks 40 to 42 in Figure 6.1). This is due to the fact that the peak of the outbreak is included in the reference values which results in a considerably high mean $\mu(t)$ for the significance test. Because of this, unusually high p-values are generated for the counts after the peak, which provide sufficient evidence for the stacking algorithm to raise an alarm. However, this also increases the number of false alarms as the machine learning approach learns to raise alarms when the count is decreasing outside an epidemic period.

To avoid this, we propose three adaptations of O_0: O_1 labels all time points until the peak (the point with maximum number of counts during the period) as positive. O_2 instead skips the time points whose count is decreasing compared to the immediate previous count (i.e., it labels all increasing counts until reaching the peak). Finally, O_3 labels only the peak of the outbreak as positive. Figure 6.1 visualizes an example outbreak with the corresponding options to label the epidemic period on the top.

6.4 Evaluation Measures

Instead of manually adjusting the α parameter of the statistical methods and examining the results individually, which is mostly done in previous works, we propose to evaluate the p-value as it is done by Kleinman and Abrams (2006).

In particular, the p-value can be interpreted as a score, which sorts examples according to their degree to which they indicate an alarm. This allows us to analyze an algorithm with ROC curves (Fawcett, 2006). A ROC curve can be used to examine the trade-off between the *true positive rate* (i.e., the probability of raising an alarm in case of an actual outbreak) and the *false alarm rate* (i.e., the probability of falsely raising an alarm when no outbreak is ongoing). In order to only focus on high specificity results (e.g., with a false alarm rate below 1%), which is of major importance for many medical applications, we only consider *partial ROC curves*. By using the partial area under the ROC curve as proposed in Ma et al. (2013), we obtain a simple measure to evaluate the performance of an algorithm, satisfying particular constraint on the false alarm rate. We refer to this measure as AUC_τ where the parameter τ defines the maximum allowed false alarm rate to be considered. It is computed as

$$AUC_\tau = \frac{\int_0^\tau ROC(f)\ df}{\tau}$$

where $ROC(f)$ denotes the true positive rate given a false alarm rate of f.

However, alarms raised in cases when the epidemic has already been detected are typically not very decisive and informative anymore. To incorporate this, we consider the *detection rate*, which represents the proportion of recognized outbreaks (i.e., the outbreaks in which at least one alarm is raised during their activity). Following Kleinman and Abrams (2006) and Jafarpour et al. (2013), we therefore use a ROC curve-like representation with the detection rate on the y-axis instead of the true positive rate, and use $dAUC_\tau$ to refer to the partial area under this curve. Figure 6.2 shows an example of the ROC-curve like representation and visualizes the area of

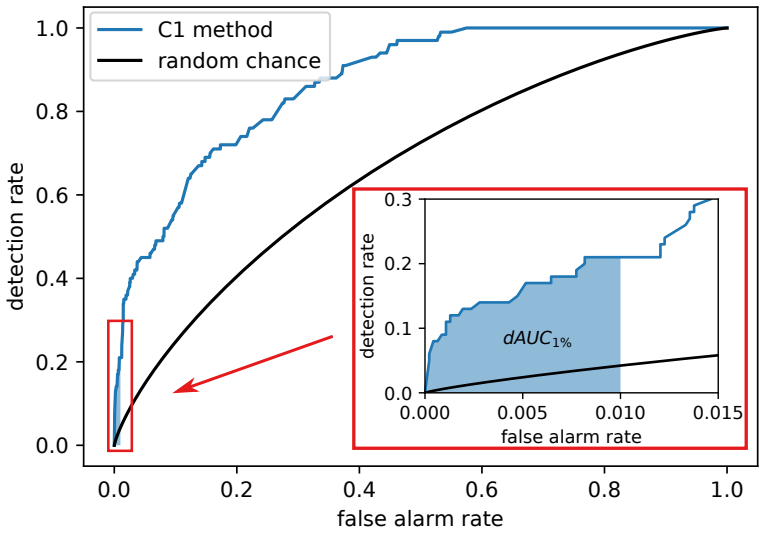


Figure 6.2: ROC curve using the detection rate on the y-axis. The better-than-chance performance is lifted above the diagonal since the detection rate is an interval-based metric.

$dAUC_{1\%}$. Kleinman and Abrams (2006) proposed to use weighted ROC curves to also incorporate the influence of the measure timeliness (mean time to detect an outbreak). However, we argue that the weighing with the timeliness introduces a trade-off (importance of timeliness over detection rate) and a loss in interpretability of the absolute numbers. If timeliness is of interest, we suggest to use AMOC curves (cf. Section 4.5) which use the timeliness on the y-axis instead of the true positive rate or the detection rate.

6.5 Evaluation

The key aspect of our experimental evaluation is to demonstrate that the fusion of p-values leads to a further improvement in performance compared to only using the binary output of the statistical algorithms. For a deeper understanding of our proposed approaches, we first performed experiments on synthetic data to evaluate the influence of our adaptions on stacking and to compare it with the underlying statistical algorithms in a controlled environment. Afterwards, we test stacking on real data in order to underline its practical utility.

6.5.1 Experimental Setup

Baselines. As an implementation baseline for the statistical methods, we have used the R package *surveillance* (Salmon et al., 2016) and adapted the implementation of the methods EARS (C1, C2, and C3), Bayes, and RKI so that they also return p-values. All methods use the previous seven time points as reference values, which is the standard configuration. We have evaluated the underlying statistical methods itself which serve as a baseline to which the stacking approaches are compared. In addition, we also evaluated the fusion method which only combines the binary outputs of the statistical methods as proposed in Jafarpour et al. (2013); Texier et al. (2019) and to which we refer to as *standard fusion*.

Measures. For all evaluations, we focus on the evaluation measure $dAUC_{1\%}$ proposed in Section 6.4. In addition, we evaluate the conventional area under partial ROC-curve $AUC_{1\%}$ to further investigate the effect of the labeling on the true positive rate.

Parameter optimization. We have evaluated three machine learning algorithms: (1) Random Forest (cf. Section 3.4.3) with the different values for the minimum number of samples per leaf $\{5, 10, 20, 30\}$, (2) Logistic Regression (cf. Section 3.4.1)

with different values for the regularization parameter $\{2^{-3}, 2^{-2}, 2^{-1}, 1, 2^1, 2^2, 2^3\}$, and (3) K-Nearest-Neighbours (cf. Section 3.4.2) with different values for the number of considered neighbours $\{1, 3, 5, 7, 9\}$. For the labeling, we have the parameters $\{O_0, O_1, O_2, O_3\}$, $\{False, True\}$ for the use of mean, and $\{0, 1, 2, 3, 4, 6, 9\}$ for the windowing where 0 represents no windowing. For each algorithm we performed a grid search in which we evaluate all possible parameter combinations. For the standard fusion approach, we have set the threshold of each single statistical method beforehand on the validation data so that it has a false alarm rate of 1% and then performed the parameter optimization. The reported results can always be attributed to one of the random forest configurations since this approach always achieved the best results in the parameter optimization, on the synthetic as well as on the real data.

Synthetic data. For the generation of synthetic data, we use the data generator proposed in Noufaily et al. (2013). In total, 42 different settings to generate time series (*test cases*) are proposed which reflect a wide range of application scenarios allowing to explicitly analyze the effects of trend (T), seasonality (S1) and biannual seasonality (S2). For each test cases, 100 time series are created, using the first 575 weeks of all 100 time series to train an machine learning model and evaluated the created model on the last 49 weeks of the time series. The parameter k of the data generator, used to estimate the number of cases per outbreak, is randomly drawn from the range $1 \ldots 10$ for each outbreak. Instead of reporting averaged $dAUC_{1\%}$ scores, which could have different scales for different test cases, we determined a ranking over the methods for each considered test case. Afterwards we computed each method's average ranking, 1 being the best rank.

For the parameter optimization on synthetic data, we created new time series for each test case by using a different random seed and used these to evaluate all parameter combinations. We wanted to optimize the performance across all test cases and, therefore, picked the parameter combination with the best average rank.

Real data. For the evaluation on real data, we rely on the reported cases for the diseases *Salmonella* (SAL) and *Campylobacter* (CAM) in Germany, which are captured by the *Robert Koch-Institut*. The reported cases are aggregated by the public health offices with respect to the 401 districts (*Landkreise* and *Stadtkreise*) in Germany and range from 2001 to 2018. Only for Berlin we obtain a finer granularity (12 sub-districts), resulting in a total of 412 time series for each disease. In addition, each reported case can be associated with a specific outbreak. The outbreaks were labeled in a retrospective manner by grouping all cases which can be assigned to a specific outbreak reason. However, not always all cases relating to a specific outbreak are reported since not all people visit the doctor nor is it possible to identify the source for catching the disease (cf. Section 4.3). Due to this, a lot of outbreaks only contain a few cases which are hard to detect. Therefore, we have generated three application

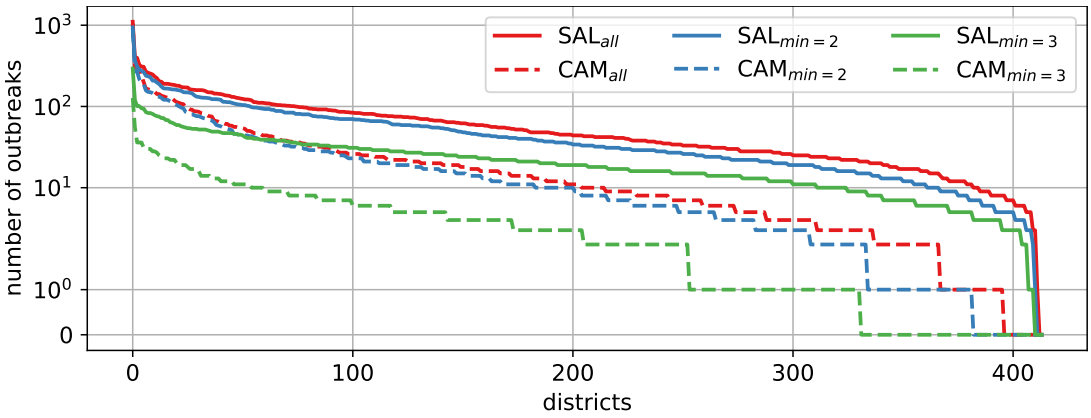


Figure 6.3: Number of outbreaks for each application scenario and each district (ordered according to number of outbreaks) in Germany from 2001 until 2018.

scenarios: (1) Use all outbreaks (*all*), (2) use the outbreaks consisting of at least two cases (*min=2*) and (3) use the outbreaks which contain at least three cases (*min=3*). We obtain the outbreak labeling by labeling all weeks positive as long as cases are reported for a particular outbreak. In Figure 6.3 the number of outbreaks for the six data configurations over all 18 years for each district is visualized.

We use the first 884 weeks (17 years) of all districts to train the machine learning model and evaluate on the last 52 weeks of all districts with the trained model. For parameter optimization, we split each time series into 18 folds $\{f_1, \ldots, f_{18}\}$ and performed for each parameter combination five evaluations $i \in \{1, \ldots, 5\}$, using the weeks of the folds f_1 to f_{11+i} of all districts as training data and evaluated on the weeks of fold f_{12+i} of all districts. Afterwards, we average the result for the measure $dAUC_{1\%}$ across the five evaluations and then choose the parameter configuration maximizing this score.

6.5.2 Results on Synthetic Data

Based on the controlled environment of the synthetic data, we first analyze the effect of the additional information and the labeling. For these experiments, we chose to evaluate *random forest* as the fusion classifier, which has proven to be robust in performance theoretically and practically (Fernández-Delgado et al., 2014; Wyner et al., 2017). Each model is, therefore, composed of 100 decision trees with a minimum number of 5 instances per leaf and default settings otherwise. In addition, we have used $\alpha = 0.5\%$ for the underlying statistical methods of the standard fusion method which performed best in our preliminary experiments. In the last experiment of this section, the fusion approaches are compared to the underlying statistical approaches using optimized parameter configurations.

Table 6.1: Comparison of including or not including the mean in the data for machine learning algorithms. Each column shows the average ranks considering different test case combinations: *overall* denotes all 42 test cases, the other columns cases (not) containing trend ($[\neg]T$), annual ($[\neg]S1$) or biannual ($[\neg]S2$) seasonality.

approach	**overall**	$\{\neg T, \neg S1, \neg S2\}$	$\{\neg T, S1, \neg S2\}$	$\{\neg T, S1, S2\}$	$\{T, \neg S1, \neg S2\}$	$\{T, S1, \neg S2\}$	$\{T, S1, S2\}$
standard fusion(no mean)	3.429	3.714	3.571	**3.000**	**3.429**	**3.286**	3.571
standard fusion(mean)	**3.357**	**2.571**	**3.286**	3.571	3.571	3.714	**3.429**
p-value fusion(no mean)	1.905	2.571	1.857	2.000	1.714	1.714	1.571
p-value fusion(mean)	**1.310**	**1.143**	**1.286**	**1.429**	**1.286**	**1.286**	**1.429**

Table 6.2: Comparison of different window sizes for the data (including the mean and using the labeling O_0). Each column shows the average ranks considering different test case combinations: *overall* denotes all 42 test cases, the other columns cases (not) containing trend ($[\neg]T$), annual ($[\neg]S1$) or biannual ($[\neg]S2$) seasonality.

approach	**overall**	$\{\neg T, \neg S1, \neg S2\}$	$\{\neg T, S1, \neg S2\}$	$\{\neg T, S1, S2\}$	$\{T, \neg S1, \neg S2\}$	$\{T, S1, \neg S2\}$	$\{T, S1, S2\}$
standard fusion($w = 0$)	9.738	9.000	9.571	**8.286**	11.143	10.571	**9.857**
standard fusion($w = 1$)	**8.738**	**8.857**	**7.000**	9.000	**7.571**	**8.857**	11.143
standard fusion($w = 2$)	10.762	10.571	10.857	10.714	10.571	10.143	11.714
standard fusion($w = 4$)	11.310	11.429	11.714	11.714	10.857	12.000	10.143
standard fusion($w = 6$)	11.619	12.714	12.286	10.286	11.571	11.857	11.000
standard fusion($w = 8$)	11.548	11.143	11.571	12.000	10.857	12.000	11.714
standard fusion($w = 12$)	11.929	12.143	12.143	13.000	11.714	11.571	11.000
p-value fusion($w = 0$)	5.000	5.714	5.000	5.714	4.429	3.714	5.429
p-value fusion($w = 1$)	**3.405**	**3.143**	**2.571**	4.571	**3.286**	4.286	**2.571**
p-value fusion($w = 2$)	4.381	5.000	4.714	4.000	4.571	4.571	3.429
p-value fusion($w = 4$)	4.667	4.143	5.000	4.000	5.143	5.286	4.429
p-value fusion($w = 6$)	4.310	5.000	4.429	3.857	3.857	3.857	4.857
p-value fusion($w = 8$)	4.000	3.000	4.000	4.714	5.000	3.857	3.429
p-value fusion($w = 12$)	3.595	**3.143**	4.143	**3.143**	4.429	**2.429**	4.286

Evaluation of additional features. The first aspect to review concerns the inclusion of the mean count over the last seven time points. Therefore, we have analyzed the effect of this feature independent of the other parameters using O_0 for the labeling of the outbreak and window size $w = 0$. The results for the average rank are displayed in Table 6.1. Comparing the standard to the p-value fusion method reveals a beneficial effect especially for the p-value approach, for which the variant including the mean achieves an average rank of 1.31 over 1.91. In contrast, the average ranks of 3.36 over 3.43 for the standard method not only shows that there are issues regarding the usage of the mean for some of the test case configurations, but also the substantial gap between using the binary outputs and the more fine-grained p-values. A closer examination reveals that the best improvement for both fusion methods can be achieved on time series without trend and seasonality. By adding effects like trend and seasonality, the mean changes over time, making it difficult for the learning algorithm to use this information. In contrast to the standard fusion, the p-value fusion method still enhances by including the mean over the previous time points.

The observation that the p-value fusion method is superior to the standard fusion can also be seen when comparing different window sizes. The results of this experiment, using O_0 for the labeling of the outbreak and not including the mean, are displayed in Table 6.2. In particular, no window configuration of the standard fusion method can outperform any of the p-value configurations with respect to the average rank. Overall, a window size of 1 performed best for both fusion approaches. Being able to compare to the most immediate previous output of the underlying statistical algorithms seems to make it easier to detect anomalies. In contrast, larger window sizes harm the overall performance, which suggests that the additional information is not relevant for detecting sudden changes and rather confuses the learner. Interestingly, on certain combinations of trend and seasonality a larger window size for the p-value fusion method seems to be beneficial. Actually, the increase of the window size also results in taking a further look back in the past allowing to detect effects like trend and seasonality achieving good results on the test cases which only contain biannual seasonality. However, the observed results for larger window sizes are inconsistent across the different test cases, making it difficult to draw valid conclusions.

Evaluation of the labeling adaptions. In addition to augmenting the input data, we have evaluated the effect of adapting the labeling of the epidemic period for the training of the stacking algorithm. The comparison shown in Table 6.3 was performed without the augmentation.

In general, we can observe that by narrowing the labeling of the outbreak on particular events (i.e., O_1, O_2 or O_3) a better performance can be achieved. This effect is clearly visible for the p-value fusion method and less obvious for the standard fusion method, for which the adaption O_1 seems to be an exception. In particular, learning only the peaks (O_3) achieved the best results for both fusion approaches. The benefit of

Table 6.3: Comparison of the different labeling strategies for the epidemics (not using the average and $w = 0$). Each column shows the average ranks considering different test case combinations: *overall* denotes all 42 test cases, the other columns cases (not) containing trend ($[\neg]T$), annual ($[\neg]S1$) or biannual ($[\neg]S2$) seasonality.

approach	**overall**	$\{\neg T, \neg S1, \neg S2\}$	$\{\neg T, S1, \neg S2\}$	$\{\neg T, S1, S2\}$	$\{T, \neg S1, \neg S2\}$	$\{T, S1, \neg S2\}$	$\{T, S1, S2\}$
standard fusion (O_0)	6.476	6.286	5.571	**4.857**	7.143	7.571	7.429
standard fusion (O_1)	6.738	7.286	6.714	6.286	6.429	7.000	6.714
standard fusion (O_2)	5.738	6.286	5.714	5.286	**5.429**	6.000	**5.714**
standard fusion (O_3)	**5.524**	**5.286**	**5.143**	5.429	6.000	**5.429**	5.857
p-value fusion (O_0)	3.762	3.857	3.857	**2.714**	4.857	4.000	3.286
p-value fusion (O_1)	3.262	2.857	4.143	4.857	2.429	2.429	2.857
p-value fusion (O_2)	2.690	3.143	3.000	3.286	2.714	2.143	**1.857**
p-value fusion (O_3)	**1.810**	**1.000**	**1.857**	3.286	**1.000**	**1.429**	2.286

Table 6.4: Results evaluated on synthetic data using optimized parameter configurations. Each column shows the average ranks considering different test case combinations: *overall* denotes all 42 test cases, the other columns cases (not) containing trend ($[\neg]T$), annual ($[\neg]S1$) or biannual ($[\neg]S2$) seasonality.

approach	**overall**	$\{\neg T, \neg S1, \neg S2\}$	$\{\neg T, S1, \neg S2\}$	$\{\neg T, S1, S2\}$	$\{T, \neg S1, \neg S2\}$	$\{T, S1, \neg S2\}$	$\{T, S1, S2\}$
C1	5.738	6.571	6.000	4.571	6.857	5.428	5.000
C2	4.905	4.000	4.429	5.143	4.714	5.429	5.714
C3	4.857	5.143	5.143	4.143	5.571	5.000	4.143
Bayes	2.881	4.143	3.143	3.429	2.000	2.000	2.571
RKI	4.024	4.142	3.571	3.857	4.714	3.571	4.286
standard fusion	4.500	3.143	4.857	5.286	3.143	5.429	5.143
p-value fusion	**1.381**	**1.143**	**1.143**	**1.857**	**1.286**	**1.429**	**1.429**

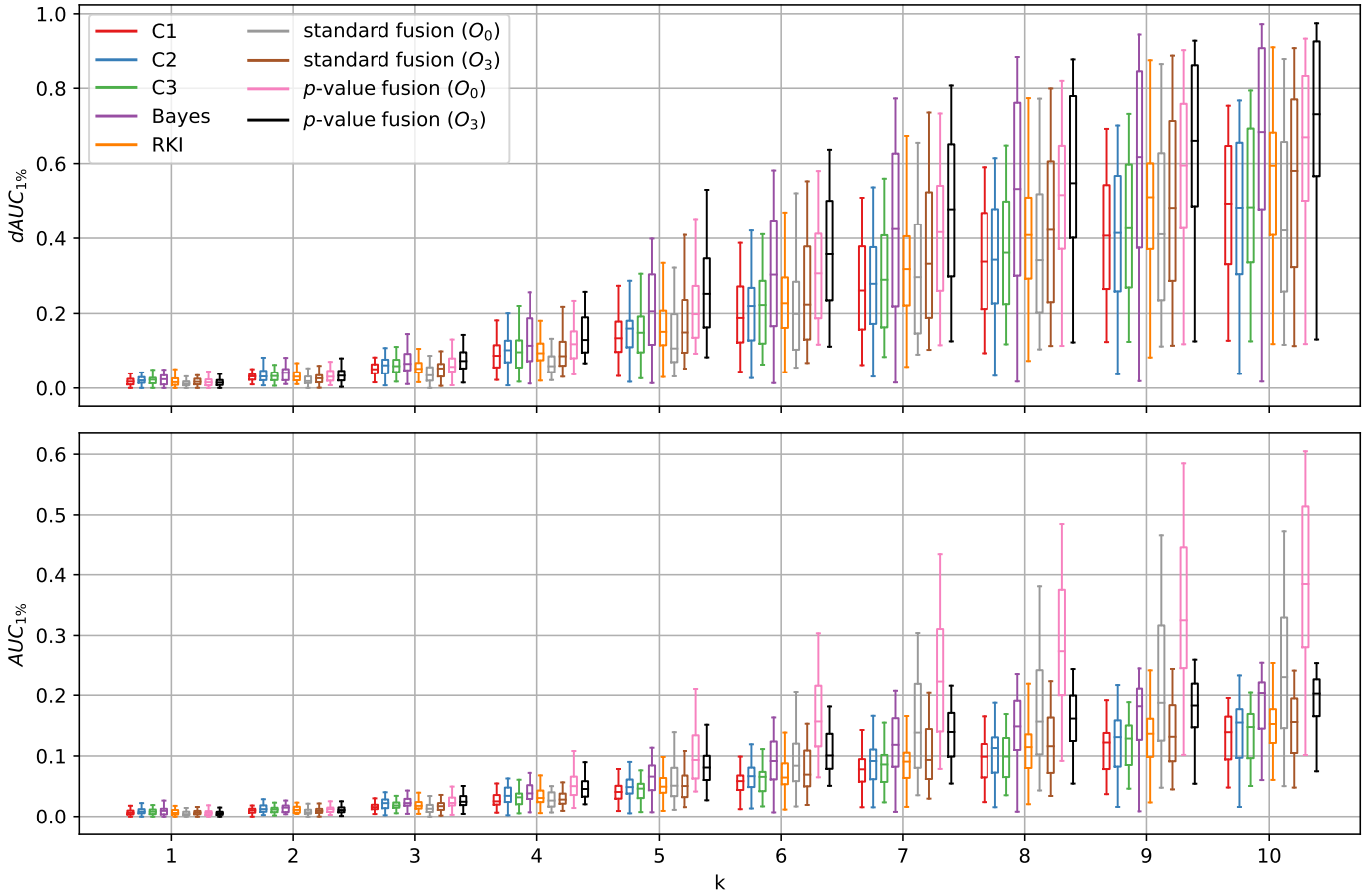


Figure 6.4: Results for the measures $dAUC_1\%$ and $AUC_1\%$. Each box plot represents the distribution of measure values for a particular method computed over all 42 test cases for a fixed outbreak size defined by the parameter k (a bigger value for k indicate more cases per outbreak).

this variant is that the learner can actually focus on the identification of strong and sudden peaks which is indeed the main goal of outbreak detection. However, in case of biannual seasonality the frequent change of the counts over the season results in many random peaks which apparently makes it difficult for the stacking approach to distinguish between an epidemic peak and a peak caused by random effects. On the test cases without trend ($\{\neg T, S1, S2\}$) outbreaks are better identifiable by also including the fading of the outbreak (O_0), whereas on the test cases which contain trend ($\{T, S1, S2\}$) the best option seems to be O_2, which includes only the increasing counts until the peak of the outbreak is reached (O_2).

Evaluation with respect to the outbreak size. Furthermore, we have evaluated the approaches with respect to the number of cases per outbreak. In contrast to the previous experiments, where the value for the parameter k (used to define the number of cases per outbreak) was randomly drawn between 1 and 10, we have fixed this parameter to a particular value for all time series of the 42 test cases. The results for the measure $dAUC_{1\%}$ across the 42 test cases with a fixed value for the parameter k is visualized as box plots, representing minimum, first quantile, mean, third quantile and maximum, in Figure 6.4. In addition to $dAUC_{1\%}$, we include the analysis of the $AUC_{1\%}$ measure and compare to the original labeling O_0 in order to further investigate the effect of the labeling on detection rate and true positive rate.

As the cases per outbreak increases all methods are more likely to obtain a better performance. While the C1, C2, C3 and RKI method achieve comparable results across all outbreak sizes, we are surprised to observe that the Bayes method has a better performance in case of larger outbreaks. This contradicts our expectation that the RKI method should obtain the best results across these methods since the Poisson assumption was specifically used to generate the synthetic data. Regarding the p-value fusion approaches, the results confirm the better overall performance across all outbreak sizes while the performance of the standard fusion approach gets worse compared to the other methods with an increasing number of cases per outbreak. This gives further evidence that the standard fusion is not ideal. A closer examination of the graphs for the measures $dAUC_{1\%}$ and $AUC_{1\%}$ reveals the difference between the adaption of the labeling for the learning. In particular, without adaption the machine learning algorithm achieves a tremendously better performance for the trade-off between the true positive rate and the false alarm rate. However, this also has an effect on the ability to detect outbreaks as discussed in Section 6.3.3, yielding a slightly worse result for the measure $dAUC_{1\%}$ than with adapting the labeling.

Comparison to the statistical surveillance baselines. Considering the results of the parameter optimization, we evaluated both fusion approaches with the adaption of the labeling O_3 and including the mean. For the p-value fusion a window size of two and for the standard fusion a window size of four is used. The results in Table 6.4

clearly show that p-value fusion performs best across all test cases, with average ranks close to 1. In line with Texier et al. (2019) and Jafarpour et al. (2013), the results show an improvement of the standard fusion approach on the time series without trend and seasonality. However, this improvement is not consistent for all compared test cases, resulting only in an average rank of 3.143. On the other test cases, the fusion of binary outputs obtains often a low rank, making it often worse than the underlying statistical algorithms. Indeed, the ability to detect outbreaks with the standard fusion approach is reduced since it is based on the output of the statistical algorithms given a particular pre-defined significance level α for them. This limits the information about sudden changes encapsulated in the training data which makes it pretty difficult for the machine learning algorithm to identify valuable patterns.

6.5.3 Results on Real Data

Comparison to the statistical surveillance baselines. Using the optimal parameters found by the parameter optimization, we obtain the results which are shown in Table 6.5. Only considering the results of the statistical methods, we can observe that the *C2* method achieves the best results for *SAL* and the worst results for *CAM*. This shows the diversity of the data but also highlights the difficulty in choosing a suitable statistical algorithm for a particular disease. Regarding our fusion approaches, we are able to achieve better results compared to the underlying statistical algorithms for the configurations *all* and *min=2*. However, the results for the configuration *min=3* are indicating a somehow unexpected behavior: While the statistical methods can double the $dAUC_{1\%}$ score compared to configuration *min=2*, the stacking approaches are unable to achieve such an improvement. Therefore, we further investigated the results to understand these effects.

Table 6.5: Results for the measure $dAUC_{1\%}$ evaluated on real data using optimized parameter configurations.

approach	$\mathbf{SAL_{all}}$	$\mathbf{SAL_{min=2}}$	$\mathbf{SAL_{min=3}}$	$\mathbf{CAM_{all}}$	$\mathbf{CAM_{min=2}}$	$\mathbf{CAM_{min=3}}$
C1	0.1545	0.1763	0.3392	0.0473	0.0478	0.0799
C2	0.1599	**0.1867**	**0.3692**	0.0389	0.0383	0.0780
C3	0.1444	0.1668	0.3294	0.0442	0.0441	0.0894
Bayes	0.1212	0.1314	0.2811	0.0398	0.0428	0.0921
RKI	0.1495	0.1730	0.3303	0.0506	0.0526	**0.1159**
standard fusion	0.1591	0.1869	0.3300	0.1459	0.1457	0.0892
p-value fusion	**0.1732**	**0.1901**	0.3341	**0.1589**	**0.1516**	**0.1300**

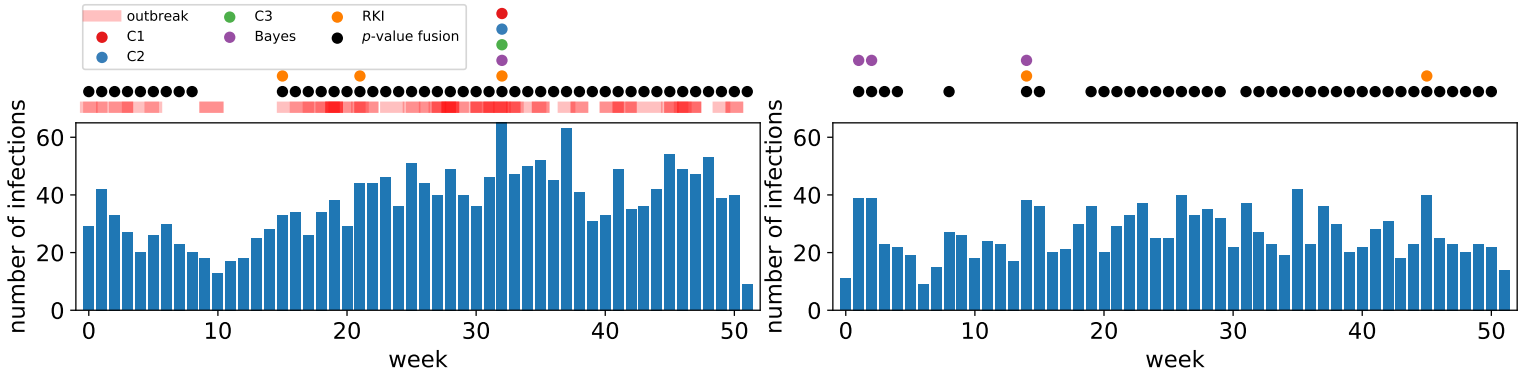
Inconsistencies. From a machine learning point of view, an important reason for low performance of learning algorithms are inconsistencies in the data. Roughly speaking, inconsistent data points can not or only hardly be discriminated by the learning

Table 6.6: Results for the measure $dAUC_{1\%}$ evaluated on the training data.

approach	$\mathbf{SAL_{all}}$	$\mathbf{SAL_{min=2}}$	$\mathbf{SAL_{min=3}}$	$\mathbf{CAM_{all}}$	$\mathbf{CAM_{min=2}}$	$\mathbf{CAM_{min=3}}$
standard fusion	0.1982	0.2013	0.2693	0.1369	0.1404	0.1808
p-value fusion	0.2745	0.3006	0.5668	0.7208	0.7359	0.7548

Table 6.7: Results of the p-value fusion approach for the measure $dAUC_{1\%}$ evaluated on the test data by including or excluding the location.

approach	location	$\mathbf{SAL_{all}}$	$\mathbf{SAL_{min=2}}$	$\mathbf{SAL_{min=3}}$	$\mathbf{CAM_{all}}$	$\mathbf{CAM_{min=2}}$	$\mathbf{CAM_{min=3}}$
p-value fusion	no	0.1732	0.1901	0.3341	0.1589	0.1516	0.1300
p-value fusion	yes	0.1867	0.1968	0.3369	0.1689	0.1694	0.1501



a) District A. **b)** District B.

Figure 6.5: Exemplary results for two districts associated to two major cities for $\mathrm{CAM_{all}}$. The red lines above the plot indicate ongoing outbreaks at the respective time steps.

algorithm, but are associated with different outcomes (e.g., *outbreak* yes/no). Our hypothesis is that such inconsistencies are the main cause for the observed results. They can be identified by analyzing how well the model can adapt to the observed data. Table 6.6 shows the training set performance of the random forest algorithm, which is generally capable of memorizing observed data well Wyner et al. (2017). The low values for SAL indicate that the dataset contains a high ratio of inconsistencies from the perspective of the learner. Especially small outbreaks are difficult to differentiate from ordinary cases.

Heterogeneous data sources. As already mentioned, in addition to undiagnosed patients, our dataset may contain cases which were erroneously not attributed to an epidemic outbreak, or even undetected outbreaks. Furthermore, the unequal distribution of outbreaks (cf. Figure 6.3) indicates that there may be some heterogeneity in the district's policies for reporting of cases and labeling of outbreaks. To support this assumption, Figure 6.5 shows the results for CAM_{all} for two districts. The alarms which would have been triggered if the false alarm rate on all evaluated districts was set to 1%, respectively for each approach separately, are visualized on top of the graphs. Furthermore, the red lines indicate the outbreaks, which are as we see sometimes overlapping. The stacking approaches achieve excellent results for district A while for district B the predictions do not fit at all. Since a global model is learned, the machine learning algorithm is not able to differentiate between the districts. Therefore, patterns are learned which work best across all districts, even though the predictions for particular districts are incorrect. This assumption is somehow confirmed by our experiments which include features which allow to differentiate between locations. As it can be seen from Table 6.7, the detection quality benefits from including this additional information.

Coarse grained observations. Apart from heterogeneous data, we also face problems due to the aggregation level of the data. In particular for the high populated districts, the reports of multiple different health offices are merged together which makes the identification of small local outbreaks difficult due to the high endemic load. Furthermore, the aggregation over a high population also raises the probability of having many small outbreaks which causes to be in an outbreak for almost all time steps as it can be seen for district A in Figure 6.5a. Especially during the learning process, the constant labeling makes it difficult for the machine learning approaches to identify consistent patterns for outbreaks, since arbitrary increases, as well as decreases of the number infections, are labeled as positive. Our analysis reveals that the p-value approach learns the pattern that there is a high probability of being in an outbreak if the number of total cases is high, which is typically the case for the data of district A, but not for district B.

6.6 Conclusions

In this chapter, we introduced an approach for the fusion of outbreak detection methods using machine learning, more specifically stacking. The original idea is to use the *alarm* or *no alarm* prediction of the underlying statistical algorithms as inputs to the learner. We improved that setup by incorporating the p-values instead, which contain more information about the certainty of an event than the simple binary outputs. In addition, we proposed to incorporate additional information in the learning data and to adapt the labeling of an outbreak in order to improve the ability to detect outbreaks. For evaluation, we proposed a measure based on ROC curves which better adapts to the specific need for a very low false alarm rate but still considers the trade-off with the detection rate.

Our experimental results on synthetic data show that the fusion of p-values improves the performance compared to the underlying statistical algorithms. Contrary to previous work, we could also observe that simple fusion of binary outputs using stacking does not always lead to an improvement. By incorporating additional information to the learning data, more specifically the mean count of the previous observations and the previous outputs of the statistical methods, the machine learning algorithm is able to capture more reliable patterns to detect outbreaks. Furthermore, the labeling of an outbreak has an influence on the performance for the classification algorithm to detect outbreaks. By setting the focus on the peak of an outbreak during the learning process, a better performance to detect sudden changes can be achieved. However, on real data, we face several issues regarding inconsistencies, heterogeneous labeling and the aggregation level of the data which all need to be considered when learning a fusion classifier.

For future work, it remains to analyze more deeply how the treatment of the outbreak annotations during training can be selected in order to optimize the detection of outbreaks. Moreover, stacking allows enriching the detection by additional signals and sources of information in a highly flexible way, such as local weather data or data from neighbor districts, which can be studied as well.

7

Non-Specific Syndromic Surveillance

We have seen in Chapter 6 how supervised learning can improve outbreak detection. However, the results we obtained on the real data are less clear due to inconsistent and heterogeneous labeling of outbreaks. Especially this can have an impact on the learnability of the model which in turn lead to unexpected results, reducing the acceptance and trustworthiness in machine learning approaches among epidemiologists.

In addition, particularly for rare infectious diseases insufficient amounts of labeled data are available to adequately learn a supervised classifier. Based on this knowledge, we turn our attention towards unsupervised machine learning techniques (cf. Section 3.1 and 3.5) which do not depend on labeled data. In particular for syndromic surveillance (cf. Section 4.2), these kind of algorithms offer a variety of opportunities to improve outbreak detection.

Rather than developing highly specialized algorithms which are based on a specific disease and assume particular characteristics of outbreak shapes (Shmueli and Burkom, 2010), we argue that the task of outbreak detection should be viewed as a general anomaly detection problem where an outbreak alarm is triggered if the distribution of the incoming data changes in an unforeseen and unexpected way. Therefore, we distinguish between *specific* syndromic surveillance, where factors related to a specific disease are monitored, and *non-specific* syndromic surveillance, where general, universal characteristics of the stream of data are monitored for anomalies.

While specific syndromic surveillance is a well-studied research area, we found that only little research has been devoted to non-specific syndromic surveillance with only very few algorithms available. In particular, the close relation to anomaly detection (cf. Section 3.6) motivated us to investigate the problem of non-specific syndromic surveillance from a machine learning perspective and to make the task more approachable for the anomaly detection community.

In this chapter, we revisit algorithms for non-specific syndromic surveillance and compare them to a broad range of anomaly detection algorithms. In addition, we propose a general framework for non-specific syndromic surveillance in which the approaches can be integrated. Due to little effort on implementing baselines in previous works

Table 7.1: Notation.

Notation	Meaning
$\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$	response attributes
$\mathcal{E} = \{E_1, E_2, \ldots, E_k\}$	environmental attributes
$\mathcal{C} \in A_1 \times A_2 \times \ldots \times A_m$	population of instances
$\mathbf{c} \in \mathcal{C}$	a single instance
t	index for the time slot
$\mathcal{C}(t) \subset \mathcal{C}$	cases of time slot t
$\mathbf{e}(t) \in E_1 \times E_2 \times \ldots \times E_k$	environmental setting for time slot t
$(\mathcal{C}(t), \mathbf{e}(t))$	information about time slot t
$\mathcal{H} = ((\mathcal{C}(1), \mathbf{e}(1)), \ldots, (\mathcal{C}(t-1), \mathbf{e}(t-1)))$	information about previous time slots
$G(\mathbf{e}(t), \mathcal{H}) = \hat{\mathcal{C}}(t)$	global model
$\hat{\mathcal{C}}(t)$	expectation for $\mathcal{C}(t)$
$\mathcal{X} = \{X_1, X_2, \ldots, X_r\}$	set of patterns
$\hat{\mathcal{X}} = \{\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_r\}$	set of expectations for the patterns
$\mathcal{L}(\mathcal{X}, \mathcal{H}) = \{L_{X_1}, L_{X_2}, \ldots, L_{X_r}\}$	local model creator
$L_X(\mathbf{e}(t))$	a local model monitoring pattern X
$\mathcal{S}_{all}$	set of all possible syndromes
$s \in \mathcal{S}_{all}$	a particular syndrome
$\mathcal{S}_{\leq n} = \{s \mid s \in \mathcal{S}_{all} \wedge \lvert s \rvert \leq n\}$	set of syndromes with max. n conditions
$s(t)$	count of syndrome s for time slot t
$\mathcal{H}_s(t) = (s(1), s(2), \ldots, s(t-1))$	time series of counts for syndrome s
$\mathcal{R} \subset \mathcal{C}$	reference set of instances

on non-specific syndromic surveillance, we propose a set of baselines relying on simple statistical assumptions which nonetheless have been widely used before in disease surveillance.

We experimentally compare the methods on an established synthetic dataset (Fanaee-T and Gama, 2015; Wong et al., 2005) and real data from a German emergency department in which we injected synthetic outbreaks. Our results demonstrate that the simple statistical approaches, which have not been considered in previous works, are quite effective and often can outperform more elaborate machine learning algorithms.

7.1 Framework for Non-Specific Syndromic Surveillance

In this section, we formulate the problem of non-specific syndromic surveillance from the perspective of machine learning and propose two modeling strategies which are presented in an unified framework. The used notation is summarized in Table 7.1.

7.1.1 Problem Definition

Syndromic data can be seen as a constant stream of instances of a population $\mathcal{C}$. Each instance $\mathbf{c} \in \mathcal{C}$ is represented by a set of attributes $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ where each attribute can be either categorical (e.g., *gender*), continuous (e.g., *age*) or text (e.g., *chief complaint*). Following the notation of Wong et al. (2005), we refer to these attributes as *response attributes*. To be able to detect changes over time, instances are grouped together according to pre-specified time slots (e.g., all patients arriving at the emergency department in one day). Hence, the instances for a specific time slot t are denoted as $\mathcal{C}(t) \subseteq \mathcal{C}$.

In addition, each group $\mathcal{C}(t)$ is associated with an environmental setting $\mathbf{e}(t) \in E_1 \times E_2 \times \ldots \times E_k$ where $\mathcal{E} = \{E_1, E_2, \ldots, E_k\}$ is a set of *environmental attributes*. Environmental attributes are independent of the response attributes and represent external factors which might have an influence on the distribution of instances $\mathcal{C}(t)$ (e.g., during the winter flu-like symptoms are more frequent). In particular, a specific characteristic of syndromic data is *seasonality*, in machine learning also known as *cyclic drift* (Webb et al., 2016). Environmental variables can help the algorithm to adapt to this kind of concept drift. Thus, the information available for time slot t can be represented by the tuple $(\mathcal{C}(t), \mathbf{e}(t))$ and the information about prior time slots can be denoted as $\mathcal{H}(t) = ((\mathcal{C}(1), \mathbf{e}(1)), \ldots, (\mathcal{C}(t-1), \mathbf{e}(t-1)))$.

The main goal of non-specific syndromic surveillance is to detect anomalies in the set $\mathcal{C}(t)$ of the current time slot t w.r.t. the previous time slots $\mathcal{H}(t)$ as potential indicators of an infectious disease outbreak. Therefore, the history $\mathcal{H}(t)$ is used to fit a model $f_{\mathcal{H}(t)}(\mathbf{e}(t), \mathcal{C}(t))$ which is able to generate a score for time slot t, representing the likelihood of being in an outbreak.

Viewed from the perspective of specific syndromic surveillance, the non-specific setting can be seen as the monitoring of all possible syndromes at the same time. The set of all possible syndromes can be defined as

$$\mathcal{S}_{all} = \left\{ \prod_{i \in \mathcal{I}} A_i \mid A_i \in \mathcal{A} \wedge \mathcal{I} \subseteq \{1, 2, \ldots, m\} \wedge |\mathcal{I}| \geq 1 \right\}$$

where $\prod_{i \in \mathcal{I}} A_i$ for $|\mathcal{I}| = 1$ is defined as $\{\{a\} \mid a \in A \wedge A \in \mathcal{A}\}$. In addition, we denote $\mathcal{S}_{\leq n} = \{s \mid s \in \mathcal{S}_{all} \wedge |s| \leq n\}$ as the set of all possible syndromes having a maximum of n conditions and $\mathcal{H}_s(t) = (s(1), s(2), \ldots, s(t-1))$ as the time series of counts for a particular syndrome $s \in \mathcal{S}_{all}$.

7.1.2 Modeling

The general approach to non-specific syndromic surveillance is to model the normal activity by analyzing $\mathcal{H}(t)$ and compare it to the set $\mathcal{C}(t)$. A significant difference between the expectation to the actual observed set $\mathcal{C}(t)$ can indicate an outbreak of an infectious disease. Especially an increase in the number of instances following a particular syndrome can be a good indicator for an outbreak, while under normal circumstances the absence is not of interest. The difference between the expectation and the current observed subset $\mathcal{C}(t)$ can be modeled in two ways, namely via *global* and *local* modeling.

While global modeling tries to solve the problem of outbreak detection with a single universal model, a local modeling approach breaks down the problem into many local tasks, each representing an expectation for a particular characteristic of $\mathcal{C}(t)$. For example, an expectation could be the count for a specific syndrome which is then compared to the actual count of the syndrome in $\mathcal{C}(t)$. The local tasks are executed independently and their results need to be aggregated afterwards, in contrast to the global modeling where the outcome is already a single result.

7.1.2.1 Global Modeling

The basic idea of global modeling is visualized in Figure 7.1. Given the information of prior time slots $\mathcal{H}(t)$, which serve as training data, and the information about the environmental attributes of the current time slot $\mathbf{e}(t)$, the learning objective of the model is to create an expectation for the distribution of cases $G(\mathbf{e}(t), \mathcal{H}(t)) = \hat{\mathcal{C}}(t)$. In the following step, the distribution $\hat{\mathcal{C}}(t)$ is compared to the actual observation of cases $\mathcal{C}(t)$. Depending on the used algorithm, the representation of $\mathcal{C}(t)$ and $\hat{\mathcal{C}}(t)$ can have arbitrary forms. For example, the information about all cases for a particular time slot can be encapsulated as one vector, as it is done by Fanaee-T and Gama (2015).

Depending on the representation of $\hat{\mathcal{C}}(t)$, statistical tests such as the normality (Fanaee-T and Gama, 2015) or the Fisher's test (Brossette et al., 1998; Wong et al., 2005) are typically used for assessing the difference between $\mathcal{C}(t)$ and $\hat{\mathcal{C}}(t)$. However, instead of making a binary final decision, it is much more preferable to directly use the p-value (Amrhein et al., 2019). The complement of the p-value can be seen as the likelihood of being in an outbreak and, therefore, contain much more information about the belief of being in an outbreak than the binary decision. This allows us to analyze the performance of the model in the evaluation more precisely and, moreover, we are able to defer the specification of the significance level during the evaluation. Depending on the results, an appropriate significance level can be selected for applying the model in practice.

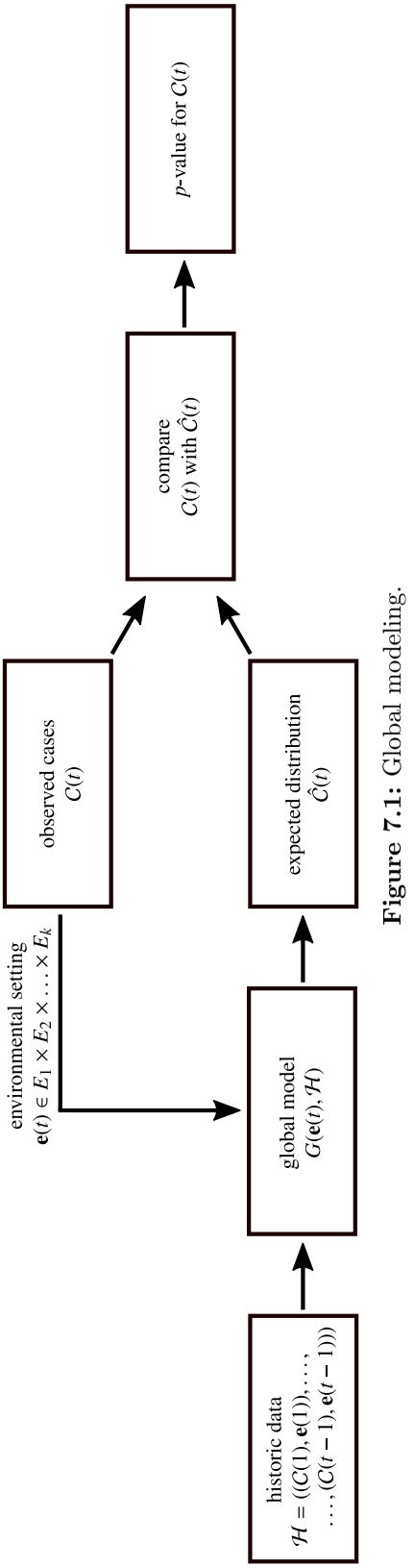


Figure 7.1: Global modeling.

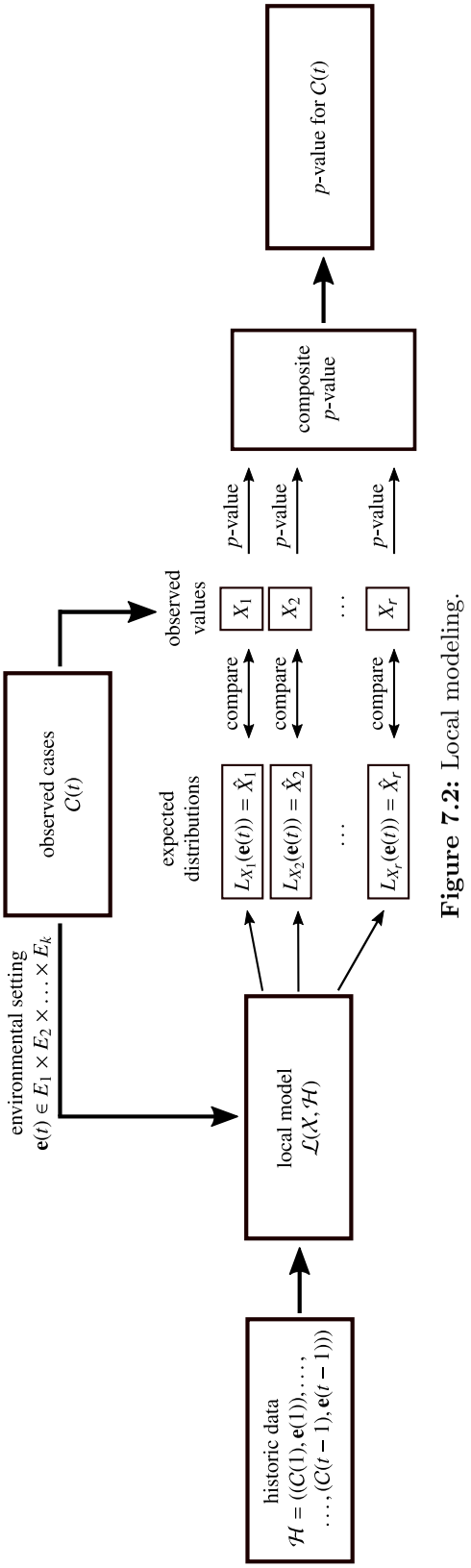


Figure 7.2: Local modeling.

7.1.2.2 Local Modeling

The major drawback of global modeling is that all the information about the cases $\mathcal{C}$ is summarized in one representation. Thus, information about individual cases, which could be a good indicator for an outbreak, might be lost. This is particularly important for detecting outbreaks of very rare diseases, for which a relative small increase of cases can be already alerting. Therefore, local modeling breaks down the problem of non-specific syndromic surveillance into several local modeling tasks, each focusing on a different characteristic of the data (cf. Figure 7.2). The general idea of composing a global model from many local models has previously been proposed by Knobbe et al. (2008).

Given some information of prior days $\mathcal{H}(t)$, local modeling generates a set of local models $\mathcal{L}(\mathcal{X}, \mathcal{H}) = \{L_{X_1}, L_{X_2}, \ldots, L_{X_r}\}$, each responsible for monitoring a specific pattern $X_i \in \mathcal{X} = \{X_1, X_2, \ldots, X_r\}$ of the data. As we will discuss in more detail in Section 7.2, most of the algorithms differ in what patterns they monitor. For example, one can use high-frequency association rules, as in DMSS (Section 7.2.1), or all possible patterns up to a certain complexity, as in WSARE (Section 7.2.2). In dependence of the environmental attributes of the current time slot $\mathbf{e}(t)$, each local model defines an expectation $L_X(\mathbf{e}(t)) = \hat{X}$ for their pattern.

A local model can monitor any kind of pattern which might be helpful to detect an outbreak of an infectious disease. Subsequently, the expectation for the patterns $\{\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_r\}$ are compared to the actual observations of the patterns $\mathcal{X}$ obtained from the data set $\mathcal{C}(t)$. Like for the global modeling, the comparison is usually performed with statistical tests which yield a p-value. As each local model yields a single p-value, the results of the local models need to be aggregated to a final p-value for the respective time slot.

Heard and Rubin-Delanchy (2018) review multiple methods for combining p-values, such as Edgington's and Fisher's methods which have already been used for syndromic surveillance (Vial et al., 2016). More of these combination methods are discussed in Section 2.3.2. Each of the combination procedures have different statistical properties, not allowing to generally prefer one particular method. In addition to these methods, we introduce as a simple baseline the naive approach that reports the smallest p-value, which basically represents the most significant observation. If a relative assessment of the findings is sufficient our proposed aggregation of taking the minimum is sufficient. Nonetheless, more sophisticated approaches, such as the exploitation of correlations, can likewise be integrated in our proposed framework. We leave the investigation for future work.

7.2 Machine Learning Algorithms

In a survey of the relevant literature we have identified only a few algorithms which relate to non-specific syndromic surveillance, described in sections 7.2.1, 7.2.2, and 7.2.3. In Section 7.2.4 we introduce a way how common anomaly detection algorithms can be applied in the setting of non-specific syndromic surveillance.

7.2.1 Data Mining Surveillance System (DMSS)

One of the first algorithms able to identify new and interesting patterns in syndromic data was proposed by Brossette et al. (1998) who adopted the idea of association rule mining (Zhang and Zhang, 2002) to the field of public health surveillance. The algorithm follows the local modeling approach where each local model L_X is represented by an association rule X. In order to detect an outbreak for time slot t, an association rule mining algorithm needs to be run on $\mathcal{C}(t)$ to obtain a set of association rules $\mathcal{X}$. Moreover, a reference set of patients $\mathcal{R} \subset \mathcal{C}$ is created by merging the instances of a selected set of previous time slots. For each association rule $X \in \mathcal{X}$ the confidence of the rule on $\mathcal{C}(t)$ is compared to the confidence of the rule computed on $\mathcal{R}$ using a χ^2 or a Fisher's test (cf. Section 2.3.1). If the confidence has significantly increased on $\mathcal{C}(t)$, the finding is reported as an unexpected event. In order to reduce the complexity, the authors propose to focus only on mining high-support association rules. An aggregation of the observations for one time slot is not performed and environmental attributes are not considered by this approach.

7.2.2 What is strange about recent events? (WSARE)

The family of *WSARE* algorithms has been proposed by Wong et al. (2005). All algorithms share the same underlying concept, namely to monitor all possible syndromes having a maximum of two conditions $\mathcal{S}_{\leq 2}$ simultaneously. Again, these algorithms can be categorized as local modeling in which each local model L_X is responsible for monitoring one particular syndrome $s \in \mathcal{S}_{\leq 2}$, hence $X = s$ and $\mathcal{X} = \mathcal{S}_{\leq 2}$. The basic idea is to create a reference set of cases $\mathcal{R} \subseteq \mathcal{C}$ on which the expected proportion for each syndrome is estimated. For each local model L_X, the expected proportion $\hat{X}$ of the syndrome X is compared to the proportion of the syndromes observed on the set $\mathcal{C}(t)$ using the χ^2 or Fisher's exact test (cf. Section 2.3.1). In order to aggregate the p-values of the statistical tests for one time slot, a *permutation test* with $1,000$ repetitions is performed. As for DMSS, the authors of the WSARE algorithm focused in their evaluation on patient data using single-day time slots.

The three WSARE algorithms only differ in the way how the reference set $\mathcal{R}$ is created. The following three versions have been considered:

WSARE 2.0 merges the instances of a selected set of prior time slots together for the reference set $\mathcal{R}$. Since their evaluation was based on single-day time slots, they combined the instances of the previous time slots 35, 42, 49 and 56 to consider only instances of the same weekday.

WSARE 2.5 merges the instances of all prior time slots together which share the same environmental setting as for the current day $\mathbf{e}(t)$. This has the advantage that the expected proportions are conditioned on the environmental setting $\mathbf{e}(t)$ and that potentially more instances are contained in the reference set $\mathcal{R}$, allowing to have more precise expectations.

WSARE 3.0 learns a Bayesian network over all recent data $\mathcal{H}(t)$ from which 10,000 instances for the reference set $\mathcal{R}$ are sampled given the environmental attributes $\mathbf{e}(t)$ as evidence. The authors make use of domain knowledge for the structure learning of the Bayesian network by not allowing parent nodes for nodes of environmental attributes. This can be done because environmental attributes only serve as evidence for the sampling. The prediction of their distribution is not of interest.

7.2.3 Eigenevent

The *Eigenevent* algorithm proposed by Fanaee-T and Gama (2015) can be categorized as a global modeling approach. Its key idea is to track changes in the data correlation structure using eigenspace techniques. Instead of monitoring all possible syndromes, only overall changes and dimension-level changes are observed by the algorithm. The global approach makes the Eigenevent algorithm less susceptible to noise resulting in a lower false alarm rate. However, this also reduces the sensitivity of detecting outbreaks which might be caused by only a few cases for a very rare disease.

In order to detect outbreaks, a dynamic baseline tensor is created using the information of prior time slots $\mathcal{H}(t)$ which share the same environmental setting $\mathbf{e}(t)$. In the case that not enough prior time slots are available, time slots with the most frequent value combinations for the environmental attributes will be added. The conducted experiments showed that this mixing improves the detection power of the algorithm for unseen value combinations of environmental attributes. In the next step, information of the instances $\mathcal{C}(t)$ and the baseline tensor are decomposed to a lower-rank subspace in which the eigenvectors and eigenvalues are compared to each other, respectively. Any significant changes in the eigenvectors and eigenvalues between the baseline tensor and the information of instances $\mathcal{C}(t)$ indicate an outbreak.

7.2.4 Anomaly Detection Algorithms

A direct application of point anomaly detection (cf. Section 3.6) is in general not suitable for syndromic surveillance because these methods aim to identify single instances $\mathbf{c} \in \mathcal{C}$ as outliers and could thus, e.g., be triggered by a patient who is over a hundred years old (Wong et al., 2005). In order to still apply point anomaly detectors to discover outbreaks, we form a dataset $\mathcal{D}$ using the syndromes $s \in \mathcal{S}_{\leq n}$ as features and the respective syndrome counts $\mathcal{H}_s(t)$ as values. Hence, each instance represents the occurrence counts of all syndromes for one particular time slot and the dataset contains $t-1$ instances in total. This dataset can be used to fit an anomaly detector which can be then applied to the instance of syndrome counts for time slot t. Hence, an outbreak could be identified by an unusual combination of syndrome counts. The basic concept of this approach can be described as global modeling since the anomaly detector aims to capture the normal behaviour in one model.

In this chapter, we consider the following anomaly detection algorithms. We refer to the respective references for a comprehensive review of these methods.

One-Class SVM (OSVM) extends the support vector machine algorithm to perform outlier detection by separating instances $\mathcal{D}$ from the complement of $\mathcal{D}$ (Schölkopf et al., 2001).

Local Outlier Factor (LOF) computes the outlier score for an instance based on how isolated the instance is with respect to the surrounding neighborhood (Breunig et al., 2000).

Gaussian Mixture Models (GMM) approximate the distribution of the dataset $\mathcal{D}$ using a mixture of Gaussian distributions. The outlier score is based on how dense the region of the evaluated instance is (Reynolds, 2009).

Copula-Based Outlier Detection (COPOD) creates an empirical copula for the multi-variate distribution of $\mathcal{D}$ on which tail probabilities for an instance can be predicted to estimate the outlier score (Li et al., 2020).

Isolation Forest constructs an ensemble of randomly generated decision trees in which anomalies can be identified by counting the number of splittings required to isolate an instance (Liu et al., 2008).

Autoencoder learns an identity function of the data through a network of multiple hidden layers. Instances which have a high reconstruction error are considered to be anomalous (Zhou and Paffenroth, 2017).

Multiple-Objective Generative Adversarial Active Learning (GAAL) constructs multiple generators having different objectives to generate outliers for learning a discriminator which can assign outlier scores to new instances (Liu et al., 2020).

7.3 Basic Statistical Approaches

In addition to the machine learning models introduced in Section 7.2, we also include statistical techniques, which are commonly used for specific syndromic surveillance (cf. Section 4.4), into our comparison and adapt them to a non-specific syndromic surveillance setting. The key idea of these adaptations is to monitor all possible syndromes $\mathcal{S}_{all}$ simultaneously based on the local modeling approach.

Similar to the WSARE algorithms, each local model L_X is responsible for monitoring one particular syndrome $s \in \mathcal{S}_{\leq n}$ where $X = s$ and $\mathcal{X} = \mathcal{S}_{\leq n}$. Therefore, a parametric distribution $P_s(x)$ is fitted in each local model for the respective syndrome $s \in \mathcal{S}$ using the empirical mean μ and the empirical variance σ^2 computed over $\mathcal{H}_s(t)$:

$$\mu = \frac{1}{|\mathcal{H}_s(t)|} \sum_{i=0}^{|\mathcal{H}_s(t)|} s(i) \qquad \sigma^2 = \frac{1}{|\mathcal{H}_s(t)| - 1} \sum_{i=0}^{|\mathcal{H}_s(t)|} (s(i) - \mu)^2$$

On the fitted distribution $P_s(x)$, a one-tailed significance test is performed in order to identify a suspicious increase of cases. For a particular observed count $s(t)$, the p-value is computed as the probability $\int_{s(t)}^{\infty} P_s(x)dx$ of observing $s(t)$ or higher counts. Thus, for evaluating a single time slot t, we obtain $|\mathcal{S}|$ p-values which need to be aggregated under consideration of the multiple-testing problem. Following Roure et al. (2007), we only report the minimum p-value for each time slot t because the Bonferroni correction can be regarded as a form of aggregation of p-values based on the minimum function. In particular, note that scale-free anomaly scores are sufficient for the purpose of identifying the most suspicious time slots. The complement of the selected p-value represents the anomaly score reported for time slot t. For our baselines we have considered the Poisson distribution, the Gaussian distribution and the negative binomial distribution which we have reviewed in detail in Section 2.2.

Modifications for adapting the sensitivity. Modeling count data with a statistical distribution is often challenging because of the different forms of count data and distributional assumptions (cf. Section 2.2). Especially for our application scenario, in which we perform multiple statistical tests in parallel, a fitted distribution which is overly sensitive to changes can cause many false alarms. In fact, if the number of monitored syndromes is much higher than the average number of cases observed each time slot, most of the syndromes are rare. Statistical tests performed on these syndromes report a very low p-value if only one case is observed in $\mathcal{C}(t)$. This problem becomes more frequent with an increasing number of rare syndromes which are monitored simultaneously, which results in reporting many unusual observations throughout the time slots.

Since outbreaks are usually associated with a high number of infections, we propose the following modifications for the baselines in order to reduce the sensitivity of statistical tests on rare syndromes. For the Gaussian distribution, we propose to use a minimal value for the standard deviation to which we refer to as σ_{min}. Moreover, for the Poisson distribution, we use a minimal value for the lambda parameter λ_{min}. The negative binomial distribution is similarly lead by the mean number of cases. Hence, we assume a minimal mean μ_{min} for the negative binomial distribution before setting the parameters as indicated. We leave the standard deviation untouched for the negative binomial distribution since manipulating the overdispersion can lead to extreme distortions in the estimation.

7.4 Experiments and Results

The goal of the experimental evaluation reported in this section is to provide an overview of the performance of non-specific syndromic surveillance methods in general, and in particular, to re-evaluate the established methods in context of the proposed base statistical approaches and the anomaly detection algorithms. We conducted experiments on synthetic data, which already have been used for the evaluation of the algorithms Eigenevent and WSARE (Fanaee-T and Gama, 2015; Wong et al., 2005), and on real data of a German emergency department (cf. Section 7.4.3). As the emergency department data do not contain any information about real outbreaks, we decided to inject synthetic outbreaks which is common practice in the area of syndromic surveillance, allowing us to evaluate and compare the algorithms in a controlled environment (cf. Section 4.5).

7.4.1 Evaluation Setup

The evaluation of syndromic surveillance methods is usually performed on a set of data streams, to which we will refer as an *evaluation set*, since a single data stream does normally not contain enough outbreaks to draw conclusions about the performance of the evaluated algorithms.

To evaluate a data stream it is split into two parts, namely a *training part*, containing the first time slots which are only used for training, and a *test part*, which contains the remaining time slots of the data stream. The evaluation is performed on the test part incrementally which means that for evaluating each time slot t the model will be newly fitted on the complete set of previously observed data points $\mathcal{H}(t) = ((\mathcal{C}(1), \mathbf{e}(1)), \ldots, (\mathcal{C}(t-1), \mathbf{e}(t-1)))$. Alarms raised during an outbreak are considered as true positives while all other raised alarms are considered as false positives.

Table 7.2: Information about the attributes of the synthetic data.

attribute	type	#values
age	response	3
gender	response	2
action	response	3
symptom	response	4
drug	response	4
location	response	9
flu level	environmental	4
day of week	environmental	3
weather	environmental	2
season	environmental	4

Table 7.3: Information about the attributes of the real data.

attribute	type	#values
age	response	3
gender	response	2
mts	response	28
fever	response	2
pulse	response	3
respiration	response	3
oxygen saturation	response	2
blood pressure	response	2
day of week	environmental	7
season	environmental	4

Synthetic data. The synthetic data consists of 100 data streams, generated with the synthetic data generator proposed by Wong et al. (2005). The data generator is based on a Bayesian network and simulates a population of people living in a city of whom only a subset are reported to the data stream at each simulated time slot. Detailed information about the attributes in the data stream is given in Table 7.2. Each data stream captures the information about the people on a daily basis over a time period of two years, i.e., each time slot $\mathcal{C}(t)$ contains the patients of one day. In average 34 instances are reported per time slot and 270 possible syndromes are contained in the set $\mathcal{S}_{\leq 2}$. The first year is used for the training part while the second year serves as the test part. Exactly one outbreak is simulated in the test part which starts at a randomly chosen day and always lasts for 14 days. During the outbreak period, the simulated people have a higher chance of catching a particular disease.

Real data. We rely on routinely collected, fully anonymized patient data of a German emergency department, captured on a daily basis over a time period of two years. We have extracted a set of response attributes and added two environmental attributes (cf. Table 7.3). Continuous attributes, such as *respiration*, have been discretized with the help of a physician into meaningful categories. In addition, we include the Manchester-Triage-System (MTS) (Gräff et al., 2014) initial assessment which is filled out for every patient on arrival. To reduce the number of values for the attribute MTS, we group classifications which do not relate to any infectious disease, such as various kinds of injuries, into a single value. In average 165 patients are reported per day and in total 574 syndromes can be formed for the set $|\mathcal{S}_{\leq 2}|$. In preparation for the injection of simulated outbreaks, we replicated the data stream 100 times. For each data stream, we used the first year as the training part and the second year as the test part in which we injected exactly one outbreak. In order to simulate an outbreak, we first uniformly sampled a syndrome from $\mathcal{S}_{\leq 2}$. In a second step, we sampled the size of the outbreak from a Poisson distribution with mean equal to the standard deviation of the daily patient visits and randomly selected the corresponding number of patients from all patients that exhibit the sampled syndrome. To avoid over-representing outbreaks on rare syndromes, only 20 data streams contain outbreaks with syndromes that have a lower frequency than one per day. In total, 29 outbreaks are based on syndromes with one condition and 71 with two.

Additional baselines. We also include the *control chart*, the *moving average* and the *linear regression* algorithms into our analysis. Compared to our *syndrome-based* statistical baselines, these *global* statistical baselines only monitor the total number of instances per time slot and therefore can only give a very broad assessment of outbreak detection performance. For a detailed explanation of these algorithms, we refer to Wong et al. (2005).

Table 7.4: Results for the $AAUC_{5\%}$ measure on the synthetic data.

name	rerun	min. p-value	permutation test	imported p-values
Eigenevent	4.993	–	–	4.391
WSARE 2.0	–	2.963	3.805	4.925
WSARE 2.5	–	1.321	1.614	1.931
WSARE 3.0	–	0.899	1.325	1.610

Implementation and parameterization. For the Eigenevent algorithm we rely on the code provided by the authors.[i] All other algorithms are implemented in Python.[ii] Parameters for the DMSS and the anomaly detection algorithms have been tuned in a grid search using 1000 iterations of *Bootstrap Bias Corrected Cross-Validation* (Tsamardinos et al., 2018) which allows to integrate hyperparameter tuning and reliable performance estimation into a single evaluation loop. The evaluated parameter combinations can be found in our repository. The WSARE, the Eigenevent, the COPOD and the statistical algorithms do not contain any parameters which need to be tuned.

Evaluation measures. For measuring the performance, we rely on the *activity monitor operating characteristic* as described in Section 4.5. Since we are interested in a very low false alarm rate, we only report the partial area under AMOC-curve for a false alarm rate less than 5%, to which we refer to as $AAUC_{5\%}$.

7.4.2 Preliminary Evaluation

In a first experiment, we replicated the experiments on the synthetic data of Fanaee-T and Gama (2015). More specifically, we imported and re-evaluated the outlier scores for the synthetic data from the Eigenevent repository (*imported p-values*) and compare these to our own results with rerunning the Eigenevent algorithm (*rerun*) and to our implementation of the WSARE algorithms. For the latter, we additionally evaluate the results of just reporting the minimal p-value for each time slot (*min. p-value*, cf. Section 7.3) instead of performing an originally proposed permutation test with 1000 repetitions (*permutation test*). The results are shown in Table 7.4. Our rerun of the Eigenevent algorithm returned slightly worse results than the imported p-values, which could be caused by the random initialization. In a personal communication, one of the authors of Eigenevent pointed out that the performance of Eigenevent depends on the random initialization of the used tensor decomposition, and suggested BetaNTF as an alternative (Fernandes et al., 2017). For the WSARE algorithms, we can observe that our implementation achieves better results than the imported p-values, probably

[i] `https://github.com/fanaee/EigenEvent`
[ii] Our code is publicly available at `https://github.com/MoritzKulessa/NSS`

due to the different Bayesian network used. In particular, the results for the minimal p-value were better than those for the more expensive permutation test. Thus, we chose to only report the minimal p-value for the WSARE algorithms in the following experiments.

7.4.3 Results

The results on the synthetic and real data are both shown in Table 7.5. For syndrome-based algorithms, the results for monitoring $\mathcal{S}_{\leq 1}$ and $\mathcal{S}_{\leq 2}$ are reported in the respective columns while results for the other methods are reported in the columns *none*. Note that the worst possible result on the synthetic data is 14 while for the real data the worst result is 1. In the first paragraphs, we will discuss the results without specifically considering the size of the syndrome sets unless needed. The effect of using $\mathcal{S}_{\leq 1}$ or $\mathcal{S}_{\leq 2}$ is discussed in the last paragraph.

Comparison between non-specific syndromic surveillance algorithms. Firstly, we analyze the results of the non-specific syndromic surveillance approaches which have been presented in Section 7.2.1 to 7.2.3. In general, the WSARE algorithms outperform the other algorithms in the group. In particular, the results of the

Table 7.5: Results for the $AAUC_{5\%}$ measure on the synthetic and real data.

category	**algorithm name**	**synthetic data**			**real data**		
		none	$\mathcal{S}_{\leq 1}$	$\mathcal{S}_{\leq 2}$	none	$\mathcal{S}_{\leq 1}$	$\mathcal{S}_{\leq 2}$
non-specific syndromic surveillance	WSARE 2.0	–	3.028	2.963	–	0.661	0.590
	WSARE 2.5	–	1.099	1.321	–	0.917	0.867
	WSARE 3.0	–	0.803	0.899	–	0.882	0.847
	DMSS	2.430	–	–	0.953	–	–
	Eigenevent	4.993	–	–	0.878	–	–
anomaly detectors	OSVM	–	1.043	1.262	–	0.468	0.495
	LOF	–	2.000	2.260	–	0.642	0.610
	GMM	–	1.117	3.547	–	0.444	0.791
	isolation forest	–	4.576	4.948	–	0.873	0.835
	COPOD	–	5.216	5.032	–	0.816	0.800
	autoencoder	–	1.521	1.643	–	0.550	0.576
	GAAL	–	7.024	6.766	–	0.792	0.866
global baselines	control chart	5.086	–	–	0.891	–	–
	moving average	7.012	–	–	0.910	–	–
	linear regression	3.279	–	–	0.819	–	–
basic statistical baselines	Gaussian	–	0.806	0.941	–	0.328	0.267
	Poisson	–	1.294	1.347	–	0.598	0.486
	negative binomial	–	0.895	0.958	–	0.299	0.216

modified versions WSARE 2.5 and WSARE 3.0 on the synthetic data show that the use of environmental attributes can be beneficial. However, the results on the real data indicate the opposite. We further investigated this finding by rerunning WSARE 3.0 on the real data without the use of environmental variables and observed a substantial improvement of the results to 0.613 for $\mathcal{S}_{\leq 1}$ and 0.570 for $\mathcal{S}_{\leq 2}$, respectively. Therefore, we conclude that the modelling of the environmental factors should be done with care since it can easily lead to worse estimates if the real distribution does not follow the categorization imposed by defined attributes.

The results of the DMSS algorithm suggest that monitoring association rules is not as effective as monitoring syndromes. In particular, the space of possible association rules is much greater than the space of possible syndromes $\mathcal{S}$ which worsens the problem of multiple testing. Especially on the real data this results in a bad performance since the high number of instances per time slot yields too many rules. Conversely, by monitoring only rules with very high support most of the outbreaks remain undetected since the disease pattern could not be captured anymore. In contrast to the results reported by Fanaee-T and Gama (2015), the Eigenevent algorithm performs poorly compared to the WSARE algorithms. A closer analysis reveals that the difference in these results can be explained by the used evaluation measure. Fanaee-T and Gama (2015) consider only p-values in the range $[0.02, 0.25]$ to create the AMOC-curve. However, exactly the omitted low p-values are particularly important when precise predictions with low false positive rates are required which is why we explicitly included this range into the computation of the AMOC-curve.

Comparison to the anomaly detection algorithms. Regarding the synthetic data, which was specifically created in order to evaluate the WSARE algorithms, we can observe that no anomaly detection algorithm can reach competitive $AAUC_{5\%}$ scores to WSARE 3.0. Considering the gap to WSARE 2.0, which in comparison to 3.0 does not distinguish between environmental settings, one reason could be that the anomaly detection algorithms are not able to take the environmental variables into account. Another reason could be the low number of training instances (one for each day) which might have caused problems, especially for the neural networks. Only the SVM, which is known to work well with only few instances, and the Gaussian mixture model are able to achieve acceptable results. These two approaches are in fact able to outperform the WSARE variants on the real data for which we already found evidence that the environmental information might not be useful.

Comparison to the baselines. In the following, we will put the previously discussed results in relation to the baselines. For the global baselines, we can observe that monitoring the total number of cases per time slot is not sufficient to adequately detect most of the outbreaks. Notably, many of the machine learning approaches do in fact not perform considerably better than these simple baselines. The comparison

to our proposed statistical baselines applied on each possible syndrome separately allow further important insights. Our main observation is that, despite its simplicity, they outperform most of the previously discussed, more sophisticated approaches. In fact, in the case of the real data the Gaussian and the negative binomial baselines achieve the best scores. On the synthetic data they are able to achieve results that are competitive to WSARE 3.0 even though the baselines do not take the environmental attributes into account.

Comparison between $\mathcal{S}_{\leq 1}$ and $\mathcal{S}_{\leq 2}$. We can make two basic observations regarding the complexity of the monitored syndromes: Firstly, the outbreaks in the synthetic data are better detected by the algorithms and baselines for non-specific syndromic surveillance when monitoring single condition syndromes $\mathcal{S}_{\leq 1}$ while for the real data we benefit from pair patterns $\mathcal{S}_{\leq 2}$. Secondly, almost no anomaly detector is able to profit from the explicit counts for $\mathcal{S}_{\leq 2}$ regardless of the dataset. For understanding the first effect, we take a closer look at the results of our proposed baselines. These approaches can only take co-occurrences between conditions into account if explicitly given or if the $\mathcal{S} \setminus \mathcal{S}_{\leq 1}$ patterns greatly affect the counts for the composing conditions. Hence, monitoring a larger set of syndromes increases the sensitivity of detecting outbreaks with complex disease patterns. However, it comes at the cost of a higher false alarm rate due to multiple testing. For the real dataset, for which we know that it contains more outbreaks based on two than on one condition, the higher sensitivity is able to outweigh the increased false alarm rate. On the other hand, the results on the synthetic dataset suggests that most of the outbreaks in the synthetic data are lead by single indicators, resulting in more false alarms when monitoring $\mathcal{S}_{\leq 2}$.

In contrast to the non-specific syndromic surveillance approaches, only some anomaly detectors benefit and only slightly from the explicit counts for $\mathcal{S}_{\leq 2}$, such as the local outlier factor algorithm and the isolation forests. This indicates that the remaining approaches, such as SVM and neural networks, already adequately consider correlations between attributes. Especially remarkable is the case of Gaussian mixture models, which achieves the best results in the group when monitoring $\mathcal{S}_{\leq 1}$ but is strongly affected by the $\mathcal{S}_{\leq 2}$ patterns.

7.5 Conclusion

In this chapter, we presented non-specific syndromic surveillance from the perspective of machine learning and gave an overview of the few approaches addressing this task. Furthermore, we introduced a way of how anomaly detection algorithms can be applied on this problem and a set of simple statistical algorithms which we believe should serve as reference points for future experimental comparisons. In an experimental

evaluation, we revisited the non-specific syndromic surveillance approaches in face of the previously not considered statistical baselines and a variety of anomaly detectors. Eventually, we found that these baselines outperform most of the more sophisticated techniques and are competitive to the best approaches in the field.

For future work it remains to improve the representation for the anomaly detectors to include domain specific knowledge or even environmental variables. Moreover, the problem of multiple testing seems to be countered by the possibility of replacing local detectors by more appropriate ones, which for instance take the seasonality of specific syndromes into account.

8

Sum-Product Networks for Non-specific Syndromic Surveillance

In Chapter 7, we have approached outbreak detection from a general unsupervised anomaly detection perspective. One of our main results is that our proposed basic statistical approaches (cf. Section 7.3) already achieved compelling performance compared to the more elaborated machine learning algorithms (cf. Section 7.2). For our last contribution in this thesis, we aim to improve upon this potential. The general idea is to enhance the basic statistical approaches by capturing correlations between syndromes in the monitored data and by incorporating environmental information into the monitoring process.

More precisely, we transfer the idea of these basic statistical approaches to *sum-product networks* (SPNs) (Poon and Domingos, 2011), a statistical and generative machine learning algorithm (cf. Section 3.5). Instead of fitting one particular distribution for each single syndrome, we use SPNs to model the joint probability distribution of syndromic data. We further introduce a technique that allows to detect anomalies by reasoning with the p-values in the SPN model. In addition, syndromic data can be enriched with information about environmental factors, such as the weather or the season, which can be used to condition the SPN on particular circumstances before p-values are computed.

We experimentally compare our proposed approach to established algorithms for non-specific syndromic surveillance (cf. Section 7.2) on a synthetic data set (Fanaee-T and Gama, 2015; Wong et al., 2005) and real data from a German emergency department in which we injected synthetic outbreaks. Our results demonstrate that SPNs can further improve upon the state-of-the-art statistical modeling techniques.

8.1 Non-Specific Syndromic Surveillance

Typically, syndromic surveillance monitors a small set of syndromes that are characteristic for a specific disease. Outbreaks of other infectious diseases that do not correspond to these pre-specified patterns cannot be detected with these systems. To overcome this problem, we focus on what we refer to as non-specific syndromic surveillance. It is an universal approach to syndromic surveillance which aim to detect any suspicious anomalies in the given data, indicating a possible outbreak of any kind of infectious disease. For a detailed formulation of non-specific syndromic surveillance and the used notation, we refer to Section 7.1.1.

8.1.1 Creation of Structured Data

For this chapter, we transform $\mathcal{H}(t)$ into a structured format, which facilitates the analysis with common machine learning algorithms. For a given set of syndromes $\mathcal{S} \subseteq \mathcal{S}_{all}$, we denote $f_{\mathcal{S}} : 2^{\mathcal{C}} \to \mathbb{N}^{|\mathcal{S}|}$ as the function that counts the number of occurrences $f_s(\mathcal{C}(i))$ for each syndrome $s \in \mathcal{S}$ in a given set of instances $\mathcal{C}(i)$ at time i. Based on the syndrome counts, we form a data set $\mathcal{D} = \{(f_{\mathcal{S}}(\mathcal{C}(i)), \mathbf{e}(i)) \mid (\mathcal{C}(i), \mathbf{e}(i)) \in \mathcal{H}(t)\}$ in which each instance represents a single time slot.

Figure 8.1 depicts an example of how the data set is created for syndromes $\mathcal{S}_{\leq 1}$. Note that in case of syndromes $\mathcal{S}_{\leq 2}$, the data set would additionally contain the columns #(male ∧ cough), #(female ∧ cough), #(male ∧ fever), and #(female ∧ fever).

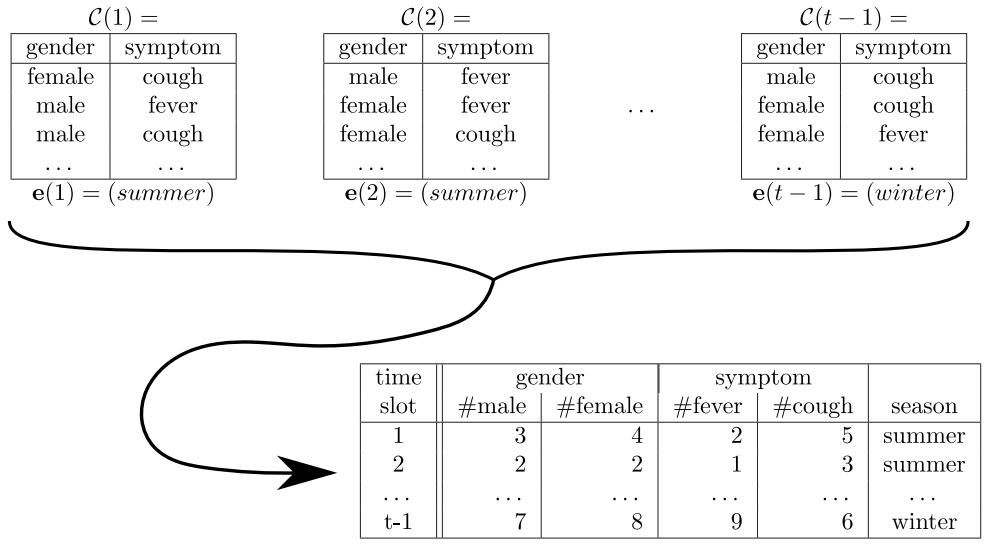


Figure 8.1: Example for the creation of a structured data set using syndromes $\mathcal{S}_{\leq 1}$.

8.1.2 Related Work

While specific syndromic surveillance is a well-studied research area, we found out that only little research has been devoted to non-specific syndromic surveillance. Brossette et al. (1998) adopt the idea of association rule mining to identify anomalous patterns in health data (cf. Section 7.2.1). Wong et al. (2005) first learn a Bayesian network over historic health data and then compare a sample of historical cases to current cases $\mathcal{C}(t)$ to detect potential outbreaks (cf. Section 7.2.2). Fanaee-T and Gama (2015) track changes in the data correlation structure using eigenspace techniques to identify anomalies (cf. Section 7.2.3). In particular, Wong et al. (2005) and Fanaee-T and Gama (2015) distinguish between indicator and environmental attributes to improve detection performance which is also known as contextual or conditional anomaly detection (Song et al., 2007). For more details on these methods, as well as an empirically comparison to common anomaly detectors and statistical modeling techniques, we refer to sections 7.2 and 7.4.

A particular result of the previous chapter is that statistical techniques for a simultaneous and individual monitoring of syndromes $\mathcal{S}_{\leq 1}$ or $\mathcal{S}_{\leq 2}$ already achieve very competitive results and often outperform more elaborate algorithms. More precisely, for each syndrome s a distribution $P_s(x)$ is fitted on $\mathcal{H}(t)$ such that $f_s(C(t)) \sim P_s(x)$. The Poisson and the negative binomial distribution are natural choices but also the Gaussian distribution is used in practice (Hutwagner et al., 2003). However, this approach has two main limitations. Firstly, independence among the monitored syndromes is assumed and, secondly, environmental factors are not taken into account.

8.2 Sum-Product Networks for Syndromic Surveillance

Most statistical techniques, including those mentioned in the previous section, model the joint probability distribution as a product of individual syndrome distributions. Clearly, this is only valid if the syndromes are independent of each other. Sum-product networks (Poon and Domingos, 2011) are an elegant way of extending this simple model by taking dependencies between the monitored syndromes and even dependencies to environmental factors into account.

A *sum-product network* (SPN) models the joint probability distribution $P(\mathcal{X})$ of a data set, where $\mathcal{X} = \{X_1, X_2, \ldots, X_m\}$ is a set of random variables, as a rooted directed acyclic graph of sum, product and leaf nodes. In this graph, the *scope* of a particular node is defined as the set of features appearing in the subgraph below that node. Formally, sum nodes provide a weighted mixture of distributions by combining nodes which share the same scope, whereas product nodes represent the factorization over independent distributions by combining nodes defined over disjunct scopes. Finally,

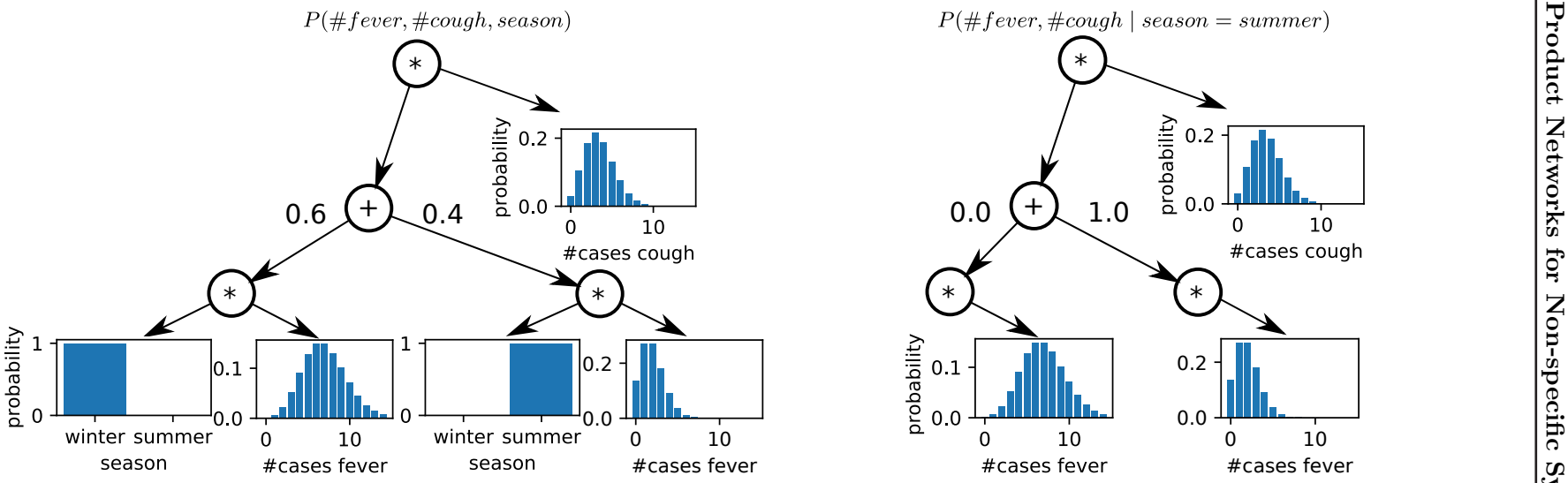


Figure 8.2: The left SPN represents $P(\#fever, \#cough, season)$ while the right SPN represents $P(\#fever, \#cough \mid season = summer)$ derived from the left.

each leaf node contains a univariate distribution $P(X)$ for a particular feature $X \in \mathcal{X}$. In this chapter, we use *mixed* SPNs (Molina et al., 2018) which allow to learn an SPN over both, continuous and discrete attributes.For a detailed explanation of SPNs, we refer to Section 3.5.

Figure 8.2 shows an exemplary SPN, representing the joint probability distribution $P(\#fever, \#cough, season)$. The top product node indicates that the distribution of *#cough* is independent of the other attributes. In contrast, the distribution of *#fever* depends on *season* and, therefore, it is split into two clusters by a sum node, one for the winter and one for the summer. Moreover, Figure 8.2 also shows the same SPN which has been conditioned on $season = summer$ to represent the conditional probability distribution $P(\#fever, \#cough \mid season = summer)$.

8.2.1 Inference of p-values in Sum-Product Networks

The main advantage of SPNs over other probabilistic models is that inference for probabilistic queries is tractable and can be computed in linear time with respect to the size of the network (Poon and Domingos, 2011). For syndromic surveillance we are particularly interested in p-values, which express the chance of obtaining data at least as extreme under a given null hypothesis. In the following, we propose novel extensions to SPNs that allow them to properly reason with p-values.

To compute the p-value for a query $q \subseteq \{X_1 \geq x_1, \ldots, X_m \geq x_m\}$ for arbitrary x_i, the conditions of q are forwarded to the leaves of the SPN. In case q contains conditions only on a subset of attributes of $\mathcal{X}$, the SPN is marginalized beforehand by simply removing all leaves on attributes which are not contained in the query. In the remaining leaves, the p-value for the respective condition is evaluated and propagated upwards. At product nodes, we use either *Fisher's* or *Stouffer's* method (cf. *aggregation of independent hypothesis tests* in Section 2.3.2) for merging independent p-values (Whitlock, 2005). At sum nodes, which encode a mixture of distributions over the same attributes, we need to merge dependent p-values. Vovk and Wang (2020) recommend to use the *harmonic mean* in case of substantial dependence among the merging p-values and suggest to use the *geometric* or the *arithmetic mean* for stronger dependencies (cf. *aggregation of dependent hypothesis tests* in Section 2.3.2). We have implemented the weighted versions of these three merging functions in order to consider the weights of sum nodes during merging. As a result, the obtained value at the root node of the SPN can be seen as a composite p-value for query q.

8.2.2 Application to Non-Specific Syndromic Surveillance

The key idea of our approach is to learn an SPN over a data set that is structured as described in Section 8.1.1. In particular, the SPN models the joint probability distribution $P(X_{\mathcal{S}}, X_{\mathcal{E}})$ where $X_{\mathcal{S}} = \{X_s \mid s \in \mathcal{S}\}$ and $X_{\mathcal{E}} = \{X_E \mid E \in \mathcal{E}\}$ are random variables associated with syndromes $\mathcal{S}$ and environmental attributes $\mathcal{E}$ respectively. For environmental attributes, categorical distributions are used in the leaves, whereas for the syndrome counts we either use Gaussian, Poisson or negative binomial distributions, which are commonly used for monitoring count data in syndromic surveillance.

To check for outbreaks in a given time slot t, we first condition the SPN on the current environmental setting to obtain $P(X_{\mathcal{S}} \mid X_{E_1} = e_1, \ldots, X_{E_k} = e_k)$ where e_i is the i-th element of $\mathbf{e}(t)$. The set of queries $\mathcal{Q}_1 = \{\{X_s \leq f_s(\mathcal{C}(t))\} \mid s \in \mathcal{S}\}$ is then evaluated on the conditioned SPN, which results in a p-value for each syndrome $s \in \mathcal{S}$. This sensitivity to changes for each individual syndrome is indeed important if the potential disease pattern for an outbreak is unknown beforehand. However, for our empirical study in Section 8.3 a single score for the evaluated time slot is required. Therefore, the p-values need to be aggregated under consideration of the multiple-testing problem (cf. Section 2.3.2). Following Roure et al. (2007), we only report the minimum p-value for each time slot t since the Bonferroni correction can be regarded as a form of aggregation of p-values based on the minimum function. In particular, note that scale-free anomaly scores are sufficient for the purpose of identifying the most suspicious time slots. The complement of the selected p-value represents the anomaly score reported for time slot t.

8.2.3 Handling of Higher Order Syndromes

Note that an SPN modeled over frequency counts of syndromes of length 1 ($\mathcal{S}_{\leq 1}$) models the dependencies between the frequency counts of individual syndromes, but it does not model the frequency of their co-occurrence. For example, if both *cough* and *fever* occur with high frequency in the current window $\mathcal{C}(t)$, it does not imply that there are many patients that exhibit both symptoms at the same time. For modeling such interactions, we have two options: First, we can directly include syndromes of length two ($\mathcal{S}_{\leq 2}$) or even higher. The obvious disadvantage is that the number of possible syndromes grows exponentially with their length. Nonetheless, we can use the SPN for making a best guess. More specifically, if we only model syndromes of length 1 ($\mathcal{S}_{\leq 1}$), we can still form the query set $\mathcal{Q}_2 = \mathcal{Q}_1 \cup \{\{X_{s_1} \leq f_{s_1}(\mathcal{C}(t)), X_{s_2} \leq f_{s_2}(\mathcal{C}(t))\} \mid s_1 \neq s_2, s_1 \in \mathcal{S}, s_2 \in \mathcal{S}\}$, and use the resulting p-values as a heuristic best guess for the p-values of syndromes $\mathcal{S}_{\leq 2}$. We will evaluate both approaches in the experimental section.

8.2.4 Interpretability

Due to the single evaluation of the queries for each time slot, we can always track which query, and therefore which syndrome, is responsible for the found anomaly. Combined with the functionality of the SPN to compute expectations, a ranking of the most suspicious anomalies can be provided to local health authorities in order to analyze and understand the found irregularities. An example of such a report can be seen in Table 8.1.

Table 8.1: Exemplary report.

syndrome	**expected count**	**observed count**	***p*-value**
fever $\wedge$ male	1	4	0.004
fever	2	5	0.017
male	5	7	0.133
fever $\wedge$ female	1	1	0.264
...	...	...	...

8.2.5 Scenario-based Modifications.

Our preliminary experiments showed that statistical tests on rare syndromes are often too sensitive to changes, causing many false alarms (cf. Section 7.3). In addition, outbreaks are usually associated with a high number of infections. Therefore, we set the standard deviation σ^2 to a minimum of 1 before fitting the Gaussian distribution in the leaves, and for the Poisson and the negative binomial distribution we set the mean μ to a minimum of 1. We leave the standard deviation untouched for the negative binomial distribution since manipulating the overdispersion can lead to extreme distortions in the estimation.

8.3 Experiments and Results

The goal of our experimental evaluation is to demonstrate that modeling of syndromic data through an SPN can further improve state-of-the-art statistical modeling techniques. To that end, we conducted experiments on synthetic data (Fanaee-T and Gama, 2015; Wong et al., 2005) and on real data from a German emergency department. As the latter did not contain any information about real outbreaks, we injected synthetic outbreaks. This common practice allows the evaluation for arbitrary types of outbreak patterns in a controlled environment. In our case the arrival of an increased number of patients sharing the same symptoms. However, of course, it cannot cover the full range of real life outbreaks, which remains a general challenge under

Table 8.2: Information about the attributes of the synthetic data.

response attributes	#values	environmental attributes	#values
age	3	weather	2
gender	2	flu level	4
action	3	day of week	3
symptom	4	season	4
drug	4		
location	9		

Table 8.3: Information about the attributes of the real data.

response attributes	#values	environmental attributes	#values
age	3	weather	2
gender	2	flu level	4
symptom	28	day of week	3
fever	4	season	4
oxygen saturation	2		
blood pressure	2		
pulse	3		
respiration	3		

the non-availability of complete and certain disease patient data. The development of more realistic evaluation strategies (or alternatively the acquisition of complete and certain patient data) remains a major challenge for the research field (cf. Section 4.5).

8.3.1 Evaluation Setup

In both scenarios, we generate 100 data streams, where each data stream captures daily information $\mathcal{C}(t)$ over a time period of two years. The time slots of the first year are used for training, whereas the second year is reserved for testing only. Each test data stream contains exactly one simulated outbreak starting on a randomly chosen day. The *synthetic data* (Table 8.2) were generated as proposed by Wong et al. (2005). In each stream, an outbreak is simulated which lasts for 14 days, during which people have a higher chance of catching a particular disease. On average, 34 patients are reported per time slot. The *real data* (Table 8.3) consist of fully anonymized patient data from a German emergency department. With the help of a physician, we have extracted a set of attributes and discretized them into meaningful categories. In addition, we enriched the syndromic data with environmental attributes matching the synthetic data. Information about the flu level has been obtained from *SurvStat*[i]

[i]Robert Koch-Institut: SurvStat@RKI 2.0, `https://survstat.rki.de`, 11.01.2021

Table 8.4: Results $AAUC_{5\%}$.

	algorithm	synthetic data $\mathcal{S}_{\leq 1}$	synthetic data $\mathcal{S}_{\leq 2}$	real data $\mathcal{S}_{\leq 1}$	real data $\mathcal{S}_{\leq 2}$
	baseline Gaussian	0.859	0.957	0.331	0.296
	baseline Poisson	1.312	1.321	0.283	0.220
	baseline negative binomial	0.964	1.021	0.259	0.216
without $\mathcal{E}$	autoencoder	1.647	1.549	0.443	0.372
without $\mathcal{E}$	one-class SVM	1.031	1.536	0.353	0.350
without $\mathcal{E}$	Gaussian mixture models	1.128	3.601	0.332	0.449
without $\mathcal{E}$	WSARE	0.907	1.066	0.333	0.281
without $\mathcal{E}$	SPN($\cdot$, $\mathcal{Q}_1$)	0.913	1.082	0.271	0.200
without $\mathcal{E}$	SPN($\mathcal{S}_{\leq 1}$, $\mathcal{Q}_2$)	1.102		0.250	
with $\mathcal{E}$	autoencoder	2.523	1.629	0.452	0.365
with $\mathcal{E}$	one-class SVM	1.519	1.427	0.392	0.347
with $\mathcal{E}$	Gaussian mix. models	3.404	4.033	0.403	0.443
with $\mathcal{E}$	WSARE	0.907	0.996	0.302	0.266
with $\mathcal{E}$	SPN($\cdot$, $\mathcal{Q}_1$)	**0.647**	0.869	0.244	**0.190**
with $\mathcal{E}$	SPN($\mathcal{S}_{\leq 1}$, $\mathcal{Q}_2$)	0.983		0.230	

and weather data from the *DWD*.[ii] On average 165 patients are reported per day. To simulate an outbreak, we first uniformly sampled a syndrome from $\mathcal{S}_{\leq 2}$. In a second step, we sampled the size of the outbreak from a Poisson distribution with mean equal to the standard deviation of the daily patient visits. To avoid over-representing outbreaks on rare syndromes, we ensured that only 20 streams contain outbreaks with syndromes that have a lower frequency than one per day.

We compare our approach to the statistical baselines, WSARE and anomaly detection algorithms which performed best in our evaluation in Section 7.4. Parameters were tuned in a grid search using 1000 iterations of *bootstrap bias corrected cross-validation* (Tsamardinos et al., 2018) which integrates hyperparameter tuning and performance estimation into a single evaluation loop. The evaluated parameters combinations for all algorithms can be found in our repository.[iii]

As a performance measure, we report the partial area under AMOC-curve for a false alarm rate less than 5% (referred to as $AAUC_{5\%}$) because of the importance of very low false alarm rate in syndromic surveillance (cf. Section 4.5). We report average $AAUC_{5\%}$ scores over all 100 data streams. Note that the worst possible result for $AAUC_{5\%}$ is 14 on the synthetic and 1 on the real data, respectively.

[ii] Deutscher Wetterdienst: Open Data, `https://www.dwd.de/opendata`, 11.01.2021

[iii] Our code is publicly available at `https://github.com/MoritzKulessa/NSS`.

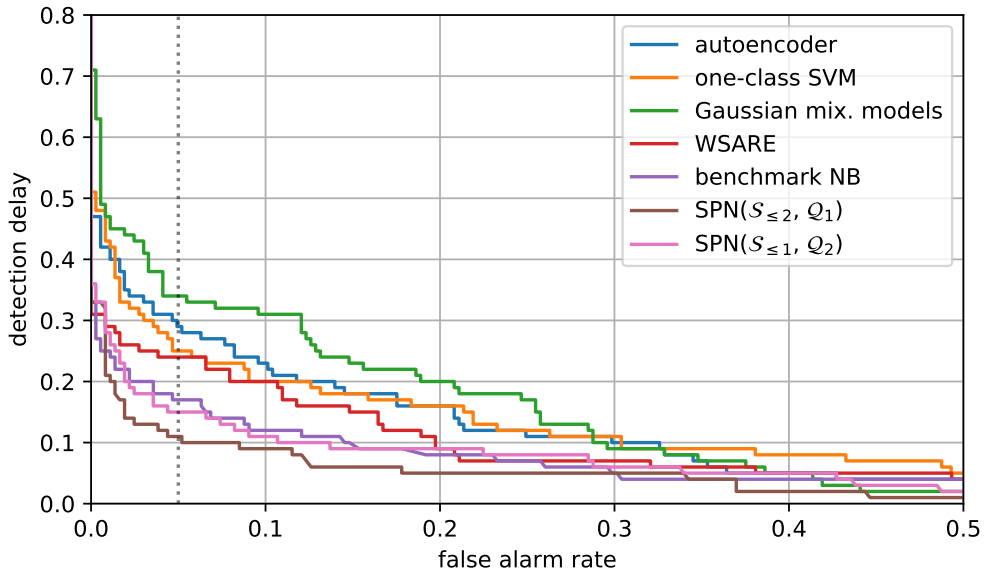


Figure 8.3: AMOC-curve for monitoring $\mathcal{S}_{\leq 2}$ on the real data.

8.3.2 Results

Comparison of algorithms. Table 8.4 shows the results of all algorithms for monitoring syndromes $\mathcal{S}_{\leq 1}$ and $\mathcal{S}_{\leq 2}$. As the consideration of environmental attributes is one of the main differences compared to the baselines, we performed additional evaluations without considering environmental attributes (cf. without $\mathcal{E}$).

We can see that the SPNs with the conventional queries $\mathcal{Q}_1$ and taking environmental variables into account outperform all other competitors, on both syndrome sets $\mathcal{S}_{\leq 1}$, $\mathcal{S}_{\leq 2}$. The AMOC curve shown in Figure 8.3 confirms this result. Clearly, SPN($\mathcal{Q}_1$) offers the best trade-off between detection delay and false alarm rate, also for the extended range from 0 to 0.5.

Going into more detail, we can observe that the improvement over the baselines is more pronounced for the synthetic than for the real data. This is in line with previous findings that indicated a higher dependency on environmental factors in the synthetic data set (cf. Section 7.4). Regarding the exclusion of environmental attributes, we expected an advantage of SPNs over the baselines due to the modeling of dependencies between syndromes. Conversely, in accompanying analyses we found that overfitting of the SPN can also result in less stable estimates. In fact, both effects seem to balance each other out in our comparison since the SPNs are on par with the baselines.

Nonetheless, the results indicate that in our analyzed data SPNs can benefit from dependencies between the syndrome patterns if they are combined with environmental factors. Similar experiments have been conducted for the anomaly detectors. Contrary to the SPNs, we can observe that these approaches do not benefit from environmental

Table 8.5: Detailed SPN results.

		synthetic data			real data		
setting	**distribution**	**average**	**harmonic**	**geometric**	**average**	**harmonic**	**geometric**
$\mathcal{S}_{\leq 1}$, $\mathcal{Q}_1$	Gaussian	0.711	0.862	0.894	0.267	0.332	0.332
	Poisson	0.660	1.317	1.130	0.276	0.244	0.243
	neg. binomial	**0.634**	0.962	1.112	**0.228**	0.235	0.234
$\mathcal{S}_{\leq 2}$, $\mathcal{Q}_1$	Gaussian	0.853	0.966	0.966	0.276	0.284	0.278
	Poisson	**0.805**	1.328	1.106	0.232	0.216	0.200
	neg. binomial	0.876	1.026	1.127	0.207	**0.178**	0.187
$\mathcal{S}_{\leq 1}$, $\mathcal{Q}_2$ Stouffer	Gaussian	1.045	1.152	1.152	0.276	0.314	0.263
	Poisson	**0.993**	2.322	1.549	0.261	0.250	0.252
	neg. binomial	1.091	1.500	1.683	0.263	0.256	**0.232**
$\mathcal{S}_{\leq 1}$, $\mathcal{Q}_2$ Fisher	Gaussian	0.943	1.110	1.112	0.265	0.295	0.268
	Poisson	**0.930**	2.099	1.414	0.262	0.254	0.253
	neg. binomial	0.968	1.372	1.538	0.253	0.256	**0.228**

information. An explanation could be the inability to condition on the given environmental attributes as all attributes are treated in the same manner by the anomaly detectors. For example, a rare environmental scenario can lead to an high anomaly score even though the observed syndromic situation might not be exceptional.

Comparison between $\mathcal{S}_{\leq 1}$ and $\mathcal{S}_{\leq 2}$. We can observe that outbreaks in the synthetic data are better detected when monitoring single condition syndromes $\mathcal{S}_{\leq 1}$ while monitoring $\mathcal{S}_{\leq 2}$ works better for the real data. As discussed in Section 8.2.3, we can approximate $\mathcal{S}_{\leq 2}$ results when using query $\mathcal{Q}_2$ on SPN($\mathcal{S}_{\leq 1}$) (last line of Table 8.4). We can see that in both cases, the approximation with SPN($\mathcal{S}_{\leq 1}, \mathcal{Q}_2$) does not reach the performance of directly modelling SPN($\mathcal{S}_{\leq 2}$, $\mathcal{Q}_1$) but in the case of real data it improves over SPN($\mathcal{S}_{\leq 1}$, $\mathcal{Q}_1$). Thus monitoring $\mathcal{Q}_2$ can be beneficial when the computational costs of direct modelling higher order syndromes are prohibitive.

Analysis of parameters of the SPN. Table 8.5 shows the results of different methods for combining p-values with respect to the distributions used in the leaves of the SPN. The columns correspond to the method for merging p-values in sum nodes while rows represent the *setting* and the used *distribution*. Note that p-values in the product node are only merged if we evaluate $\mathcal{Q}_2$, in which case, we tested merging with Fisher's or Stouffer's method.

Most notably, we observe that a simple weighted average of p-values works best on the synthetic data regardless of the other parameter settings. Following the theoretic results of Vovk and Wang (2020), we can only hypothesize that this is the case due to strong dependencies between the attributes. In contrast, the results are less clear on the real data set. For instance, regarding the negative binomial distribution, the arithmetic mean seems to be more preferable when using $\mathcal{S}_{\leq 1}$ whereas the harmonic mean achieves the highest score on $\mathcal{S}_{\leq 2}$. With respect to $\mathcal{Q}_2$ the results suggest a slight advantage of the Fisher's method over the Stouffer's method on both data sets.

In summary, the results exhibit clear differences between the merging options, but these seem to be highly dependent on the data, distributions and architectures used. An approach that goes beyond the proposed parameter selection and makes these decisions at each inner node of the SPN in a data-driven way could be a way of further exploiting these gaps. We leave these extensions for future work.

8.4 Conclusion

In this chapter, we proposed the use of SPNs for modeling the joint probability distribution of syndromic data. The main technical contribution is a method for propagating p-values in SPNs in order to detect anomalies as potential indicators for an

outbreak of an infectious disease. In addition, the SPN can consider environmental factors, such as the season, the weather, or the current level of influenza infections, which may increase or decrease the awareness of outbreaks with particular disease patterns.

Our empirical study revealed that our proposed approach outperforms state-of-the-art algorithms in the field of non-specific syndromic surveillance, hence, on the task of detecting emerging diseases. In particular, by taking correlation between the monitored syndromes and environmental factors into account, the performance of our approach improved substantially.

9

Summary and Conclusions

Infectious disease surveillance is of great importance for the prevention of major outbreaks which otherwise would have severe consequences on public health and economy. One of the most effective approaches to contain the spread of a disease is an early detection of the outbreak which allows to apply control measures as soon as possible. Therefore, either the confirmed cases or the cases with early symptoms of a particular disease are monitored with statistical methods. Mainly based on hypothesis testing, these methods automatically raise an alarm if an unexpected increase in the number of infections is observed.

This thesis contributes in a variety of ways to improve outbreak detection based on data-driven models learned over historic data. In addition to presenting an approach that supports epidemiologists on the basis of data-driven suggestions for syndrome definitions, we have developed algorithms that are better at detecting sudden increases in the number of infections than common statistical methods. Throughout our research, we focused on traditional as well as syndromic surveillance and even extended the application scenario to also include outbreak detection of emerging diseases with an beforehand unknown disease pattern.

9.1 Summary

In the first part, including chapters 2, 3, and 4, concepts of probability theory, statistical hypothesis testing, machine learning and disease surveillance are reviewed to provide a broad background knowledge about the subject of this thesis.

Based on this knowledge, we first set our focus on syndromic surveillance (cf. Section 4.2). A particular challenge in this area is the definition of disease patterns since they highly depend on the infectious disease and the health-related data source under surveillance. To support epidemiologists in this process, we have presented a novel, data-driven approach in Chapter 5 to discover such patterns in historic data. The key idea of the proposed algorithm is to extract indicators from the health-related data source which correlate with the reported number of infections in the respective geographic region.

We have evaluated our approach on patient-based data of 12 emergency departments. In a first experiment, we could show that our proposed method is capable to reconstruct synthetic generated disease patterns with varying characteristics. Furthermore, we have conducted experiments to discover real disease patterns for three infectious diseases. Our results suggest that a correlation-based learning approach for the extraction of disease patterns can identify meaningful indicators that are closely related to a particular disease under surveillance. However, these extracted disease patterns also often include indicators which do not relate to the respective disease.

In our next contribution, we set the focus on traditional surveillance which is the monitoring of confirmed infections (cf. Section 4.1). In Chapter 6, we proposed an approach for this scenario which combines the output of multiple statistical methods using a machine learning technique called stacking. Instead of relying only on the binary output (*alarm* or *no alarm*) of the statistical algorithms, we propose to make use of their p-values for training a fusion classifier. In addition, we also show that augmenting additional features and adapting the labeling of an epidemic period may further improve performance. For comparison and evaluation, a new measure is introduced which captures the performance of an outbreak detection method with respect to a low rate of false alarms more precisely than previous works.

We have performed experiments on synthetic data to evaluate our proposed approach and the adaptations in a controlled setting. Furthermore, we used the reported cases for the disease *Salmonella* and *Campylobacter* from 2001 until 2018 all over Germany to evaluate on real data. The experimental results show a substantial improvement on the synthetic data when p-values are used for learning. The results on real data are less clear. Inconsistencies in the data, which occur under real conditions, make it more challenging for the learning approach to identify valuable patterns for outbreak detection.

One of the major drawbacks of common disease surveillance is that only known diseases are monitored. In particular, for syndromic surveillance major efforts are spent on the definition of syndromes which only relate to specific diseases. Consequently, outbreaks of emerging diseases are likely to be missed by such surveillance systems. We addressed this problem in Chapter 7 by framing outbreak detection as a general anomaly detection task. The general idea is to monitor all available data to detect any kind of infectious disease outbreaks. We have termed it non-specific syndromic surveillance and provided an overview of this area from the perspective of machine learning.

Based on our proposed unified framework with local and global modeling techniques, we revisited published approaches and applied common anomaly detection algorithms to the problem. In addition, we also present a set of statistical modeling techniques which can serve as baselines in future works. In an experimental comparison of different approaches to non-specific syndromic surveillance we found that these simple

statistical techniques already achieve competitive results and sometimes even outperform more elaborate machine learning approaches.

The compelling performance of the aforementioned statistical baselines for non-specific syndromic surveillance has shown us that there is still potential for improvement. Inspired by the concept of these baselines, we proposed an approach based on sum-product networks in Chapter 8 which models the joint probability distribution of syndromic data. On the one hand, this enables us to capture correlations of the monitored syndromes. On the other hand, it even allows us to incorporate environmental information into the monitoring process. In contrast to the conventional inference of probabilities or densities in a sum-product network, we presented a novel approach to properly reason with p-values in order to detect anomalies.

The conducted experiments on synthetic and real data showed that our approach is able to outperform state-of-the-art techniques for detecting outbreaks of emerging diseases. In particular, the performance of our approach enhanced substantially by taking correlation between the monitored syndromes and environmental factors into account.

9.2 Conclusions

The presented work shows the potential of machine learning to improve outbreak detection. However, throughout the years of our research we faced several challenges when implementing data-driven approaches in this field. In the following paragraphs, we summarize these challenges and conclude our work.

First of all, the knowledge of experts is still indispensable for the evaluation and supervision of machine learning methods. In particular for our first contribution (cf. Chapter 5), the extracted disease patterns need to be validated based on the semantic meaning of the indicators. Such kind of semantic information is difficult to incorporate into the learning process, especially if the algorithm should be applied to a variety of different infectious diseases. As a solution to this problem, we propose that domain experts are required to interact with the algorithms to successively refine syndrome definitions.

Furthermore, the lack of labeled data is one of the major problems in the area of outbreak detection. The key takeaway of our second contribution (cf. Chapter 6) is that even if labeled data is available, the annotation is often inconsistent and heterogeneous since a precise norm for the labeling does not exist. Especially this has an impact on the learnability of the model which in turn lead to unexpected results, reducing the acceptance and trustworthiness in machine learning approaches among epidemiologists. In addition, particularly for rare infectious diseases insufficient amounts of labeled

data are available to adequately learn a supervised classifier. To overcome these challenges, we suggest to shift the attention towards unsupervised learning which does not depend on labeled data.

Indeed, with our framework to non-specific syndromic surveillance presented in Chapter 7 we aim to make the task of unsupervised outbreak detection more approachable for the machine learning community. Based on a general anomaly detection perspective, we even extend common disease surveillance by including the surveillance of emerging diseases, a research area not extensively studied yet. On the one hand, it relates to conditional anomaly detection since environmental attributes can be integrated into the monitoring process. On the other hand, it is a special type of anomaly detection in which we are only interested in unexpected increases in the number of cases for any kind of symptom pattern.

Another key advantage of this general approach to outbreak detection is transferability. While syndromic surveillance requires to design disease patterns specifically for each data source and disease under surveillance, non-specific syndromic surveillance algorithms only require to choose a set of epidemiologically relevant attributes. Hence, they are easy to implement on a variety of different data sources. Even if such approaches are not integrated into the active surveillance system, they can always be taken into consideration as additional evidence. For example, they can assist epidemiologists to further support decision making in difficult epidemiological situations (Althouse et al., 2015).

Among our contributions, we can observe that the inclusion of additional information into the monitoring process facilitates the detection of outbreaks. While for our stacking approach (cf. Chapter 6) further indicators of the univariate time series (e.g., mean over the last counts) helps, we could also show in Chapter 8 that conditioning on environmental information improves outbreak detection performance. Therefore, we conclude that machine learning techniques offer an elegant approach to enrich the detection by additional signals and sources of information in a highly flexible and data-driven way. With respect to the statistical methods, we suggest to make use of their p-values instead of binary decisions according to a pre-specified significance level. On the one hand, it requires less specification while, on the other hand, the learning algorithm is able to differ between the outputs more precisely.

9.3 Perspectives

Throughout this thesis, we have introduced minor and major improvements to disease surveillance. However, we only scratched the surface of what is possible with data-driven approaches. Under consideration of the presented approaches and the

aforementioned conclusions a variety of exciting avenues for future work exist which we will discuss in the following paragraphs.

Most importantly, more work can be spent on non-specific syndromic surveillance. In particular, in our research we only focused on improving common statistical methods with machine learning. If viewed from a different perspective (cf. Section 4.7), we believe that outbreak detection performance can be further enhanced with more reliable methods to identify clusters of infections. Indeed, the presented and revisited techniques based on sum-product networks, Bayesian networks, eigenspace techniques and association rule mining (cf. Section 7.2 and Chapter 8) already show the diversity the problem can be encountered with. Especially our proposed baselines for non-specific syndromic surveillance presented in Section 7.3 can serve as reference points for future experimental comparisons. This may foster the development of algorithms in this newly emerging field.

A major drawback of most reviewed and proposed approaches in this thesis is that they are mainly designed to identify sudden increases of infections in the monitored data stream. However, slight but constant changes over time in the underlying probability distribution of the syndromic data can be indicative for an ongoing outbreak as well. In particular, for our stacking approach presented in Chapter 6, we could show that the mean over the last time points and the previous outputs of the statistical methods can improve outbreak detection performance. Such kind of indicators can also be integrated in the other proposed approaches or considered during development of more advanced techniques.

Furthermore, an interesting avenue for future work is also the inclusion of geo-spatial information (Robertson et al., 2010). For example, residential information of patients captured in emergency departments can be used to detect increases of infections with respect to a particular area. If such detailed information is not available, information about the location can still be used to enhance the surveillance of multiple data sources. For example, spatial correlations of pharmaceutical purchases can be used in a region-wide monitoring of pharmacies.

Regarding Chapter 5, further work can be spent on technical improvements for our algorithm that help to prevent overfitting (cf. Section 5.4). This may include the use of pruning techniques to remove problematic clauses from rules or the use of different correlation measures during the learning process. Furthermore, we believe that an interactive learning approach to discover disease patterns is indispensable to prevent the inclusion of undesired indicators in the patterns. The general idea is to implement a syndrome explorer in which epidemiologists can create disease patterns interactively based on data-driven suggestions.

With respect to our contribution presented in Chapter 8, we believe that sum-product networks can also be used for multivariate surveillance (cf. Section 4.6). Particularly

in traditional surveillance the monitoring of multiple infectious diseases can be unified in one model. Such a model can be further enhanced by taking environmental information into account.

Beyond outbreak detection, we think that our novel approach to reason with p-values in sum-product networks also contribute towards anomaly detection in general (cf. Section 3.6). Especially, if compared to density-based approaches, p-values are superior due to their scale invariance. Moreover, they even make it possible to perform one-sided anomaly detection by considering only left-sided or right-sided p-values. In addition, sum-product networks can easily be extended to perform conditional anomaly detection (Song et al., 2007) and the probability distributions used in the leaves can be changed as desired.

Curriculum Vitae

Education

Master of Science in Computer Science at TU Darmstadt,
April 2015 – August 2018

Bachelor of Science in Computer Science at TU Darmstadt,
April 2012 – March 2015

Academic Employment

Research Assistant, Department of Computer Science, TU Darmstadt,
September 2018 – September 2021

Professional Memberships

Guest Scientist, Department of Infectious Disease Epidemiology,
Robert Koch Institute, April 2019 – September 2021

Publications

M. Kulessa and E. Loza Mencía. Dynamic classifier chain with random decision trees. In *Proceedings of the 21st International Conference on Discovery Science*, pages 33–50, 2018

M. Kulessa, A. Molina, C. Binnig, B. Hilprecht, and K. Kersting. Model-based approximate query processing. *arXiv:1811.06224*, 2018

M. Kulessa, B. Hilprecht, A. Molina, K. Kersting, and C. Binnig. Towards model-based approximate query processing. In *Working Notes of the 1st International Workshop on Applied AI for Database Systems and Applications (held in conjunction with VLDB 2019)*, 2019a

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Improving outbreak detection with stacking of statistical surveillance methods. In *Workshop Proceedings: Epidemiology Meets Data Mining and Knowledge Discovery (held in conjunction with ACM SIGKDD 2019)*, 2019b

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Improving the fusion of outbreak detection methods with supervised learning. In *Proceedings of the 16th International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 55–66, 2020

B. Hilprecht, A. Schmidt, M. Kulessa, A. Molina, K. Kersting, and C. Binnig. Deepdb: Learn from data, not from queries! *Proceedings of the VLDB Endowment*, 13(7):992–1005, 2020

S. Bohlender, E. Loza Mencía, and M. Kulessa. Extreme gradient boosted multi-label trees for dynamic classifier chains. In *Proceedings of the 23rd International Conference on Discovery Science*, pages 471–485, 2020

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Revisiting non-specific syndromic surveillance. In *Proceedings of 19th International Symposium on Intelligent Data Analysis*, pages 128–140, 2021a

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. A unifying framework and comparative evaluation of statistical and machine learning approaches to non-specific syndromic surveillance. *Computers (Special Issue: Artificial Intelligence for Health)*, 10(3):32, 2021b

M. Kulessa, B. Wittelsbach, E. Loza Mencía, and J. Fürnkranz. Sum-product networks for early outbreak detection of emerging diseases. In *Proceedings of the 19th International Conference on Artificial Intelligence in Medicine*, pages 61–71, 2021c. *(nominated for best student paper award)*

F. Busch, M. Kulessa, E. Loza Mencía, and H. Blockeel. Combining predictions under uncertainty: Random decision trees. In *Proceedings of the 24th International Conference on Discovery Science*, pages 78–93, 2021

E. Loza Mencía, M. Kulessa, S. Bohlender, and J. Fürnkranz. Tree-based dynamic classifier chains. *Machine Learning*, 2022. To appear

M. Rapp, M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Correlation-based discovery of disease patterns for syndromic surveillance. *Frontiers in Big Data*, 4, 2022

Bibliography

B. M. Althouse, S. V. Scarpino, L. A. Meyers, J. W. Ayers, M. Bargsten, J. Baumbach, J. S. Brownstein, L. Castro, H. Clapham, D. A. Cummings, et al. Enhancing disease surveillance with novel data streams: Challenges and opportunities. *European Physical Journal on Data Science*, 4(1):1–8, 2015.

V. Amrhein, S. Greenland, and B. McShane. Scientists rise up against statistical significance. *Nature*, 567:305–307, 03 2019.

F. Ansaldi, A. Orsi, F. Altomonte, G. Bertone, V. Parodi, R. Carloni, P. Moscatelli, E. Pasero, P. Oreste, and G. Icardi. Emergency department syndromic surveillance system for early detection of 5 syndromes: A pilot project in a reference teaching hospital in Genoa, Italy. *Journal of Preventive Medicine and Hygiene*, 49(4):131–135, 2008.

S. Bansal, G. Chowell, L. Simonsen, A. Vespignani, and C. Viboud. Big data for infectious disease surveillance and modeling. *Journal of Infectious Diseases*, 214: 375–379, 2016.

R. J. Barro, J. F. Ursúa, and J. Weng. The coronavirus and the great influenza pandemic: Lessons from the "spanish flu" for the coronavirus's potential effects on mortality and economic activity. Technical report, National Bureau of Economic Research, 2020.

S. Bay and M. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5:213–246, 2001.

G. Bédubourg and Y. Le Strat. Evaluation and comparison of statistical methods for early temporal detection of outbreaks: A simulation-based study. *PLOS ONE*, 12 (7):1–18, 2017.

Q. Bi, K. E. Goodman, J. Kaminsky, and J. Lessler. What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*, 188(12):2222–2239, 2019.

A. Bifet, R. Gavaldà, G. Holmes, and B. Pfahringer. *Machine learning for data streams: With practical examples in MOA*. MIT press, 2018.

C. M. Bishop. *Pattern recognition and machine learning*. Springer Science & Business Media, 2006.

L. Bjerring and E. Frank. Beyond trees: Adopting MITI to learn rules and ensemble classifiers for multi-instance data. In *Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence*, pages 41–50, 2011.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

T. S. Boender, W. Cai, M. Schranz, T. Kocher, B. Wagner, A. Ullrich, S. Buda, R. Zöllner, F. Greiner, M. Diercke, et al. Using routine emergency department data for syndromic surveillance of acute respiratory illness before and during the COVID-19 pandemic in Germany, week 10-2017 and 10-2021. *medRxiv*, 2021.

S. Bohlender, E. Loza Mencía, and M. Kulessa. Extreme gradient boosted multi-label trees for dynamic classifier chains. In *Proceedings of the 23rd International Conference on Discovery Science*, pages 471–485, 2020.

E. Bouchouar, B. M. Hetman, and B. Hanley. Development and validation of an automated emergency department-based syndromic surveillance system to enhance public health surveillance in Yukon: A lower-resourced and remote setting. *BMC Public Health*, 21(1):1–13, 2021.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 93–104, 2000.

S. Brossette, A. Sprague, J. Hardin, K. Waites, W. Jones, and S. Moser. Association rules and data mining in hospital infection control and public health surveillance. *Journal of the American Medical Informatics Association*, 5:373–81, 1998.

D. L. Buckeridge. Outbreak detection through automated surveillance: A review of the determinants of detection. *Journal of Biomedical Informatics*, 40(4):370–379, 2007.

D. L. Buckeridge, H. Burkom, M. Campbell, W. R. Hogan, A. W. Moore, et al. Algorithms for rapid outbreak detection: A research synthesis. *Journal of Biomedical Informatics*, 38(2):99–113, 2005.

J. W. Buehler, R. S. Hopkins, J. M. Overhage, D. M. Sosin, and V. Tong. Framework for evaluating public health surveillance systems for early detection of outbreaks. `https://www.cdc.gov/mmwr/preview/mmwrhtml/rr5305a1.htm`, 2008. [Online; accessed 14-July-2020].

H. Burkom, L. Ramac-Thomas, S. Babin, R. Holtry, Z. Mnatsakanyan, and C. Yund. An integrated approach for fusion of environmental and human health data for disease surveillance. *Statistics in Medicine*, 30(5):470–479, 2011.

F. Busch, M. Kulessa, E. Loza Mencía, and H. Blockeel. Combining predictions under uncertainty: Random decision trees. In *Proceedings of the 24th International Conference on Discovery Science*, pages 78–93, 2021.

P. Chakraborty, P. Khadivi, B. Lewis, A. Mahendiran, J. Chen, P. Butler, E. Nsoesie, S. Mekaru, J. Brownstein, M. Marathe, and N. Ramakrishnan. Forecasting a moving target: Ensemble models for ILI case count predictions. In *Proceedings of the SIAM International Conference on Data Mining*, pages 262–270, 2014.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

Y. Chevaleyre and J. Zucker. A framework for learning rules from multiple instance data. In *Proceedings of the 12th European Conference on Machine Learning*, pages 49–60, 2001.

T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997.

V. F. Dirmyer. Using real-time syndromic surveillance to analyze the impact of a cold weather event in New Mexico. *Journal of Environmental and Public Health*, 2018.

G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 43–52, 1999.

W. Duivesteijn, A. J. Feelders, and A. Knobbe. Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98, 2016.

V. L. Edge, F. Pollari, L. King, P. Michel, S. A. McEwen, J. B. Wilson, M. Jerrett, P. N. Sockett, and S. W. Martin. Syndromic surveillance of norovirus using over the counter sales of medications related to gastrointestinal illness. *Canadian Journal of Infectious Diseases and Medical Microbiology*, 17(4):235–241, 2006.

H. Fanaee-T and J. Gama. Eigenevent: An algorithm for event detection from complex data streams in syndromic surveillance. *Intelligent Data Analysis*, 19:597—616, 2015.

D. Farrow, L. Brooks, S. Hyun, R. J. Tibshirani, D. Burke, and R. Rosenfeld. A human judgment approach to epidemiological forecasting. *PLOS Computational Biology*, 13(3):1–19, 2017.

C. Faverjon, M. G. Andersson, A. Decors, J. Tapprest, P. Tritz, A. Sandoz, O. Kutasi, C. Sala, and A. Leblond. Evaluation of a multivariate syndromic surveillance system for west nile virus. *Vector-Borne and Zoonotic Diseases*, 16(6):382–390, 2016.

T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8): 861–874, 2006.

T. Fawcett and F. Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, 1999.

S. Fernandes, H. Fanaee-T., and J. Gama. The initialization and parameter setting problem in tensor decomposition-based link prediction. In *Proceedings of the International Conference on Data Science and Advanced Analytics*, pages 99–108, 2017.

M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.

R. A. Fisher. *Statistical methods for research workers.* Oliver and Boyd, Edinburgh and London, 1934.

R. D. Fricker. Syndromic surveillance. *Wiley StatsRef: Statistics Reference Online*, 2014.

R. D. Fricker, B. Hegler, and D. Dunfee. Comparing syndromic surveillance detection methods: Ears' versus a cusum-based methodology. *Statistics in Medicine*, 27(17): 3407–3429, 2008.

J. Fürnkranz, D. Gamberger, and N. Lavrač. *Foundations of rule learning.* Springer Science & Business Media, 2012.

M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: A review. *ACM Sigmod Record*, 34(2):18–26, 2005.

R. Gens and D. Pedro. Learning the structure of sum-product networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 873–880, 2013.

E. Goldstein, B. J. Cowling, A. E. Aiello, S. Takahashi, G. King, Y. Lu, and M. Lipsitch. Estimating incidence curves of several infections using symptom surveillance data. *PLOS ONE*, 6(8):1–8, 2011.

I. Gräff, B. Goldschmidt, P. Glien, M. Bogdanow, R. Fimmers, A. Hoeft, S.-C. Kim, and D. Grigutsch. The german version of the manchester triage system and its quality criteria-first assessment of validity and reliability. *PLOS ONE*, 9(2):e88995, 2014.

D. Grennan. What is a pandemic? *Journal of the American Medical Association*, 321 (9):910, 2019.

K. P. Hartnett, A. Kite-Powell, J. DeVies, M. A. Coletta, T. K. Boehmer, J. Adjemian, A. V. Gundlapalli, et al. Impact of the COVID-19 pandemic on emergency department visits—United states, January 1, 2019–May 30, 2020. *Morbidity and Mortality Weekly Report*, 69(23):699–704, 2020.

S. I. Hay, D. B. George, C. L. Moyes, and J. S. Brownstein. Big data opportunities for global infectious disease surveillance. *PLOS Medicine*, 10(4):e1001413, 2013.

N. A. Heard and P. Rubin-Delanchy. Choosing between methods of combining-values. *Biometrika*, 105(1):239–246, 2018.

R. Heffernan, F. Mostashari, D. Das, A. Karpati, M. Kulldorff, and D. Weiss. Syndromic surveillance in public health practice, New York City. *Emerging Infectious Diseases*, 10:858–864, 2004.

L. Held, M. Höhle, and M. Hofmann. A statistical framework for the analysis of multivariate infectious disease surveillance counts. *Statistical Modelling*, 5(3):187–199, 2005.

K. J. Henning. What is syndromic surveillance? *Morbidity and Mortality Weekly Report*, 53:7–11, 2004.

J. M. Hilbe. *Modeling count data*. Springer Berlin Heidelberg, 2011.

K. M. Hiller, L. Stoneking, A. Min, and S. M. Rhodes. Syndromic surveillance for influenza in the emergency department–a systematic review. *PLOS ONE*, 8(9): e73832, 2013.

B. Hilprecht, A. Schmidt, M. Kulessa, A. Molina, K. Kersting, and C. Binnig. Deepdb: Learn from data, not from queries! *Proceedings of the VLDB Endowment*, 13(7): 992–1005, 2020.

K. Hope, D. N. Durrheim, D. Muscatello, T. Merritt, W. Zheng, P. Massey, P. Cashman, and K. Eastwood. Identifying pneumonia outbreaks of public health importance: Can emergency department data assist in earlier identification? *Australian and New Zealand Journal of Public Health*, 32(4):361–363, 2008.

R. S. Hopkins, C. C. Tong, H. S. Burkom, J. E. Akkina, J. Berezowski, M. Shigematsu, P. D. Finley, I. Painter, R. Gamache, V. J. D. R. Vilas, and L. C. Streichert. A practitioner-driven research agenda for syndromic surveillance. *Public Health Reports*, 132(1):116–126, 2017.

H. Hughes, R. Morbey, T. Hughes, T. Locker, T. Shannon, C. Carmichael, V. Murray, S. Ibbotson, M. Catchpole, B. McCloskey, et al. Using an emergency department syndromic surveillance system to investigate the impact of extreme cold weather events. *Public Health*, 128(7):628–635, 2014.

L. Hutwagner, W. Thompson, G. Seeman, and T. Treadwell. The bioterrorism preparedness and response early aberration reporting system (EARS). *Journal of Urban Health*, 80(1):i89–i96, 2003.

L. Hutwagner, T. Browne, G. Seeman, and A. Fleischauer. Comparing aberration detection methods with simulated data. *Emerging Infectious Diseases*, 11(2):314–316, 2005.

A. I. Ising, D. A. Travers, J. MacFarquhar, A. Kipp, and A. E. Waller. Triage note in emergency department-based syndromic surveillance. *Advances in Disease Surveillance*, 1:34, 2006.

O. Ivanov, M. M. Wagner, W. W. Chapman, and R. T. Olszewski. Accuracy of three classifiers of acute gastrointestinal syndrome for syndromic surveillance. In *Proceedings of the AIMA Symposium*, pages 345–349, 2002.

M. Jackson, A. Baer, I. Painter, and J. Duchin. A simulation study comparing aberration detection algorithms for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 7(1):6, 2007.

N. Jafarpour, D. Precup, M. Izadi, and D. Buckeridge. Using hierarchical mixture of experts model for fusion of outbreak detection methods. In *Proceedings of the AMIA Annual Symposium*, pages 663–669, 2013.

N. Jafarpour, M. Izadi, D. Precup, and D. L. Buckeridge. Quantifying the determinants of outbreak detection performance through simulation and machine learning. *Journal of Biomedical Informatics*, 53:180–187, 2015.

N. Japkowicz and M. Shah. *Evaluating learning algorithms: A classification perspective*. Cambridge University Press, 2011.

F. V. Jensen. *An introduction to Bayesian networks*. UCL press, 1996.

R. S. Jhangiani, I.-C. A. Chiang, C. Cuttler, D. C. Leighton, et al. *Research methods in psychology*. Kwantlen Polytechnic University, 2019.

L. Jiang, Z. Cai, D. Wang, and S. Jiang. Survey of improving k-nearest-neighbor for classification. In *Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 679–683, 2007.

K. Johnson, A. Alianell, and R. Radcliffe. Seasonal patterns in syndromic surveillance emergency department data due to respiratory illnesses. *Online Journal of Public Health Informatics*, 6(1), 2014.

K. E. Jones, N. G. Patel, M. A. Levy, A. Storeygard, D. Balk, J. L. Gittleman, and P. Daszak. Global trends in emerging infectious diseases. *Nature*, 451(7181): 990–993, 2008.

M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

K. Kalimeri, M. Delfino, C. Cattuto, D. Perrotta, V. Colizza, C. Guerrisi, C. Turbelin, J. Duggan, J. Edmunds, C. Obi, et al. Unsupervised extraction of epidemic syndromes from participatory influenza surveillance self-reported symptoms. *PLOS Computational Biology*, 15(4):e1006173, 2019.

E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM International Conference on Data Mining*, pages 1–11, 2001.

B. Khaleghi, A. Khamis, F. Karray, and S. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

H.-Y. Kim. Statistical notes for clinical researchers: Chi-squared test and Fisher's exact test. *Restorative Dentistry and Endodontics*, 42(2):152–155, 2017.

K. Kleinman and A. Abrams. Assessing surveillance using sensitivity, specificity and timeliness. *Statistical Methods in Medical Research*, 15(5):445–464, 2006.

A. Knobbe, B. Crémilleux, J. Fürnkranz, and M. Scholz. From local patterns to global models: The LeGo approach to data mining. In *Workshop Proceedings: From Local Patterns to Global Models (held in conjunction with ECML/PKDD-08)*, volume 8, pages 1–16, 2008.

M. Kulessa and E. Loza Mencía. Dynamic classifier chain with random decision trees. In *Proceedings of the 21st International Conference on Discovery Science*, pages 33–50, 2018.

M. Kulessa, A. Molina, C. Binnig, B. Hilprecht, and K. Kersting. Model-based approximate query processing. *arXiv:1811.06224*, 2018.

M. Kulessa, B. Hilprecht, A. Molina, K. Kersting, and C. Binnig. Towards model-based approximate query processing. In *Working Notes of the 1st International Workshop on Applied AI for Database Systems and Applications (held in conjunction with VLDB 2019)*, 2019a.

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Improving outbreak detection with stacking of statistical surveillance methods. In *Workshop Proceedings: Epidemiology Meets Data Mining and Knowledge Discovery (held in conjunction with ACM SIGKDD 2019)*, 2019b.

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Improving the fusion of outbreak detection methods with supervised learning. In *Proceedings of the 16th International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 55–66, 2020.

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Revisiting non-specific syndromic surveillance. In *Proceedings of 19th International Symposium on Intelligent Data Analysis*, pages 128–140, 2021a.

M. Kulessa, E. Loza Mencía, and J. Fürnkranz. A unifying framework and comparative evaluation of statistical and machine learning approaches to non-specific syndromic surveillance. *Computers (Special Issue: Artificial Intelligence for Health)*, 10(3):32, 2021b.

M. Kulessa, B. Wittelsbach, E. Loza Mencía, and J. Fürnkranz. Sum-product networks for early outbreak detection of emerging diseases. In *Proceedings of the 19th International Conference on Artificial Intelligence in Medicine*, pages 61–71, 2021c. *(nominated for best student paper award).*

M. Kulldorff, F. Mostashari, L. Duczmal, W. Katherine Yih, K. Kleinman, and R. Platt. Multivariate scan statistics for disease surveillance. *Statistics in Medicine*, 26(8):1824–1833, 2007.

R. Lall, J. Abdelnabi, S. Ngai, H. B. Parton, K. Saunders, J. Sell, A. Wahnich, D. Weiss, and R. W. Mathes. Advancing the use of emergency department syndromic surveillance data, New York City, 2012-2016. *Public Health Reports*, 132(1): 23S–30S, 2017.

E. Lau, B. Cowling, L. Ho, and G. Leung. Optimizing use of multistream influenza sentinel surveillance data. *Journal of Emerging Infectious Diseases*, 14:1154–1157, 2008.

J. T. Leek and J. D. Storey. A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences*, 105(48):18718–18723, 2008.

E. L. Lehmann and J. P. Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2005.

Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu. Copod: copula-based outlier detection. In *Proceedings of the 20th IEEE International Conference on Data Mining*, pages 1118–1123, 2020.

M. A. Lindquist and A. Mejia. Zen and the art of multiple comparisons. *Psychosomatic Medicine*, 77(2):114–125, 2015.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 413–422, 2008.

Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528, 2020.

W. B. Lober, L. J. Trigg, B. T. Karras, D. Bliss, J. Ciliberti, L. Stewart, and J. S. Duchin. Syndromic surveillance using automated collection of computerized discharge diagnoses. *Journal of Urban Health*, 80(1):i97–i106, 2003.

J. S. Lombardo and D. L. Buckeridge. *Disease surveillance: A public health informatics approach*. John Wiley and Sons, 2012.

T. Lotze, G. Shmueli, and I. Yahav. Simulating multivariate syndromic time series and outbreak signatures. *Robert H. Smith School Research Paper No. RHS-06-054*, 2007.

E. Loza Mencía, M. Kulessa, S. Bohlender, and J. Fürnkranz. Tree-based dynamic classifier chains. *Machine Learning*, 2022. To appear.

H. Ma, A. I. Bandos, H. E. Rockette, and D. Gur. On use of partial area under the ROC curve for evaluation of diagnostic performance. *Statistics in Medicine*, 32(20): 3449–3458, 2013.

K. D. Mandl, J. M. Overhage, M. M. Wagner, W. B. Lober, P. Sebastiani, F. Mostashari, J. A. Pavlin, P. H. Gesteland, T. Treadwell, E. Koski, et al. Implementing syndromic surveillance: A practical guide informed by the early experience. *Journal of the American Medical Informatics Association*, 11(2):141–150, 2004.

M. L. McHugh. The Chi-square test of independence. *Biochemia Medica*, 23(2): 143–149, 2013.

Z. Mnatsakanyan, H. Burkom, J. Coberly, and J. Lombardo. Bayesian information fusion networks for biosurveillance applications. *Journal of the American Medical Informatics Association*, 16(6):855–863, 2009.

A. Molina, S. Natarajan, and K. Kersting. Poisson sum-product networks: A deep architecture for tractable multivariate poisson distributions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017.

A. Molina, A. Vergari, N. Di Mauro, S. Natarajan, F. Esposito, and K. Kersting. Mixed sum-product networks: A deep architecture for hybrid domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

C. Molnar, G. Casalicchio, and B. Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431, 2020.

D. Montgomery, E. Peck, and G. Vining. *Introduction to linear regression analysis*. John Wiley and Sons, 2001.

D. M. Morens and A. S. Fauci. Emerging infectious diseases: Threats to human health and global stability. *PLOS Pathogens*, 9(7):e1003467, 2013.

P. Muchaal, S. Parker, K. Meganath, L. Landry, and J. Aramini. Big data: Evaluation of a national pharmacy-based syndromic surveillance system. *Canada Communicable Disease Report*, 41(9):203, 2015.

K. P. Murphy. *Machine learning: A probabilistic perspective*. MIT Press, 2012.

M. L. Nolan, H. V. Kunins, R. Lall, and D. Paone. Developing syndromic surveillance to monitor and respond to adverse health events related to psychoactive substance use: Methods and applications. *Public Health Reports*, 132(1):65S–72S, 2017.

A. Noufaily, D. Enki, P. Farrington, P. Garthwaite, N. Andrews, and A. Charlett. An improved algorithm for outbreak detection in multiple surveillance systems. *Statistics in Medicine*, 32(7):1206–1222, 2013.

A. Noufaily, R. Morbey, F. Colón-González, A. Elliot, G. Smith, I. Lake, and N. McCarthy. Comparison of statistical algorithms for daily syndromic surveillance aberration detection. *Bioinformatics*, 10(1), 2019.

P. K. Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10(2), 2009.

J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, 1988.

K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Proceedings of IEEE International Conference on Computer Vision Workshops*, pages 689–690, 2011.

M. Rapp, E. Loza Mencía, J. Fürnkranz, V.-L. Nguyen, and E. Hüllermeier. Learning gradient boosted multi-label classification rules. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 124–140, 2020.

M. Rapp, M. Kulessa, E. Loza Mencía, and J. Fürnkranz. Correlation-based discovery of disease patterns for syndromic surveillance. *Frontiers in Big Data*, 4, 2022.

S. Ray and D. Page. Multiple instance regression. In C. E. Brodley and A. P. Danyluk, editors, *Proceedings of the 18th International Conference on Machine Learning*, pages 425–432, 2001.

B. Y. Reis and K. D. Mandl. Time series modeling for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 3(1):1–11, 2003.

B. Y. Reis and K. D. Mandl. Syndromic surveillance: The effects of syndrome grouping on model accuracy and outbreak detection. *Annals of Emergency Medicine*, 44(3): 235–241, 2004.

B. Y. Reis, M. Pagano, and K. D. Mandl. Using temporal context to improve biosurveillance. *Proceedings of the National Academy of Sciences*, 100(4):1961–1965, 2003.

D. A. Reynolds. Gaussian mixture models. *Encyclopedia of Biometrics*, 741:659–663, 2009.

C. Robertson, T. A. Nelson, Y. C. MacNab, and A. B. Lawson. Review of methods for space–time disease surveillance. *Spatial and Spatio-Temporal Epidemiology*, 1 (2):105–116, 2010.

J. Roure, A. Dubrawski, and J. Schneider. A study into detection of bio-events in multiple streams of surveillance data. In *NSF Workshop on Intelligence and Security Informatics*, pages 124–133, 2007.

M. Salmon, D. Schumacher, and M. Höhle. Monitoring count time series in R: Aberration detection in public health surveillance. *Journal of Statistical Software*, 70 (10):1–35, 2016.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7): 1443–1471, 2001.

J. Shao. *Mathematical statistics*. Springer Science and Business Media, 2003.

G. Shmueli and H. Burkom. Statistical challenges facing early outbreak detection in biosurveillance. *Technometrics*, 52(1):39–51, 2010.

L. Simonsen, J. R. Gog, D. Olson, and C. Viboud. Infectious disease surveillance in the big data era: Towards faster and locally relevant systems. *Journal of Infectious Diseases*, 214(supplements 4):S380–S385, 11 2016.

X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, 2007.

K. A. Spackman. Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the 6th International Workshop on Machine Learning*, pages 160–163, 1989.

J. Suyama, M. Sztajnkrycer, C. Lindsell, E. J. Otten, J. M. Daniels, and A. B. Kressel. Surveillance of infectious disease occurrences in the community: An analysis of symptom presentation in the emergency department. *Academic Emergency Medicine*, 10(7):753–763, 2003.

G. Texier, R. Allodji, L. Diop, J. Meynard, L. Pellegrin, and H. Chaudet. Using decision fusion methods to improve outbreak detection in disease surveillance. *BMC Medical Informatics and Decision Making*, 19(1):38, 2019.

K. Ting and I. Witten. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271–289, 1999.

P. A. Trott. International classification of diseases for oncology. *Journal of Clinical Pathology*, 30(8):782, 1977.

I. Tsamardinos, E. Greasidou, and G. Borboudakis. Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Machine Learning*, 107(12): 1895–1922, 2018.

P. Velardi, G. Stilo, A. E. Tozzi, and F. Gesualdo. Twitter mining for fine-grained syndromic surveillance. *Artificial Intelligence in Medicine*, 61(3):153–163, 2014.

F. Vial, W. Wei, and L. Held. Methodological challenges to multivariate syndromic surveillance: A case study using Swiss animal health data. *BMC Veterinary Research*, 12(1):288, 2016.

V. Vovk and R. Wang. Combining p-values via averaging. *Biometrika*, 107(4):791–808, 2020.

G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016.

M. C. Whitlock. Combining probability from independent tests: The weighted Z-method is superior to Fisher's approach. *Journal of Evolutionary Biology*, 18(5): 1368–1373, 2005.

R. Winkelmann. *Econometric analysis of count data*. Springer Science & Business Media, 2008.

I. H. Witten and E. Frank. *Data mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers, 2005.

N. D. Wolfe, C. P. Dunavan, and J. Diamond. Origins of major human infectious diseases. *Nature*, 447(7142):279–283, 2007.

D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

W.-K. Wong, A. Moore, G. Cooper, and M. Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *Proceedings of the 18nth National Conference on Artificial Intelligence*, pages 217—223, 2002.

W.-K. Wong, A. Moore, G. Cooper, and M. Wagner. What's strange about recent events (WSARE): An algorithm for the early detection of disease outbreaks. *Journal of Machine Learning Research*, 6:1961–1998, 2005.

S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 78–87, 1997.

T.-S. J. Wu, F.-Y. F. Shih, M.-Y. Yen, J.-S. J. Wu, S.-W. Lu, K. C.-M. Chang, C. Hsiung, J.-H. Chou, Y.-T. Chu, H. Chang, et al. Establishing a nationwide emergency department-based syndromic surveillance system for better public health responses in taiwan. *BMC Public Health*, 8(1):1–13, 2008.

A. Wyner, M. Olson, J. Bleich, and D. Mease. Explaining the success of AdaBoost and Random Forests as interpolating classifiers. *Journal of Machine Learning Research*, 18(48):1–33, 2017.

C. Zhang and S. Zhang. *Association rule mining: Models and algorithms*. Springer-Verlag, 2002.

C. Zhou and R. C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.