Using Reinforcement Learning to Improve Clinical Decision Making in Neonatal Care

Dominique Doorhof

Faculty of Science, Information and Computing Sciences
Utrecht University, The Netherlands

A thesis presented for the degree of
Master of Science

July 5, 2018

Supervised by:

Dr. Matthieu Brinkhuis — Utrecht University

Dr. Martijn Ludwig — Deloitte.

MSc. Simone Cammel — LUMC Leiden University Medical Center

Contents

List of Figures

Chapter 1

Introduction

Health care is coming to a new era. Now that technology has advanced, able to handle large amounts of data, and we collect more and more biomedical data, new opportunities and challenges rise in health care research (Miotto et al., 2017). Examples of these biomedical data sources are clinical imaging, electronic health records (EHRs), genomes, and also wearable devices. This data can be used to develop reliable medical tools, for example decision support systems for physicians. Next to this application of the data, new discoveries in health care can be made by exploring the associations among all these data sources. One of the areas where health care is getting more advanced is the automated drug dosing of intensive care unit (ICU) patients. There are several issues that should be addressed and machine learning can be a great tool to do so. These issues are discussed in the following paragraphs.

High heterogeneity in patient response. Every human body is different and therefore every patient can respond differently to certain medication. The Target Controlled Infusion (TCI) system that is used to control the infusion rate of drugs relies on a precomputed drug-patient interaction model. These models, also known as pharmacokinetic/pharmacodynamic (PK/PD) models, characterize the distribution of the drug within the body (pharmacokinetics), as well as the effect of the drug (pharmacodynamics). PK/PD models are developed based on trials with patients that do not necessarily fit all the target patient's characteristics but only some patient-specific parameters, which can include gender, height, weight, and age, which have to be provided by the clinician (Moore et al., 2004). Although this system has showed a positive effect in comparison to manual control, for example at the infusion of Propofol for Direct Laryngoscopy and Bronchoscopy (Passot et al., 2002), there are a lot of improvements that can be made.

The use of TCI is not recommended for the paediatric population as there are still hardware limitations, lack of integrated PK/PD studies and target monitoring issues (Anderson, 2010; Anderson & Hodkinson, 2010). Currently, most of the TCI systems perform open-loop control, which means the system is not equipped with a feedback mechanism of the patient's reaction. Current research of closed-loop TCI systems is mostly about sedation of the patient and uses the bispectral index (BIS) as the control variable (Moore et al., 2004). In neonatal care closed-loop systems are used for oxygen titration to automatically stay within the target range of oxygen saturation measured with pulse oximetry (van Zanten, 2017). When it comes to a feedback system that includes the combined effect of multiple medications, more variables should be included that can provide feedback about the patient's response. By better phenotyping of patients, using a combination of multiple biomedical data sources, personalized treatments can be improved (Silverman & Loscalzo, 2013).

Mismanagement of drugs. Drug dosing is complex for adults, but even more complex for newborns. Newborns have a delayed absorption of gastric emptying, lower renal and liver activity than adults and a body composition of 80-90% water (Bressan et al., 2013). This makes that they react different to medicine than adults. There is a lack of evidence to support most of the medication use in neonates: often there is no reference standard for doses of off-label and unlicensed medication while they are being used for neonates (Chedoe et al., 2007). Next to this, neonates are not a homogeneous group, because they are born at different gestational ages and their weight and length vary widely. These issues can lead to mismanagement of drugs where the newborns receive an incorrect dose. This may cause short-term side effects and potentially has irreversible damage that has not been reported in the literature for lack of prospectively collected data (Bressan et al., 2013).

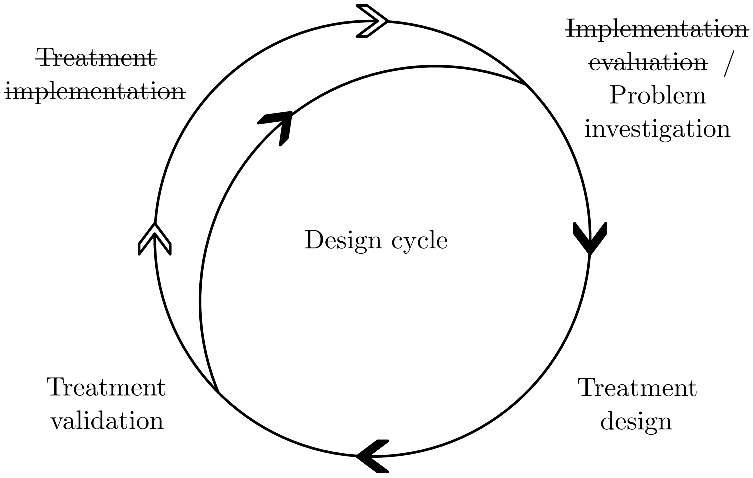

~~Treatment implementation~~
~~Implementation evaluation~~ / Problem investigation
Design cycle
Treatment validation
Treatment design

Figure 1.1: Design cycle (J. Wieringa, 2014)

High treatment costs. Next to the improvements to the patient's health the optimization of drug dosing also has the potential to lower costs. First, the treatment costs that include the amount of medicine used can be lowered. For example, the costs that are caused by the overdosing of patients that do not respond to medicine. One research mentioned that an optimal policy for erythropoietin (EPO) dosage can save between 100 and 200 Euro per patient per year (Martín-Guerrero et al., 2009). Secondly, the hospital invests a considerable amount of money in alleviating side-effects directly related to the treatment. Therefore, reducing side-effects by lowering medication dosing could save money. And finally, mismanagement of drugs can lead to unnecessarily extending a patient's length of stay in the hospital. All these factors can generously reduce costs for a hospital with thousands of patients a year.

It is clear there are multiple opportunities to use machine learning to improve the patient's quality-of-life and reduce costs of treatment in the hospital. Although these opportunities are applicable to machine learning in general, this research will use a specific type of machine learning: reinforcement learning. Reinforcement learning is a computational approach to learning by interacting with our environment, which is neither supervised or unsupervised. It can be thought of as the nature of learning for humans. The machine is not told what actions to take, but tries different actions and assesses the rewards of those actions. The ultimate goal for the machine is to take the actions that maximise the future reward. The reinforcement learning problem is extensively described by Sutton & Barto (1998) who are the primers in this field. Section 2.2 will explain reinforcement learning, and it's advantages and challenges.

Sepsis is a life-threatening complication of an infection, occurring when the body is trying to fight the infection. Adults and neonates on the intensive care unit are susceptible for sepsis as their body's defence system is affected. This research will focus on neonatal late-onset sepsis, which is explained in Section 2.1. The goal of this research is to improve the treatment of late-onset neonatal sepsis at the NICU department of the Leiden University Medical Centre. This research attempts to make improvements by earlier detection of sepsis and optimizing antibiotics dosing.

1.1 Research Questions

After having discussed problems around drug dosing of intensive care unit patients and previous research to the implementation of reinforcement learning at the intensive care unit to solve these issues, we introduce the following research question that is leading in this work:

Research Question: *How can reinforcement learning optimize the treatment policy of premature neonates in neonatal care?*

To answer this main research question five sub questions will be investigated. In this section the research methodology will be presented by using the design cycle as proposed by J. Wieringa (2014). The design cycle (shown in Figure 1.1) is part of the engineering cycle, a rational problem-solving process with five tasks, which will be described in hereafter. The first step of the design cycle is the *problem investigation*. The problem to be treated will be investigated using the following question:

Question 1: *What is the current treatment policy of premature newborns in neonatal care?*

First the stakeholders and their goals have to be identified. Thereafter, the problem is investigated, by describing the phenomena, defining how and why it is caused, and evaluating and explaining the effects on the stakeholders' goals. There are many ways to investigate the problem, of which the following will be used in this research:

- **Literature research**: extracting information from scientific, professional and technical literature. The research will be done by manual search, in the following topics: *Reinforcement Learning, Neonatal Intensive Care Unit, Personalized Treatment, Health Care Analytics, Biomedical Informatics, Drug Dosing Optimization.*
- **Case study**: one case is studied in depth, using multiple data sources, to obtain a detailed insight into the problem environment. In this research the case will be at the neonatology department of the Leiden University Medical Center. The prediction of sepsis occurrence at the neonatal intensive care unit will be investigated. The case study will be of a descriptive nature: it illustrates the problem that is occurring and discusses how the stakeholder perceive it. In order to examine the problem, two data sources will be collected:
    1. *Interviews*: a couple of experts will be asked questions about the area of interest in the research. The interview questions are based on the research question and will be open in order to get a better understanding of the phenomena.
    2. *Attendance at the nurse transfer meetings*: during this research the researcher will attend the meetings that clinicians have in the morning to discuss how the night shift went and what is planned for the day. By attending these meetings approximately once a week the researcher is able to gain knowledge about the daily issues the nurses at the neonatal intensive care unit face, without interfering with their daily routine.
    3. *Archichal Data*: data that is already collected by the organization. In this case the medical data about the patients, for example from monitoring equipment or laboratory results, that are stored by the hospital. The data will be anonymized in order to protect the privacy of the patients. The data is described in **??**.

The next phase of the design cycle is treatment design. During this phase multiple artifacts are designed in order to answer the following research questions:

Question 2: *How can reinforcement learning analyse the current treatment policy?*

Question 3: *How can reinforcement learning be used to optimize the treatment policy?*

Question 4: *How does the learned policy compare to the current policy?*

In this research two machine learning algorithms will be presented and compared. The process of designing a machine learning algorithm can be visualized by the CRISP-DM Reference Model (*CRoss-Industry Standard Process for Data Mining*, introduced by Shearer (2000)). Similar to the Design Cycle by Wieringa, the CRISP-DM model is circular and multiple iterations of the steps will be performed. Although some steps might seem similar, for example the *Treatment Validation* from the Design Cycle and the *Evaluation* step in the CRISP-DM model, the scope in which both models will be used is different. Where the Design Cycle is used as a model for the complete research, the CRISP-DM model describes only the processes within the Treatment Design phase of the Design Cycle.

The CRISP-DM model (Figure 1.2) defines the process into six phases (Shearer, 2000):

- *Business understanding.* In the first phase the problem determined in the *Problem Investigation* step of the design cycle has to be converted into a data mining problem. The treatment requirements will be defined, which are the desired properties of the to-be-designed treatment (as described by J. Wieringa (2014)). These can be functional requirements, which are requirements for desired functions of the artifact, or nonfunctional requirements, which are requirements that have a specified nonfunctional property. A nonfunctional property, sometimes called a quality property, is any property that is not a function, for example utility, accuracy, efficiency, security, reliability and usability. The business understanding is presented in Section 3.1.

- *Data understanding.* The data understanding phase consists of collecting the initial data, describing and exploring the data and verifying the quality of the data. For the data describing step the hospital will provide a database description and the support of a medical PhD that can explain the semantics of the data. The results of this step will be presented in Section 3.2.

- *Data preparation.* In this phase the data will be selected, cleaned, constructed, integrated and formatted. This is critical for health care data as it is highly heterogeneous, ambiguous, noisy and often incomplete (see Section 2.2.1). Selecting what data to use is important to reduce the dimensionality of the data. Data cleaning includes getting rid of errors and the removal of redundancies. Data preprocessing consists of renaming, rescaling, discretization, abstraction, aggregation and adding new attributes. These transformations can be automated. The results of this step will be presented in Section 3.2.

- *Modeling.* After the data is prepared the modeling phase starts. First, the appropriate modeling technique has to be selected which can satisfy the requirements. In this research reinforcement learning is the chosen type of machine learning. Different versions of reinforcement learning will be generated and assessed. During the modeling phase it might be necessary to step back to the data preparation phase, for example to adjust the data formatting. The modeling phase is described in Section 3.3.

- *Evaluation.* This step assesses if the model meets the business objectives that were discovered in the first phase (and presented in Section 3.1). Next to this, the model might be tested on real-world applications to see if implementation would positively influence the business problem. The whole process has to be reviewed to see if no mistakes are made when creating the model. Evaluation of the process can be found in Chapter 4.

- *Deployment.* The last phase of the data mining process is the deployment of the created model. The size and importance of this phase in the process depends on the requirements. In this step it has to be planned how to deploy the model, and how to monitor and maintain the deployment after implementation. This should all be recorded in a final report. Possible deployment of the model is presented in Section 5.1.4.

During the last phase of the design cycle, *treatment validation*, the following research question will be investigated:

Question 5: *How can the reinforcement learning implementation improve the clinicians decision making?*

The validation of a treatment is to assess if the designed artifact would contribute to the stakeholders' goals that are defined during the problem investigation phase. The goal is to predict what effect the artifact has on it's environment, before implementing it in the real-world. There are many methods to validate an artifact:

- *Expert Opinion*: The artifact is presented to a panel of experts, who will conceptualize how the artifact will interact with the problem and predict what effects it would have.

- *Single-Case Mechanism Experiment*: The artifact will be validated by feeding it test scenarios (from archichal data) and observing the responses.

- *Technical Action Research*: The artifact is validated like a single-case mechanism experiment, but in a real-world scenario to help the client.

- *Statistical Difference-Making Experiments*: The artifact is validated by comparing average outcome of treatments to different samples.

For this particular research the first two are considered to be within the scope of this research. In Chapter 5 it will be explained why these validation methods are different for reinforcement learning in

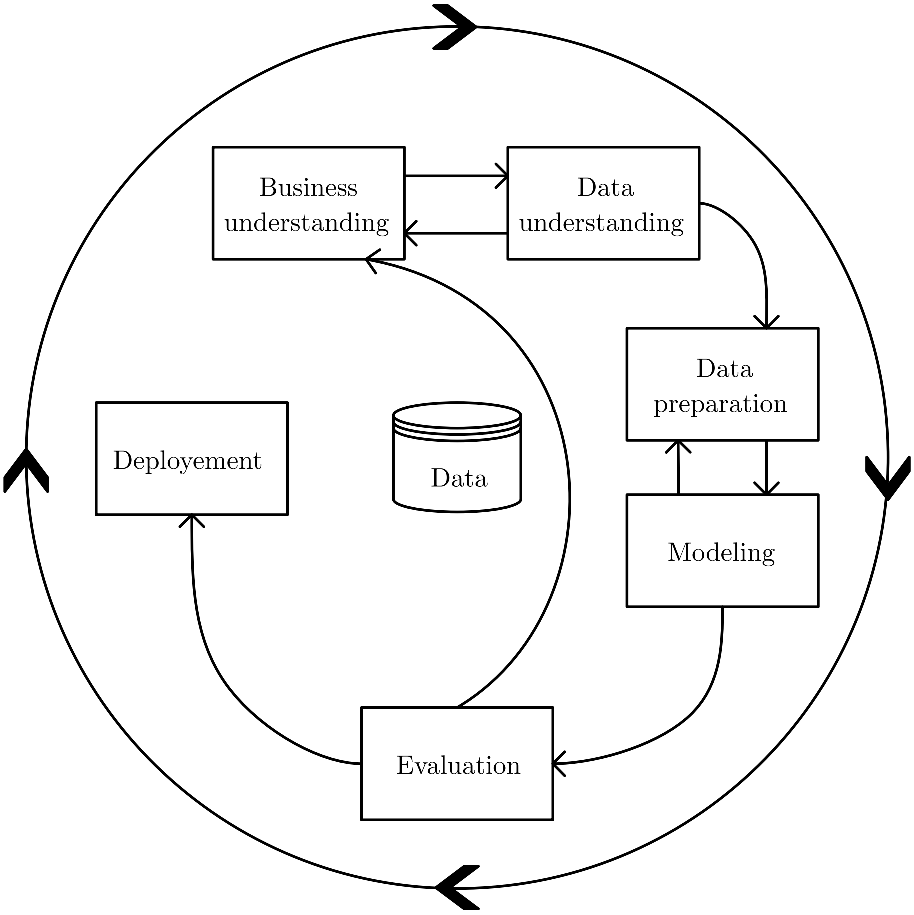


Figure 1.2: CRISP-DM Reference Model (Shearer, 2000)

relation to other machine learning algorithms or predictive models. The latter two methods can only be conducted if the first two methods indicate the algorithm generates a positive effect on the problem environment.

The next phase of the full engineering cycle is the *treatment implementation*, where the problem is to be treated with one of the designed artifacts. This phase should not be confused with the model implementation phase of the CRISP-DM process. The implementation of a treatment is defined as "the application of the treatment to the original problem context", in this case the usage of the machine learning algorithm in practice.This is not feasible, as this research investigates a problem that affects human beings in a hospital setting. Therefore, the algorithm has to be extensively tested before it can be implemented, which is not part of the scope of this research project.

The final phase is the *implementation evaluation* in which it is evaluated wheter the treatment has been successful. This can be investigated the same way the problem investigation was done, by letting the stakeholders in the field assess how the treatment influences the real-world problem. As the treatment will not be implemented in practice, this step will be left out as well.

This research project will be restricted to the first three tasks and will therefore use the design cycle (shown in Figure 1.1) instead of the full engineering cycle.

Chapter 2

Background and Related Work

This chapter will discuss literature on the subjects that are relevant for this research:. The first section investigates the problem to be treated in the neonatal intensive care unit (NICU). The second section will explain the type of machine learning we will use to treat the problem: reinforcement learning (RL). Thereafter we mention related work that used reinforcement learning in healthcare. Finally, we introduce literature about clinical decision support systems (CDSS).

2.1 Neonatal Intensive Care Unit

The Neonatal Intensive Care Unit takes care of preterm infants, who are born before the gestational age of 37 weeks, and sick a-term born newborns that need hospitalization. According to the World Health Organization (2017) every year approximately fifteen million preterm babies are born worldwide, which is more than one in ten babies . For the last 15 years in The Netherlands on average 7 to 8 % of all newborns are born preterm (De Staat van Volksgezondheid en Zorg, 2017). Since 2010 newborns from the gestational age of 24 weeks are treated according to the *Perinatal Policy at Extreme Preterm Birth* (Nederlandse Vereniging voor Kindergeneeskunde, 2010).

The first two years. Currently a heated discussion is going on whether extreme premature newborns of the age of 24 or 25 weeks should be treated, after research showed the development of the 185 preterm newborns that were born in the first year after the new policy was introduced. This research (De Kluiver et al., 2013), involving all ten NICUs in The Netherlands, was first published in 2013 and showed that the survival rate of the group of the 185 preterm newborns was 43% at 24 weeks and 61% at 25 weeks. After two years the researchers did a follow-up research (Aarnoudse-Moens et al., 2017) where they tested the development of 78 of the 95 infants that were still alive at the age of two years. Overall, only one fourth of the 185 preterm born infants grew up to be a healthy two year old without any impairments. Due to medical technological advancements preterm newborns can be treated at a younger gestational age, and chances of survival have grown during the last couple of years, but still doctors cannot predict which newborn will survive and if they will grow up to be a healthy child.

Adult life. In The Netherlands there is a long term project running for over 30 years called POPS (Project On Preterm and Small for gestational age infants) (TNO, 2017). A cohort of 1.338 infants born in 1983 at the gestational age of less than 32 weeks and/or with a weight less than 1500 grams has been investigated after birth and at the age of 19, 28 and 30+ years. Some of the conclusions of the POPS-19 research are: (1) one third of the original group from 1983 died before reaching the age of 19, (2) one third of the survivors has severe problems, one third has small problems and one third has no problems as a consequence of the prematurity, and (3) a quarter of the surviving group had to follow special education. At this moment the research group is applying for funding to perform the 30+ study.

2.1.1 Sepsis

“The term neonatal sepsis is used to designate a systemic condition of bacterial, viral, or fungal (yeast) origin that is associated with haemodynamic changes and other clinical manifestations and results in substantial morbidity and mortality” (Shane et al., 2017). Neonatal sepsis can be classified as either early-onset sepsis, which usually appears within the first 72 hours of life, or late-onset sepsis (LOS), which occurs beyond 3 to 7 days of age. The focus in this research will be on the prediction of late-onset sepsis in order to timely start the treatment and limit short and long term effects.

Early-onset sepsis. An early-onset sepsis is usually caused by bacteria in the placenta or in the uterus from the vaginal environment following membrane rupture, or by pathogenix bacteria during the passage through the birth canal. The most common bacteria associated with EOS are *Streptococcus agalactiae* (GBS) and *Escherichia coli* (Shane et al., 2017).

Late-onset sepsis (LOS). Late-onset sepsis is attributed to organisms acquired from interaction with the hospital environment. The main causative micro-organisms are Gram-positive organisms, including *coagulase-negative staphylococci* and *streptococci.* LOS occurs more often for premature born infants, as they usually require a longer hospitalization, more invasive interventions, surgery and respiratory support. Because of the strong affects of sepsis on neonates it is important to recognise developing sepsis early. The frequency and severity of apnea (see Section A.1.1) is mentioned in research as one of the signs of a developing sepsis, but this is not a good predictor as it can be also present in non-septic preterm infants with other complications. Next to this, the vital signs that can imply sepsis but not exclusively are respiratory rate, temperature, blood pressure and heart rate.

Proven vs Clinical sepsis. Within the definition of late-onset sepsis, there is a difference between clinical sepsis and culture-proven sepsis. In other research (Griffin et al., 2007) proven sepsis is defined as "clinical signs of sepsis and a positive blood culture prompting five or more days of antibiotic therapy" and clinical sepsis as "clinical signs of sepsis with a negative blood culture prompting five or more days of antibiotic therapy". In this research they defined a *Clinical Illness Score* to identify clinical signs of sepsis. The candidate findings that appeared in the final score were:

- Severe apnea requiring positive pressure ventilation or 50% increase in apneic episodes over 24h in an extubated infant stable for three days;
- increased ventilatory support and FiO_2 by 25%;
- temperature instability ($> 38°C$ or $< 36.2°C$) twice in 8 hours;
- lethargy or hypotonia;
- feeding intolerance (feedings held for $> 24h$) in an infant tolerant of advancing or full feeds for 3 days;
- immature/total neutrophil (I:T) ratio > 0.2;
- white blood cell count $> 25,000$ or $< 5,000/mm^3$;
- hyperglycemia ($>$ 180 mg/dL).

Sepsis protocol (LUMC). The NICU department of the LUMC has a written treatment policy, visualized in two charts, for the use of antibiotics when an infection is suspected. The first part of the treatment is deciding which medication has to be used, based on how long the infant has been hospitalized and additional research. This process is visualized in a flow chart (Figure 2.1). After two days the blood cultures are evaluated in order to decide whether to continue the antibiotics or end the treatment (Figure 2.2).

2.1.2 Sepsis prediction

Using different machine learning algorithms to predict late-onset sepsis from off-the-shelf medical data has showed to be successful (Mani et al., 2014). The predictive models developed exceeded the treatment sensitivity and specificity of clinicians.

Heart Rate Characteristics (HRC). Multiple studies found that reduced variability and transient decelerations of the heart rate both indicate a high chance of sepsis (Moore et al., 2011; Griffin et al., 2007). The heart rate characteristics index (HRC-index) takes the variability and decelerations of the heart rate into account and was used to calculate the risk of a neonate developing sepsis in the next 24 hours. Looking at the HRC-index is not fully reliable though, since not only infectious causes (including

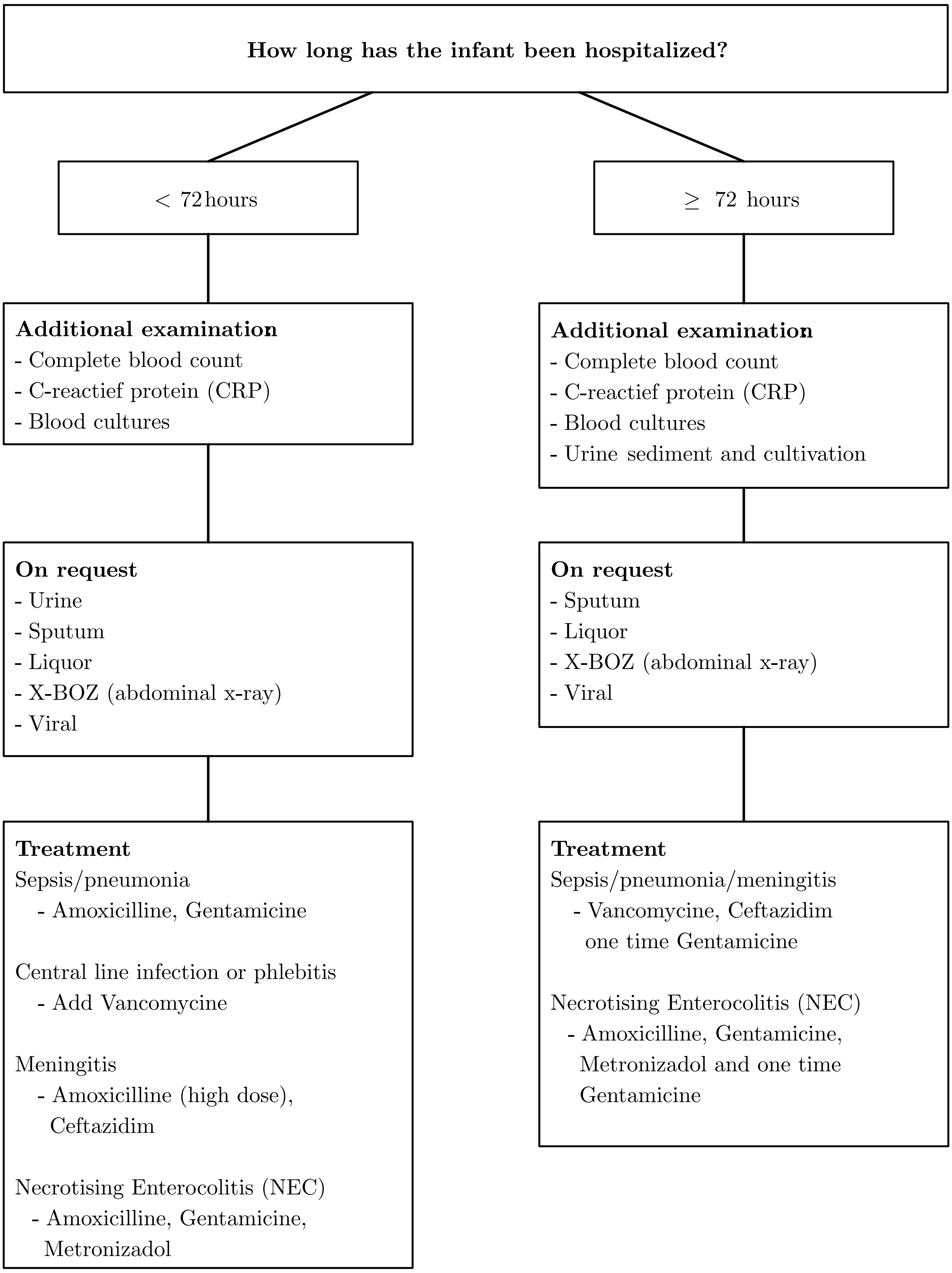

How long has the infant been hospitalized?
< 72 hours
≥ 72 hours
Additional examination
- Complete blood count
- C-reactief protein (CRP)
- Blood cultures
Additional examination
- Complete blood count
- C-reactief protein (CRP)
- Blood cultures
- Urine sediment and cultivation
On request
- Urine
- Sputum
- Liquor
- X-BOZ (abdominal x-ray)
- Viral
On request
- Sputum
- Liquor
- X-BOZ (abdominal x-ray)
- Viral
Treatment
Sepsis/pneumonia
- Amoxicilline, Gentamicine
Central line infection or phlebitis
- Add Vancomycine
Meningitis
- Amoxicilline (high dose), Ceftazidim
Necrotising Enterocolitis (NEC)
- Amoxicilline, Gentamicine, Metronizadol
Treatment
Sepsis/pneumonia/meningitis
- Vancomycine, Ceftazidim one time Gentamicine
Necrotising Enterocolitis (NEC)
- Amoxicilline, Gentamicine, Metronizadol and one time Gentamicine

Figure 2.1: Antibiotics policy

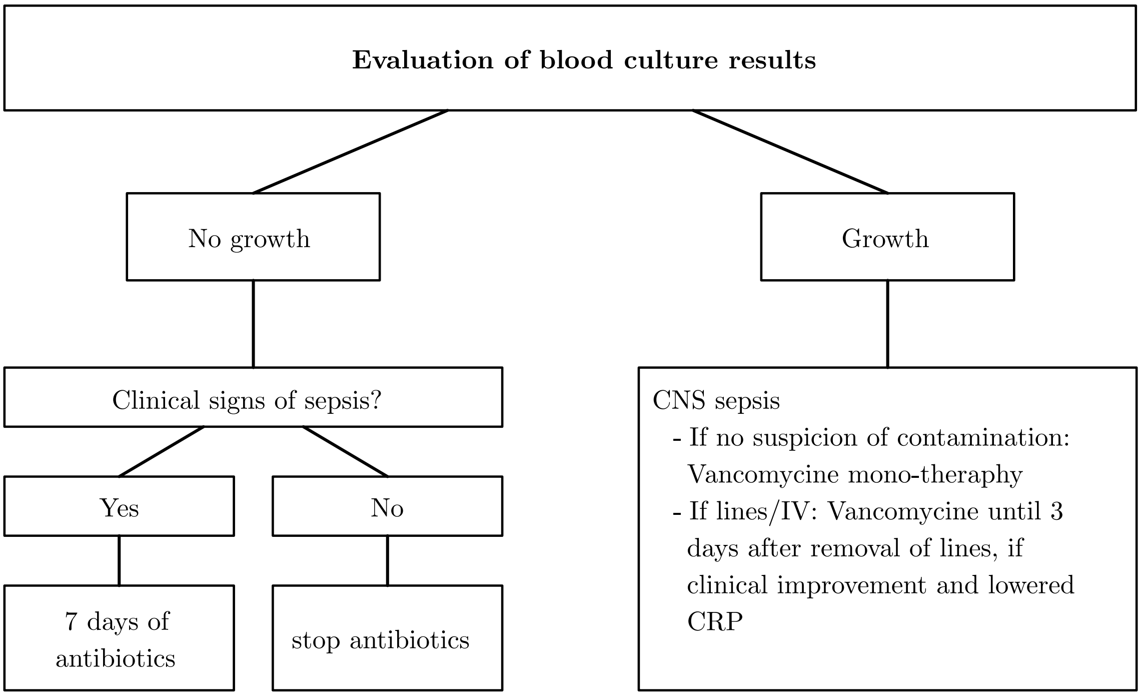

Evaluation of blood culture results
No growth
Growth
Clinical signs of sepsis?
Yes
No
7 days of antibiotics
stop antibiotics
CNS sepsis
- If no suspicion of contamination: Vancomycine mono-theraphy
- If lines/IV: Vancomycine until 3 days after removal of lines, if clinical improvement and lowered CRP

Figure 2.2: Evaluation of blood culture results

LOS) are associated with a high score on the HRC-index. Other causes for a high HRC-index are surgery, acute respiratory deterioration without infection, or no apparent clinical correlation at all (Gilfillan & Bhandari, 2017).

Oxygen saturation. Preterm infants often need supplemental oxygen for a prolonged period. Monitoring the arterial oxygen saturation is mostly performed using pulse oximetry, by analysing the saturation (SpO_2). Because the therapeutic ranges for oxygen therapy in preterm infants are very small, preterm infants are regularly exposed to hypoxemia or hyperoxemia (van Zanten et al., 2015). *Hypoxemia* (or *hypoxia*) is a decrease in blood saturation (SpO_2) of $\leq 80\%$ for ≥ 10 seconds. It is called *hyperoxemia* (or *hyperoxia*) when the blood saturation (SpO_2) is $\geq 95\%$ for ≥ 10 seconds. Hypoxemia leads to an increased risk of several morbidities, including retinopathy of prematurity (ROP), impaired growth, long term cardio-respiratory instability, and adverse neurodevelopmental outcome. Hyperoxemia increases the risk of high oxygen levels, which is toxic to cells and an important risk factor for the development of bronchopulmonary dysplasia and ROP (Saugstad & Aune, 2011), and is associated with cerebral palsy (Askie et al., 2011). Therefore, the target range for SpO_2 in preterm infants is usually set at $85\% - 95\%$ (van Zanten et al., 2015).

The fraction of inspired oxygen (FiO_2) is manually or automatically titrated to maintain the SpO_2 within the target range, while trying to avoid hypoxemia and hyperoxemia. However, premature infants frequently have fluctuations in SpO_2 due to respiratory instability and immaturity, and require continuous titration of FiO_2 Claure & Bancalari (2015). Several studies have shown that automatic FiO_2 control improved the time within the target range by reducing the occurrence and duration of hyperoxemia, but it has only little effect in reducing hypoxemia (van Zanten et al., 2015; Van Kaam et al., 2015). Periods with hyperoxemia occur mostly when oxygen is increased for ABCs (occurrence of Apnoea, Bradycardia, and Cyanosis, which are explained in Appendix A). In those cases, the hyperoxemic periods last longer than the duration of bradycardia and hypoxemia. A study tried to lower the periods of hypoxemia by narrowing the target range from $85\% - 95\%$ to $90\% - 95\%$. They found an increase in median SpO_2 and a rightwards shift in the distribution of SpO_2. However, no change was found in time spent between 90% and 95%, and in frequency and duration of hypoxemic events (van Zanten et al., 2017).

Since nearly all preterm infants need supplemental oxygen, the SpO_2 is constantly measured. When a neonate is developing sepsis, often the frequency of apnoea episodes increases. When this happens, the SpO_2 will decrease more often than normally. Since the SpO_2 is constantly monitored, fluctuations in SpO_2 may be a predictive factor for late onset sepsis.

Artemis project. One of the first and biggest data analytics projects in neonatal care is Artemis (Catley et al., 2010), which is a framework for the real-time analysis of times series physiological data streams from multiple devices for multiple patients. It employs IBM's InfoSphere Streams, which is a software platform that enables the development and execution of applications that process information of multiple streams of high volume and high rate data. It combines data streams from physical monitor devices as well as from the hospitals Clinical Information Management System (CIMS), the Electronic Health Record (EHR) and Laboratory Information Systems (LIS). This platform can be used to detect clinically significant conditions based on real-time and retrospective analysis in order to support clinical decision making. In past research it has been used to detect apnea (Catley et al., 2010), changes in sleep-wake cycling (Eklund et al., 2014), retinopathy of prematurity (Courtney et al., 2013), neonatal spells (Thommandram et al., 2014), late-onset neonatal sepsis (McGregor et al., 2012), and pain (Naik et al., 2013).

Complications of preterm birth. There are many complication that newborns can have as a result of preterm birth. Appendix A explains some complications to the respiratory system and the cardiovascular system, and also describes the most common treatments for these complications.

2.2 Reinforcement Learning

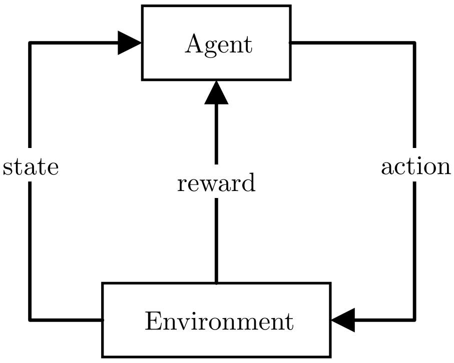


Figure 2.3: Reinforcement Learning

The RL problem can be represented in a diagram which contains five elements (Figure 2.3). The *agent* is the learner that selects the *actions* to take. It interacts with the *environment*, which contains everything outside the agent. From the environment the agent receives a numerical *reward* and a new representation of the environment's *state*. Next to the agent and the environment, there are four main elements of a reinforcement learning system:

- *Policy*: A stochastic rule by which the agent selects actions as a function of states, or the agent's behaviour function. It corresponds to what for humans would be called a set of stimulus-response rules or associations.

- *Reward function*: It maps states with rewards of being in that state, indicating the intrinsic desirability of the state in an immediate sense. In the biological system this relates to pleasure and pain. The policy can be dependent of the reward function.

- *Value function:* A prediction of future reward, or the amount of reward an agent can expect to accumulate over the future starting from that state. It indicates the long-term desirability of states after taking into account the states that are likely to follow.

- *Model*: mimics the behaviour of the environment. It predicts the next state and next reward given a state and action. Models are used for planning, which means deciding the course of action by considering future states before they happen.

When implementing reinforcement learning a couple of choices have to be considered, which will be discussed in the following sections.

Exploration vs Exploitation

One of the challenges of reinforcement learning is the trade-off between exploration and exploitation (Sutton & Barto, 1998). *Exploitation* is making the best decision given the current information the agent has retrieved using the *greedy* strategy. But how does it know that there might not be a better decision, given that he did not exhaust all of the possible options (which might not be possible in problems with a large action and state space set)? It doesn't, therefore it also has to *explore* other options, because the best long-term strategy may involve short-term sacrifices. The agent has to gather sufficient information to make the best overall decision. One approach to handle this problem is the *epsilon-greedy* strategy, which selects a random action on a fraction 'epsilon' of the time steps. The balance between exploration and exploitation is important in the case of modelling a medication treatment as the best policy does not only rely on the patient's outcome (improving health), but also on the amount of medication given to the patient. As medication can have side effects one of the goals is to give the lowest amount of medication that would still improve the patient's health.

On-policy vs Off-policy

An *on-policy* method estimates the value of a policy while using it for control, while in *off-policy* methods these are separates (Sutton & Barto, 1998). In off-policy methods the estimation policy, the policy that is evaluated and improved, can be unrelated to the policy used to generate behavior, called the *behavior policy.* The advantage of this is that the behavior policy can continue to explore all possible action even if the estimation policy is deterministic. In this research both on-policy and off-policy methods are used. The on-policy method is used to evaluate the clinicians policy based on historical data. For this we use *Sarsa*, an on-policy temporal difference control method (Sutton & Barto, 1998). Sarsa stands for $s_t, a_t, r_t, s_{t+1}, a_{t+1}$), which reflects the value (reward) of the transition from state-action pair to state-action pair. For the off-policy TD control algorithm we use Q-learning as developed by Watkins & Dayan (1992). In this case the learned action-value function Q directly approximates the optimal value function regardless of the actions taken in the current policy that is being followed.

Discrete vs Continuous Spaces

Estimates of value functions for discrete state and action spaces are represented as a table with one entry for each state-action pair. This is limited to tasks with a small number of states and actions, but cannot be used for when action or state spaces are continuous. Building large tables would require a lot of memory and time to fill them in, and the learner will encounter states it has not experienced before. Therefore some sort of generalization has to be formed, where previously experienced states can produce a good approximation of ones that have not been seen yet. A method to generalize from examples that can be used in RL is *supervised-learning function approximation*, where each backup is treated as a training example. Using continuous state-space models to capture a patient's physiological state allows for discovery of high-quality treatment policies (Raghu et al., 2017).

Online vs Offline

There are two different ways of making updates when computing the value function. In *online* updating, the updates are done during the episode as soon as the increment is computed. In *offline* updating, on the other hand, the increments are accumulated 'on the side' and not used to change value estimates until the end of the episode. Applying reinforcement learning to optimize treatments using offline sampled data can be a challenge, as models can only be fit to a retrospective dataset (Raghu et al., 2017). Exploration of state spaces is limited to those that already exist in the dataset, which makes learning the truly 'optimal' policy for a new patient difficult.

Feature selection vs Autoencoding

An important part of developing a reinforcement learning model, or any model, is the selection of features that are the best predictors. An autoencoder is a neural network that represents the data as a function of the input data. It is forced to prioritize the features of the input data that best represent the data and are therefore most useful (LeCun et al., 2015). Especially in healthcare this is a challenge as the patient's state can be represented as a high dimensional continuous vector without clear structure (Raghu et al., 2017). One approach is that of the *Deep patient*, where the patient is represented by a set of general features, which are inferred automatically from a large-scale EHR database processed by a deep neural network composed of a stack of denoising autoencoders (Miotto et al., 2016).

2.2.1 Challenges

When applying machine learning to health care, there are some challenges to face (Miotto et al., 2017). Some of the challenges are not specific to the domain of health care, but are also common for other industries, for example the challenge of *data volume* and *temporality.* Next to this, *data quality* is an important challenge as health care data are highly heterogeneous, ambiguous, noisy and incomplete. Two challenges that require specific attention when handling healthcare data are:

Domain complexity. Problems in biomedicine and health care are complicated. The diseases are highly heterogeneous and for most of the diseases there is still no complete knowledge on their causes and how they progress. There are four challenges when modelling patient-level healthcare time series data (Pham et al., 2016):

- *Long term dependencies:* future illness may depend on historical illness, and often effects of treatment cannot be immediately detected.
- *Representation of admission:* an admission episode consists of a variable-size discrete set containing diagnoses and interventions.
- *Episodic recording and irregular timing:* hospital admissions vary in size and only portray a specific time episode in a patients life, ranging from days to weeks.
- *Confounding interactions between disease progression and intervention:* medical records are a mixture of the course of illness, the developmental and the intervening processes.

Interpretability. In health care, not only the quantitative algorithmic performance is important, but also the reason why the algorithm works is relevant. This can be hard to explain when using deep learning models that are often described as 'black boxes'. The interpretability is crucial when convincing a clinician to take actions recommended by a predictive system. Clinicians are held accountable by law and by the GMC (Rocheteau, 2012): they can only carry out a recommendation if they can justify that decision to themselves. Therefore the clinical decision support system should be able to communicate the reasoning behind their recommendation.

2.2.2 Advantages of RL

There are a couple of advantages of reinforcement learning that can address some of these challenge mentioned before and that make it suitable to use with health care data.

Long term effect. RL can handle sequential data where there is no one-to-one correspondence between actions and outcomes. This makes reinforcement learning well-suited for the analysis medication dosing data, where multiple treatments are performed and effectiveness cannot be immediately detected (Nemati et al., 2016). The optimization process is made over sequences of doses instead of isolated doses, which is crucial to include the drug long-term effects (Escandell-Montero et al., 2014).

No ground truth needed. The RL agent can learn from suboptimal examples, because it learns in a natural way where it does not require prior knowledge of optimal performance in the form of a model (Martín-Guerrero et al., 2009). No ground truth is needed of a what is a 'good' treatment (Raghu et al., 2017). Because the agent does not learn via 'scripted' observation/action sequences, the likelihood of developing a brittle, overtrained controller is reduced, resulting in more generalized control that is better equipped to handle uncertainty and variability (Moore et al., 2011).

2.3 Reinforcement Learning in Healthcare

This approach is inspired by research paper *Continuous State-Space Models for Optimal Sepsis Treatment - a Deep Reinforcement Learning Approach* (Raghu et al., 2017). In this paper a new approach is proposed to deduce optimal treatment policies for septic patients by using continuous state-space models and deep reinforcement learning. The network architecture used in this model is a Dueling Double-Deep Q Network (Dueling DDQN) with two different latent state representations as inputs: one created by ordinary autoencoders and one created by sparse autoencoders. For the action spaces they defined a 5 x 5 action space for the medical interventions covering the space of intravenous (IV) fluid (volume adjusted for fluid tonicity) and maximum vasopressor (VP) dosage in a given 4 hour window. Evaluation of the proposed model on past ICU patient data showed that the model could reduce patient mortality in the hospital by 1.8 - 3.6%, over observed clinical policies, from a baseline mortality of 13.7%. It must be noted that this research is applicable to intensive care unit patients of at least 15 years old and does not include the neonatal patients.

Another research that discusses the implementation of reinforcement learning to the intensive care unit is *A Reinforcement Learning Approach to Weaning of Mechanical Ventilation in Intensive Care Units* (Prasad et al., 2017). This work aims to develop a decision support tool that uses available patient information to predict time-to-extubation readiness and recommend a personalized regime of sedation dosage and ventilator support. They used off-policy reinforcement learning algorithms to determine the best action at a given patient's state from sub-optimal historical ICU data. They compared treatment policies from fitted Q-iteration with extremely randomized trees and with feed forward neural networks. The policies learnt show promise in recommending weaning protocols with improved outcomes, in terms of minimizing rates of re-intubation and regulating physiological stability.

In *Optimal Medication Dosing from Suboptimal Clinical Examples: A Deep Reinforcement Learning Approach* (Nemati et al., 2016) they present a clinician-in-the-loop sequential decision making framework, which provides an individualized dosing policy adapted to each patient's evolving clinical phenotype. They employed retrospective data from the publicly available MIMIC-II intensive care unit database, and developed a deep reinforcement learning algorithm that learns an optimal heparin dosing policy from sample dosing trails and their associated outcomes in large electronic medical records. Using separate training and testing datasets, the model was observed to be effective in proposing heparin doses that resulted in better expected outcomes than the clinical guidelines.

No existing research can be found that implements reinforcement learning in neonatal care, therefore this research will be the first in the field.

2.4 Clinical Decision Support System

As machine learning techniques have evolved and the quantity of healthcare data is growing exponentially, the implementation of clinical decision support systems (CDSS) in our hospitals and healthcare institutions seems like a logical next step. The current CDSSs are based on statistical analysis or decision trees and cannot cope with the full complexity of handling long decision sequences. There is a need for more advanced CDSSs (Rocheteau, 2012) that will be able to reduce the workload for clinicians by taking over certain tasks, so clinicians can focus on the tasks that require human cognitive and social skills. One possible method that can improve CDSSs is reinforcement learning which is able to analyse problems that involve sequences of decisions where the effects of certain actions are not directly visible in the data. As mentioned before RL is applied to disciplines as gaming and robotics, but the examples in healthcare analytics are sparse.

For the development of a successful CDSS three steps are crucial (Rocheteau, 2012): (1) Development of a realistic simulation for exploring healthcare policies, (2) compatibility with EHR software, and (3) acceptance in the medical community. The first step will be addressed in this research by applying a reinforcement learning model to healthcare data and assessing the performance with clinicians. Compatibility with EHR software is possible, but not yet implemented in all hospitals because of a lack of resources and funding. To encourage the further integration of clinical decision support systems the third step has to be addressed: the use of artificial intelligence has to be accepted in the medical community. A major obstacle is the view that CDSSs are 'black boxes': the clinicians don't understand how they work and therefore cannot base their decision on the CDSS.

In *A Framework to Design Successful Clinical Decision Support Systems* (Zikos, 2017) seven principles are mentioned to consider when designing a clinical decision support system. A CDSS should...

1. mimic the cognitive process of clinical decision makers
2. provide recommendations with longitudinal insight
3. 'know' the time when decisions will be made
4. provide predictions in a dynamic manner

5. should be outcome-based, with a historical decision bias
6. model a-priory known interactions between clinical attributes
7. take caution when reducing the data dimensionality

A clinical decision support system based on reinforcement learning will address principle 1. by imitating the cognitive process of decision making, principle 2. by considering decision sequences, principle 3. and 4. by developing a model that makes decisions on the current available data, and principle 5. by using a reward system that is outcome-based. These issues are standard addressed by reinforcement learning, where principle 6. and 7. require more involvement with clinicians to determine which attributes are important and where data dimensionality can be reduced.

Chapter 3

Methods

This chapter will explain what data and tools are used to create the two reinforcement learning models, following four phases of the CRISP-DM cycle: business understanding, data understanding, data preparation and modelling.

3.1 Business Understanding

The first step of our modelling process is to understand what porblem our model should treat. The previous chapter described the literature around the to be treated problem. This section however will summarise the problem investigation and conclude with a list of requirements for our model.

As described in the problem investigation, infants that are born preterm can have a lot of problems caused by underdevelopment. One of these problems is the high chance of late-onset neonatal sepsis (LONS), as explained in Section 2.1. There are multiple challenges around diagnosing and treating sepsis that should be considered when building a machine learning model. These are described in the next paragraphs.

Definition of sepsis. Defining late-onset sepsis can be difficult as symptoms can vary between patients and they may overlap with symptoms of other diseases and are therefore a-specific. Earlier in this research it is explained that sepsis can be proven (based on positive blood cultures) or can be clinical when showing the clinical symptoms of sepsis with negative blood cultues. In this research we investigate clinical sepsis and do not attempt to make claims about the sepsis being proven or not. The diagnosis of sepsis will be based on the following vitals signs: temperature, heart rate, respiration rate, oxygen saturation, fraction of inspired oxygen and blood pressure. These vitals signs are input for the state of the environment in our RL model.

Timely diagnosis of sepsis. As explained earlier it is import to recognize a developing sepsis early. But because symptoms of sepsis vary and patients this is challenging. During the interviews with clinicians that treat the newborns with sepsis they were not able to define the specific boundaries of when they diagnose an infant with sepsis. This is understandable as their is no standard definition of sepsis and clinician make their diagnosis based on intensive observation of the condition of each patient. In order to evaluate if our model can predict the diagnosis of sepsis, we have decided on a time of diagnosis in consultation with the clinicians: the moment antibiotics (which are typical for sepsis) are given to the patient, that is the moment the sepsis is diagnosed by the clinician. The RL model built will not directly provide a prediction of when sepsis starts, but will ultimately provide a treatment advice when it predicts a possibility of sepsis.

Treatment of sepsis. Another challenge is the treatment of the sepsis, mainly because until the blood cultures are evaluated (48 hours or longer), it is not clear if it is indeed a proven sepsis. And even if it is determined to not be a proven sepsis but clinical signs are still present, the treatment can be continued. The protocol of the LUMC (to be found in Section 2.1.1) provides some guidance on how to treat a possible sepsis. However this protocol does not define the dose of antibiotics given. Suggested by the clinicians this is because each case of sepsis is different and demands a personal approach when considering the medication dosing. In our research the reinforcement learning model will evaluate the policy executed by the clincians and will attempt to find the optimal treatment policy. The two types of antibiotics given when a sepsis is suspected, Vancomycine and Ceftazidim, are the two actions that can be taken by the agent in the RL model.

The goal of this research is to improve the treatment of late-onset neonatal sepsis at the NICU department at the Leiden University Medical Centre. Improving the treatment has underlying requirements:

1. Earlier detection of possible clinical sepsis
2. Determining the optimal antibiotics dosing dependent on the situation
3. Prevent unnecessary medication overdose
4. Earlier decision making about whether to continue with the antibiotics treatment
5. Reducing unnecessary (blood culture) examination

The data mining objective is to predict the medication dose for each specific patient at a given time based on the optimal policy developed by a reinforcement learning model. The goals would be successfully achieved when the optimal policy would give the patients less amount of antibiotics during their stay while resulting in a similar or better patient outcome.

3.2 Data Understanding & Preparation

During this research data from the Neonatology Intensive Care Unit (NICU) of the Leiden Universitair Medical Centre (LUMC) will be used. The NICU of the LUMC stores the vital signs of the infants every minute. On request the database is made available by the ICT Business Intelligence Unit of the LUMC. This was the first time the NICU database (with the exception of personal data) was made available for a data science project. Therefore the collection of the data was an extensive process which required an amount of time that could not be estimated beforehand. For this research the researcher will have access to the full NICU database, which is anonymized in order to protect the privacy of the patients and apply to the regulations involved with the use of patient data. Standard documentation is provided of the MetaVision database tables. The database contains data of 3722 NICU patients with a total of 3957 hospital stays from October 2011 to May 2018.

Features	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**	**unit**
Birth Weight	973.75	255.30	430.0	763.0	950.0	1158.0	2120.0	grams
Gestational age	191.50	10.63	168.0	184.0	193.0	200.0	209.0	days
Chonological age	23.83	20.25	0.0	9.0	18.0	34.0	139.0	days
HR	161.46	10.65	107.4	154.7	162.0	168.7	200.8	min
RR	48.19	9.73	9.0	41.5	47.6	54.6	113.0	min
FiO_2	15.61	14.03	0.0	0.0	21.0	24.8	100.0	%
SpO_2	94.45	2.60	21.0	92.7	94.3	96.4	100.0	%
$Temperature$	36.80	0.40	30.5	36.6	36.8	37.0	38.6	Celsius
$Weight$	1317.07	525.59	428.0	973.0	1200.0	1519.3	5285.0	grams
ABP (measured)	0.15	0.36	0.0	0.0	0.0	0.0	1.0	bool
Vancomycin dose	1.03	2.72	0.0	0.0	0.0	0.0	42.0	mg
Ceftazidim dose	2.49	7.87	0.0	0.0	0.0	0.0	100.0	mg

Table 3.1: Data descriptives

Data Selection

The data from the NICU database contains data from infants born at different gestational ages. In this research only the infants that are born before 30 weeks of pregnancy are relevant as they are more vulnerable which increases the occurrence of sepsis. Of all the infants in the dataset 653 were born before the gestational age of 30 weeks. Next to this we only included patients that were admitted to the neonatal intensive care unit within one day after birth. Another requirement is that the patient stayed

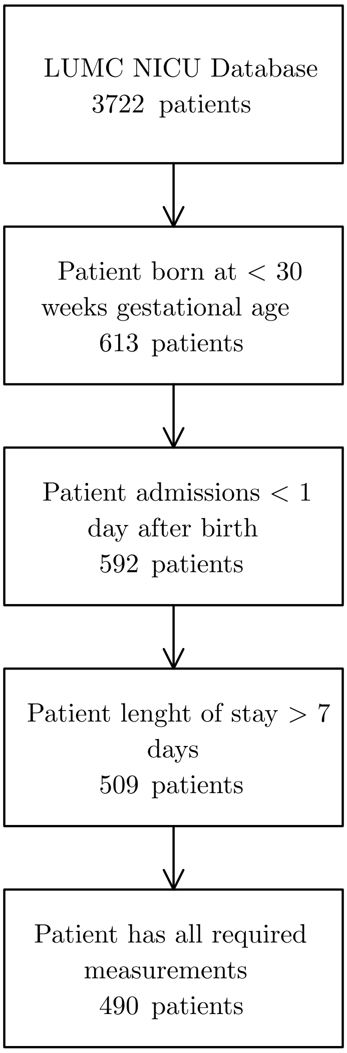

LUMC NICU Database
3722 patients
Patient born at < 30 weeks gestational age
613 patients
Patient admissions < 1 day after birth
592 patients
Patient lenght of stay > 7 days
509 patients
Patient has all required measurements
490 patients

Figure 3.1: Inclusion diagram

for at least 7 days at the NICU, in order to have enough data points for out machine learning algorithm. After removing the data of patients that did not have all the required measurements (i.e. vital signs), there were 490 patients in the target group. The selection process is visualized in an inclusion diagram Figure 3.1.

Data Description

As the NICU database contains over 6600 parameters in various categories, a selection has to be made on what parameters to use for our research. For this research we have chosen to focus on the vital signs of the patients. As the infants at the neonatal intensive care unit are a vulnerable group they are intensively monitored. The NICU database contains multiple measurements of the patient's temperature, heart beat, blood pressure, oxygen saturation, breathing pace, but also 120 parameters about respiration. For this research we will use three types of parameters: demographics, vital signs and medication. Table 3.1 shows the parameters used and their mean, standard deviation, minimum, maximum, quantile cuts and the unit used.

Three demographics are used. *Birth weight* is the measured weight of the patient right after birth. This is a static parameter. The *gestational age* is the difference between the birth date and the date of conception, in other words, the duration of the pregnancy. In this research we focus on infants born after 30 weeks of pregnancy or less. Gestational age is also a static parameter. The *chronological age* is the difference between the birth date and the date of measurement. This is a dynamic parameter, and differs for each patient at each time. The terms *chronological age* and *length of stay* are used interchangeably, as for our group of patients which are admitted directly after birth, they are equal. Of the final group of participants the distribution of gestational age, birth weight and length of stay are visualized (Figure 3.2).

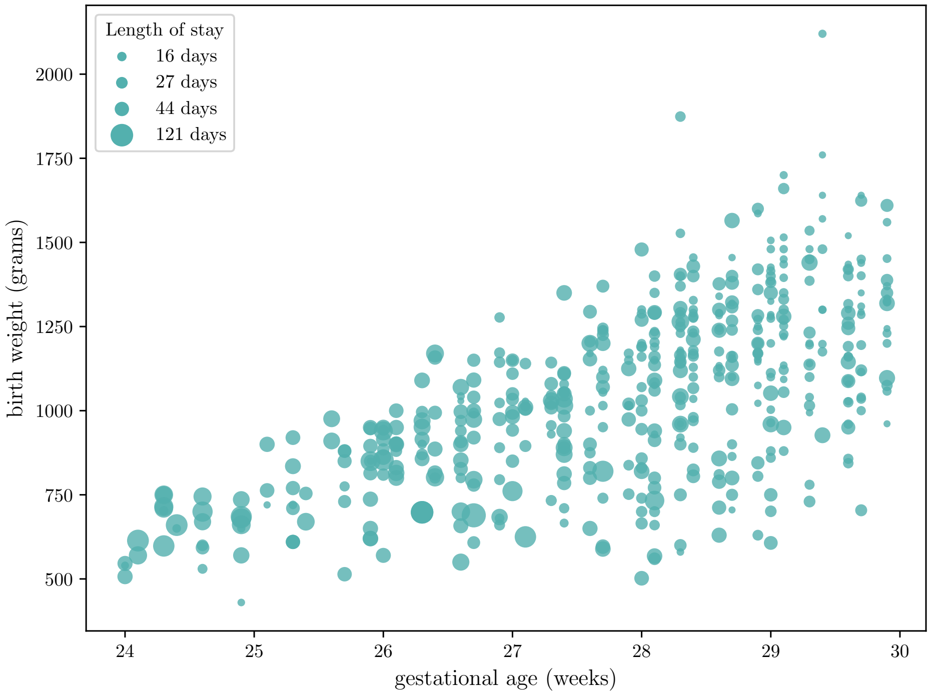


Figure 3.2: Scatterplot of the three variables *birth weight*, *gestational age* and *length of stay*

Six parameters are used to represent vital signs, which are dynamic variables. All vital signs have a saved measurement every minute. The actual measurement frequency is higher for most vital signs, but due to restrictions of the current data storage architecture only one measurement is saved every minute. The first vital sign is the *heart rate (HR)*, which is measured in beats per minute. The *repiration rate (RR)* is measured as the number of breaths taken per minute. FiO_2 and SpO_2 are both parameters that are related to the oxygen levels in the blood. Where SpO_2 is the blood oxygen saturation of the patient, FiO_2 represents the fraction of inspired oxygen the patient receives. Both have measurements saved every minute. See Section 2.1.2 for more information about these parameters and how they are used in neonatal care. The percentage of days the patients received additional inspired oxygen (Figure E.1) is visualized, which shows that around ten percent of the group received respiratory support during their entire stay. The *temperature* of the patient is measured on multiple parts of the body: on the skin, nose or ear, axillary or rectally. These measurements are combined into one parameter in order to achieve the lowest missing rate (Section 3.2). The infants is measured on average once a day during their stay at the NICU, which is saved in the *weight* parameter. The weight is measured in grams. The *arterial blood pressure (ABP)* is the mean blood pressure measured when the infants have a catheter inserted into an artery, i.e. an artery line. Because of the high percentage of missing values for this parameter (see Section 3.2) the parameter is represented by a boolean.

The two final parameters, *Vancomycin* and *Ceftazidim*, are the two types of medication the patients receive in order to treat an late onset sepsis (as explained in Section 2.1.1). Both are saved in the database with a start time, a medication dose and a frequency of administration. During the data integration step (Section 3.2) this is transformed to a medication dose per 6 hours. The descriptives in Table 3.1 describe the data after this transformation. The final scatter plot (Figure 3.3) shows the total amount of Vancomycine and Ceftazidim the infants received during their stay.

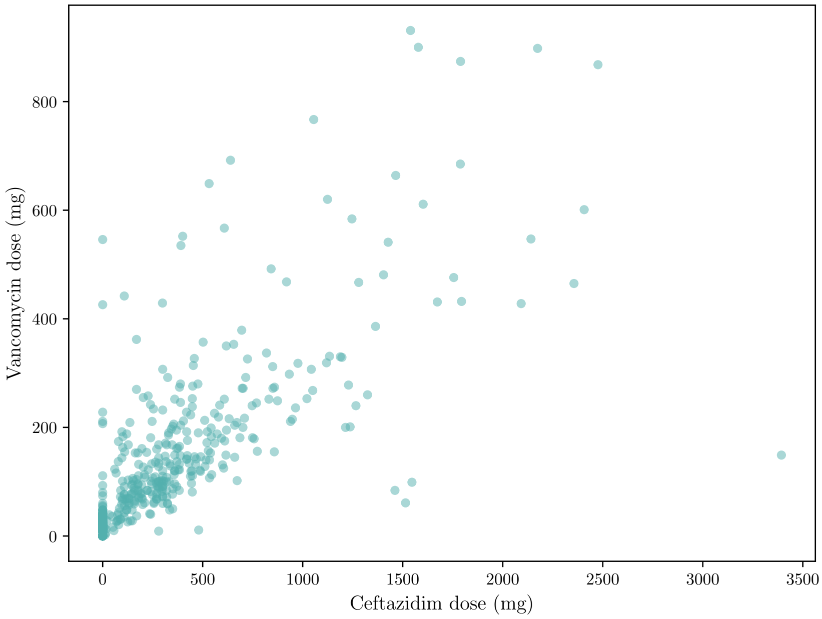


Figure 3.3: Scatter plot of the total *Vancomycin* dose and *Ceftazidim* dose during the patients stay

Data Quality

As the database of the NICU at the LUMC was not yet used for data science purposes, there were no previous assumptions about the data quality. Initial exploration showed a high percentage of missings for most of the vitals signs. After further investigation it appeared that the vital signs were saved in multiple parameters. This is dependent of the type of medical measuring equipment that is used. After combining the different parameters that measured the same vital signs, there was still missing data which is shown in Figure 3.4. The missing percentage is calculated as the percentage of 6 hours blocks (explained in Section 3.2) that have no value for that vital sign. Of the vital signs ABP (ambulatory blood pressure) has the highest percentage of missing data with over 80%. This can be explained as blood pressure is not constantly measured for neonates, but only when they have an artery line inserted. The second highest missing percentage is for the weight of the infants. As the infant is not weighted within every 6 hour block the missings appear naturally. On average the infants are weighed every 24 hours, especially if they are born with a low birth weight. FiO_2 has explainable missings as the value for the inspired oxygen is only noted when the infant receives respiratory support. Temperature, respiration rate, heart rate and SpO_2 should be measured constantly. Missings for these variables can arrive from the infants being detached from the measuring equipment, for instance when they are being held in the parents arms.

Data Integration

For this research the data integration was executed before the data cleaning and data construction in order to reduce the size of the dataset before performing actions on it. The data source contains one value every minute for every vital sign. As this would produce a really big data set to apply machine learning on, we have binned the data into 6 hour time windows. This reduces the complexity of the problem at hand and the resources necessary to handle these amounts of data. The binning of the data happened during the extraction from the database and was coded into the SQL query to average all values from within the six hours. The data of the medication (*Vancomycin* and *Ceftazidim*) was also transformed to those 6 hours bins.

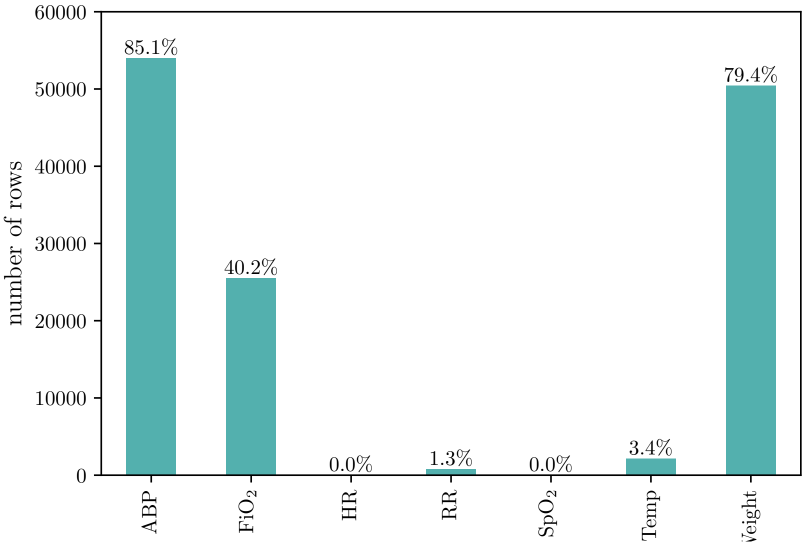


Figure 3.4: Chart of percentage missings per variable

Data Cleaning

In Section 3.2 we described the quality of the data and the occurrence of missing values in the data. In this section we describe how the missing data of each vital sign was handled. First of all, rows with all variables missing at the beginning or end of the admission were removed. Empty rows at the beginning represent the time before the infant was set up with the measuring equipment and the empty rows at the end represent the time that the infants is already detached from the measuring equipment but not yet discharged from the department. The variable with the highest percentage of missing data was the blood pressure ($> 80\%$). This variable was not imputated but transformed into a binary attribute, value 1 for 'measured' and 0 for 'not measured'. This also solved the situation were no blood pressure was measure for the infant during the entire stay. The weight measurement also had a hight percentage of missing values, but this was less problematic as on average the data contained one measurement every 24 hours. We filled the gaps with linear interpolation treating the values as equally spaced. This is acceptable for the variable as weight is not expected to dramatically change every hour. The missing data for the inspired oxygen, FiO_2, meant that the infant did not receive additional respiratory support and could be filled with the value '0'. The remaining variables, HR, RR, SpO_2 and Temperature, were filled in with linear interpolation.

The state features are scaled to a 0,1 range, using the *scikit-learn* Python package. The action features needed no further data formatting after being discretized during the data integration process.

Data Construction

After the binning of the data it had to be prepared in order to be used with reinforcement learning. Section 2.2 explained the different elements in a RL model: states, actions and rewards.

State space In this case the state is represented by the demographics and vitals signs of the patient for the six hour bin, yielding a 10 x 1 feature vector for each patient at each time step.

Chronological age	Gestational age	Birth weight	Heart rate	Respiration rate	FiO_2	SpO_2	Temperature	Weight	ABP
static	static	static	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic	dynamic

Table 3.2: 10 x 1 state space with variable type

Action space The action space is discretised into a 5 x 5 action space for the antibiotics treatment with *Vancomycin* and *Ceftazidim.* These medications were discretized into quatiles based on all non-zero dosages of each drug and one bin for no dosage at that moment. The binning is visualized in Figure D.1. For the *Vancomycin* dosing the quantile cuts are at 4,5 and 7. For *Ceftazidim* the cuts are at 12, 20 and 27. The binning of both antibiotics resulted in a discrete set of 25 possible actions, as seen in Figure 3.5.

Ceftazidim

Vancomycin

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19
4	20	21	22	23	24

Figure 3.5: Action matrix

Reward function The reward function has to represent whether the state space the patient is in is desired. The desired states are where the patient shows no signs of possible clinical sepsis. The clinical signs of sepsis are described in Section 2.1.1 and include temperature instability and increased ventilatory support. As these two variables are included in dataset we haven chosen for a reward function based on temperature ($°C$) and inspired oxygen (FiO_2). The desired temperature for newborns is from 36$°C$ to 38$°C$, with the ideal temperature being 37$°C$. The additional inspired oxygen (FiO_2) should be as low as possible, with 0% being most optimal but around 20% being acceptable. The final code for the shaped reward function can be found in Appendix B.

3.3 Modelling

To analyse the current policy and find the optimal policy we will use two different reinforcement learning models: an on-policy model and an off-policy model. The on-policy model Sarsa (as explained in Section 2.2) will be used to model the policy used by the clinicians based on historical data. Q-learning, the off-policy algorithm, is used to find the optimal policy for sepsis medication dosing for neonates. Although the algorithms differ the model are based on a similar network architecture: a Dueling Double Deep Q Network. Both models will share the following characteristics:

Deep Q-Network

Introduced by the researchers of DeepMind Technologies in 2013, the Deep Q-Network was first used to play Atari games(Mnih et al., 2013). It was the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network trained with a variant of the Q-learning algorithm. The network architecture of the Q Network will be fully-connected with two hidden layers of size 128.

Separate Target Network

Two years after the initial DQN was introduced by DeepMind the updated version of DQN was published, which showed the importance of experience replay and a separate target network (Mnih et al., 2015). The target network is second network that is used to generate the target Q-values that will be used to compute the loss for every action during training.

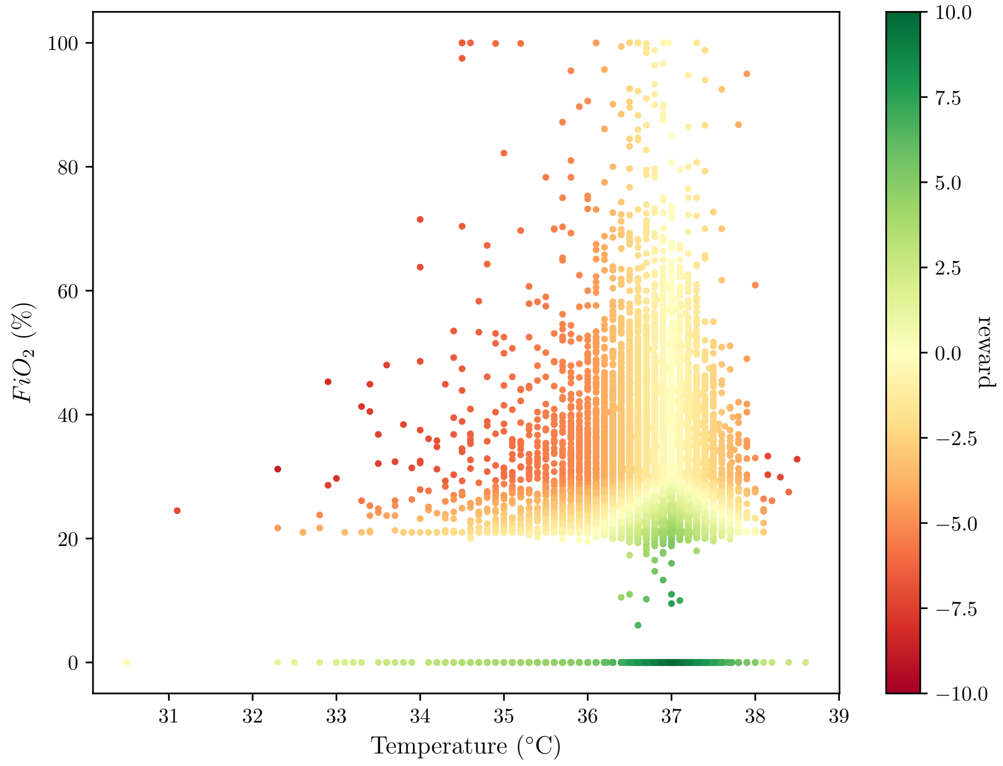


Figure 3.6: Visualization of shaped reward function

Double Q-learning

A regular Deep Q-Network often overestimates the Q-values of the potential actions to take in a given state, which could endanger the chance of the agent being able to learn the optimal policy. The addition of the Double Q-learning algorithm (Van Hasselt et al., 2016) to the DQN was introduced in 2015 by the researchers of DeepMind, part of Google since 2014, in order to reduce the overestimation. The algorithm was originally proposed by van Hasselt (2010) in a tabular setting, which is used to construct the new Double DQN algorithm. In standard Q-learning and DQN the same values are used to both select and evaluate an action, which results in overoptimistic value estimation. The approach of Double DQN to reduce the overestimation is to decouple the action choice from the Q-value generation, which means that the main network is used to choose an action and the target network to generate the Q-value.

Dueling Network Architecture

As of 2016 most of the approaches for reinforcement learning used standard neural networks, until an 'alternative but complementary approach' was introduced by Wang et al. (2015) that focused on innovating a neural network architecture that is better suited for model-free RL. The proposed network architecture consists of two streams that represent the value and advantage functions, instead of the single-stream of the DQN. The dueling architecture has a value $V(s)$ function and an advantage $A(s, a)$ function, whose output is later combined into a state-action value $Q(s, a)$. The advantage function represents the quality of the chosen action relative to other possible actions and the value function represent the quality of the current state. The benefit of this separation is that it allows the model to better differentiate actions from one another and learn faster. The key insight behind this is that in some states the choice of action has no effect on what happens and therefore it is unnecessary to estimate the value of each possible action choice in that state. This also means that the estimation of state values is of great importance.

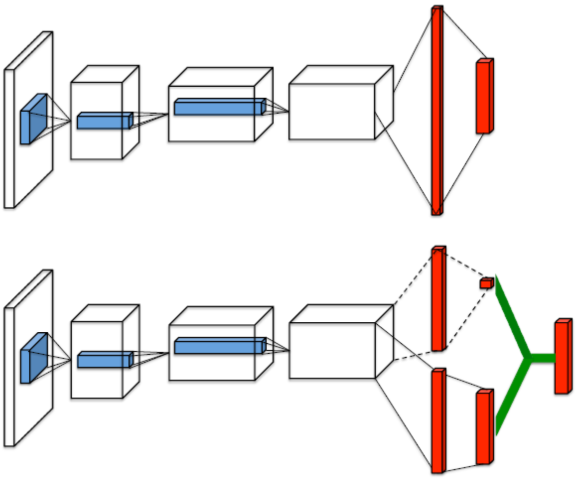


Figure 3.7: Normal DQN vs Dueling DQN architecture (Wang et al., 2015)

Prioritized Experience Replay (PER)

In experience replay the agent's experiences are stored as a tuple of $< state, action, reward, nextstate >$ (Schaul et al., 2015). During training of the model mini-batches of experiences are drawn from the memory to learn from. This way the model will be more robust by preventing it only learns from what it is immediately doing in the environment, but also learn from past experiences. The idea behind prioritizing experiences is that the RL agent can learn more effectively from some transitions than from others. PER allows the agent to replay experiences at a different frequency than what they were originally collected in and allowing it to choose experiences with a bigger learning value. However, prioritization introduces bias by changing the distribution of experiences, which is corrected by using weighted importance sampling.

Batch Normalization

Batch normalization has been introduced by Ioffe & Szegedy (2015) to accelerate training of deep neural networks by reducing internal covariate shift. They define this phenomenon as 'the change in the distribution of network activations due to the change in network parameters during training'. Simplified this means the input to each layer is affected by parameters in all the preceding input layers. This slows down training because the layers need to continuously adapt to the new distribution. Batch normalization accelerates the training by putting in a normalization step that fixes the means and variances of layer inputs.

Leaky-ReLU activation function

Leaky ReLU (Rectified Linear Units) is an activation function designed by Maas et al. (2013) to improve the performance of deep neural network acoustic models for speech recognition. An activation function checks whether to consider a neuron as activated or not. A simple version of this is the threshold based activation function. An improvement to this is using a linear function where activation is proportional to the input. Using a linear function causes problems for deep neural networks, because if all layers are linear the final activation function of the last later is just a linear function fo the first layer. This undermines the reasoning for stacking multiple layers as the whole network is equivalent to a single layer network with one linear activation function. Moving on to the Sigmoid Function, stacking layers are justifiable again as it is a non-linear activation function. Although the Sigmoid function has been widely used, it still has a particular problem that can be improved: the problem of 'vanishing gradients'. Towards the end of the sigmoid the gradient is small or is even vanished, causing the network to slow down or stop learning. As a reaction to this problem the ReLU is non-linear as are combinations of ReLU. The ReLU function gives an output x if x is positive and zero otherwise. The activation function is sparse as only the neurons with a positive output are activated. But even the ReLU function has a downside: the dying ReLU problem. Neurons that are not active have a zero gradient and can possibly stop responding because the gradient-based optimization algorithm will not adjust their weights. The idea behind Leaky ReLU is that it allows for a small, non-zero gradient when the unit is not active. It sacrifices the hard-zero sparsity described earlier for more robust optimization.

Parameters

- Alpha (α): the learning rate α determines with what factor the model overrides old information with new information. Setting it to 0 means that the Q-values are never updated, hence nothing is learned. Setting a high value such as 0.9 means that learning can occur quickly. The alpha of our prioritized experience replay is set to 0.6.
- Epsilon (ε): the ε-greedy algorithm describes the balance between exploration and exploitation. During the training process it selects random actions with a probability epsilon. In our model the epsilon of our prioritized experience replay is 0.01.
- Gamma (γ): the discount factor γ determines the present value of future rewards by defining that a reward received k steps in the future is worth γ^{k-1} of its immediate worth. In our model the discount factor is set to 0.99.
- Number of steps. This models has been trained with 100.000 steps, which took around 8 hours for both models on one GPU each.
- Batch size. The batch size which the model processed during training was 32.
- TAU: rate to update the target network toward primary network. The variable is set to 0.001 which means that the target network is updated with the amount of 1/1000 every time step which is roughly equivalent to fully updating the network every 1000 steps.
- Reward threshold. Regularization in the form of a reward threshold was added to the network that penalises the network when it produces rewards that are above the reward threshold, to ensure reasonable Q-value predictions. The reward threshold in our model is set to 100.

3.3.1 Tools used

All the code is written in Python. *Pandas, Numpy and scikit-learn* are used for data preparation. The *Matplotlib* library is used for visualization. And finally *Tensorflow* is used for creating the reinforcement learning models.

The code used for this research is based on the code created by Raghu et al. (2017). Their research introduced a reinforcement learning model for sepsis treatment for adults at the intensive care unit. As the problem is similar the code was a guideline for the code used in this research. The code used by Raghu et al. (2017) is published on `https://github.com/aniruddhraghu/sepsisrl`.

3.3.2 Model Assessment

After both models have ran for eight hours on one GPU each, we will now assess the performance of the models based on five values: Q-value, Q-loss, average loss, mean absolute error and convergence.

Q-value

One of the most important values when evaluating the performance of a reinforcement learning model is the average action-value, or Q-value, as the goal of our model is to improve the value of actions taken. The history of the Q-values of both models during 100.000 training epochs is displayed in Figure 3.8. The chart shows that the Q-value of the DQN model is higher than the Q-value of the SARSA model. This is occurs naturally when training a RL model because the goal is to discover the optimal actions to take based on the actions that are taken in the examples.

Loss and Error

During the training the loss is obtained by taking the sum of squares difference between the target and prediction Q-values. The mean abs error is the mean of the absolute error values of a processed batch during training. The absolute error is the absolute difference between the Q values from the main network and the Q values of the target network.

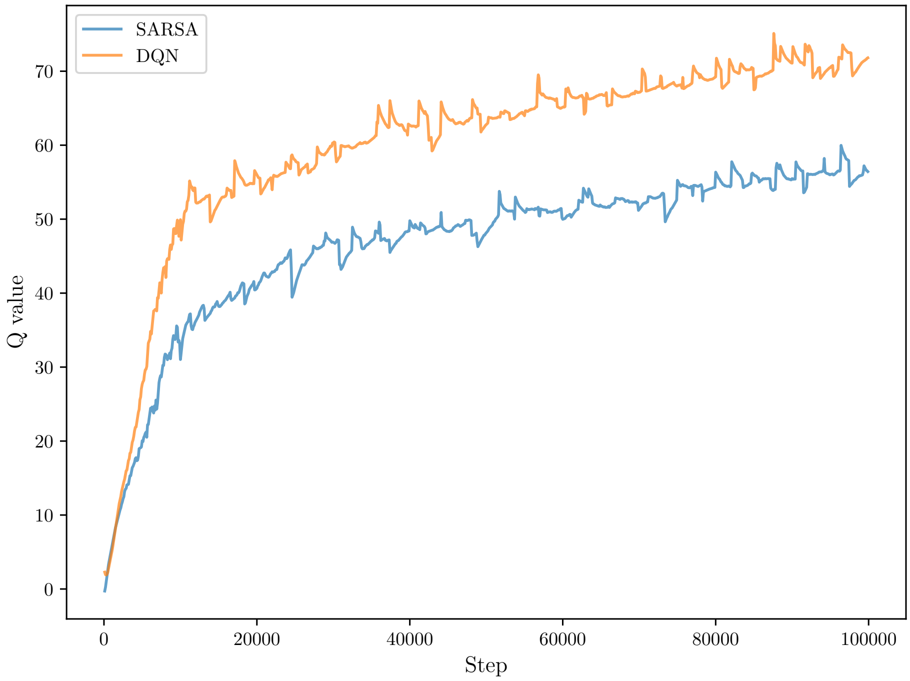
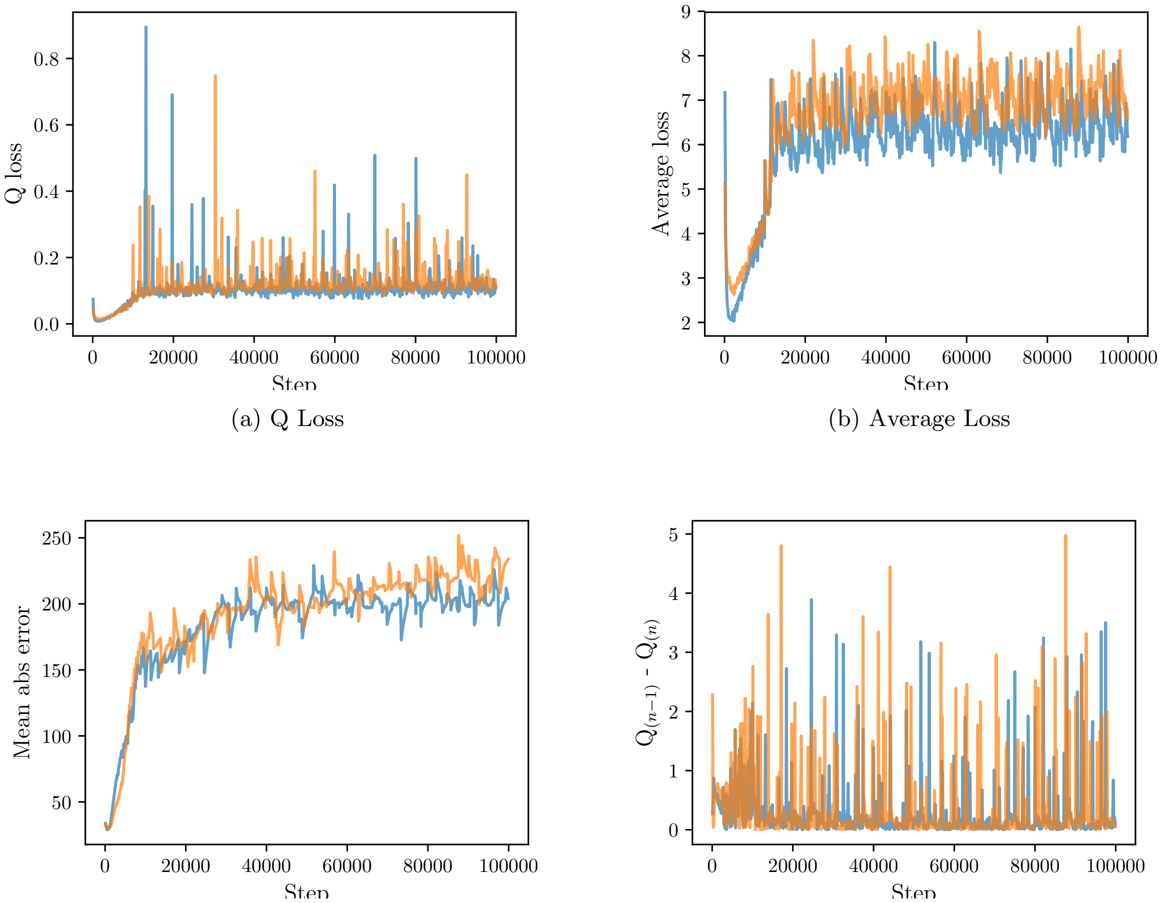


Figure 3.8: Q Value

Convergence

The convergence chart in Figure 3.9d shows us the difference between $Q_{(n)}$ and $Q_{(n-1)}$. The chart shows that these values do not converge over the 100.000 steps that we have run and once every couple of steps we see a peak. This can be explained by the trade-off between exploration and exploitation (as discussed in Section 2.2). In our model the epsilon, the fraction of the time steps in which it selects a random action, is set to 0.01 and is not reduced during training. Gradual reduction of epsilon during training would result in faster convergence of model performance.



(a) Q Loss

(b) Average Loss

(c) Mean abs error

(d) Convergence of $Q_{(n)}$ and $Q_{(n-1)}$

Figure 3.9: Evaluation metrics

Chapter 4

Results

In this chapter we will compare the optimal policy with the clinician's policy. First we discuss whether validation studies on reinforcement learning models are possible. Then we show multiple charts that show the comparison of the physicians policy and the optimal policy. And finally we show a performance measurement of the reward differences between the two policies. The evaluation in Section 4.2.1 is based on the evaluation in the paper about reinforcement learning for optimal sepsis treatment by Raghu et al. (2017). The performance measurement that we use in Section 4.2.5 is inspired by the analysis performed in the research by Nemati et al. (2016).

4.1 Validation

Validation of supervised learning algorithms has been extensively discussed in literature. In a validation study the predictive performance of the model is observed without intervening with the decision-making (Kappen & Peelen, 2016). The predicted risks are compared with the actually observed risks to assess the applicability of the model. The model can be internally validated as well as externally. Internal validation, which is the minimal requirement for accepting a model, is the predictive performance in the population from which the model was developed. To asses the predictive performance of the model the data is split into a training set and a validation set beforehand, or split multiple times in case of cross-validation. The predicted risk from the model based on the training set is compared to the observed risk in the validation set. This assesses how well the model performs on unseen data. Because a model will typically perform best on the data that was used to develop the model, or data from the same environment, it should also be tested for external validity: the predictive performance of the model is estimated in a new cohort of patients. External validation assesses the generalizability of the model.

Unfortunately, validation of reinforcement learning models is less straightforward. As we do not have a model of the environment, which in this case is the patient, we do not know how it will react to the actions taken by the optimal policy. Another possible problem is that the model is trained only on the states and action seen in the historical data, and during online use it might encounter states it has not seen before and there is no way to tell how the model will react to these states. Examples of reinforcement learning models being validated with hypothetical simulation models are the research about cancer clinical trials (Zhao et al., 2009) and about optimization of anemia treatment in hemodialysis patients (Escandell-Montero et al., 2014).

This research does not include either an internal or external validation study. Possibilities for internal and external validation will be discussed in Section 5.1 about future work. The following section will discuss the possibilities for verification of our reinforcement learning model.

4.2 Verification

As mentioned in the previous section, validation of reinforcement learning is not straightforward and no discussion has been found examining the validation of reinforcement learning models where there is no model of the environment or the RL model cannot simply be tested in a real-life scenario. These two properties are often true for health care environments: physiological models of patients are not always available and clinical trials take time to set up and perform. If validation of this research is not possible without performing a clinical trial, what can be done to evaluate the reinforcement learning model performance. According to a publication by Van Wesel & Goodloe (2017) there are multiple ways to verify reinforcement learning models, both offline and online. The core problem of verification in computer science is to verify that a given system satisfies the specification. In practice this means that we apply the machine learning model and check whether the output are as expected. Hereby we can prove that the implementation of the system is correct, but cannot say anything about the (predictive) performance of the model. The next sections show multiple offline verifications of our reinforcement learning model based on historical data.

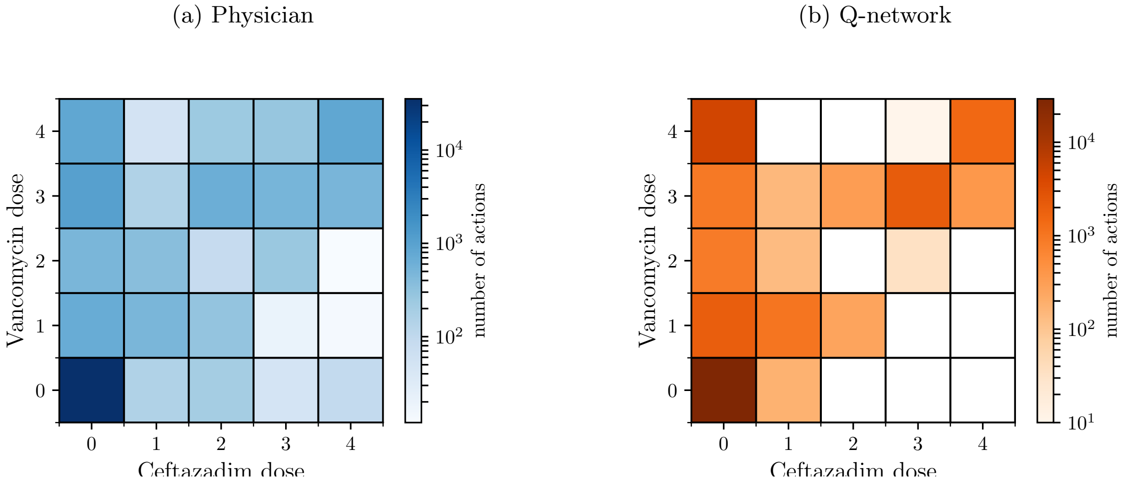


Figure 4.1: 2D histogram of actions selected by (a) the physician and (b) the Q-network

4.2.1 2D histogram of actions taken

During training the Sarsa model has evaluated the actions taken by the physician. In Figure 4.1a the results are plotted against the action map in a 2D histogram with a logarithmic scale. It is directly evident from the 2D histogram that the action where no *Ceftazidim* dose and no *Vancomycin* dose is administered is most common. This can be explained by the fact that the number of time bins where no sepsis occurs is most prevalent. This also explains the need for a logarithmic scale as the zero-zero action is zo prevalent, the other actions all fell in the same size category and no distinctions could be made. Furthermore it can be stated that there are more times where only *Vancomycin* is given to the patient than the occurrence of only *Ceftazidim* dose. This is consistent with the written sepsis protocol of the LUMC which states that *Vancomycin* mono-therapy is given when the blood culture results are positive but no contamination is suspected (Figure 2.2).

The 2D histogram of the mapped actions by the optimal policy can be seen in Figure 4.1b, again with a logarithmic scale to show internal differenced of non zero-zero actions. Similar to the physicians actions the Q-network choose the zero *Vancomycin* and zero *Ceftazidim* most often. It can be assumed that the model did this with the same reason: no occurrence of sepsis at those time steps. Also the *Vancomycin* mono-therapy is one of the most prevalent actions chosen by the Q-network. *Ceftazidim* mono-therapy is almost never chosen as an optimal action.

As the two 2D histogram are hard to compare seen separately, we created a 2D histogram of the difference in actions taken by the two policies (Figure 4.2). Blue represents the physician's policy and red the optimal policy. This figure shows a couple of interesting results. First off, the zero-zero dose is more often picked by the physicians than by the optimal policy. Second, the *Vancomycin* mono-therapy is more frequently picked by the optimal policy than by the physicians policy. Finally, the optimal policy shows a preference for a high dose of both medications.

4.2.2 Action to action mapping

Although the policy comparison in the previous section provided an overview of the differences in strategy of both policies, it did not show us exactly which action were replaced by which other action. Therefore we created the chart in Figure C.1. The actions taken by the optimal policy are mapped to the actions taken by the physician in a matrix. On the vertical axis are the action taken by the physician and on the horizontal top axis are the actions taken by the Q network. Instantly it can be seen that most of the

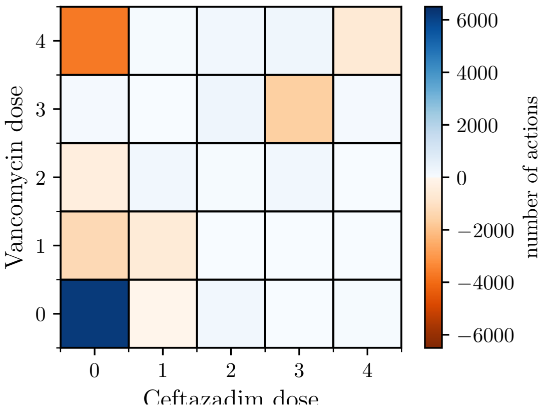


Figure 4.2: 2D histogram of difference in actions selected by the physician (blue) and the Q-network (orange)

switches are from the physician's zero action to a non-zero action by the DQN. In practice this means that there are multiple opportunities where the physician gives no medication but the DQN thinks the patient should be given medication. If these moments represent situations where the DQN detects sepsis earlier than the physician, this would check one of the requirements of our model. This can be validated during online testing of the model (described in Section 5.1.3). Another observation can be made about the action switches. The DQN zero column shows that some non-zero action of the physician are replaced by zero action from the DQN. In other words, in some situations were the physician did give medication to the patients the DQN would advise to give no medication. In practice this could apply to the states were the patient is healthy again, but the medication is not yet stopped by the physician. If the DQN is indeed able to stop the medication at an earlier time than the physician, then this would approve another requirement for our model, which is to reduce the medication given to the patient. To check whether the DQN is more compliant to this requirement than the physician, this should be tested during online learning of the model.

4.2.3 Action sequences

The previous analyses did not take the time dimension in consideration. By mapping action sequences when using a model-free algorithm, we can discover the underlying model of the system. If the environment can be modelled as a Markov decision process, the requirements can be quantified in a probabilistic temporal logic specification (Van Wesel & Goodloe, 2017). Temporal logic have been an important research subject within the study on logical formalisms for specifying and verifying real-time systems. More in-depth information can be found in the overview paper about *Real-time and Probabilistic Temporal Logics* (Konur, 2010). During runtime verification of the temporal logic specification can be performed by observing the output actions of the DQN agent and classifying whether they fall within the specification. For this research we did not model the Markov decision process or create a temporal logic specification due to time limitations.

Figure C.2 shows an visualization of the first week of 50 random patients. Visualizing these action sequences could already show some structure of the underlying model. An improvement to this visualization would be to split the combined Vancomycin and Ceftazidim action number. That way it would offer direct insight into the dosing of both medications.

4.2.4 State to action mappings

When the actions taken at each state are mapped, when the model is not learning online, rules can be discovered (Van Wesel & Goodloe, 2017). These rules can then be checked with the requirements of the model in order to perform verification. An example of this could be that for a certain patient state, a high dose of Vancomycin is given with a zero dose of Ceftazidim. Checking this with the NICU antibiotics protocol could verify whether this is state to action mapping is appropriate. State to action mapping can also be performed when the model is learning online, keeping track of the changes in these mappings.

A state to action mapping is not performed in this research as it would require discrete states opposed to the continuous states used. This is possible to add to the research, but restricted by time limits.

4.2.5 Reward difference

Evaluation of performance of the two modelled policies is difficult as there is no model of the environment. In other words, as we do not have a model of how the infant would react to the actions picked by the two different policies we can not evaluate whether the actions picked by the optimal policy lead to a better outcome for the patient. One form of evaluating the performance of the optimal policy against the physicians policy is to calculate how much the actions taken by the policies differ. Figure 4.3 shows the percentual difference of both policies plotted against the cumulated reward. The points in this scatter plot represent the different patients in our dataset. Based on this plot we performed a linear regression analysis, which seems to show a correlation between the two plotted values. Important to note is that the values of the reward are dependent on how the reward function is shaped. Therefore no conclusions can be made about the clinical performance of the model.

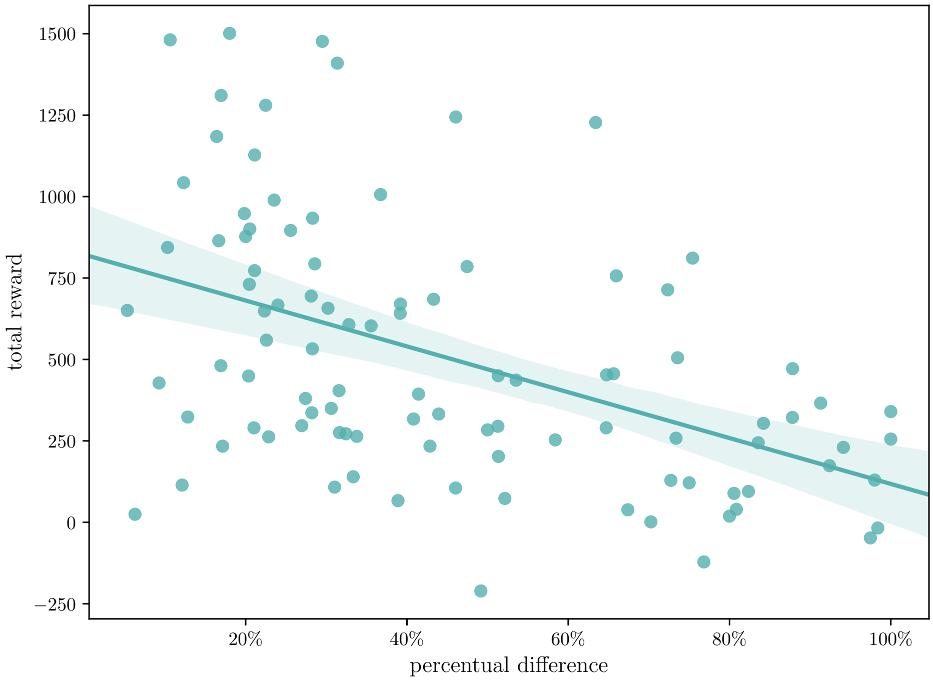


Figure 4.3: Linear regression analysis chart of reward difference

Chapter 5

Discussion

In this research we have investigated the possibility of improving sepsis treatment in neonatal care by using reinforcement learning. First we defined the current sepsis treatment according to the literature, which answered our first research question. We discovered that the treatment of sepsis for neonates can be quite complicated. This is due to the fact that neonates all differ in age and developmental stage, and that the definition of sepsis is not unambiguous. For this research we decided to focus on the dosing of the two most used antibiotics for sepsis treatment in neonates: *Vancomycin* and *Ceftazidim*.

Next to our study on neonatal sepsis, we researched the field of reinforcement learning (RL) and clinical decision support system in order to understand how these subjects could help improve the treatment. The next three research questions were about how RL could analyse the current treatment, optimize the treatment policy and compare that to the current policy. We have developed two models, one to analyse the current policy of the physician and one to learn an optimal policy from historical data. Assessment of the model showed that the optimal model performed better on model specific evaluation metrics. Therefore in terms of the model performance, the learned optimal model outperforms the physicians model.

The final question is whether the learned optimal policy improves the clinicians decision making. This question is more difficult to answer as the clinical impact of the model cannot be evaluated. As we do not have a model of the environment, in this case the patient, we cannot evaluate the reaction of the environment to the actions taken by the model. It is not possible to run simulations in order to see whether the actions taken will lead to better patient outcome. Although we have verified the model and it's output meets the expectations, we have to run clinical trials to validate the model.

5.1 Future work

During this research it gradually became evident that the field of reinforcement learning applied to health care data is relatively unknown and turned out to be more complicated than seemed at first. The list of limitations grew and thereby also the list of future work. The future work based on this research is divided into four subjects. First of all, a better definition of the problem statement would improve this work. In this case that includes the definition of sepsis, which has been explained to be complicated in Section 2.1. The second option to advance our research is to improve the model. The state space, action space and reward function of the reinforcement learning model can each be further improved. A third improvement is the execution of clinical trials which can help to finally evaluate the clinical impact, rather than only verify the model performance as done during this research. The final step of the future work is actual deployment of the model as a decision support system. All these improvements are described in the following sections.

5.1.1 Definition of sepsis

As mentioned before the definition of sepsis ambiguous. Different definitions are used in research and the clinicians could not provide a straightforward definitions of sepsis during the interviews. Neonatal sepsis is not straightforward. The symptoms vary and could also be an indication for other diseases. Using a different definition during the modelling process might possible lead to better or worse results. Next to this, there is the difference between clinical sepsis and proven sepsis. Where the first is diagnosed based on symptoms in the clinical signs, the latter is proven by a positive blood culture. This research was focused on clinical sepsis, as it is important to treat a sepsis on time, whether it is proven or clinical. Research into proven sepsis could also lead to improvements. Although treatment of both proven and non-proven sepsis are equally important, it might be useful to see if an advanced machine learning model can detect differences between them in the patient state. When a proven sepsis can be detected without the need of collecting blood from the patient, this improves the patient well being as blood collection from newborns is a tough intervention because their bodies are so vulnerable.

5.1.2 Improvement of the model

As this paper showed a simple implementation of reinforcement learning, there are multiple possibilities for improvement of the model. These improvements can be made on multiple parts of the RL model: the state space, the action space and the reward function. The next paragraphs discuss the improvement possibilities for each of the elements.

State space

The state space that was used for this research was a simple 7 x 1 feature vector representing the vital signs of the patient for each six hour time step. This small feature vector (comparable research used a 47 x 1 vector (Raghu et al., 2017)) provides the model with a simplified representation of the patient. Adding more variables to the feature vector might lead to a better performance of the model, which is a more personalized treatment. Another improvement to the state space could be achieved by giving the features different weights according to their importance. The importance of the features should be derived from clinical research. One example of this is giving more importance to ventilatory support (FiO_2) as opposed to heart rate as increased ventilatory support is seen as one of the important variables in sepsis prediction. A third improvement is reduction of the time bin sizes. In this research we used 6 hour bins to aggregate the state of the patient. Using smaller time bins, for instance every hour, might improve the results. Especially when the model will be used in real time you want more evaluation points during the day. Next to this, the neonatal intensive care unit of LUMC will possibly make a transition from 1/60 Hertz frequency data points to 1 Hertz data storage. This would make it possible to keep track of the heart rate and respiratory rate in more detail, and detect apnoea and bradycardia events. As these events can be indicators of possible sepsis, detecting them can improve the models performance. Next to this the way that the data is imputed can influence the performance of the model. In this research most of the parameters are imputed by using interpolation, but there is research available that handled missing medical data using multiple imputation(Wells et al., 2013). As simple imputation methods can lead to misleading results, multiple imputation is recommended by preventing exclusion of data rather than creating false data (Janssen et al., 2010).

Action space

The action space can be improved in multiple ways. The first improvement would be to relate the dosing of the antibiotics to dosing given according to the protocol. In this research we calculated the dosing per 6 hour bin. In practice the medication is not given every 6 hour, but dosing are given with a frequency per 24 hours. For example Vancomycin can be given once per 24 hours, but when increasing the dose, it will be given twice every 24 hours or even three times per 24 hours. When relating the actions recommended by the model to actions used in practice, the physicians will be able to directly interpret

Reward function

Further shaping of the reward function might improve the effectiveness of a reinforcement learning model. The reward function of our model is shaped based on the patients temperature ($°C$) and ventilatory support (FiO_2), as they are possible indicators of sepsis. Other indicators (as described in Section 2.1.1) could be integrated in the reward function. Another possibility is to use *Inverse Reinforcement Learning* (IRL) to learn an unknown reward function from observed behaviour of an agent (Ng et al., 2000). This algorithm assumes that the agent behaves optimal, but in many cases this is not true. For example when a human learns a tasks by trial and error, the observed behaviour will contain behaviour that lead to failure. Therefore other researchers introduced *Inverse Reinforcement Learning from Failure* (IRLF) to address this problem (Shiarlis et al., 2016). This could improve the reward function by minimizing learning from ‘failed’ behaviour, or in practice treatments that did not work. Another paper addresses the problem of multitask inverse reinforcement learning (Dimitrakakis & Rothkopf, 2011), where the motivations of different experts that try to solve the same task are considered to be part of the problem. This relates to the different physicians having different opinions about how the treat an patient, within this subject group, or when combining data from different hospitals. Next to this, our reward function gives no reward or punishment based on the final state. In comparable research this is done based on the patients survival. Another improvement could be giving negative reward for medication use, as one of the motives for this research is to reduce medication use on these patients. Our current model does not have a negative relation with medication use and might give more medication if it improves the reward achieved. In practice, giving more medication could lead to negative long term effects.

5.1.3 Clinical trials

The final question in this research is whether a reinforcement learning implementation can improve clinical decision making in health care. As described in Chapter 4 validation of the optimal policy learned by the RL model is difficult and without retrieving new data we can not conclude whether our learned policy would perform better than the current physicians policy. In order to answer our the previously mentioned question, we should run clinical trials to determine it's effect on clinical decision making. A cluster randomized trial there is one intervention group and one 'care-as-usual' group. The physicians in the intervention group receive information from the prediction model in a non-intrusive matter: the model runs alongside the existing systems and the physicians could compare their approach to the suggestions by the model and choose whether they agree with the model. At the end of the trial, the physicians are then asked about their experience with the model and how the model influenced their decision making. Although this is subjective feedback, and therefore hard to assess the performance numerically, it will provide some information about the effects of the model on clinical practice. This process is extensively described in a PhD research by (Kappen, 2015) about the chances and challenges of prediction models and decision support.

5.1.4 Deployment

An important part of the problem treatment is the implementation of a decision support system. In this current research the reinforcement learning agent has only been able to learn on historical, offline data. In the future we want the agent to learn online and be able to adapt to every specific patient. Therefore we need an online implementation with a front-end system for the physicians to work with.

There are a couple of recommendations for a successful implementation of a prediction model in clinical practice (Kappen et al., 2016):

1. Adding an actionable recommendation to the predicted risk (directive prediction model).
2. Presentation of the predicted risk should be automated and smoothly integrated with the physician's workflow.
3. The reasoning and research evidence of the underlying prediction model to show how risks are actually estimated should also be available to physicians.
4. A prediction model will be better perceived by physicians when it predicts outcomes that are relevant to them and their patients.

In order to confirm with the first recommendation the CDSS should provide a recommendation of which antibiotic to give and in what amount. For example it should say to the physician "Give the patient a 8mg Vancomycin dose and repeat every 6 hours". This provides the physician with an actionable recommendation which is in alignment with their clinical practice. Second, the model should be implemented into the current workflow of the clinician. Preferably in the system they already use. Adding another separate interface could cause the physicians being reluctant to use it as it adds steps to their process. The implementation of the third step is more difficult as the underlying algorithm of the reinforcement learning is possibly hard to understand for physicians. Research has to be put into illustrating the inner processes of reinforcement learning models. As mentioned before the physician has to fully understand how the recommendation by the model is established in order to be able to make a decision. Making visible which features were most important to the model could be a first step towards this. The final recommendation for successful implementation of a CDSS is that it should explain what the predicted outcome is for the patient. For example it should state that when using this amount of medication for this duration, it expects an improvement in the patient's health in the next couple of hours. If the model could be specific on what effects the antibiotics would have on the patient's vital signs, it could help the physician decide whether the recommendation is correct for this patient and if it would have negative side effects. For example a lowered oxygen saturation which would require to increase the respiratory support.

To conclude, the clinical decision support system should make a recommendation, explain what the recommendation is based on and illustrate the expected outcome for the patient's health.

References

Aarnoudse-Moens, C., Rijken, M., Swarte, R., Andriessen, P., ter Horst, H., Mulder-de Tollenaer, S., ... Weisglas-Kuperus, N. (2017). Follow-up na 2 jaar van kinderen geboren bij 24 weken. *Nederlands Tijdschrift voor Geneeskunde*, *161*, D1168.

Anderson, B. J. (2010). Pediatric models for adult target-controlled infusion pumps. *Pediatric Anesthesia*, *20*(3), 223–232.

Anderson, B. J., & Hodkinson, B. (2010). Are there still limitations for the use of target-controlled infusion in children? *Current Opinion in Anesthesiology*, *23*(3), 356–362.

Askie, L. M., Brocklehurst, P., Darlow, B. A., Finer, N., Schmidt, B., & Tarnow-Mordi, W. (2011). Neoprom: Neonatal oxygenation prospective meta-analysis collaboration study protocol. *BMC pediatrics*, *11*(1), 6.

Bressan, N., James, A., & McGregor, C. (2013). Integration of drug dosing data with physiological data streams using a cloud computing paradigm. In *Engineering in medicine and biology society (embc), 2013 35th annual international conference of the ieee* (pp. 4175–4178).

Catley, C., Smith, K., McGregor, C., James, A., & Eklund, J. M. (2010). A framework to model and translate clinical rules to support complex real-time analysis of physiological and clinical data. In *Proceedings of the 1st acm international health informatics symposium* (pp. 307–315).

Chedoe, I., Molendijk, H. A., Dittrich, S. T., Jansman, F. G., Harting, J. W., Brouwers, J. R., & Taxis, K. (2007). Incidence and nature of medication errors in neonatal intensive care with strategies to improve safety. *Drug safety*, *30*(6), 503–513.

Claure, N., & Bancalari, E. (2015). Closed-loop control of inspired oxygen in premature infants. In *Seminars in fetal and neonatal medicine* (Vol. 20, pp. 198–204).

Courtney, K., et al. (2013). Analysis of continuous oxygen saturation data for accurate representation of retinal exposure to oxygen in the preterm infant. *Enabling Health and Healthcare Through ICT: Available, Tailored and Closer*, *183*, 126.

De Staat van Volksgezondheid en Zorg. (2017). *Geboorten: kerncijfers.* Retrieved 2018-02-23, from `https://www.staatvenz.nl/kerncijfers/geboorten`

De Kluiver, E., Offringa, M., Walther, F., et al. (2013). Perinataal beleid bij extreme vroeggeboorte. *Nederlands Tijdschrift voor Geneeskunde*, *157*, A6362.

Dimitrakakis, C., & Rothkopf, C. A. (2011). Bayesian multitask inverse reinforcement learning. In *European workshop on reinforcement learning* (pp. 273–284).

Eklund, J. M., Fontana, N., Pugh, E., McGregor, C., Yielder, P., James, A., ... McNamara, P. (2014). Automated sleep-wake detection in neonates from cerebral function monitor signals. In *Computer-based medical systems (cbms), 2014 ieee 27th international symposium on* (pp. 22–27).

Escandell-Montero, P., Chermisi, M., Martínez-Martínez, J. M., Gómez-Sanchis, J., Barbieri, C., Soria-Olivas, E., ... others (2014). Optimization of anemia treatment in hemodialysis patients via reinforcement learning. *Artificial intelligence in medicine*, *62*(1), 47–60.

Gilfillan, M., & Bhandari, V. (2017). Biomarkers for the diagnosis of neonatal sepsis and necrotizing enterocolitis: Clinical practice guidelines. *Early human development*, *105*, 25–33.

Griffin, M. P., Lake, D. E., O'shea, T. M., & Moorman, J. R. (2007). Heart rate characteristics and clinical signs in neonatal sepsis. *Pediatric research*, *61*(2), 222–227.

Hasselt, H. V. (2010). Double q-learning. In *Advances in neural information processing systems* (pp. 2613–2621).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Janssen, K. J., Donders, A. R. T., Harrell Jr, F. E., Vergouwe, Y., Chen, Q., Grobbee, D. E., & Moons, K. G. (2010). Missing covariate data in medical research: to impute is better than to ignore. *Journal of clinical epidemiology*, *63*(7), 721–727.

J. Wieringa, R. (2014). *Design science methodology for information systems and software engineering.*

Kappen, T. H. (2015). *Prediction models and decision support: Chances and challenges.* Utrecht University.

Kappen, T. H., & Peelen, L. M. (2016). Prediction models: the right tool for the right problem. *Current Opinion in Anesthesiology*, *29*(6), 717–726.

Kappen, T. H., Van Loon, K., Kappen, M. A., Van Wolfswinkel, L., Vergouwe, Y., Van Klei, W. A., ... Kalkman, C. J. (2016). Barriers and facilitators perceived by physicians when using prediction models in practice. *Journal of clinical epidemiology*, *70*, 136–145.

Kneepkens, C., van Rijswijk, H., Pieters, R., de Beaufort, A., van Heurn, L., Drexhage, V., ... van Kaam, A. (2005). *Neonatologie.* Bohn Stafleu van Loghum.

Konur, S. (2010). Real-time and probabilistic temporal logics: An overview. *arXiv preprint arXiv:1005.3200*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (Vol. 30, p. 3).

Mani, S., Ozdas, A., Aliferis, C., Varol, H. A., Chen, Q., Carnevale, R., ... Weitkamp, J.-H. (2014). Medical decision support using machine learning for early detection of late-onset neonatal sepsis. *Journal of the American Medical Informatics Association*, *21*(2), 326–336.

Martín-Guerrero, J. D., Gomez, F., Soria-Olivas, E., Schmidhuber, J., Climente-Martí, M., & Jiménez-Torres, N. V. (2009). A reinforcement learning approach for individualizing erythropoietin dosages in hemodialysis patients. *Expert Systems with Applications*, *36*(6), 9737–9742.

McGregor, C., Catley, C., & James, A. (2012). Variability analysis with analytics applied to physiological data streams from the neonatal intensive care unit. In *Computer-based medical systems (cbms), 2012 25th international symposium on* (pp. 1–5).

Miotto, R., Li, L., Kidd, B. A., & Dudley, J. T. (2016). Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, *6*, 26094.

Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2017). Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Moore, B. L., Doufas, A. G., & Pyeatt, L. D. (2011). Reinforcement learning: a novel method for optimal control of propofol-induced hypnosis. *Anesthesia & Analgesia*, *112*(2), 360–367.

Moore, B. L., Sinzinger, E. D., Quasny, T. M., & Pyeatt, L. D. (2004). Intelligent control of closed-loop sedation in simulated icu patients. In *Flairs conference* (pp. 109–114).

Naik, T., Bressan, N., James, A., & McGregor, C. (2013). Design of temporal analysis for a novel premature infant pain profile using artemis. *Journal of Critical Care*, *28*(1), e4.

Nederlandse Vereniging voor Kindergeneeskunde. (2010). *Nvk richtlijn: Vroeggeboorte, perinataal beleid bij extreme vroeggeboorte.*

Nemati, S., Ghassemi, M. M., & Clifford, G. D. (2016). Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *Engineering in medicine and biology society (embc), 2016 ieee 38th annual international conference of the* (pp. 2978–2981).

Ng, A. Y., Russell, S. J., et al. (2000). Algorithms for inverse reinforcement learning. In *Icml* (pp. 663–670).

Nissen, M. D. (2007). Congenital and neonatal pneumonia. *Paediatric respiratory reviews*, *8*(3), 195–203.

Passot, S., Servin, F., Allary, R., Pascal, J., Prades, J.-M., Auboyer, C., & Molliex, S. (2002). Target-controlled versus manually-controlled infusion of propofol for direct laryngoscopy and bronchoscopy. *Anesthesia & Analgesia*, *94*(5), 1212–1216.

Pham, T., Tran, T., Phung, D., & Venkatesh, S. (2016). Deepcare: A deep dynamic memory model for predictive medicine. In *Pacific-asia conference on knowledge discovery and data mining* (pp. 30–41).

Prasad, N., Cheng, L.-F., Chivers, C., Draugelis, M., & Engelhardt, B. E. (2017). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint arXiv:1704.06300*.

Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., & Ghassemi, M. (2017). Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422*.

Rocheteau, E. (2012). What will british healthcare look like in 20 years' time? *analysis*, *7*, 11–13.

Saugstad, O. D., & Aune, D. (2011). In search of the optimal oxygen saturation for extremely low birth weight infants: a systematic review and meta-analysis. *Neonatology*, *100*(1), 1–8.

Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

Shane, A. L., Sánchez, P. J., & Stoll, B. J. (2017). Neonatal sepsis. *The Lancet*.

Shearer, C. (2000). The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, *5*(4), 13–22.

Shiarlis, K., Messias, J., & Whiteson, S. (2016). Inverse reinforcement learning from failure. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems* (pp. 1060–1068).

Silverman, E. K., & Loscalzo, J. (2013). Developing new drug treatments in the era of network medicine. *Clinical Pharmacology & Therapeutics*, *93*(1), 26–28.

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (Vol. 135). MIT Press Cambridge.

Thommandram, A., Eklund, J. M., McGregor, C., Pugh, J. E., & James, A. G. (2014). A rule-based temporal analysis method for online health analytics and its application for real-time detection of neonatal spells. In *Big data (bigdata congress), 2014 ieee international congress on* (pp. 470–477).

TNO. (2017). *Project on preterm and small for gestational age infants in the netherlands.*

Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Aaai* (pp. 2094–2100).

Van Kaam, A. H., Hummler, H. D., Wilinska, M., Swietlinski, J., Lal, M. K., Te Pas, A. B., ... others (2015). Automated versus manual oxygen control with different saturation targets and modes of respiratory support in preterm infants. *The Journal of pediatrics*, *167*(3), 545–550.

Van Wesel, P., & Goodloe, A. E. (2017). Challenges in the verification of reinforcement learning algorithms.

van Zanten, H. A. (2017). *Oxygen titration and compliance with targeting oxygen saturation in preterm infants* (Unpublished doctoral dissertation). Leiden University.

van Zanten, H. A., Pauws, S. C., Stenson, B. J., Walther, F. J., Lopriore, E., & te Pas, A. B. (2017). Effect of a smaller target range on the compliance in targeting and distribution of oxygen saturation in preterm infants. *Archives of Disease in Childhood-Fetal and Neonatal Edition*, fetalneonatal–2016.

van Zanten, H. A., Tan, R. N., van den Hoogen, A., Lopriore, E., & te Pas, A. B. (2015). Compliance in oxygen saturation targeting in preterm infants: a systematic review. *European journal of pediatrics*, *174*(12), 1561–1572.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292.

Wells, B. J., Chagin, K. M., Nowacki, A. S., & Kattan, M. W. (2013). Strategies for handling missing data in electronic health record derived data. *eGEMs*, *1*(3).

World Health Organization. (2017). *Preterm birth: fact sheet.* Retrieved from `http://www.who.int/mediacentre/factsheets/fs363/en/`

Zhao, Y., Kosorok, M. R., & Zeng, D. (2009). Reinforcement learning design for cancer clinical trials. *Statistics in medicine*, *28*(26), 3294–3315.

Zikos, D. (2017). A framework to design successful clinical decision support systems. In *Proceedings of the 10th international conference on pervasive technologies related to assistive environments* (pp. 185–188).

Appendices

Appendix A

Complications of preterm birth

Information about these complications and treatments are mostly based on a Dutch book about Neonatology written by paediatricians (Kneepkens et al., 2005). If another source is used it will be references to.

A.1 Respiratory system

A complication of pre-term birth is underdevelopment of the respiratory system. This may lead to the following complications described in the next subsections.

A.1.1 Apnea

The absence of spontaneous breathing for a consecutive period of at least 15 seconds. This is usually caused by immaturity of the respiratory system. There are three types of apnea: (1) central apnea: the respiratory muscles are not activated by the respiratory center in the brainstem, (2) obstructive apnea: the respiratory muscles are activated but the (upper) airways are obstructed and the air can't enter the lungs, and (3) mixed apnea: most common type of apnea, which is a combination of both central and obstructive apnea. Apnea occurs at almost 100 percent of the infants born before 32 weeks. *Diagnosis*: One of the symptoms is central cyanosis, which is the bluish or purplish discolouration of the skin, due to hypoxia (see next section). Another symptom is bradycardia (see Section A.2). *Treatment*: As apnea can be caused by a variety of neonatal syndromes, it can be hard to determine what causes the apnea. In some situations apnea can be remedied by repositioning the infant, removing nasal mucus, or better regulation of the body temperature. In other cases antibiotics are needed to treat an untreated infection which can be cause of the apnea. One of the most used types of medication used to treat apnea are *methylxanthines*, for example *caffeine* or *theofylline*. These increase the heartbeat which causes the heart to pump the blood through faster and work as a bronchodilator, a substance that dilates the bronchi and bronchioles, decreasing resistance in the respiratory airway and increasing airflow to the lungs. Apnea generally resolves as the preterm infant matures.

A.1.2 Respiratory Distress Syndrome (RDS)

RDS is associated with surfactant deficiency combined with structural immaturity of the lungs. Symptoms are tachypnea (abnormally rapid breathing), dyspnea (shortness of breath that comes with grunting, flaring nostrils and subcostal retractions of the chest) and cyanosis. Infants born before 28 weeks have approximately 80 percent chance of developing RDS. *Diagnosis*: The above mentioned symptoms usually develop in the first six hours after birth and progress during the first 24 hours. Important is to differentiate between RDS, transient tachypnea of the newborn (TTN) and pneumonia. The most significant diagnostic for RDS is to examine radiographic imaging of the thorax. This shows whether the lungs suffer from underaeration and have a reduced lung volume, which are indications of RDS. *Treatment*: RDS is an acute illness treated with respiratory support (oxygen, positive airway pressure, ventilator, or surfactant) as needed and improves in 2 to 4 days and resolves in 7 to 14 days. Because RDS is difficult to distinguish from other infections, infants with respiratory distress are generally treated with antibiotics. Next to this the infants are treated with a modified natural surfactant.

A.1.3 Neonatal pneumonia

Opposed to adult pneumonia neonatal pneumonia is not clearly defined and often hard to identify (Nissen, 2007). It is an infection in the area of the lungs due to compromised lung defences of premature neonates. There are two types of pneumonia: (1) congenital pneumonia, which is caused by an ascending infection in the birth canal of the mother, and (2) nosocomial pneumonia, which evolves during the hospitalization of newborns. *Diagnosis*: Both types of pneumonia have show different symptoms. Congenital pneumonia is similar to the RDS, with tachypnea, dyspnea and cyanosis, and even with a thorax image it can be hard to distinguish the two. Infants with nosocomial pneumonia show aspecific symptoms and a thorax image shows similarities with bronchopulmonary dysplasia (BPD). *Treatment*: Neonatal pneumonia is treated with antibiotics, supplemental oxygen and, when it hard to distinguish from RDS, a modified surfactant.

A.1.4 Transient Tachypnea of the Newborn (TTN)

Respiratory problems due to slow absorption of the fluid in the fetal lungs. The symptoms are similar to those of RDS, but in contrast to RDS tachypnea is more present than dyspnea. TTN occurs mostly at almost full term infants and most infants are no longer showing symptoms after 24 hours. *Diagnosis*: Although the clinical course of TTN is characteristics, addition investigation can be done to differentiate it from other pulmonary complications. A radiographic image of the thorax will show a normal aeration of the lungs but a large lung volume, which distinguishes it from RDS. *Treatment*: In most cases supplemental oxygen is sufficient.

A.2 Cardiovascular system

Next to the respiratory system, is also the cardiovascular system of pre-term infants often underdeveloped. The next subsections describe possible complications of this.

A.2.1 Bradycardia

Reduction of heart rate. For premature neonates < 100 beats per minute and < 80 beats per minute for full-term neonates. One of the causes for bradycardia is hypoxia (see section Section A.2). *Diagnosis*: Analysis of the heart rate (HR) can demonstrate bradycardia. If the bradycardia occurs without the manifestation of hypoxia there might be an atrioventricular block (AV block), which can be diagnosed based on an electrocardiography (ECG). *Treatment*: As bradycardia is a result of other complications, the treatment of these complication is essential in order to manage the bradycardia.

A.2.2 Tachycardia

High heart beat frequency of > 180 beats per minute for premature infants and > 150 beats per minute for full-term neonates. *Diagnosis*: Analysis of the heart rate (HR) can show if the infant suffers from tachycardia. *Treatment*: Short term tachycardia is not unusual for newborns, but if the tachycardia persist for a longer period the underlying cause should be investigated.

A.2.3 Persistent Pulmonary Hypertension of the Newborn (PPHN)

PPHN is defined as failure of the normal circulatory transition that occurs after birth. It is a syndrome characterized by marked pulmonary hypertension that causes right-to-left shunting of blood at the foramen ovale and ductus arteriosus and hypoxia. *Diagnosis*: Can be caused by for example pneumonia, sepsis or RDS and usually leads to cyanosis. *Treatment*: The underlying causes of the PPHN should be investigated and treated. One of the most frequent treatments is respiratory support.

Appendix B

Reward function

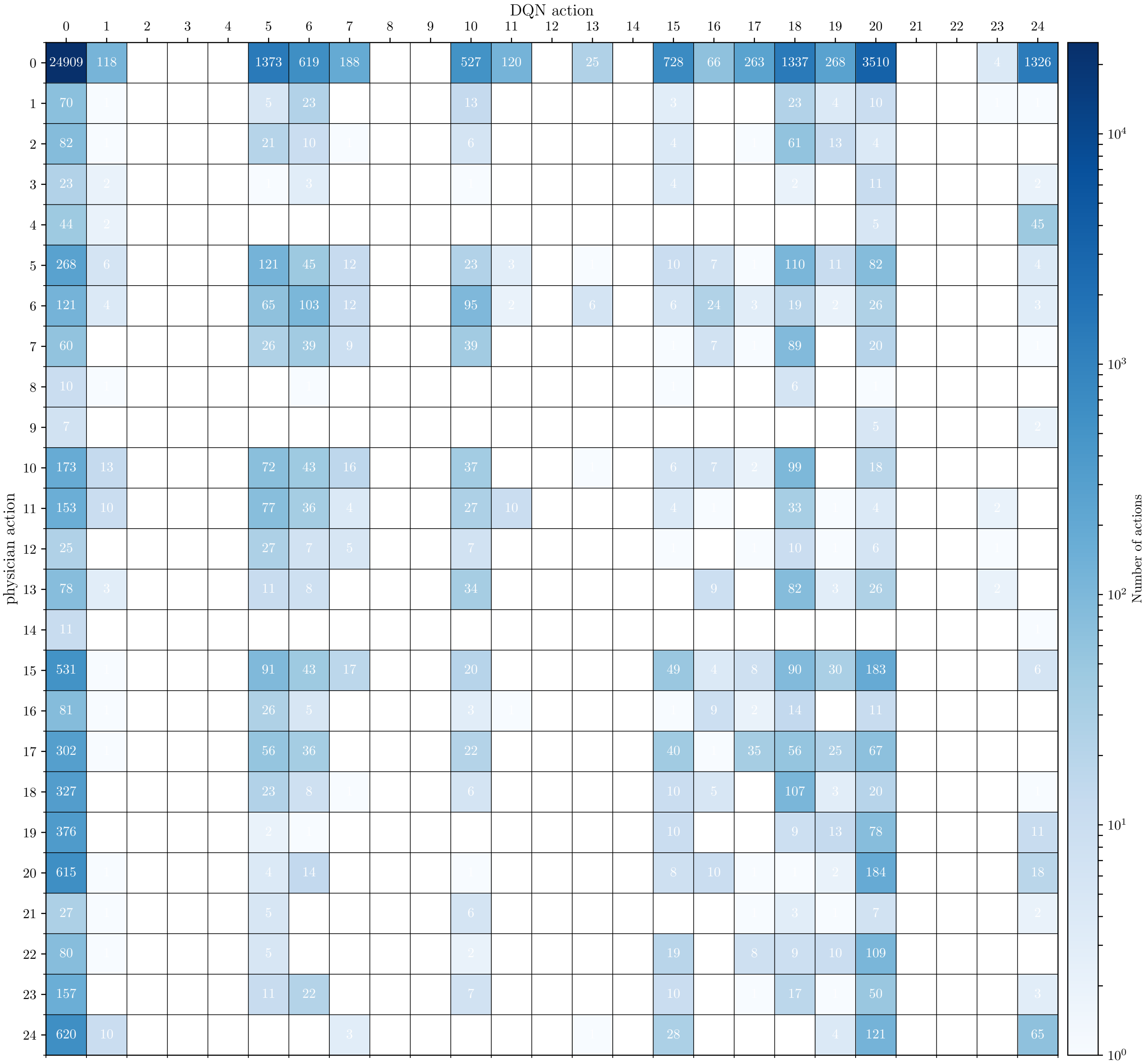
```
def calc_temp_reward(row):
        temp_reward = 0
        if (row.Temp_Avg_Int >= 36) & (row.Temp_Avg_Int <= 38):
                temp_reward = (1- abs(37 - row.Temp_Avg_Int)) * 5
        if (row.Temp_Avg_Int < 36):
                temp_reward = abs(36 - row.Temp_Avg_Int) * -1
        if (row.Temp_Avg_Int > 38):
                temp_reward = abs(36 - row.Temp_Avg_Int) * -1
        return temp_reward

def calc_FiO2_reward(row):
        FiO2_reward = 0
        if row.FiO2_Avg <= 20:
                FiO2_reward = ((20 - row.FiO2_Avg) * 0.25)
        else:
                FiO2_reward = ((row.FiO2_Avg-20) * -0.5)
        if FiO2_reward < -5:
                FiO2_reward = -5
        return FiO2_reward

data['temp_reward'] = data.apply(calc_temp_reward, axis=1)
data['FiO2_reward'] = data.apply(calc_FiO2_reward, axis=1)
data['reward'] = data['temp_reward'] + data['FiO2_reward']
```

Appendix C

Verification



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	24909	118				1373	619	188			527	120		25		728	66	263	1337	268	3510			4	1326
1	70	1				5	23				13					3			23	4	10			1	1
2	82	1				21	10	1			6					4		1	61	13	4				
3	23	2				1	3				1					4			2		11				2
4	44	2																			5				45
5	268	6				121	45	12			23	3		1		10	7	1	110	11	82				4
6	121	4				65	103	12			95	2		6		6	24	3	19	2	26				3
7	60					26	39	9			39					1	7	1	89		20				1
8	10	1					1									1			6		1				
9	7																				5				2
10	173	13				72	43	16			37			1		6	7	2	99		18				
11	153	10				77	36	4			27	10				4	1		33	1	4			2	
12	25					27	7	5			7					1		1	10	1	6			1	
13	78	3				11	8				34						9		82	3	26			2	
14	11																								1
15	531	1				91	43	17			20					49	4	8	90	30	183				6
16	81	1				26	5				3	1				1	9	2	14		11				
17	302	1				56	36				22					40	1	35	56	25	67				
18	327					23	8	1			6					10	5		107	3	20				1
19	376					2	1									10			9	13	78				11
20	615	1				4	14				1					8	10	1	1	2	184				18
21	27	1				5					6							1	3	1	7				2
22	80	1				5					2					19		8	9	10	109				
23	157					11	22				7					10		1	17	1	50				3
24	620	10						3						1		28				4	121				65

Figure C.1: Action to action mapping

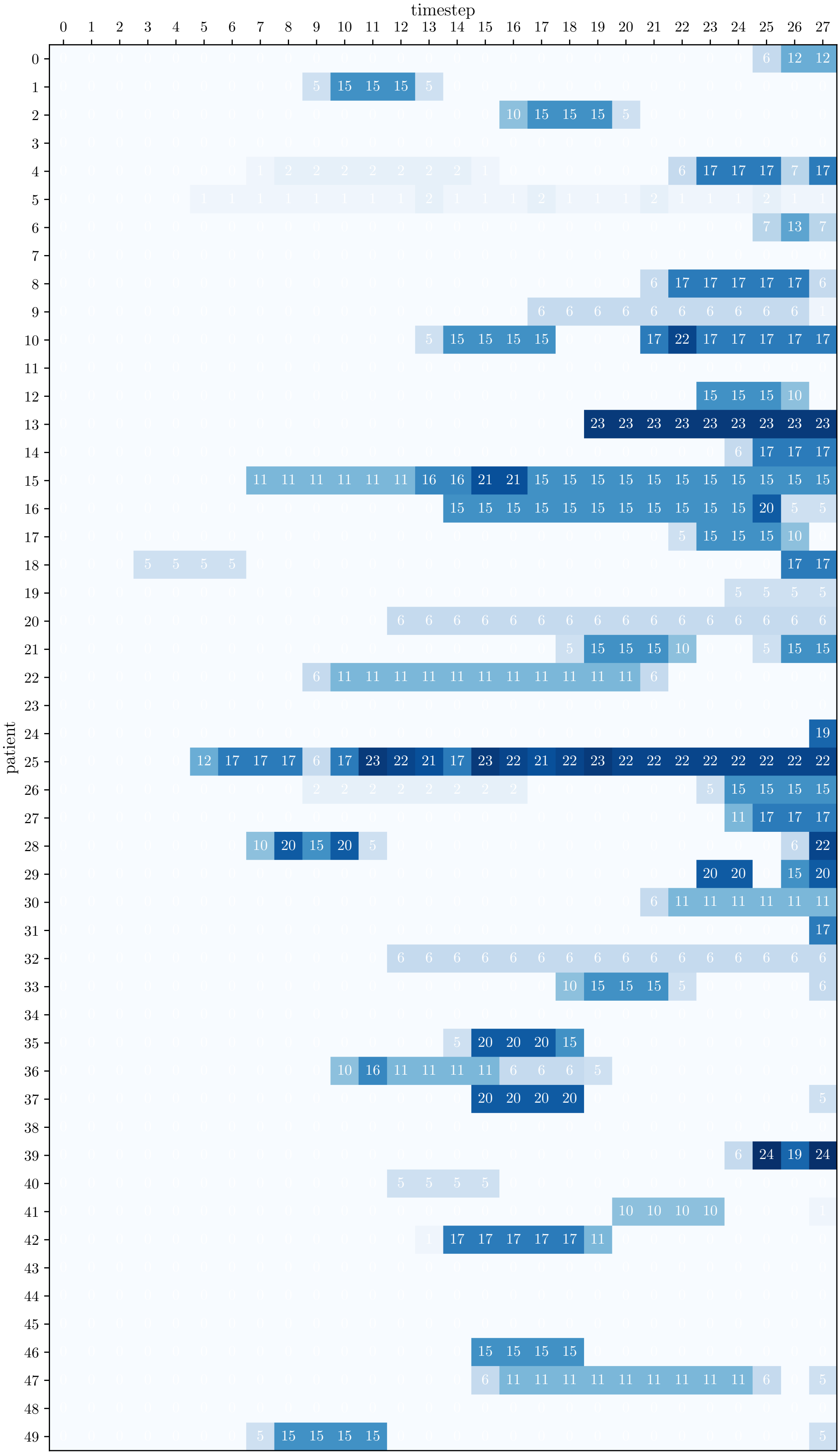
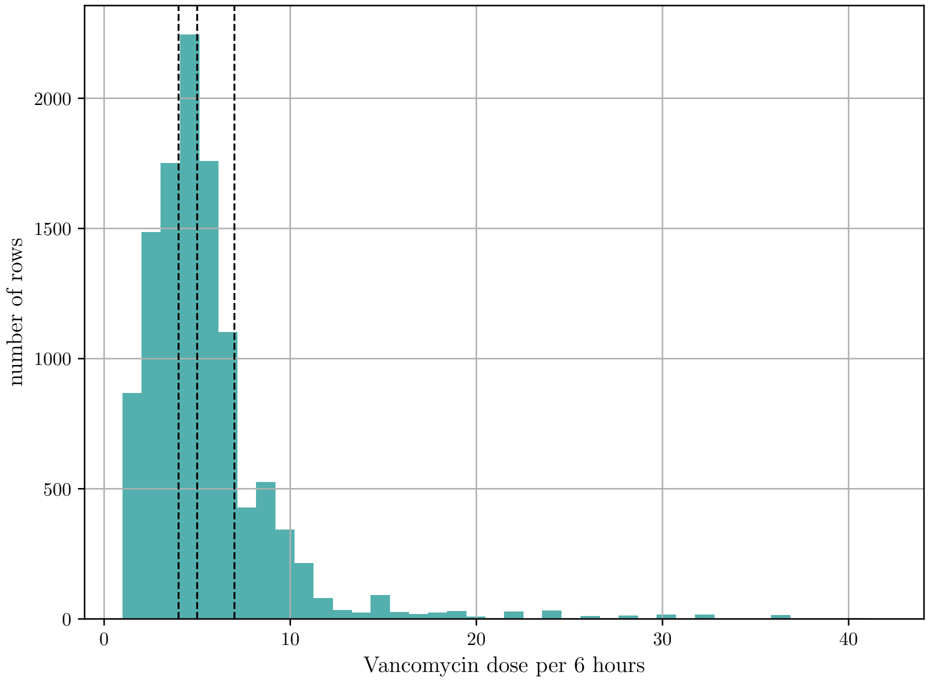


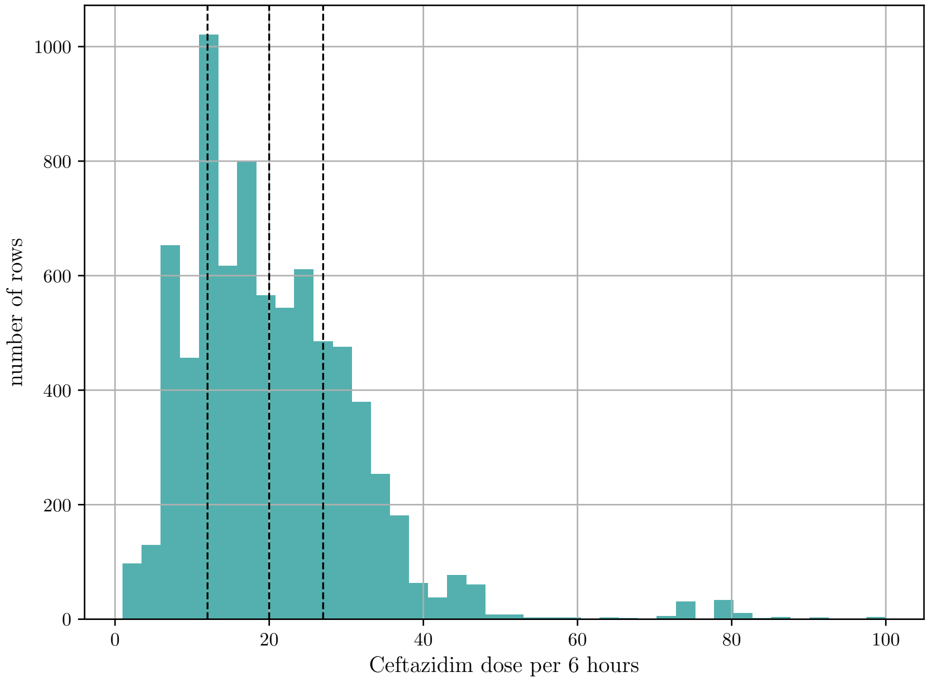
Figure C.2: Action sequences

Appendix D

Medication doses



(a) Vancomycin dosing (excluding zero dose) with the quantile cut points at 4,5 and 7



(b) Ceftazidim dosing (excluding zero dose) with the quantile cut points at 12, 20 and 27

Figure D.1: Histograms of medication dosing per 6 hours

Appendix E

Respiration

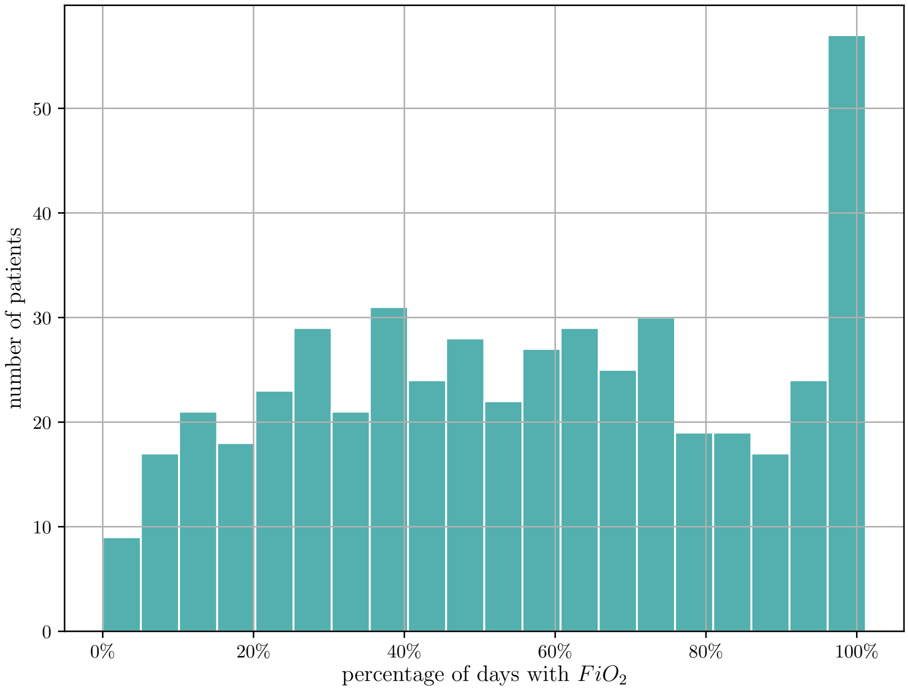


Figure E.1: Percentage of days with additional inspired oxygen (FiO_2)

Appendix F

Density

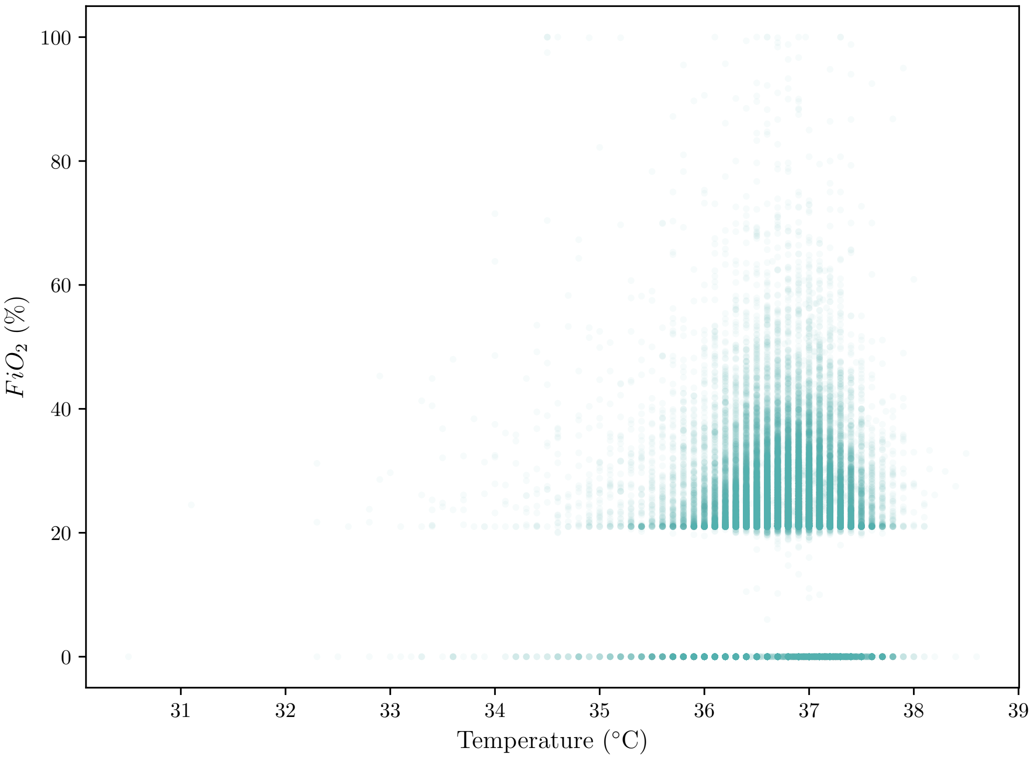


Figure F.1: Density chart of the patients' states