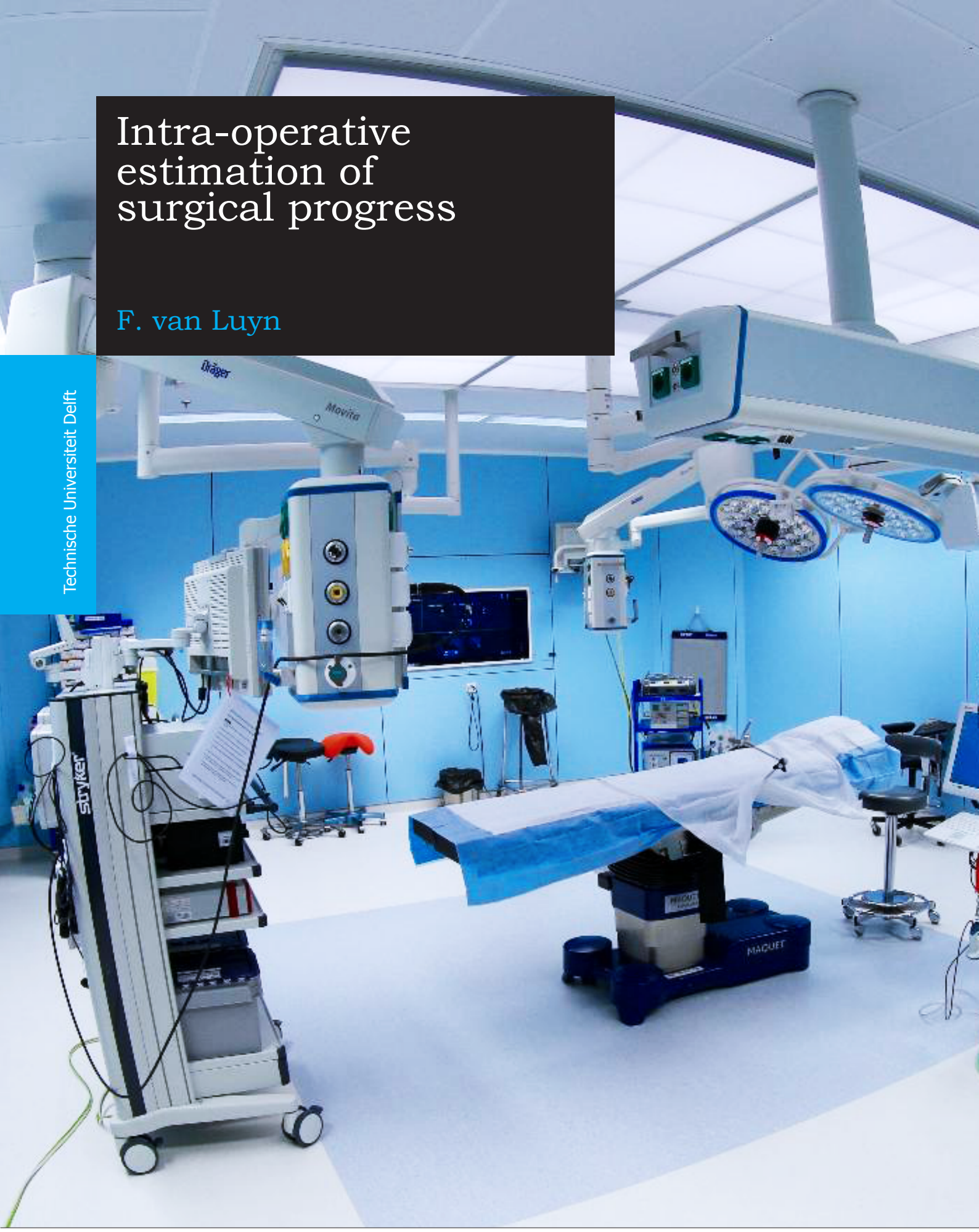
Intra-operative estimation of surgical progress

F. van Luyn

Technische Universiteit Delft



TU Delft Delft University of Technology

Challenge the future

Intra-operative estimation of surgical progress

by

F. van Luyn

in partial fulfillment of the requirements for the degree of

Master of Science
in Biomedical Engineering

at the Delft University of Technology,
to be defended publicly on Friday April 28, 2017.

Supervisor:	dr. ir. J. J. van den Dobbelsteen	
Thesis committee:	Prof. dr. J. Dankelman	TU Delft
	F. C. Meeuwsen, M.D.	TU Delft
	dr. J. F. P. Kooij	TU Delft

TU Delft Delft University of Technology

Acknowledgements

This thesis is the culmination of a project that started in the beginning of 2016. I had just returned from a year of studying and working abroad in London and was looking for a challenging master's project that would combine a technical assignment with a clear clinical application. I found this in the DORA (Digital Operating Room Assistant) project, an effort already spanning multiple years and several PhD and master projects, in which numerous organizations have cooperated in advancing workflow in the operating room. After a review of literature, the aim was to bring real-time instrument tracking into the OR using RFID-technology. However, plan and reality did not really align - which as I am told is often the case in research. Although I since have set foot into an actual OR several times, the research lying in front of you has been done safely from behind the screen of my computer, using simulations on previously recorded data. Nevertheless, I believe an important step has been made in realizing real-time phase recognition through this work. Not only due to the promising results, but also due to the fact that an enthusiastic and capable team has been formed, including people from academia, business and the hospital, who will undoubtedly continue this fascinating project in the future.

Of course, this thesis has only been finished due to the efforts of many. Annetje, thank you for introducing me to the subject and very enthusiastically supervising me during the final months of your stay at the TU Delft. Frédérique, thank you for taking up this role when I was suddenly left without a daily supervisor. John, thank you for your supervision during the whole project, your sharp questions during our enjoyable meetings allowed me to keep a clear overview of the project and the end-goals. Thanks to Jenny, for agreeing to lead my examination committee and your interest in this research, also during the project. I would furthermore like to thank Arjan for the technical help, especially by allowing me to work remotely and the thank rest of the MISIT lab for their friendly welcome, whenever I walked in. I am grateful to Mathijs Blikkendaal and the Leiden University Medical Centre, who unreservedly provided the necessary data set to perform this research project and whose feedback was valuable. The modelling work in this thesis has much benefited from suggestions by David Tax. Thanks to Maarten van der Elst, Marion Poot, Rick Schoffelen and everyone else from the Reinier de Graaf Gasthuis for their enthusiastic welcoming of this research project and to dr. Roos and dr. Andriessen for showing me the OR during actual surgeries. Also, thanks to the industry partners in this project, Van Straten Medical and Bexter, as your insights and materials were of much use. An extra mention for Bart van Straten, for his pioneering in this project. Finally, a special thanks to my friends from Delft, you have made my years at the university an unforgettable experience, and to my parents for their unconditional trust and support.

Fabian van Luyn
The Hague, April 2017

Contents

List of abbreviations

The following abbreviations are used throughout this thesis:

Technical abbreviations

ACC: Accuracy

ANN: Artificial neural network

CART: Classification and regression trees

DTW: Dynamic time warping

FN/FP: False negative/positive

HLT/LLT: High-/low-level task

HMM: Hidden Markov model

KNN: K-nearest neighbor

LLR: Log-linear regression

LR: Linear regression

MAE: Mean absolute error

MAPE: Mean absolute percentage error

MARS: Multivariate adaptive regression splines

MCMC: Monte Carlo Markov Chain

MDA: Mean decrease in accuracy

MDI: Mean decrease in impurity

RF: Random Forest

RFID: Radio-Frequency identification

RMSE: Root mean squared error

SEN: Sensitivity

SPC: Specificity

SVM: Support vector machines

TN/TP: True negative/positive

Clinical abbreviations

CAS: Computer-aided surgery

LH: Laparoscopic hysterectomy

OR: Operating room

SPM: Surgical process model

List of Figures

List of Tables

Abstract

Driven by rising health care costs due to factors including advancing technology and an aging population, cost-effectiveness has become an increasingly important aspect of care delivery, with the operating room (OR) being a specific area of interest. By means of a surgical process model (SPM), OR systems can gain an understanding of clinical context and surgical workflow, hereby generating ample opportunities to improve OR logistics and surgical care. Applications of SPM's include intra-operative end-time predictions, improved surgical training and assessment, computer-aided surgery and increased autonomy in robotic surgery.

This thesis evaluates the use of an SPM for intra-operative recognition of surgical phases in laparoscopic hysterectomy cases (n=40), based on manually annotated instrument usage data. Using a Random Forest model, an out-of-sample accuracy of 77% is achieved. The phase-recognition model is shown to predict surgical end-times with a mean absolute error of 16 minutes and is additionally found useful in the task of surgical phase extraction. Further research should specifically be aimed at replicating the promising simulated findings of this thesis in-vivo, using intra-operative sensor recordings in the OR.

1

Introduction

Driven by increasing health care costs due to factors including an aging population, cost-effectiveness has become an increasingly important aspect of care delivery. With 60% of hospital-admitted patients being treated in the operating room (OR) and surgical care taking up to 40% of the hospital budget, the OR is a specific area of interest for improving hospital efficiency [1, 2]. At the same time, another important driver of increasing costs in health care is the advancement of technology. Over the last decades the OR has evolved into a complex environment, filled with high-tech devices [3]. However, these technological advances have also made information and communication technology ubiquitously available within the OR, which will ultimately allow for improved care and less cost-intensive surgical interventions [4].

The field of *computer-aided surgery* (CAS) explores the numerous ways in which computer-systems can aid the medical team before and during surgical procedures [5]. Examples are assistive technologies for surgical navigation and automatic adaption of the OR settings to a specific procedure, for instance by changing the configuration of the OR table, the position and display of the OR screens and the level of illumination within the OR [6]. Other applications of CAS can be found in the display of additional patient-specific medical information and measurements of vital signs to help the clinical team in their decision making. Future applications of CAS include active robotic assistance, automated surgical reporting and augmented reality [3, 7, 8].

Figure 1.1: Over the last decades, the operating room (OR) has been filled with high-tech devices, as shown in this photo of the recently renewed OR of the Reinier de Graaf Gasthuis in Delft, The Netherlands. Image accessed online 10/03/2017: https://www.reinierdegraaf.nl/algemeen/nieuws/reinier-de-graaf-ziekenhuis-en-tu-delft-bekrachtigen-innovatieve-samenwerking/)

Computer-aided surgery has the potential to increase surgical outcomes, by detecting adverse events and helping clinicians with clinical tasks and decision making. It also has the potential to reduce health care costs by increasing OR workflow and efficiency, for example by predicting surgical end-times [6, 9].

Context-Aware Operating Room

In order for CAS systems to aid the surgical team in a meaningful way, these systems need to receive information from a *surgical process model (SPM)*, a formalized representation of the surgical procedure [3]. An SPM is built to autonomously detect and recognize different steps in the surgical workflow, hereby realising situational, context-awareness. This *context-awareness* results for example in knowledge about the surgical phase and the specific tasks performed based on information from instruments used and anatomical structures involved [7]. Just as a human surgical assistant would interpret the actions of the surgical team and act accordingly, an SPM acquires, processes and interprets data.

The context-aware OR uses data that is recorded during the actual surgery, to make real-time inferences on the current surgical process. Intra-operative data may be obtained from many sources, of which video and instrument tracking are the most prominently used [3]. Video recordings, either from the OR as a whole or the operative field and laparoscopic view specifically, provide most information. However, real-time processing and analyzing of video is challenging and the recordings induce additional privacy concerns for patient and physicians [9, 10]. Instrument tracking can be realized using currently available technology, for example through the use of Radio-Frequency Identification (RFID) tags [6, 9, 11].

Next to the empowerment of CAS systems, the SPM finds use in optimization of hospital logistics, for example by predicting surgical case durations [12–14], surgical skill assessment and training [15, 16] and surgical robotics [17, 18]. Based on the projected use of the model, the structure of an SPM may vary from a sequential list of surgical phases, to a complex ontology describing relations between each minor step in the procedure, together with the staff and anatomical structure involved [7, 19].

1.1. Problem Statement and Research Goal

The advantages of a context-aware operating room range from increased OR efficiency to better clinical outcomes, but to reap these benefits a surgical process model is needed that is able to understand clinical context from intra-operative surgical data. The problem can therefore be stated as follows:

Problem Statement

In order to realize a context-aware operating room, a surgical process model is needed that can infer surgical context from intra-operative surgical data.

Some of the possibilities created by a context-aware OR, such as autonomous robotic surgery, are a distant prospect, because of the high context-awareness and detailed surgical process models needed for successful implementation. Other applications, such as the predictions of surgical end-times and the generation of surgical training material, can thrive based on a relatively simple phase detection system, that can automatically detect major events within a surgery. In addition, it can be noted that instrument detection currently provides a promising trade-off between complexity in automatic processing and possibilities for generating surgical context. As instrument tracking can be implemented in-vivo in an operating room using currently available technology, this thesis will focus on surgical phase recognition based on intra-operative data of surgical instrument use, leading to the following research question:

Research Question

What is the performance of surgical phase recognition models based on intra-operative data of instrument use, with particular regard to application in surgical end-time prediction?

In order to recognize surgical phases during a procedure, a model needs to be developed that infers the phases from instrument usage information, leading to the first research goal. Furthermore, the aim is to assess the applications of the phase recognition system for clinical practice, hence simulations of in-vivo clinical tasks will also be performed.

Research Goals

1. *Predict surgical phases intra-operatively based on real-time data of instrument use.*
2. *Evaluate the surgical phase recognition model on simulated clinical tasks, including surgical end-time predictions.*

1.2. Thesis Outline

The next chapter provides additional *Background* into the topics of this thesis. The chapter includes a look into modelling surgical processes and the choices that can be made in terms of model granularity and formalization, data acquisition and analysis and model application, as well as a review of previous literature on these topics.

The chapter on *Methods and Materials* outlines the approach to surgical phase recognition in laparoscopic hysterectomy procedures. A distinction is made between an approach using multinomial classification models and an approach based on a Hidden Markov-model of the surgical workflow.

The *Results* chapter shows the performance of the different classifiers and the state-space model. The selected Random Forest model is applied to the clinical tasks of end-time prediction and surgical phase extraction.

In the *Discussion*, the performance of the selected phase recognition model is compared with previous literature and the application to the simulated clinical tasks of surgical end-time recognition and surgical phase extraction are reviewed, leading to a *conlusion* of the current work and suggestions for further research.

In *Appendix A*, preliminary work on prototyping an RFID-based instrument-tracking system is detailed, a plausible approach for putting the phase recognition system into actual clinical practice. The work includes a set-up using off-the-shelf RFID technology and a custom MATLAB implementation allowing for live instrument tracking. *Appendix B* provides some details on the implementation of the phase recognition models.

2

Background[1]

2.1. Surgical Process Models

Surgeries are inherently variable, due to differences in patient anatomy, severity of the condition, the preferences of the surgeon and a myriad of other factors. As a result, the process and workflow within two distinct surgical cases is never exactly the same and the clinical team often needs to adapt to changing and unexpected circumstances. The variable environment of the OR collides with the way that computer systems operate, as computers need rigid characterizations and clear-defined structure to be able to interpret information. Surgical process models (SPM's) are designed to overcome this gap, by simplifying and structuring the surgical process and reflecting an interesting part of this process into a (semi-)formal representation [21]. The surgical process itself is broadly defined as a set of related procedures and actions that collectively realise a surgical objective [21]. The research of SPM's has been increasing rapidly over the past years and although this section will introduce relevant concepts, please see Lalys and Florent (2013) for a recent review [3].

Two important aspects of an SPM are the *granularity* and the *formalization*. Granularity refers to the level of abstraction and detail in which the surgical process is described (Figure 2.1). The naming of the different levels of detail can be somewhat arbitrary, hence nomenclature in this thesis aims to follow conventions in previous literature [3, 22, 23].

A surgical procedure can first be distinguished into different phases, which are defined by the major events occurring during the surgery. Each phase consists of certain steps, which are sets of activities that together achieve a surgical objective. An activity is a single physical task, for example cutting or palpating. Activities in turn consist of motions, which can for example be described by the movement of a hand through space. At the most granular level, we see low-level (sensor) data. An important observation regarding increasing granularity, is that while increasing detail, it can obscure surgical meaning. The low-level data, for example, has the most detail, but lacks a direct link to the purpose of the actions of the surgical team. At the same time, surgical phases describe the most important events in the surgery but have little detail of the actual events. An alternative nomenclature discriminates high-levels tasks (HLTs), which most often refer to phases, and low-level tasks (LLTs), referring to low level information such as data of instrument use or visual features [24, 25].

Formalization is another important aspect of an SPM and describes the way in which this information is structured (Figure 2.2). This formalization is to key to allowing human-machine communication, as computers generally only understand information structured in a pre-defined way. In attempts to capture medical and surgical information into a computer-readable form, several methods haven been proposed.

[1] Selected parts of the current chapter of this thesis are composed of relevant sections that, with adaptations, have been obtained from the literature review *'Online estimation of surgical progress'*, as written by the same author in May 2016 [20].

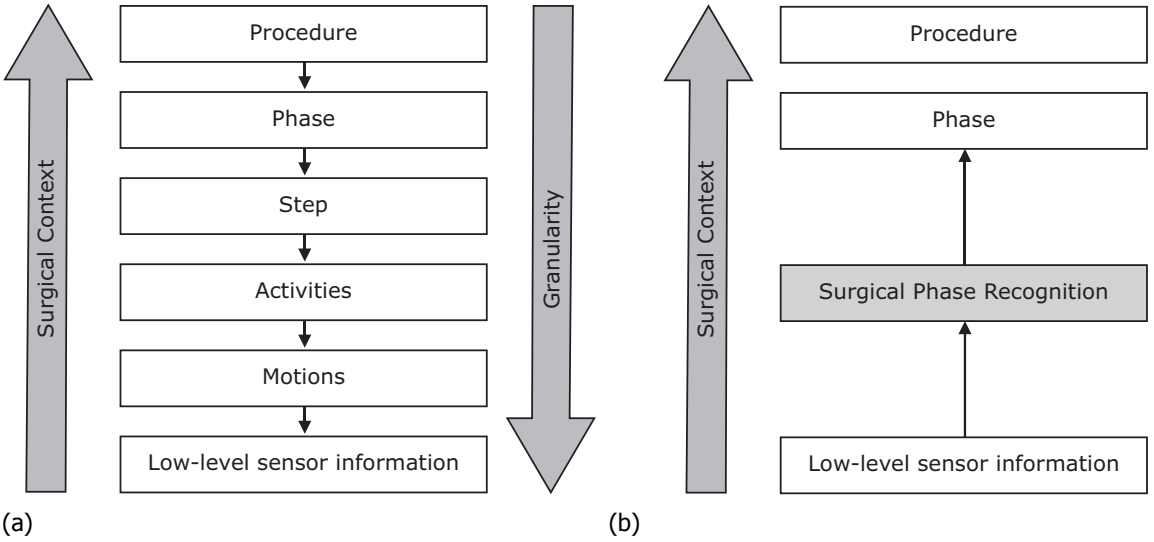


Figure 2.1: A surgical procedure can be decomposed into more detailed descriptions (left figure). Decomposition increases the granularity, but often reduces the surgical context. Previous literature commonly distinguishes phases, steps, activities, motions and low-level (sensor) information. The aim of a surgical phase recognition system (right figure) is to reconstruct the phase information, based on low-level information, often acquired using sensor recordings of intra-operative data. Figure adapted from [3].

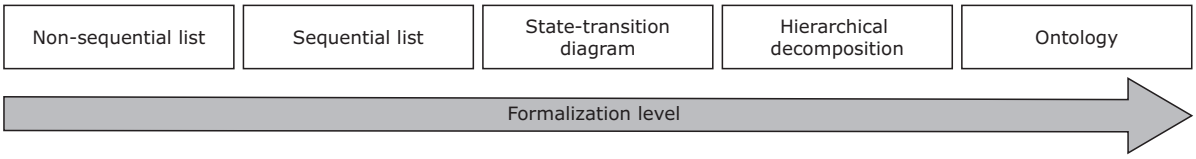


Figure 2.2: Different levels of formalization of surgical process models can be chosen, often based on the projected use of the SPM. The simplest way is to represent the surgical process model is in a non-sequential, or unordered, list. Sequential lists, state-transition diagrams, hierarchical decompositions and ontologies each increase the formalization level by adding more structure and= concepts to the SPM. Figure adapted from [3].

The simplest way to formalize a process is by means of classification into a non-sequential list, for example a list of surgical phases. Due to the time aspect in surgeries, one can logically extend this to a sequential or ordered lists, a list where one surgical phase follows another. Another extension towards increased formalization is the *state-transition diagram*. In this model, a limited set of states is defined in which a system can reside. The notions of system and state here are broad, for example the complete OR could resemble the system, having the states represent surgical phases. Next, *hierarchical decomposition* adds the notion of granularity to the state-transition diagram and specifies the hierarchical relation between for example phases, steps and activities. Finally, the surgical data can be represented using an *ontology*. The term ontology is derived from philosophy where it describes study into the nature of being. In informatics, the term is used to describe a formalization of knowledge and concepts into classes and their relations and functions [26]. Several ontologies have been designed specifically to model surgical progress, such as OntoSPM and LapOntoSPM [7, 19]. In these surgical ontologies, specific concepts relating to surgical processes and actions, surgical tools and instruments, patient anatomy, surgical team and clinical measurements are defined, together with their relations to each other. Ontologies therefore constitute the most elaborate and complex formalizations of SPMs.

2.2. Surgical Phase Recognition

Surgical phase recognition models can be seen as a subset of surgical process models, where the granularity is set at the level of surgical phases. Since a surgical phase is a rather coarse classification of the surgical process, the applications can be found mostly in the areas of predicting of surgical case durations, surgical training and automatic generation of post-operative reports. The following sections provide a brief overview of previous literature on these applications.

2.2.1. Predicting surgical case durations

A major application of surgical phase detection is the optimization of OR use by predicting surgical case durations. Although it is known that surgical duration is determined by a broad range of factors such as patient characteristics, individual surgical skills and occurrence of complications, the current methods of OR planning are often based only on either average surgery durations or estimates by the surgical staff [27]. As both average surgery duration and estimates of the surgical staff provide suboptimal predictive value on the real duration of the surgery, this limited approach on OR planning leads to inconsistencies between planned and actual surgery durations [28, 29].

Surgical procedures running over time will cause subsequent procedures to be delayed or cancelled. Next to the uncertainty and discomfort this brings to the patient, it also increases time-pressure for the surgical staff and brings additional distractions into the OR, as the planning staff will need to consult with the surgical team for updates on how the operation is proceeding. Both distractions and time-pressure are named as important stress factors in the OR, which is in turn linked to poorer surgical performance [30, 31]. On the other hand, surgical procedures that run shorter than expected will cause the OR room to be empty, as preparation and delivery of the next patient will not yet have been finished. As both types of inaccuracies in surgical duration estimation have undesirable outcomes, predicting end-times is a popular research topic.

In their 2008 research, Padoy et al. [32] create a surgical phase recognition system for laparoscopic cholecystectomy with the aim of predicting surgical end-times. A Hidden-Markov model with 14 phases allowed for a phase detection with over 90% accuracy and a prediction error of end-times below ten minutes while roughly halfway into the surgery. Franke et al. (2013) predicted remaining intervention time in discectomies and brain tumor resections, based on an ontological surgical process model [13] . The resulting model predicted end-time with a mean accuracy of 13 minutes for the discectomy and 29 minutes for brain tumor removals, decreasing towards the end of the surgery. Some authors directly predict the end-times of the surgery from the intra-operative data, without first using a surgical progress model. For example, Nakamura et al. (2013) predicted the end-times of brain tumor resections using information from surgical navigation and tumor characteristics [14]. Guédon et al. (2016) used a system based on usage of the electrosurgical device to determine end-times of laparoscopic cholecystectomies [12, 33]. In a different approach to operative planning, Bhatia et al. (2007) monitored surgical room occupancy rather than predicting surgical end-times. Using video data in combination with SVM and HMM models, the OR occupancy could be predicted with 99% accuracy [34].

2.2.2. Surgical skill assessment and training

Another application of surgical process models is to objectively assess surgical skill. As highlighted in a review by Reiley et al. (2011), surgical process models (termed 'statistical language models' by the authors) are the most promising way of assessing the surgical skill, over simply recording motions [15]. For example, Rosen et al. (2006) used a detailed model describing the motions in tying an intracorporeal knot during minimally invasive surgery. In an animal study, the performance of surgeons at different training levels was assessed by looking at similarity in which the model states where traversed when performing the knot in a pig [16]. In Leong et al. (2006) the trajectories of medical instruments in a laparoscopic box trainer where used to asses the skill level of the surgeon, using a Hidden-Markov Model [35]. As expected, most of the work on surgical skills assessment uses models with relatively high granularity such as steps and motions, as these contain most information on the dexterity and skill of the surgeon [15].

As video has shown to be an effective tool in surgical training, another application in surgical training could be the automatic generation of a labelled database of surgical videos [9, 24, 36]. Based on the surgical process model, the endoscopic video can be automatically sliced and labelled with the correct surgical phase, allowing novice surgeons to easily look up difficult phases within a surgery for review.

2.2.3. Other applications of surgical phase recognition

Robotic assistance is an often cited application of SPMs. Although surgical robots are quite common nowadays in hospitals, these robots work based on direct input of the surgeon. Semi-autonomous robots could alleviate a part of the surgical workload, by completing self-determined actions based on their understanding of the surgical procedure. Ko et al. (2007, 2010) used a surgical phase model of laparoscopic cholecystectomies to create an intelligent interaction with a laparoscopic robot [17, 18]. Based on the current phase of the surgery, which was determined using instrument tracking on the endoscopic video, the system determined the ideal view for the surgeon, such as tracking the tool tip or showing an interesting anatomical structure. The robot then automatically moved the endoscopic camera to the desired view point. Weede et al. (2013) designed closed-loop control cognitive robot system based on knowledge from a surgical model, for assisting in trocar placement and for camera guidance [37]. Using the model predictions to determine the position of the endoscopic camera allowed for a 30% reduction in camera movements and led to increased instrument visibility [38].

Although, to the extent of the authors knowledge, no research has specifically focused on automated surgical reporting, the possibility of pre-filling post-operative reports based on surgical process models has been discussed [24]. The information from the surgical process model could be used to automatically fill parts of the post-operative report, saving time for the medical staff.

Another application of surgical phase recognition lies within triggering events in the OR. In their 2008 study, Padoy et al. used surgical phase recognition as a trigger for switching the OR lights on and off [32]. Other applications could be an automatic change of settings of the monitors and the surgical table based on the surgical phase or the start or end of the operative procedure.

2.3. Clinical scope

Research into surgical process models has focused mostly on modelling single procedures, in which ease of data recording, standardisation of the procedure and regular occurrence of the procedure have been major drivers. Most popularly researched are models of laparoscopic cholecystectomy [39–49]. Other laparoscopic procedures found in previous literature are laparoscopic ovarian endometrioma [50] and laparoscopic myomectomy [51]. Another type of surgery with easily accessible intra-operative data is robotic surgery, as the movements of robotic manipulators are often recorded and stored digitally. Researched procedures include robot-assisted hysterectomy [9], robot-assisted radical prostatectomy [52] and robotic endoscopic coronary artery bypass [53].

2.3.1. Laparoscopic hysterectomy

This thesis will feature a surgical phase recognition system applied to laparoscopic hysterectomy, which is the minimally invasive removal of the uterus. In the U.S., over 600,000 hysterectomies are performed yearly, rendering it the most common gynecologic surgical procedure [54]. A retrospective study in an American hospital showed that between 2004 and 2012, the fraction of hysterectomies performed in a minimally invasive manner increased from 8% to 93%, driven by the fact that laparoscopic hysterectomy (LH) results in a decreased post-operative recovery time and shorter length of hospital stay [55, 56]. The most common indications for hysterectomy are uterine leiomyomas (41%), followed by endometriosis (18%), uterine prolapse (15%) and cancer (9%) [57]. In laparoscopic hysterectomy, first the uterine arteries are exposed and transected. The uterus is separated from the vagina and morcellated if it is too large to extract in whole. After specimen retrieval, the vaginal cuff is sutured and the patient is closed up. For a more elaborate description of the LH procedure please refer to Einnarson et al. (2009) [58] and Table 3.2.

2.4. Intra-operative data

A surgical phase recognition system needs to acquire data to use as an input to the model. In choosing the right data sources, there are several aspects to consider. The data should have predictive value for the phase of the surgery and be available for measurement. Furthermore, most applications of surgical phase recognition are aimed at providing information or assistance during the surgery. These models therefore need *intra-operative data*, indicating data that is recorded and processed during the actual surgery, ideally in (near) real-time.

Previous research has seen several sources of intra-operative data, of which video recordings and instrument usage tracking have been most popularly researched [3]. Other sources of intra-operative data include medical device or apparatus use, patient monitoring and monitoring surgeon activity, such as tracking hand movements.

2.4.1. Video recordings

Bhatia et al. (2007) estimated the OR state (one of 'empty', 'transitioning' or 'in-use') using video. The states could be accurately predicted in real-time (1 second) from relevant features of the OR video [34]. Lalys et al. (2012) predicted phases of cataract surgery using only microscopic video data. The microscopic video was automatically analyzed using shape, colour, texture and other features and processed using Hidden Markov Models and Dynamic Time Warping [24]. These visual cues could provide an phase identification accuracy of 91% for HMM and 94% for DTW. Twinanda et al. (2016) studied task recognition on laparoscopic cholecystectomy procedures using "EndoNet", a convolutional neural network (CNN) [49]. Based on laparascopic video, the CNN automatically extracts relevant features. These are first used as an input to an support vector machine (SVM) model, of which the output is in turn used to detect surgical phases using a Hidden Markov Model. The reported overall accuracy was 92% using an offline model and 81% using an online model. A novel approach using video data to estimate surgical phases was used by Tran et al. (2016), by retrieving optical flow vectors from video [59]. Optical flow describes the pattern of motion of objects, surfaces, and edges between video frames. The retrieved vectors are then simplified into four directions (up, down, left, right). The optical flow vectors are used as an input to Latent Dirichlet Allocation and Hidden Markov Models (HMM), where the best model yielded 73% accuracy.

2.4.2. Instrument tracking

As a surgery progresses through different phases, the surgeon often uses a task-specific set of tools. Hence, tracking the usage of medical instruments has become a popular approach for recognizing surgical phases. Although instrument usage data is simpler than video data or kinematic data, it can be used to detect surgical phases with high accuracy, as shown by the previous literature reviewed below.

Ahmadi et al. (2006) predicted the states of a laparoscopic cholecystectomy based on usage information of surgical instruments [39]. A model using 17-binary inputs indicating the use of each instrument was able to detect changes in 14 surgical phases within a range of 5 seconds in 92% of the cases. Instruments were weighed according to their synchronisation across different surgeries, giving higher weight to instruments that were used consistently over many surgeries. Unfortunately the authors do not report which instruments are most relevant. It needs to be noted that the instrument use was extracted from OR video using manual labelling.

Using RFID technology, Agarwal et al. (2007), created a system that could track the location of medical instruments in the operating theatre [60]. Fusing this information with tracking of staff location and drug location, a a low-level event record was created. Based on a set of seventeen pre-defined rules, a medical encounter record was generated, listing medically relevant events, which could for example be the surgery nearing the end.

Padoy et al. (2008) used real-time monitoring of endoscopic camera and instrument use to detect in which of the 14 phases of laparoscopic cholecystectomy the case is proceeding [32]. The model was cross-validated on 11 laparoscopic cholecystectomies and provided a detection rate of 93%. According to the authors, the model could reliable identify relevant events (such as the end of surgery time). Again, the instrument use was labelled manually from video recordings. In a follow-up study by the same group, Padoy et al. (2012), used Dynamic Time Warping (DTW) and Hidden Markov Models

(HMM)to predict phases in laparoscopic cholecystectomies [47]. The online models reached an accuracy of over 90% and the model could predict remaining surgery time with mean prediction error below 5 minutes, when the surgery was in the tenth of fourteen phases. Other research by the same group used similar approaches in predicting surgical phases of laparascopic cholecystectomy using binary tool usage data, with some adjustments to model and data recording setup (e.g. [41]).

Bouarfa et al. (2011) [43] used a Bayesian Hidden Markov Model to detect high-level surgical tasks based on low-level sensor data in laparoscopic cholecystectomy. The algorithm could detect high-level tasks (i.e. phases in the surgical procedure, for example ”clipping and dissection” or ”gallbladder removal”) with 90% accuracy when using noise-free sensory inputs. The sensory data used included binary variables indicating instrument use, which were manually generated from video streams. Simulation of signal noise resulted in significant decrease in model accuracy.

Nakamura et al. (2013) predicted duration of brain tumor resection using mean removal speed [14]. By tracking the instrument location in real-time and combining this information with previously recorded MRI-scans of the tumor size and location, the algorithm was able to compute surgery process, with an average error of 14 minutes ($\pm$ 9 minutes standard deviation) over the whole surgery, declining towards the end.

In a study on two surgical procedures (lumbar discectomy and brain tumor resections), Franke et al. (2013), proposed a model based on low-level surgical tasks to predict intervention time [13]. The surgical task was defined as a combination of an actor, activity, instrument, anatomical structure and intervention phase. For example, a surgical task could be the nurse disinfecting the skin using a swab during preparation. The model was able to predict surgery duration with an error of around 10 minutes for the discectomy and around 15 minutes for brain tumor removal. The quintuple task variables were annotated manually by human observers. A recent study by the same research group included patient status and device usage to further improve the model on predicting phases in brain tumor removal [61], again using human observers to generate input data.

Stauder et al. (2014) [48] detected surgical phases of laparoscopic cholecystectomy using a random forest (RF) model based on instrument usage data and various other data sources. An accuracy of 65% was obtained when using seven distinct surgical phases, this improved to over 85% for three surgical phases. The authors noted that even features that appear to carry very little information can have a high impact on the classification, such as CO2 pressure in the abdominal cavity, suction bag weight and whether the surgical light is switched on. In a follow-up study by the same group, again a Random Forest (RF) model was used on laparoscopic cholecystectomy to detect one of seven surgical phases [62]. Based on intra-abdominal pressure, suction and irrigation bag weights, table inclination and binary data of tool use, acquired using RFID tags, phases were detected with a precision of 87.6% and a sensitivity of 75.4%.

In a study on forty lumbar discectomies, Maktabi et al. (2015) estimated surgery duration using frequency domain analysis of surgical activity time-series [63]. A total of 35 operational (instruments used), spatial (treated body parts) and organizational (executing person) binary time series was generated. After transformation to frequency domain, signal features were used to assess surgical duration, which the best signals achieving around >20% error. The time series were manually generated by an observer.

Malpani et al. (2016) detected phases in a robot-assisted hysterectomy using system events [9]. Using information from the Da Vinci surgical robot, the use of tools and the built-in camera could be recorded automatically. In a set of 24 surgeries, the model was able to detect surgical phases with an accuracy in the range from 66%-76% using three different classifiers.

In a recent study, Guédon et al. (2016) estimated elective laparoscopic cholecystectomy duration in real-time using information from an electrosurgical device [12]. Several features of the electrosurgical device activation pattern were extracted, including the first and last time of activation, the number of activations and the total duration of the activation, which were all found to positively affect classifier performance. Various pre-operative data sources (including patient age, BMI and operating surgeon) were tested but did not increase model accuracy.

2.4.3. Other sources of intra-operative data

Other sources of intra-operative data include monitoring the use of medical apparatus, patient monitoring and monitoring surgeon activity.

Apparatus use
Next to surgical instruments, it is possible to track the usage of *medical devices* present in the OR. In a study on sixty neurosurgical cases, Franke et al. (2015) created a model using, amongst other sources, information on usage of apparatus present in the operating room [61]. The neuro-navigation device, ultrasound device, and neurophysiology monitor were classified as either ”not used yet”, ”in use”, ”likely to be required”, ”not likely to be required”, ”not required”, and ”unused”. The assigned states where used in combination with Hidden Semi-Markov Models to predict the surgical phase.

Patient monitoring
In their 2007 study, Agarwal et al. suggest using *patient monitoring* to detect the status of the patient during the surgery [60]. They report using data streams from pulse oximetry and vital signs monitors tracking heart rate and blood pressure. It is not reported how the data is retrieved from the patient monitoring systems to be used in the model, or how important the patient data turned out to be for realizing predictions.

Surgeon activity and hand tracking
Instead of tracking the patient or solely the instruments, an approach can be to track the *activity of the surgical team*. In the previously described tracking system, Agarwal et al. (2007), equipped the surgeon and nursing staff with RFID tags to track their location [60]. Based on eye movements of the surgeon, James et al. (2007), developed a model to recognise phases in a porcine laparoscopic cholecystectomy [45]. The eye-gaze data contains information of underlying surgical activity and an accuracy of 66% was reported using an artificial neural network (ANN) model. This improved to 75% by adding data relating to the instrument use, similar to the binary usage signals described before. Loukas & Georgiou (2013) used hand kinematics as an input for a model predicting surgical phases [46]. The hand movements of the surgeon where tracked by placing orientation sensors on the instruments of a Virtual Reality simulator for laparoscopic training. A precision between 59% and 91% was achieved for the distinct phases of a VR-simulated cholecystectomy surgery. Forestier et al. (2015) applied a decision tree model on a data set of 22 lumbar disc herniation surgeries [64]. The input data used considered of one data triplet per hand of the surgeon, which consisted of the action, anatomical structure and instrument. For example: `(cut, muscle, scissors)`. The triplets were manually labelled by a human observer, sensory recordings were simulated by adding noise to the manually generated labels.

2.5. Models in surgical phase recognition

An extensive range of machine learning and pattern recognition techniques can be applied to the field of surgical phase recognition. The selected model closely relates to the formalization of the surgery and the underlying assumptions (Table 2.1). When viewing surgical phases as a nominal variable, a non-sequential list of major events within the surgery, a classification approach is appropriate. A regression analysis typically views the phases as a ratio-variable, indicating that the phases do not only do have an ordering, but they also have identical distances between them. Distance in this sense indicates a rather abstract concept, meaning that, for example, the third phase in the procedure should be as much alike to the fourth, as the first phase is alike to the second. Finally one can view the surgery as a stochastic process, always residing in a certain state (phase) and having a certain probability of moving towards another phase, in which a state-space model is appropriate.

Each of the modelling approaches has a different set of applicable machine learning and pattern recognition techniques, which are concisely introduced in the following sections.

Modelling problem	**Example technique**	**Formalization level**	**Underlying assumptions**
Classification	CART, RF, KNN, SVM	Non-sequential list	The surgery consists of a finite set of phases.
Regression	LR, LLR, MARS	Sequential list	The surgery consists of a finite set of phases, that have a specific ordering and identical distance between phases.
State-space model	HMM	State-transition diagram	The surgery consists of a finite set of phases, that have a specific ordering. The surgery is a stochastic process residing in a certain phase and having a certain probability of moving from one phase to another.

Table 2.1: A phase recognition system can be build using different strategies, each relating to a different definition of the surgical phases and underlying model assumptions.

2.5.1. Classification

The aim of a classifier is to assign a new object to one out of set of two or more classes, based on a learned set of rules [65]. An object (x), which can be anything ranging from MRI-images, heart sounds to the state of the operating room at some point time, can be described by several its properties, or features:

$$x_i = (x_{i,1}, ..., x_{i,d}), x_{i,j} \in \mathbb{R}^{\mathbb{d}} \tag{2.1}$$

The classifier then learns a decision boundary, that is essentially a function (f) mapping the features of the object to an output (y) that has a single class (ω).

$$f : \mathbb{R}^n \rightarrow y, y \in \{\omega_1, \omega_2, ..., \omega_n\} \tag{2.2}$$

In previous research on SPMs, popular classification methods have been Support Vector Machines (SVM) [33, 34, 49], Artifical Neural Networks (ANN) [9, 49, 66] and decision tree methods such as CART and Random forest (RF) [9, 48, 64, 67, 68].

Support Vector Machines (SVM)

Support Vector Machines (SVM) is a widely used technique for classification, which has also been applied in surgical process models (e.g. [12, 34]). In principle, SVM is a binary classifier that is trained to have

a linear decision surface. The decision surface is constructed in such a way that it provides the largest distance between two sets of data points, belonging to different classes. By using so called kernels, SVM can be extended to accommodate non-linear relations. By training multiple one-vs-all classifiers, SVM can be extended for use in a classification problem featuring more than two classes.

Artificial Neural Networks (ANN)
Artificial neural networks comprise a whole set of models, that are inspired by the functionality of the brain. The model typically consists of input layers, hidden layers and output layers, each containing neuronal units. During the training phase, connections between these neurons are made and are given certain weights, either positive (excitatory) or negative (inhibitory). Research into ANN has been enormous of the past decades and the specific implementations and variations of ANN go beyond the scope of this thesis. In general, it can be stated that an ANN is able to capture complex, non-linear relationships. However, as a black-box model, an ANN is often hard to interpret.

In SPM literature, several authors have used artificial neural networks. James et al. (2007) used a Parallel Layer Perceptron (PLP) topology to recognize progress in a porcine laparoscopic cholecystectomy using eye-gaze data [45]. Devi et al. (2012) used artificial neural networks and adaptive neuro-fuzzy inference systems (ANFIS) to predict surgery durations in an ophthalmology department [66].

Decision Trees and Random Forests
Decision trees can be subdivided in classification and regression trees, with the difference that classification trees provide a categorical output, where regression trees provide a continuous estimate of the dependent variable. The acronym CART for *classification and regression trees* was first coined by Breiman (1984) and is typically used to refer to his specific implementation of decision trees [69]. A decision tree can be visualized as a graph, where each node represents a subset of the data and poses a certain question (e.g., $x_1 < 5$). The answer to this question is used to further split the data set, with egdes leading to another question at the following node. Finally, this leads to the so called leaf node, which gives either a categorical or numerical prediction of the outcome variable. The CART algorithm chooses the data split that leads to the largest decrease in *Gini-impurity (G)*, with the Gini-index of a dataset t given by [70]:

$$G(t) = \sum_{j=1}^{n} p_j^t (1 - p_j^t) \tag{2.3}$$

With p_j^t being the probability, that is the proportion, of class j within data set t. The Gini impurity ranges from zero for a data set that purely contains a single class to a maximum impurity of $1 - \frac{1}{n}$, for a uniform distribution with n possible values. Alternative measures of impurity use information gain, a metric derived from entropy. The graph-like structure of the CART model allows the model to grasp non-linear relationships and renders it easily interpretable visually.

An extension of decision trees are *Random Forests* (RF) [71], which have been used in surgical process modelling [48, 68]. As an ensenmble model, the random forest model consists of a collection of decision trees. Each decision tree is trained on a random subset of the training set and considers a random subset of features at each split. The prediction of each tree counts as a vote for a certain overall prediction. The modal (in case of classification) or mean (in case of regression) prediction of all trees provides the final prediction of the model. Random Forests usually outperform single decision trees, but this comes at the expense of interpretability. RF models are able to deal well with missing features and are robust to noise, due to the randomly sampled objects and features in de underlying decision trees. Another advantage is that the model can be used to assess feature importance, by observing the mean decrease in accuracy (MDA) or mean decrease in Gini-impurity (MDI), caused by a specific feature [72].

2.5.2. Regression

In a regression problem, the model aims to predict a continuous output (y), based on several independent variables. A simple multivariate linear regression model can be formalized as follows:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_i x_i + \epsilon \tag{2.4}$$

Where β_i indicates the coefficient, x_i the independent variables used to predict the dependent variable y and ϵ indicates a random noise term, which is often assumed to be normally distributed with zero mean. The advantages of linear regression are that the model is simple and linear in the parameters, so it can be fitted algebraically using the linear least-squares method, rendering the method computationally fast. The influence of the independent variables on the dependent variable can be observed from the regression equation, so the model can be easily interpreted. A downside of the linear regression model is that it assumes a linear relation between the input variables and output, which is of course not always the case.

Non-linear relationships between the dependent variable and one or more independent variables can cause poor fit of a regular linear regression model. A variant, the log-linear model, is obtained by predicting the log-transformation of the dependent variable.

$$ln(y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_i x_i + \epsilon \tag{2.5}$$

Splines are another possible to solution to the problem of non-linear relations, and are continuous functions formed by connecting a series of segmented basis functions. The points where the segments are connected are called knots or hinges.

$$y = \sum_{j=1}^{n} \beta_{1,j} B_{1,j} x_1 + \ldots + \sum_{j=1}^{n} \beta_{i,j} B_{i,j} x_i + \epsilon \tag{2.6}$$

Where $\beta_{i,j}$ indicate the linear coefficients and $B_{i,j}$ the hinge functions, which are defined only for a certain part of the curve and zero for all other values of x. A popular variant of splines are the *Multivariate adaptive regression splines (MARS)*, a method first coined by Friedman (1991) [73]. The *MARS* model is defined using linear splines, but splines can use higher order basis functions. ShahabiKargar et al. (2014) used the MARS model to predict surgery duration, highlighting the advantages that MARS can search a large number of variables and their possible non-linear interactions [68].

2.5.3. State-space models

State-space models explicitly incorporate the time element, by modelling the object as having a certain state at each point in time. In surgical process models, these are often Markov-chain based [32, 41–43, 61, 74]. A discrete-time Markov-chain is a mathematical representation for a series of events. The representation consists of a certain amount of states, that are described by probabilities of transferring from one state to another or remaining in the current state. A defining characteristic of a Markov process is that the probability of moving between states is only defined by the current state of the system, not by any other state in the history of the event-chain. The system is, in other words, memory-less.

Hidden Markov Models (HMM) are an extension of the Markov chain model that feature observable outputs (symbols) generated by a set of hidden states. In terms of surgical phase recognition, the real phase (e.g. suturing) might be not directly measurable (hidden), but the outputs that these states emit (e.g. the use of a suture stapler) are observable by measurement. Given a time series of these observed outputs, the path through the hidden states of the HMM with the maximum a posteriori likelihood can be found using the Viterbi-algorithm [75].

Padoy et al. (2008) used a 14-state left-to-right HMM, indicating that the state transition could only move to higher states, resulting in a classification accuracy between 84.4% and 94.4% on 10 cases of laparoscopic cholecystectomy [32]. Bouarfa et al. (2011) obtained 90% accuracy, using a five-state HMM, again on ten cases of laparoscopic cholecystectomy [43]. Blum (2010) showed only 47-53% accuracy on a 14-state HMM using video features obtained by dimensionality reduction techniques on laparoscopic video [42]. An HMM-based technique used to detect whether the OR was in use reported by Bhatia et al. (2007) achieved an accuracy of 99% [34].

2.6. Validation and performance

One of the most important aspects of modelling is out-of-sample validation, which involves the partitioning of the data into test and training sets. The model is generated based on the training data, validation of the model is performed on a set of unseen test data. This procedure reduces the risk of overfitting the model to the training data, as highlighted in Figures 2.3 and 2.4.

A single split into a test and training data set is the simplest way of performing such out-of-sample validation and can be significantly affected by which observations have randomly been allocated to which data set. A procedurally better alternative is k-fold cross-validation, in which the data is split into k folds, which each acts as out-of-sample test set once, while the model is trained on the remaining data. A special case of k-fold cross-validation is leave-one-out cross-validation, where k is equal to the size of the data set.

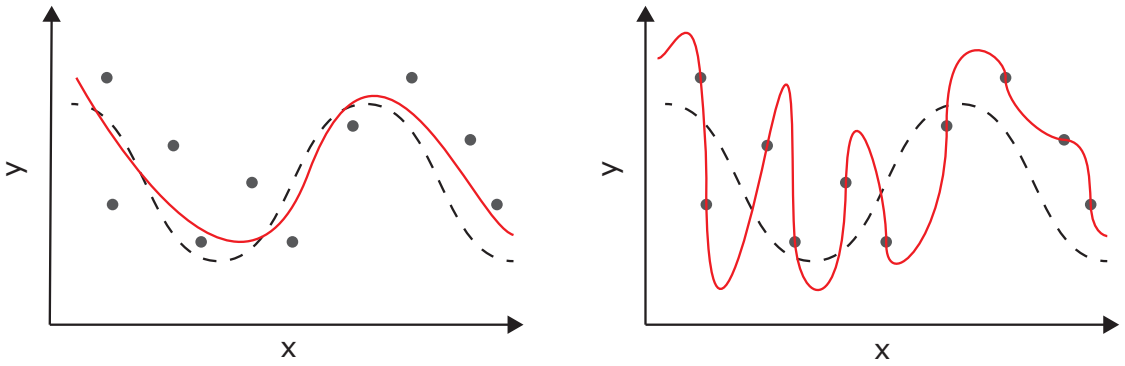


Figure 2.3: Overfitting poses a major risk for increasing the out-of-sample prediction error and is caused by an overly complex model that does not generalize well to unseen data. This figure shows an example of two models of different complexity fitted to noisy observations (dots) of an underlying process (dashed line). The model (red line) on the left side has a moderate prediction error, but would perform similarly if ten other points were randomly sampled from the underlying process. The more complex model (right figure) has zero error on the training sample, but will have considerable errors if applied to another set of unseen data points. The complex model on the right has thus been overfit to the training data. Figure adapted from [76].

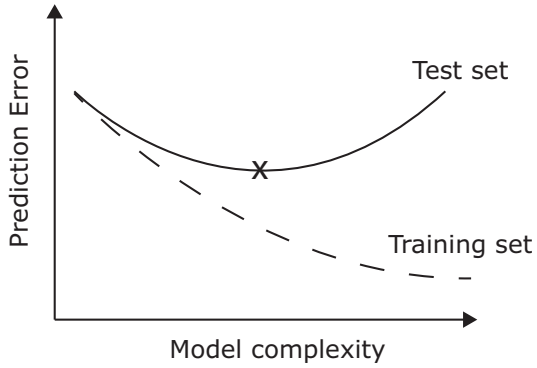


Figure 2.4: A commonly observed pattern in the prediction error when increasing model complexity is that the training set error (dotted line) continues to decrease, because the increasingly complex model allows to fit to outliers in the training data. The test set prediction error (solid line) will however at reach an optimum complexity, as further increasing model complexity will result in overfitting. This phenomenon highlights the importance of out-of-sample validation. Figure adapted from [77].

Another important consideration is the choice of a *performance metric* for use in the out-of-sample validation. Several metrics can be identified to describe the error. In case of a *numerical prediction*, commonly reported metrics are the root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). These metrics can be defined as follows, with y_t being the true values and y_t^* the predicted values:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - y_t^*)^2}{n}} \tag{2.7}$$

$$MAE = \frac{\sum_{t=1}^{n}|y_t - y_t^*|}{n} \tag{2.8}$$

In the case of predicting surgical durations, the RMSE and MAE commonly refer to the amount of minutes that the prediction is off compared to the real duration of the surgery. Such a performance metric makes it hard to compare the errors on surgeries with different lengths, as it can be imagined that a five minute error on a short surgery is of considerably more importance that the same absolute error on a surgical case taking several hours. A scaled metric such as the *mean absolute percentage error (MAPE)* can overcome this disadvantage:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - y_t^*}{y_t^*}\right| \tag{2.9}$$

When the model has a *categorical output*, for example in surgical phase recognition, other performance metrics are used. In case of a binary classifier four outcomes are possible: true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Based on the proportions of these cases the following performance metrics are commonly calculated:

$$SPC = \frac{TN}{TN + FP} \tag{2.10}$$

$$SEN = \frac{TP}{TP + FN} \tag{2.11}$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.12}$$

Specificity (SPC) is also known as false positive rate. The sensitivity (SEN) is sometimes called true positive rate or recall. Finally, the accuracy (ACC) gives the fraction between false and true predictions of the model. For a multinomial classification problem, the specificity and sensitivity are calculated class-wise. The accuracy can be calculated per class, but also gives a measure of the overall performance and is widely reported in literature on surgical phase recognition.

3

Materials and Methods

3.1. Recording and transformation of surgical data

The data set used in the current research contains 40 cases of laparoscopic hysterectomy (LH), which were recorded between November 2010 and April 2012 in the Bronovo Hospital in The Hague, The Netherlands for the purpose of a study into surgical flow disturbances by Blikkendaal et al. (2017) [78]. The procedures were recorded using three cameras and four audio signals using an audiovisual recording system (MPEG Recorder 2.1, Noldus Information Technologies, Wageningen, The Netherlands). Analysis of the procedures was performed using The Observer XT 11.5 software (Noldus Information Technologies, Wageningen, The Netherlands), by two residents of the department. Inter-observer agreement was established by comparing six procedures between the two observers, allowing the rest of the annotations to be done by one observer only.

The LH surgery was separated into 10 surgical phases and 36 surgical steps based on the method of peri-operative analysis of surgeries by Den Boer et al. (2002) [23, 78], see Table 3.2 for a description. The annotated event log was exported to a plain-text file for further analysis and contained start- and end-points of all observed surgical steps, together with the instruments used in said steps (Table 3.1). Further data transformation and model generation was performed using the R programming language (R Foundation for Statistical Computing, Vienna, Austria) [79] and RStudio IDE (RStudio inc., Boston, U.S.A.) [80].

Grasper/forceps	Bipolar coagulation	Ultrasound coagulation	Probe (Palpateur)
Irrigation/Suction	Needle driver	Suture stapler	Morcellator
Hasson cannula	Veress needle	Monopolar coagulation	Monopolar loop (Lina loop)

Table 3.1: The use of twelve surgical instruments and devices was annotated during the forty LH procedures, based on the audiovisual recording system featuring four cameras and two microphones.

3.1.1. Data transformation

First, the event log was summarized to contain a single entry for each surgical step, describing the start-points and duration. The event log contained entries representing very short surgical steps, with an annotated duration of zero seconds. These steps were arbitrarily set to a duration of five seconds, so the steps would remain visible in the log when converting to a time-based structure. An example showing the structure of the event log can be found in Table 3.3.

In order to allow real-time simulation of surgical phase recognition, the event log was converted to a time-based log containing an entry for each point in time. For the classification, a time interval of one second was chosen, theoretically allowing for a 1Hz prediction. For the state-space approach with

Phase	**Step**
1. Create CO2 pneumoperitoneum	1.1 First incision and insert Veress or Hasson
	1.2 Insufflate the abdomen
2. Insert access ports	2.1 Insert first (optical) port
	2.2 Insert laparoscope
	2.3 Inspect abdomen (active bleeding, 360 look, operatability)
	2.4 Insert second port under direct sight
	2.5 Inspect and judge operatability/unexpected pathology)
	2.6 Insert third port under direct sight
	2.7 Insert fourth port under direct sight
3. Preparation operative area	3.1 Dissect adhesions to uterus/ovaria/intestine in pelvis
	3.2 Mobilize intestine out of pelvis
4. Expose uterine arteries	4.1 Dissect ligaments and mobilize uterus
	4.2 Skeletonized uterine arteries
	4.3 Push off bladder
	4.4 Identify location of ureters
5. Transect uterine arteries	5.1 Transect left uterine artery
	5.2 Transect right uterine artery
	5.3 Check color of uterus
	5.4 Check if bladder and arteries are skeletonized enough
6. Separate uterus from vagina	6.1 Colpotomy
	6.2 Pneumoperitoneum is lost
7. Specimen retrieval	7.1 Morcellated uterus
	7.2 Extract uterus through vagina
8. Closure of the vaginal cuff	8.1 Insert needle
	8.2 Suture vaginal cuff
	8.3 Extract needle
9. Final check and irrigation	9.1 Check hemostasis
	9.2 Check vaginal cuff stump
10. Close-up patient	10.1 Remove instruments
	10.2 Remove accessory operating ports (under direct sight)
	10.3 Check access wounds/bleeding
	10.4 Release CO2 from abdomen
	10.5 Remove laparoscope and first trocar port
	10.6 Suture port wounds
	10.7 Remove draping

Table 3.2: Intra-operative surgical phases and steps commonly occurring during a laparoscopic hysterectomy procedure. Table copied from Blikkendaal et al. (2017) [78], based on earlier work by Den Boer et al. (2002) [23]

Hidden Markov Models (HMM), the time-discretization is an important parameter in constructing the model, hence the time interval was optimized between five and sixty seconds. The structure of the time-based log for 1 Hz predictions is shown in Table 3.4.

In the original event log, phases showed overlap in several occasions and during other time intervals the phase was undefined. Following from the definition of both classification and state-space models, it is a requirement that the phase is uniquely defined at each evaluated time. To satisfy this requirement in the time-based log, double entries were aggregated into a single entry by combining the instrument usage data using the logical `OR`-operator and assigning the phase with the highest numerical label. In case of missing time entries, the time-based log was filled by means of the last observation carried forward procedure.

Date	Time	Phase	Step	Duration (s)	GF	BC	UC
29-07-2013	14:53:13	5	5.2 Transect right uterine artery	84	0	1	0
29-07-2013	14:55:50	5	5.3 Check color of uterus	5	0	0	0
29-07-2013	15:01:00	5	5.5 Prepare dorsal sacro-uterine	83	0	0	1
29-07-2013	15:04:16	5	5.3 Check color of uterus	5	0	0	0
29-07-2013	15:06:29	2	2.7 Insert 4th port under direct sight	5	0	0	0
29-07-2013	15:10:49	6	6.1 Colpotomy	2606	1	1	1

Table 3.3: Selected columns of the operative event log, showing the date and time of the surgery, the phase and step in the surgical procedure, the duration of the surgical step and a binary indicator for instrument use during that step, with 1 indicating usage (3 out of 12 annotated instruments are shown). As can be observed from this excerpt of the event log, the end of one event does not necessarily lead up to the start of the next event, indicating that the phase is undefined at certain time-points. These missing time spans are filled in a later processing step by means of last-observation carried forward. GF: Grasper/Forceps, BC: Bipolar Coagulation, UC: Ultrasound coagulation. The total event log contained 2697 rows over all 40 procedures.

ID	Surgical Time (s)	Phase	Step	GF	BC	UC
1	726	2	15	0	0	0
1	727	2	15	0	0	0
1	728	4	19	1	1	0
1	729	4	19	1	1	0
1	730	4	19	1	1	0

Table 3.4: Example showing selected columns of the time-based log, which was created by interpolation of the operative event log. It shows the numerical ID of the surgery, the elapsed time within this procedure (starting from first incision), the phase and step in the surgical procedure and a binary indicator for instrument use during that step (3 out of 12 annotated instruments are shown). GF: Grasper/Forceps, BC: Bipolar Coagulation, UC: Ultrasound coagulation. When using time steps of one second, the total time-based log contained 293,631 entries.

3.2. Classification model

In the classification approach, the model assigns objects (represented by a single row of the time-based log) to a certain class (a surgical phase), based on a learned set of rules [65]. The object (x), can be described by several of its properties, or features:

$$x_i = (x_{i,1}, ..., x_{i,d}), x_{i,j} \in \mathbb{R} \tag{3.1}$$

With all features being real-valued. The classifier learns a decision boundary, essentially a function (f) mapping the features of the object to an output (y), which is one out of a set of pre-defined classes (ω).

$$f : \mathbb{R}^d \rightarrow y, y \in \{\omega_1, \omega_2, ..., \omega_n\} \tag{3.2}$$

Based on criteria including interpretability and ability to handle non-linear relations, the following classifiers were selected: Decision Tree (CART), Random Forest (RF) and K-nearest neighbor (KNN). Please see the previous chapter for a more elaborate discussion of different classifiers.

3.2.1. Feature engineering

A single entry in the time-based log (Table 3.4) does not capture all relevant information that a classifier could use to learn the patterns that distinguish phases. Therefore, extra features are derived from the indicators of instrument use to improve the classification performance (Table 3.5). To simulate real-time application, only retrospective information is used to construct the features.

From the binary instrument usage vectors from the start of the procedure until the current time (t), the cumulative usage up until t is derived. As the instrument usage is discretized with a time step of one second, the cumulative usage is given by the sum of the vector from time 1 to $\mathtt{t}$. Furthermore, by checking whether the cumulative usage is larger than zero, it is noted whether the instrument has been used during the current procedure.

Another set of features encodes changes in the use of the instruments. The backward difference of a signal I is given by $I(t) - I(t-1)$. For the binary instrument usage signals, the backward difference takes a value of either 0 (no change), $+1$ (instrument now in use) or -1 (instrument not in use anymore). By choosing different time lags (1 second, 1 minute, 5 minutes, 10 minutes) the usage history of the instrument is taken into account.

Epochs of instrument use describe the sessions of consecutive use of an instrument. These can be found by counting the times when the backward difference of the instrument with a one-sample delay is equal to one.

Finally several summarizing features are derived. By summing the instrument vectors at time t, we obtain the total number of instruments currently in use. Similarly, by summing over the usage indicators, we obtain the total number of different instruments that have been used in the current procedure. A total of 99 features are generated for every time step and appended to the time-based log (Table 3.5).

3.3. Hidden Markov Model approach

The state-space description of the Hidden Markov Model describes the different states and the probabilities of transferring between states. The time-step with which the model is discretized hence matters for the probabilities. The time-log for the HMM is generated by sampling each nth row from the time-based log (Table 3.4), which has an entry for every second. The time-step of the discretization is an optimizable parameter of the Hidden Markov Model.

Feature	Description	Count	Type	R implementation (at time `t`)
Surgical time	Elapsed time since surgery onset, in seconds.	1	Integer	`t`
Instrument	Instrument currently in use	12	Binary	`Instrument[t]`
InstrumentC	Cumulative used time of instrument in this procedure	12	Integer	`sum(Instrument[1:t])`
InstrumentUsed	Instrument used in current procedure	12	Binary	`sum(Instrument[1:t])>0`
InstrumentD1, ..., InstrumentD600	Backward difference per instrument with delay of 1, 60, 300 and 600 seconds	48	Categorical (-1,0,1)	`Instrument[t] - Instrument[t-Delay]`
InstrumentE	Epochs of use per instrument	12	Integer	`sum((Instrument[2:t] - Instrument[1:(t-1)])==1)`
numInstruments	Total number of instruments currently in use	1	Integer	`sum(Instrument1[t] + ... + InstrumentN[t])`
numInstruments-Used	Total different instruments used in this procedure	1	Integer	`sum((sum(Instrument1[1:t])>0) + ... + (sum(InstrumentN[1:t])>0))`

Table 3.5: Description of the total of 99 features used in the classification model. The R implementation shows how to derive the value for the feature at time `t` from the given instrument usage data. The instrument usage is given by binary vectors `Instrument[t]` that take value `1` when the instrument is used at time `t` and value `0` when not in use. The `Instrument` refers to one out of twelve tracked instruments (Table 3.1)

The HMM (λ) is defined by three matrices: $\lambda = (A, B, \pi)$. Given N states and M possible observation symbols, A is an NxN matrix containing the state transition probabilities and B is the NxM observation probability matrix. Both A and B can be inferred directly from the discretized data set, by observing the state-transitions and the instrument use in each state. As the surgery always starts in the first surgical phase, the π matrix is given by:

$$\pi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.3}$$

An HMM typically models one observation sequence ($\mathcal{O} = (\mathcal{O}_0, \mathcal{O}_1, ..., \mathcal{O}_t)$), with each observation ($\mathcal{O}_t$) coming from a discrete set of M observation symbols. Therefore, the instrument usage vectors need to be combined into one discrete set of symbols. Given the twelve annotated instruments (Table 3.1), there are $2^{12} = 4096$ combinations of outputs. The observation symbols are created by first joining the twelve binary instrument vectors into one 12-digit binary number. Next, this binary number is converted to a decimal representation, yielding the observation symbol (Table 3.6). Only a small fraction of possible observations is observed, largely reducing the size of the B matrix.

Instruments used	Binary representation	Observation symbol
Grasper/Forceps and Bipolar coagulation	110000000000	3072
Grasper/Forceps, Needle driver, Suture stapler	100001100000	2144
Hasson cannula	000000001000	8

Table 3.6: Example of instrument usage patterns and the corresponding observation symbol used in the Hidden Markov Model. With 12 annotated instruments, there are 4096 observable symbols, although given the data, the observation probabilities are zero for the overwhelming majority of symbols. The construction of the binary representation follows the ordering of instruments as in Table 3.1

3.4. Model optimization

In order to find the optimal model, parameter optimization is performed using a grid search and 10-fold cross-validation (Table 3.7). The model performance is assessed by the out-of-sample accuracy, defined as the fraction of correct predictions on an unseen set of test data. The mean absolute error (MAE) is also calculated. Model optimization is performed on a standard personal computer (3.60 GHz quad-core CPU, 8 GB RAM) and average computational times are noted for the model training and prediction.

Model	Parameter	Description	Values evaluated
Decision Tree (CART)	Complexity parameter	Multiplier for regularization penalty per added split in the decision tree	$2^k \cdot 10^{-5}$, with $k = 0, 2, .., 17$
Random Forest (RF)	Features per split	The amount of features considered per split	$1, \lfloor e^{0.69+0.39k} \rfloor$, with $k = 0, 1, .., 10$
K-Nearest Neighbor (KNN)	Amount of neighbors	Amount of closest neighbors deciding on the prediction	$1, 21, 41, 61$
Hidden Markov Model (HMM)	Time discretization	Time-step in seconds used to obtain the discretize the Markov model	$1,5k$ with $k = 1, ..., 12$

Table 3.7: Summary of the model optimization strategy, showing the optimized parameter and the values evaluated during grid-search. All values of the parameter are evaluated for 10 mutually exclusive folds, each containing 4 surgeries. The evaluated RF-parameters are log-spaced integers between 1 and 99.

Decision Tree (CART)

For CART the complexity parameter is optimized, which governs as a stop-criterion and regularization parameter for growing the tree. The loss-matrix ($L(i,j)$) of the decision tree defines the penalty for wrongly classifying phase i as phase j and, when penalizing uniformly, if given by:

$$L(i,j) = \begin{cases} 0, & \text{if } i = j \\ 1, & \text{if } i \neq j \end{cases} \tag{3.4}$$

Taking the previously defined loss-matrix and using the class-distributions in the training set as the prior probabilities for each class, the model risk $R(T)$ is equal to the proportion of misclassified objects, or the error rate [70]. The cost of the model R_{cp}, quantifying the trade-off between accuracy and model complexity is then defined as:

$$R_{cp}(T) = R(T) + cp \cdot |T| \cdot R(T_1) \tag{3.5}$$

Where $R(T)$ is the error rate in the model, $R(T_1)$ is the error rate in a decision tree with no splits (i.e. a a decision tree always predicting the class that is present most often in the training data), cp the complexity parameter and $|T|$ the amount of splits. The tree will only grow as long as the model cost decreases ($R_{cp}(T_n) < R_{cp}(T_{n-1})$). In other words, the downside of an increase in model complexity should be sufficiently offset by the increase in accuracy. The default value of $cp = 0.01$, indicates that the error rate should decrease by at least 1% for each added split. For large data sets, a smaller cp value might be beneficial [70]. The complexity parameter is evaluated between 10^{-5} and 1.

Random Forest (RF)

A Random Forest is an ensemble model created by the combination of n decorrelated CART decision trees [71]. To ensure decorrelation of the trees, each tree contains only a random sample of the data. Furthermore, at each split in the tree, a random subset of features is evaluated for deciding the best split. The amount of features to select at each split (m_{try}) is one of the most important parameters in RF. In contrary to CART, the decision trees that make up the RF model are unpruned (fully grown), so the complexity parameter does not need to be optimized. The default value for the number of selected features is $m_{try} = floor(\sqrt{D})$, with D being the amount of features of the object [81]. During the

optimization $n = 100$ trees are grown for each RF model.

The importance of individual features is assessed using the mean decrease in accuracy (MDA) and mean decrease in impurity (MDI) measures [72]. In calculating the MDA, the values of a selected feature are randomly permuted and these adjusted feature vectors are used to make predictions. The rate with which the model error increases gives an estimate of the importance of this feature. The MDI uses the Gini-impurity criterion to assess how well the ideal split of the feature is able to separate the classes in the data set, hereby lowering impurity. The impurity decreases are summed over all splits performed by the feature and normalized by the number of trees in the RF model.

K-nearest neighbors (KNN)

The KNN-algorithm is a simple classifier that uses the majority class of the k nearest neighboring data points to classify an object. As the KNN classifier uses the euclidean distance between object features to determine the closest neighbor, the relative scaling of features is important and KNN requires an extra step in data transformation. Therefore all features are normalized to a range of 0-1 using the following transformation:

$$x_{i,norm} = \frac{x_i - min(x)}{max(x) - min(x)} \tag{3.6}$$

Because of the duplication occurring in the transformation from event log to time-based log, several objects have highly similar features. In order to prevent excessive ties between neighbors, uniformly distributed additive noise is applied to make each object unique. The maximum magnitude of the noise is 10^{-3}. Following conventions originating in binary classification by KNN, odd values for k are evaluated, being 1, 21, 41, 61. The amount of parameters searched in KNN is limited due to computational constraints.

Hidden Markov Model (HMM)

In generating the Hidden Markov Model, the time-based log is sampled at a certain frame rate. The step size of this time discretization (t_{step}) is the optimized parameter in the HMM. A larger time-step decreases the time resolution of the model, as the prediction frequency is limited by t_{step}. A priori, a time step of 30 seconds is considered the maximum for 'real-time' estimation. To observe the effects of a more coarse discretization, time steps until 60 seconds are evaluated.

3.5. Model selection and comparison

The model parameter with the highest average out-of-sample accuracy over ten folds is selected. The accuracy is a very common way of comparing results in classifiers and is simply given by the fraction of true predictions [82].

As sphericity cannot be assumed when comparing the accuracy of learning algorithms, Friedman's ANOVA is used to assess model effects, which is the non-parametric version of the repeated-measures ANOVA [83, 84]. Because the assumptions of a paired t-test render it inappropriate for classifier performance comparison, McNemar's Chi-squared test is used as a post-hoc test [85, 86]. The McNemar test is used to compare frequencies between matched samples using the following test-statistic:

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{3.7}$$

Where b and c follow from a contingency table of the true and false predictions of both models. The amount of samples where Model 1 is false and Model 2 is true is given by b, where c conversely gives the amount of samples where Model 1 has a true prediction, but Model 2 provides a false one. The test statistic is then evaluated on a one degree of freedom χ^2-distribution to test the null hypothesis that $p(b) = p(c)$. The family-wise error-rate is controlled using the Bonferroni correction.

3.6. Model evaluation

The selected phase recognition model is evaluated on two tasks related to clinical practice in the OR: the automatic prediction of surgical end-times and the automatic generation of training material by clipping endoscopic video based on the predictions of phase.

3.6.1. Surgical End-Time Prediction

The surgical phase model is used for the task of surgical end-time prediction. For this, a second model is obtained that uses the phase predictions to estimate the remaining surgical time. The end-time prediction is given by a multiple linear regression model using the elapsed surgical time, the phase, the amount of seconds that the surgery has been in that phase and the interaction terms between phase and seconds in phase as independent variables.

To evaluate the end-time prediction, models are trained using k-fold cross validation on both the true phases and the phases estimated by the selected phase recognition model. The performance of both models is assessed by the mean average error (MAE) of the end-time prediction. Furthermore, the mean absolute percentage error (MAPE) is calculated to simplify comparisons with previous literature.

3.6.2. Generation of training material

For the generation of training material, a prediction of the phase timings is needed. As each phase often occurs multiple times during a surgical procedure, the longest consecutive run of each phase within each procedure is found. Start and end times of the longest run are compared for the ground-truth phases and the phases predicted by the selected model. The mean average error (MAE) is reported for both start and end times.

4

Results

4.1. Laparoscopic Hysterectomy

The analyzed laparoscopic hysterectomies (n=40) were shown to have an average surgery time of 128 minutes (±27 minutes standard deviation), with the individual surgical phases also showing a high variance in duration between cases (Figure 4.2). In 33 of the LH cases, all ten phases occured. The preparation of the operative area (phase 3) was omitted in seven cases, the closure of the vaginal cuff (phase 8) in two cases. Although each surgery started in the first phase and ended in the last phase, phase transitions occurred 19 (±6 S.D.) times per surgery on average. Most transitions, 70%, were between adjacent states, such as a transition from state one to state two. Over all surgical cases, 68% of the state transitions were towards higher phases. A trace of the surgical phase during a representative case is shown in Figure 4.1.

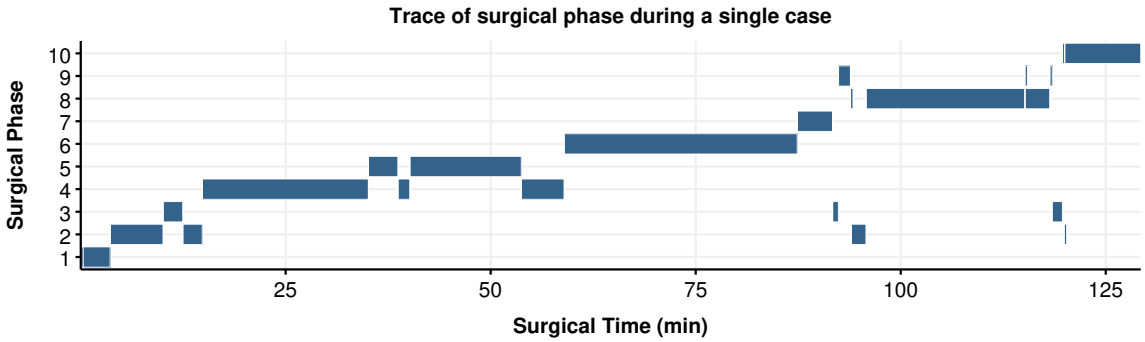


Figure 4.1: Progression of the surgical phase during a representative laparoscopic hysterectomy case. The shown case has a median case duration (129 minutes) and features 22 phase transitions, which is slightly above the average of 19.

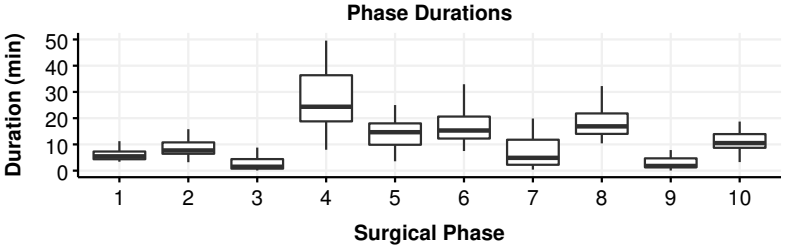


Figure 4.2: The duration of surgical phases is different per phase, but also varies strongly between surgical cases. The fourth phase, exposing the uterine arteries, takes the longest time to complete on average (29 min ±13 min S.D.), whereas the ninth phase - final check and irrigation - has the shortest time span (3 min ±3 min S.D.)

Instrument Use

The patterns of used instruments and devices differ per surgical phase (Figure 4.3). With nine different phases, the grasper and forceps are most broadly used throughout the surgery, followed by the bipolar and ultrasound coagulation tools, which were both observed in six distinct surgical phases. Five tools and devices were exclusively used in one phase: the Hasson cannula and Veress needle (phase 1), the monopolar coagulation device and monopolar loop (phase 6) and the morcellator (phase 7). Some tools are observed systematically across different cases: the bipolar coagulation device is used in phase 4 and 5 in all 40 cases, the grasper/forceps in 39 cases during the fourth phase, the needle driver in 39 cases during phase 8 and the ultrasound coagulation device in 38 cases during phase 6.

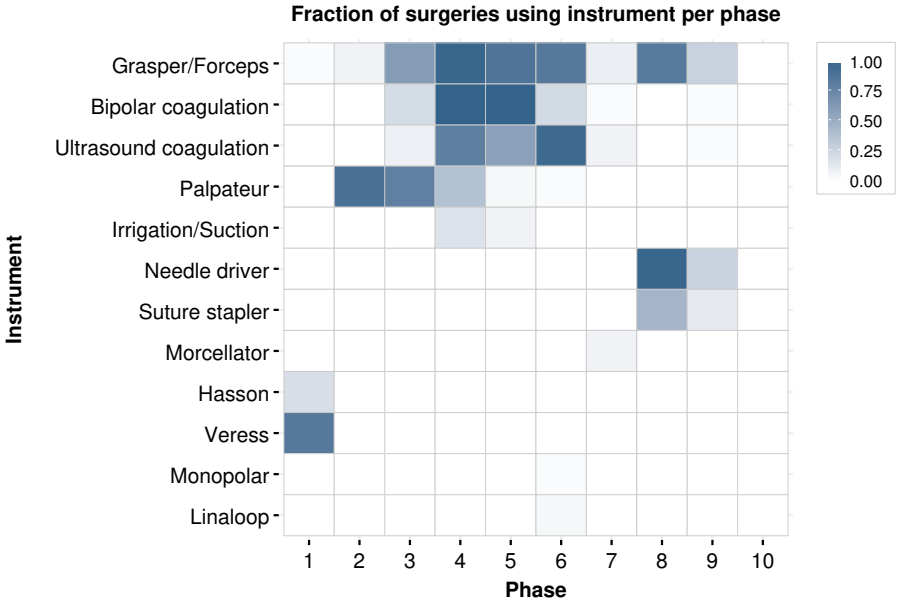


Figure 4.3: Heatmap showing the frequency of instrument use per surgical phase. The fraction indicates the share of procedures during which the instrument or tool was used in the specified phase, with one indicating use in all forty LH cases. Grasper/Forceps are observed in nine out of ten phases, while the morcellator, Hasson cannula, Veress needle, monopolar coagulation and monopolar loop are only used in a single phase.

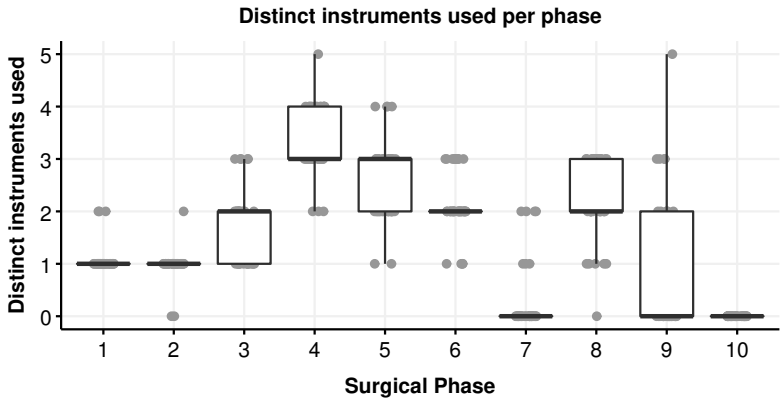


Figure 4.4: The amount of distinct instruments and tools used varies strongly per phase. In the fourth phase, 3.3 instruments are used on average, while in the final phase, no instruments are recorded during any of the cases. The points show the amount of instruments observed for each procedure.

4.2. Model optimization

Decision tree (CART), Random Forest (RF) and k-nearest neighbor (KNN) and the Hidden Markov model (HMM) are evaluated for a set of parameters in order to select the classifier with the highest performance. The models are optimized using 10-fold cross-validation, with the same ten folds of the test and training sets used for each of the model optimizations and all surgical cases being present in the test set exactly once per model. The optimization results are shown in the following sections.

4.2.1. Decision tree (CART)

For *CART*, the complexity parameter is optimized (Figure 4.5). The ideal value of cp was found to be $5.12 \cdot 10^{-3}$, yielding an accuracy of 75.5% ($\pm 5.7\%$ S.D.) and a mean absolute error of 0.46 phase ($\pm 0.23\%$ S.D.).

4.2.2. Random Forest (RF)

The *RF* model is trained by varying m_{try} (Figure 4.6). The ideal value was found to be 6 randomly sampled features per split, providing an accuracy of 76.8% ($\pm 5.2\%$ S.D.) and a mean absolute error of 0.39 phase (± 0.13 phase S.D.).

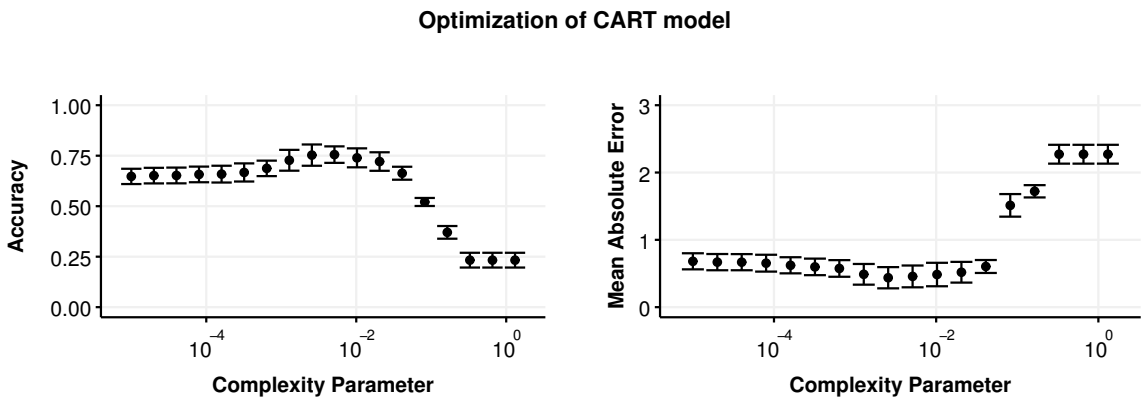


Figure 4.5: Optimization of the CART model using 10-fold cross-validation on a grid-search of 18 cp parameters ranging from $10^{-}5$ to 1. The best performing model ($cp = 5.12 \cdot 10^{-3}$) yielded an accuracy of 75.5% ($\pm 5.7\%$ S.D.) and a mean absolute error of 0.46 phase ($\pm 0.23\%$ S.D.). Error bars indicate 95% confidence interval of the mean.

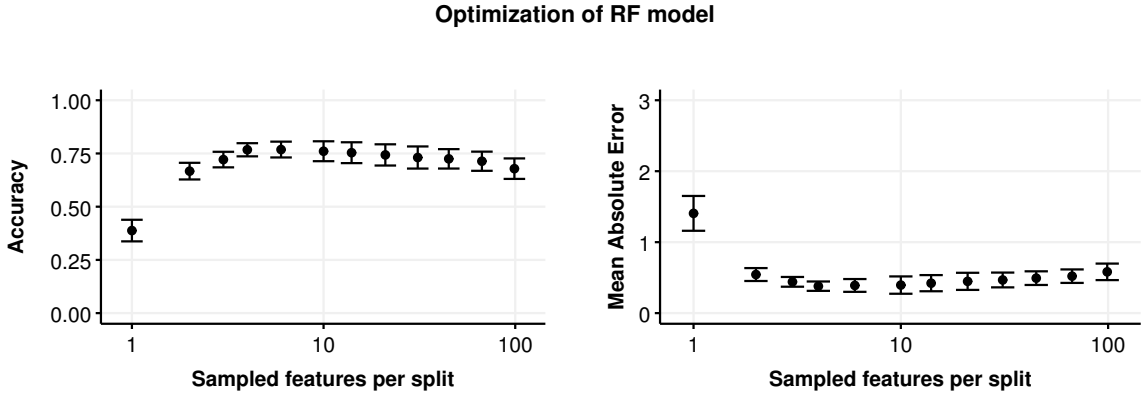


Figure 4.6: Optimization of the RF model using 10-fold cross-validation on a grid-search of 12 log-spaced parameters ranging from 1 to 98. The optimum model ($m_{try} = 6$) showed an accuracy of 76.8% ($\pm 5.2\%$ S.D.) and a mean absolute error of 0.39 phase (± 0.13 phase S.D.). Error bars indicate 95% confidence interval of the mean.

4.2.3. k-Nearest Neighbor (KNN)

Contrary to the other classifiers, KNN uses all training data as the actual model. Due to computational constraints, the amount of evaluated parameters for KNN was limited to four (Figure 4.7). The optimized KNN model (k =) showed an accuracy of 63.6% ($\pm 6.2\%$ S.D.) and a mean absolute error of 0.64 phase (± 0.13 phase S.D.).

4.2.4. Hidden Markov Model (HMM)

The HMM is tuned by varying the time step (Figure 4.8). The accuracy and mean absolute error of the HMM are both shown to improve with longer time steps, however the trade-off is with the prediction frequency, which is limited by the size of the time step. The maximum value of t_{step} was determined to be 30 seconds, resulting in a selected model with an accuracy of 61.5% ($\pm 12.1\%$ S.D.) and a mean absolute error of 1.13 phase (± 0.81 phase S.D.).

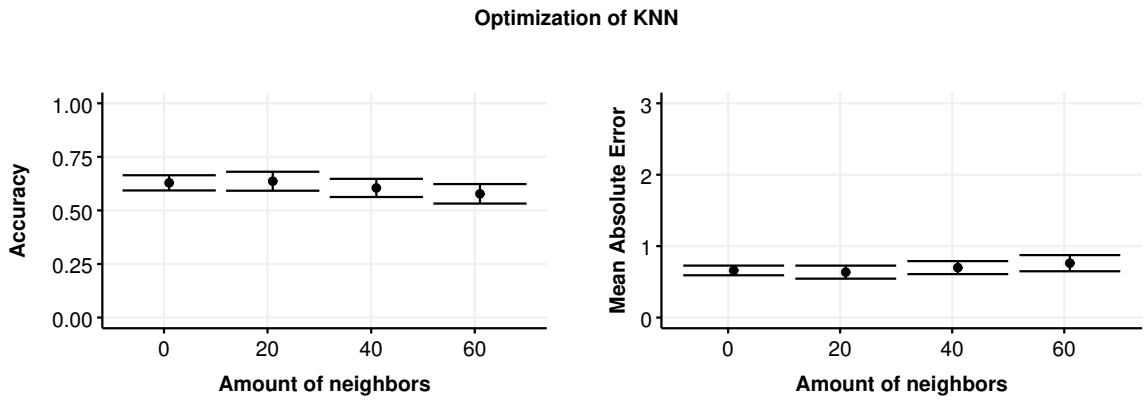


Figure 4.7: Optimization of the KNN model using 10-fold cross-validation for 1, 21, 41 and 61 neighbors. The optimum model (k =) showed an accuracy of 63.6% ($\pm 6.2\%$ S.D.) and a mean absolute error of 0.64 phase (± 0.13 phase S.D.). Error bars indicate 95% confidence interval. Due to the significant computation time, limited parameters were tried during optimization of KNN (Table 4.1). Error bars indicate 95% confidence interval of the mean.

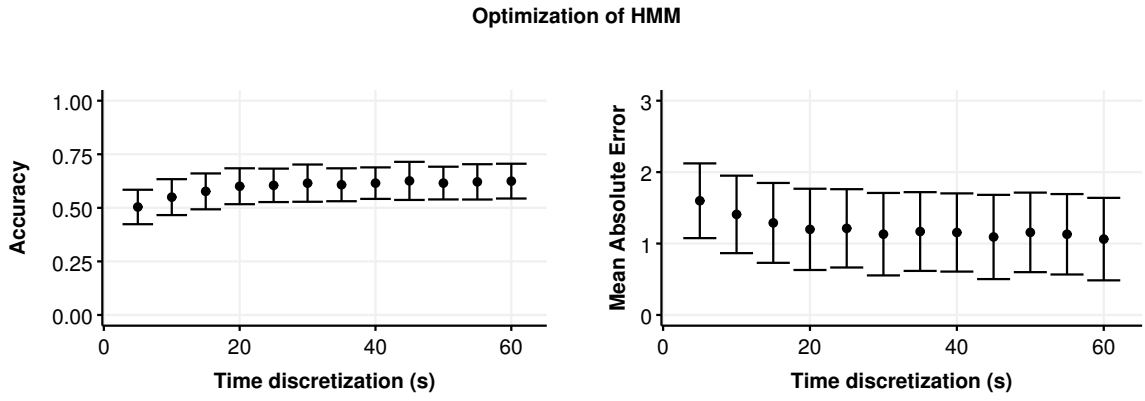


Figure 4.8: Optimization of the HMM using 10-fold cross-validation for discretization (t_{step}) ranging from 5-60 seconds. Although accuracy is shown slightly increase with increasing time step size, a value t_{step} of 30 seconds is chosen to allow relatively frequent predictions. The selected HMM model shows an accuracy of 61.5% ($\pm 12.1\%$ S.D.) and a mean absolute error of 1.13 phase (± 0.81 phase S.D.). Error bars indicate 95% confidence interval of the mean.

4.3. Model selection

Comparing the CART, RF and KNN classifiers and the HMM (Table 4.1), we observe superior performance of the RF model in both accuracy and mean absolute error. The measures of model performance show to be correlated (r=-0.86) and the RF model in both ACC and MAE is followed in descending performance by CART, KNN and HMM respectively.

The model type has a significant effect on the accuracy ($\chi^2(3) = 21.4$, $p < 0.001$, Figure 4.9). By analyzing contingency tables, Random Forest is shown to have a significant higher out-of-sample accuracy than CART ($\chi^2(1) = 12$, $p < 0.01$), KNN ($\chi^2(1) = 20605$, $p < 0.001$) and HMM ($\chi^2(1) = 20964$, $p < 0.001$).

Looking at the aspect of timing, the CART model performs best, with a total computational time of under one minute and the fastest prediction time. This is followed by the RF model, HMM model and KNN model respectively. KNN and HMM take most time in the prediction phase, whereas in CART and RF the majority of time is spent in creating the model based on the training data.

Based on the performance and time aspects, the RF model is chosen for as most suitable for the application tasks of surgical end-time prediction and generation of surgical training material.

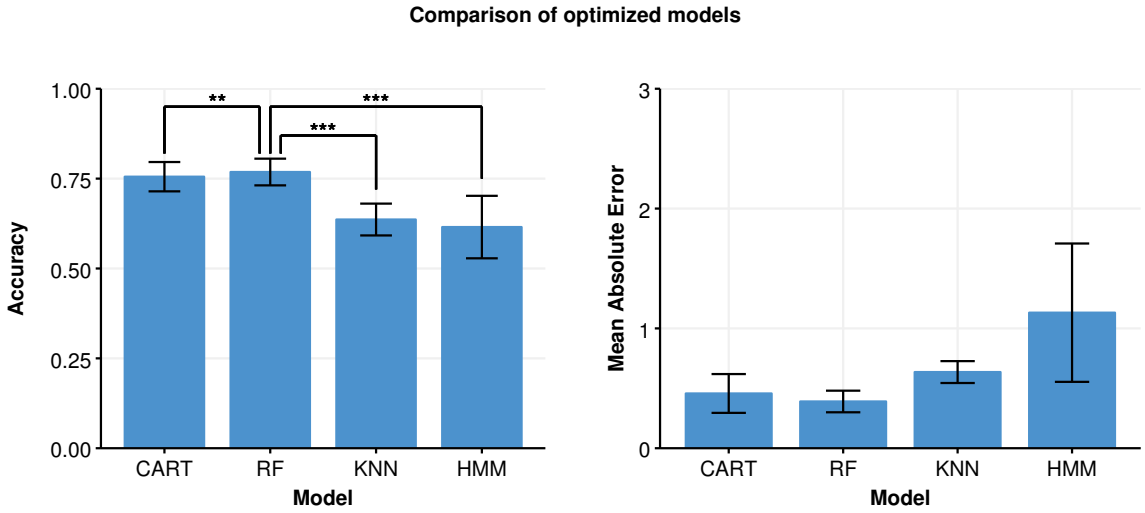


Figure 4.9: Comparison between the CART, RF, KNN and HMM models. With an accuracy of 76.8% and mean average error 0.39 phase RF scores significantly better than other tested models. Error bars indicate 95% confidence interval of the mean. (Indicated levels of significance: ** = p<0.01, *** = p<0.001).

Model	Parameter	Optimal value	ACC [%] (±S.D.)	MAE [phase] (±S.D.)	Computation time [min] (Train/Predict)
CART	cp	$5 \cdot 10^{-}3$	75.5% (±5.7% S.D.)	0.46 (±0.23 S.D.)	**0.80** (0.78/**0.02**)
RF	m_{try}	6 features	**76.8%** (±5.2% S.D.)	**0.39** (±0.13 S.D.)	5.73 (5.70/0.03)
KNN	k	21 neighbors	63.6% (±6.2% S.D.)	0.64 (±0.13 S.D.)	82.56 (0.05/82.51)
HMM	t_{step}	30 seconds	61.5% (±12.1% S.D.)	1.13 (±0.81 S.D.)	2.89 (**0**/2.89)

Table 4.1: Comparison of performances of the optimized models. The optimized RF model performs best in terms of both the accuracy and the mean absolute error, with a small margin from the CART model. The accuracy and mean absolute error are correlated (r=-0.86) and the model ranking is identical for both. In terms of computation time, the CART model performs best in total and prediction time. Computation times are defined as the average training time for one fold (36 training cases) and the prediction time for one complete case.

4.4. Model characteristics

The selected phase recognition model is a Random Forest model with one hundred trees ($n = 100$) and six considered features per split ($m_{try} = 6$). A prediction of the model on a single surgical case is shown in Figure 4.10.

The overall performance was shown to be 76.8%, however the performance differs per phase (Figure 4.11). Six of the phases are predicted accurately over 80% of their duration; phase 1 (81%), phase 2 (81%), phase 6 (86%), phase 7 (85%), phase 8 (91%), phase 10 (90%). The performance in phase 9 is lowest with an error rate of 99.7%. Again, the MAE is shown to be strongly correlated to ACC (r=-0.93), and hence shows a similar performance pattern across the different phases.

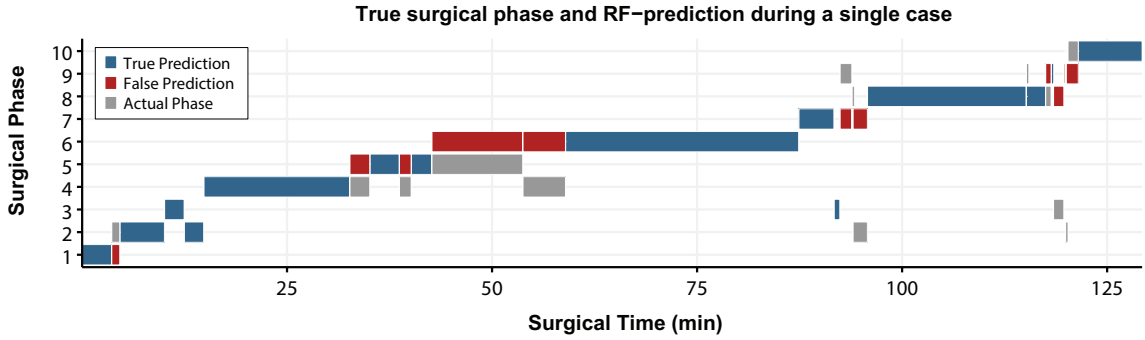


Figure 4.10: Progression of the surgical phase during a representative laparoscopic hysterectomy case (duration of 129 minutes), shown together with the prediction of the optimized RF model. The shown case has an ACC of 78.1% of and and MAE of 0.37 phases.

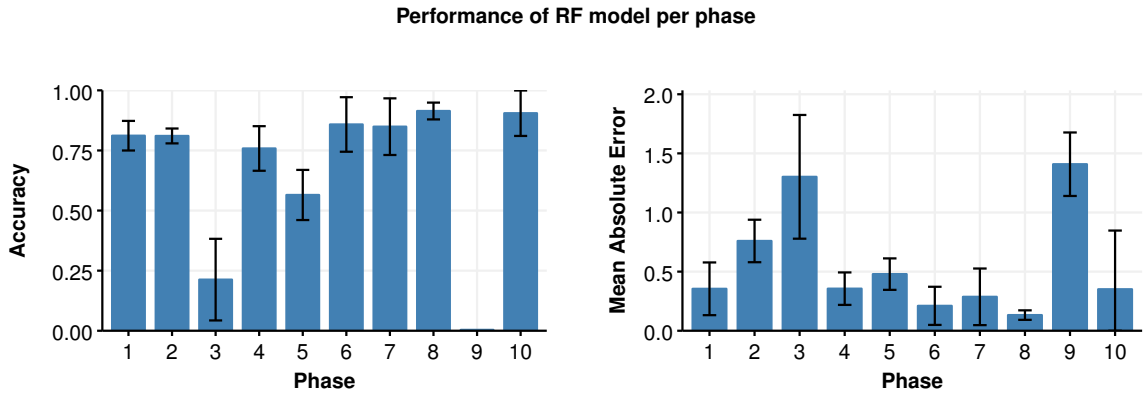


Figure 4.11: The performance of the optimized RF model differs visibly per phase, ranging from 91% accuracy in phase 8 to 0.03% in phase 9. The accuracy and mean absolute error measures of model performance are strongly correlated (r=-0.93). Error bars indicate 95% confidence interval of the mean.

Variable importance is assessed using the Mean Decrease in Accuracy (MDA) and Mean Decrease in Impurity (MDI) (Figure 4.12). Over all features (n=99), the two importance measures are strongly correlated (r=0.98). When looking at the ten most important features the bipolar coagulation device, ultrasound coagulation device, grasper/forceps and needle driver are the most important tracked instruments. Furthermore, the elapsed surgical time and the number of instruments currently in use and used in total during the procedure are important features in the RF model.

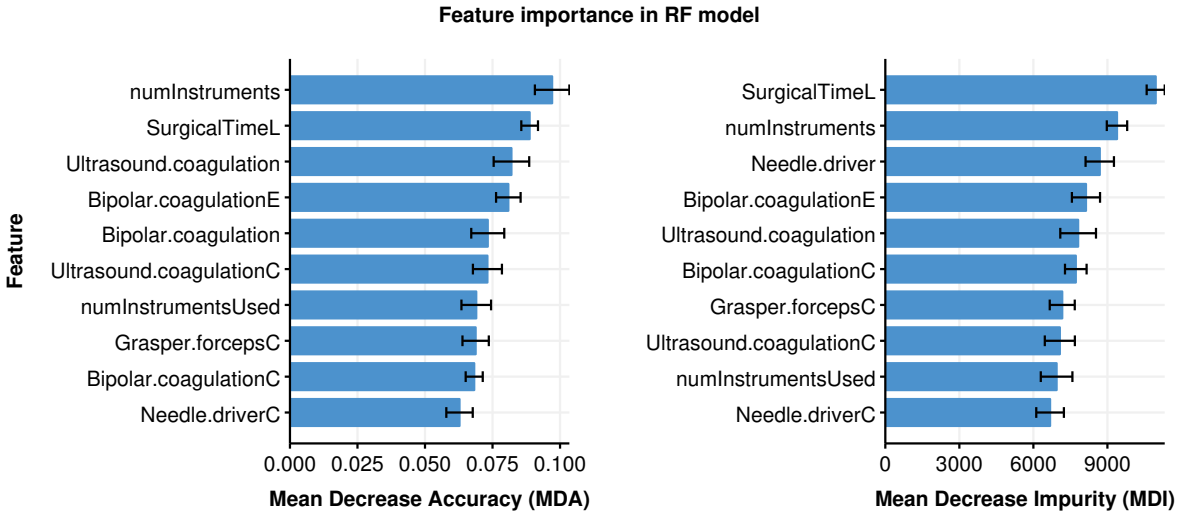


Figure 4.12: Ten most important features in the RF model, according to the mean decrease in accuracy (MDA) and mean decrease in impurity (MDI) measures. Shown MDA and MDI are averages over the 10 folds, error bars indicate the 95% confidence interval of the mean.

4.5. Model Evaluation

The model performance was evaluated by application to two simulated clinical tasks: surgical end-time prediction and phase extraction, for the purpose of surgical training.

4.5.1. Surgical end-time prediction

A multiple linear regression model was used to predict surgical end-times. The models use surgical time left as the dependent variable, predicted by surgical time passed, phase, duration within the phase and the cross terms between the phase and duration within the phase. Two linear regression models were trained, using the true annotated phase data and using the predicted phase data (Table 4.2). A model using ground-truth phases explained 75% of the variance ($R^2 = 0.75$, $F(20, 293610) = 4.38 \cdot 10^4$, $p < 0.001$) and most variables showed to significantly predict surgical end-time. The linear regression model using the phases as predicted by the selected RF model showed significance for all independent variables and a slightly higher $R^2 = 0.78$ ($F(20, 293610) = 5.10 \cdot 10^4$, $p < 0.001$).

Variable	**True Phase Model**				**Predicted Phase Model**			
	β	t	p	sig	β	t	p	sig
Intercept	7338.74	464.15	<0.001	***	6950.94	417.83	<0.001	***
Phase2	-664.36	-31.90	<0.001	***	-246.50	-12.00	<0.001	***
Phase3	-882.12	-34.23	<0.001	***	-1276.03	-22.87	<0.001	***
Phase4	-803.72	-45.30	<0.001	***	-432.58	-24.02	<0.001	***
Phase5	-1984.12	-99.12	<0.001	***	-1966.07	-97.47	<0.001	***
Phase6	-3164.90	-162.04	<0.001	***	-3194.76	-162.11	<0.001	***
Phase7	-3899.89	-172.10	<0.001	***	-4140.93	-189.98	<0.001	***
Phase8	-4643.99	-217.49	<0.001	***	-5502.04	-249.67	<0.001	***
Phase9	-4770.53	-168.69	<0.001	***	-6662.72	-68.94	<0.001	***
Phase10	-5709.33	-237.48	<0.001	***	-6590.37	-272.19	<0.001	***
SecondsInPhase	0.06	1.05	0.294		2.10	29.58	<0.001	***
SurgicalTime	-0.17	-85.16	<0.001	***	0.06	29.04	<0.001	***
Phase2*SecondsInPhase	0.01	0.14	0.885		-0.94	-12.37	<0.001	***
Phase3*SecondsInPhase	-1.82	-25.15	<0.001	***	-13.61	-24.86	<0.001	***
Phase4*SecondsInPhase	-0.58	-10.81	<0.001	***	-3.20	-45.10	<0.001	***
Phase5*SecondsInPhase	-0.52	-9.14	<0.001	***	-3.03	-41.91	<0.001	***
Phase6*SecondsInPhase	-0.13	-2.34	0.019	*	-2.54	-35.74	<0.001	***
Phase7*SecondsInPhase	-0.19	-3.28	0.001	**	-3.06	-42.67	<0.001	***
Phase8*SecondsInPhase	-0.27	-4.94	<0.001	***	-2.48	-34.69	<0.001	***
Phase9*SecondsInPhase	-0.93	-10.46	<0.001	***	-3.29	-4.15	<0.001	***
Phase10*SecondsInPhase	-0.17	-2.81	0.005	**	-2.77	-37.70	<0.001	***

Table 4.2: Multiple linear regression models of surgical end-time, based on the true annotated phases and the phases as predicted by the RF model. Coefficients in this table are a result of training on the total set of 40 procedures and are therefore an approximation of the coefficients in the ten different models used for evaluation on the different folds of the data set. ('***' <0.001, '**' <0.01, '*' <0.05)

Results

Using 10-fold cross-validation, the multiple linear regression model using ground-truth phases showed a mean absolute error of 16.2 minutes (±14.2 minutes S.D.) over all cases. With a MAE of 15.6 minutes (±12.9 minutes S.D.), the regression model based on the RF-predicted phases performed slightly better. The MAPE of the end-time prediction was found to be 13.7% for predictions using ground-truth phases and 13.2% for phase predictions based on the RF model.

Both multiple linear regression models for surgical end-time prediction based on phase information compare favourably to a baseline model always predicting the mean case duration, as the baseline shows mean absolute error of 27 minutes. Both models decrease in error between the onset of the surgery ($MAE_{TRUE} = 24.6 \pm 14.9$ minutes S.D., $MAE_{RF} = 26.3 \pm 15.0$ minutes S.D.) and surgery completion ($MAE_{TRUE} = 5.6 \pm 4.2$ minutes S.D., $MAE_{RF} = 5.9 \pm 5.5$ minutes S.D.) with the models showing similar trends (Figure 4.13).

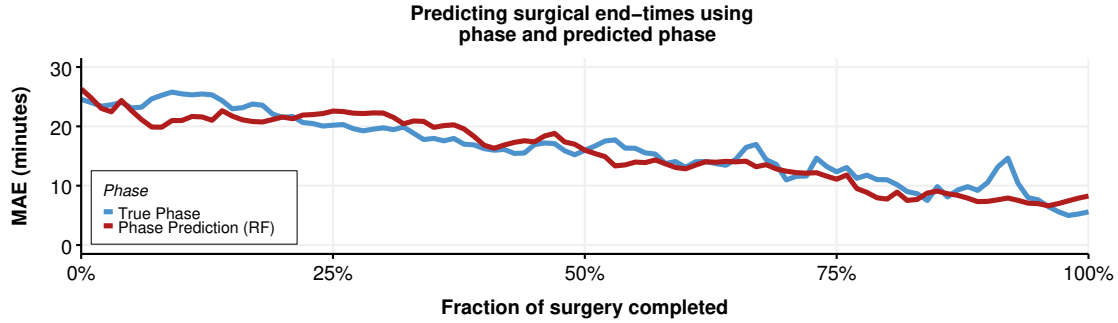


Figure 4.13: Mean absolute prediction errors of the surgical-end time prediction models as a function of surgery completion. The models based on the true phase and the phase prediction show similar performance.

When observing the error relative to the time until surgery completion, again similar performance of the models based on true and predicted phases is seen (Figure 4.14). Two hours before the end of the surgery, the end-time is predicted with an $MAE_{TRUE} = 17.8$ minutes (± 9.6 minutes S.D.) and $MAE_{RF} = 17.8$ minutes (± 14.9 minutes S.D.) for respectively the models based on the true and predicted phase information. This error stays rather constant for 60 minutes ($MAE_{TRUE} = 18.1 \pm 12.6$ minutes S.D., $MAE_{RF} = 16.0 \pm 14.0$ minutes S.D.) and 45 minutes ($MAE_{TRUE} = 17.0 \pm 14.4$ minutes S.D., $MAE_{RF} = 17.4 \pm 11.7$ minutes S.D.). At 30 minutes before the end of the surgery the error drops to $MAE_{TRUE} = 13.0 \pm 15.3$ minutes S.D. for the ground-truth model $MAE_{RF} = 12.6 \pm 13.2$ minutes S.D. for the RF-based model.

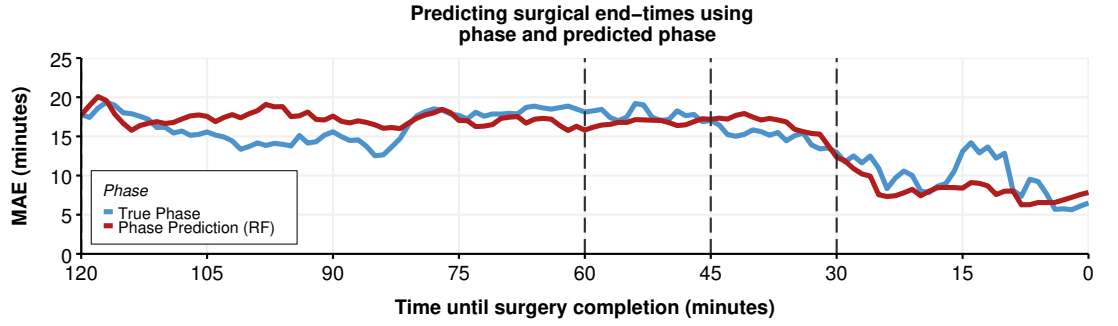


Figure 4.14: Mean absolute prediction errors of the surgical-end time prediction models as a function of the time until surgery completion. The models based on the true phase and the phase prediction show similar performance. Vertical lines highlight the performance at 60, 45 and 30 minutes before surgery completion.

4.5.2. Phase Extraction

The phase extraction algorithm extracts the longest consecutive epoch of each phase within each procedure, for example for use in the generation of training material. Related to the varying performance of the RF model on predictions for the different phases (Figure 4.11), the onset and ending of the extracted phases show different errors (Figure 4.15). The MAE for the start and ending of each phase are strongly correlated (r=0.94). The lowest errors (MAE = 0) are logically given at the start-time of phase 1 and the end-time of phase 10, as these are already bounded by the surgery duration. For the start of the phase, all phases except for 3, 5 and 9 have an average error smaller than four minutes. The recognition of the end-times is worse, with only phase 1, 2, 6, 8 and 10 having an average error under four minutes.

The median performance is observed in phase 6, the separation of the uterus from the vagina. Figure 4.16 shows the phase extraction applied to a random sample of ten surgical cases.

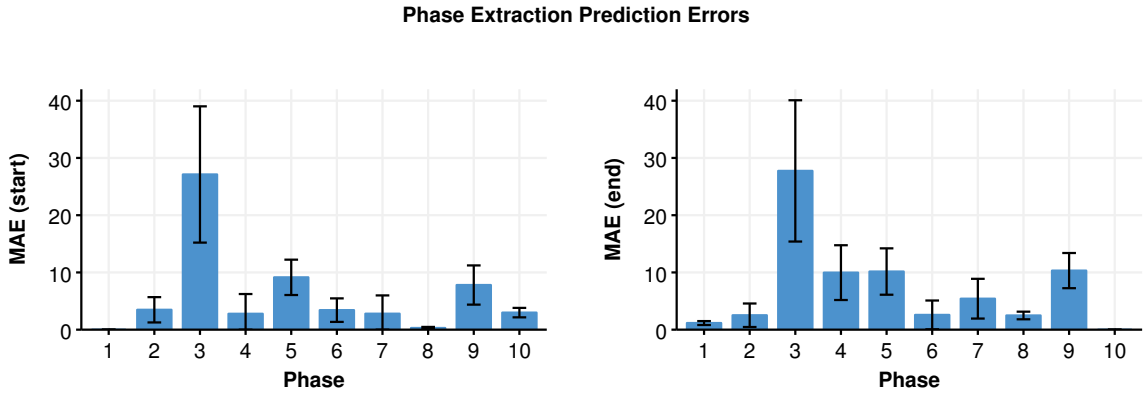


Figure 4.15: Prediction errors for the task of automatically extracting the longest consecutive run of surgical phases. The MAE for the start and ending of each phase are strongly correlated (r=0.94).

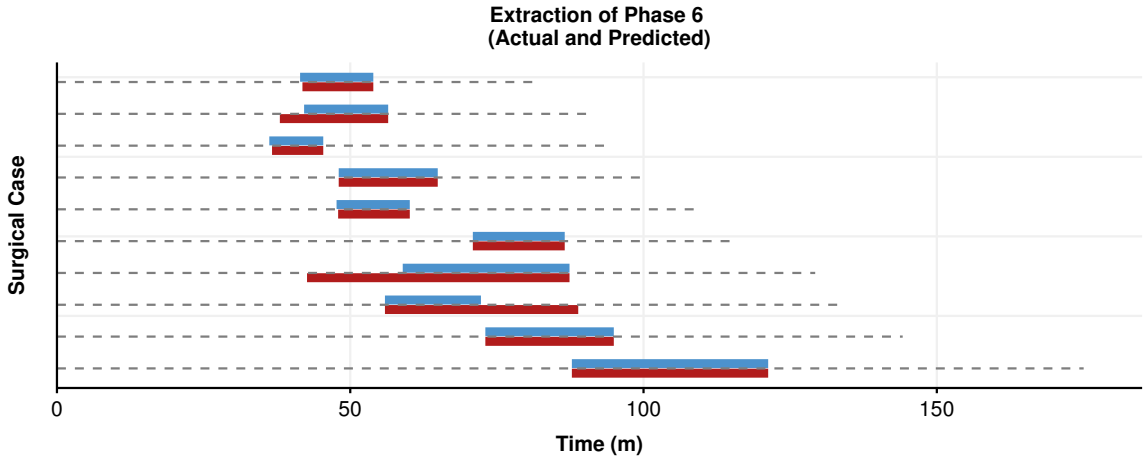


Figure 4.16: Sample of ten LH cases, showing the extraction of surgical phase six, the separation of the uterus from the vagina. The case duration (dotted line) is shown, together with the true (blue) and predicted (red) phase durations. The shown sample has prediction errors of $MAE_{start} = 130$ seconds and $MAE_{end} = 100$ seconds.

5

Discussion

This study demonstrated an intra-operative approach to recognize surgical phases in 40 laparoscopic hysterectomy cases based on manually annotated instrument usage data, with simulated application to surgical end-time prediction and surgical phase extraction.

Exploratory analysis of the data set underlined that laparoscopic hysterectomy is a suitable procedure for surgical phase recognition, due to the significant duration and variability. The cases show a surgical time over two hours on average and a substantial variance in case duration (128 ±27 minutes S.D.). Also, the case duration variance is shown to originate from variation in several phases, such as the fourth phase, indicating the exposing of the uterine arteries and the eighth phase, the closure of the vaginal cuff, both of which have duration distributions with standard deviations above 10 minutes. This variation in phase durations highlights that phase recognition in LH is a non-trivial task. Finally, with nineteen phase transitions on average per procedure, the surgical workflow in LH is shown to be more complex than a mere succession of states, also indicating the usefullness of surgical workflow modelling in laparoscopic hysterectomy.

Analysis of the instrument use over the different phases show that the instrument use naturally exhibits patterns that are linked to the different tasks within the procedure. Some instruments are used in specific phases only (e.g. the Hasson cannula, Veress needle and morcellator), whereas others have shown to be used broadly across the procedure (e.g. the grasper and forceps). As was shown later by the predictive models, these patterns in instrument use exemplify that instrument usage data can indeed be used for surgical phase prediction.

5.1. Model optimization

The phase recognition model was generated using four different approaches: decision trees (CART), Random Forest (RF), k-nearest neighbor (KNN) and Hidden Markov models (HMM), for each model selecting the parameter resulting in the highest out-of-sample accuracy in 10-fold cross-validation. The mean absolute error (MAE) showed to be correlated with the accuracy, suggesting that both performance measures should give similar indication of the model performance.

In the CART model, large values of the complexity parameter (approx. $cp > 0.01$) generated trees that were too concise, resulting in a loss of performance as detailed patterns were not recognized. A model trained with a cp value of 1 showed to always predict the fourth phase, since this phase has the highest prior probability due to its duration. Conversely, a small complexity parameter (approx. $cp < 0.001$) resulted in trees that were too detailed and overfitted to the available training data, causing lower out-of-sample performance. The optimized cp parameter of 0.005 is somewhat lower than the default value $cp = 0.01$, which can be explained by the large sample size used [70].

When optimizing m_{try} in the Random Forest model, low values resulted in an ensemble containing decision trees with very suboptimal splits, as too little features were considered. On the other hand,

high values of m_{try} resulted in overly correlated trees, inreasing the risk of overfitting to the training data. The standard value of $m_{try} = \sqrt{m}$ (with m being the amount of features), is in the range of the found parameter $m_{try} = 6$. The lower optimal value found for m_{try} in the current experiment might be due to the correlation of features, since most features are directly derived from the twelve binary indicators of instrument use.

In KNN, a small k is susceptible to noise, as a nearby outlier is one of few neighbors considered in the classification. A large value of k results in too much smoothing, leading to a decrease in out-of-sample performance. A guideline for k as the square root of the amount of samples proved too large in our empirical study, possibly because of the strong correlation between objects, as the time-based log was generated by duplication of event-log entries.

The HMM model shows increasing performance for a larger discretization time step, hence creating a trade-off between prediction frequency and accuracy. This trade-off in the HMM model might be circumvented by asynchronous processing, where the data set is sampled in such a way that only transition points, where the observable instrument use changes, remain. Bouarfa et al. (2009) show that an HMM trained with asynchronous processing leads to a more robust and discriminative HMM, with better classification performance [25]. As changes in instrument are fairly limited during surgical cases, asynchronous processing will also lead to decreased computational load. Another solution to improve HMM performance might be to use a left-to-right HMM, which is a simplified surgical model that only allows the transition to the adjacent state with a higher numerical label.

5.2. Computation time

Although computational time depends strongly on the specifics of the used hardware and software implementation, the durations are indicative for the relative computational load of the models, as well as for the allocation of computation time to the training and prediction task (Table 4.1). In clinical practice, retraining of the model is only needed when new training cases are added. As additional training cases need to be labelled, or at least checked, by human observers before incorporation into the data set, the time duration of model training is of lesser importance. With a computation time of model generation ranging between 0-6 minutes, all models have suitable training times for use in clinical practice.

As the actual phase recognition needs to be done intra-operatively for most purposes, the time taken to obtain predictions is of more importance. The RF and CART models take 1-2 seconds to provide phase labelling for a complete surgical case, consisting of over 7.5k individual predictions on average. Hence, both CART and RF are able to predict surgical phase based on real-time data. The prediction frequency of HMM is limited by the time discretization of the model, currently set at 30 seconds. For the KNN model, the real-time predictions are problematic. The KNN model is in essence just a collection of all training objects and the phase prediction for a novel object requires computation of the cartesian distances to all these training points. Therefore, the computational time for prediction is far larger than the other models and will only increase with expansion of the data set.

The computation time for CART and RF is already sufficiently low as it is expected that most time will be consumed by the real-time processing of sensor signals, for example acquired using an RFID-tracking set-up. For HMM, asynchronous processing should be explored to obtain higher prediction frequency and simultaneously decrease prediction error and computational load [25].

5.3. Model selection

Considering the experimental results and theoretical properties, the Random Forest model is deemed most suitable for the application in surgical phase recognition in laparoscopic hysterectomy. On the current data set, the models realizes the best performance in terms of accuracy and mean absolute prediction error and achieves sufficient computational speed (Section 5.2). Another useful feature in RF is the inherent measures of feature importance using MDA and MDI (Section 5.5). Furthermore, the RF models have shown to be resistant to noisy inputs [71], which is a convenient advantage when selecting a model that must work with sensor data. Finally, as an RF model consists of a set of independent regression trees, it is inherently suitable for parallel processing architectures, which increases the scalability of the model [87].

5.4. Model performance

The out-of-sample accuracy of the RF model (77%) is in the range of previous literature on SPM in general, who report recognition rates ranging from 70% up to 99% [3]. However these performances are very hard to compare due to differences in the sources and the amount of data used as well as the granularity and types of models employed. For example, sensor data and annotated data will likely result in different findings. Also, models predicting a large amount of steps might underperform when compared to models predicting only few surgical phases.

According to the knowledge of the author, there is one previous study featuring surgical process modelling on hysterectomy procedures, although this study was performed in robot-assisted rather than laparoscopic hysterectomies. In this research, Malpani et al. (2016) report an accuracy of 72-74% for RF models and 70-76% for a temporal convolutional neural network (tCNN) in recognizing five phases of the hysterectomy procedure. Both models use overlapping time windows of 10-60 seconds width [9]. Previous research using Random Forest models for surgical phase recognition report an overall prediction accuracy of 69% in detecting seven phases of a laparoscopic cholecystectomy procedure [48] and 84% in detecting four phases in lumbar disc herniation [64].

The current study features ten surgical phases, which is higher than the amount of phases observed in previous literature and as such renders the classification task more challenging. Still, the accuracy of 77% is in the range of previous findings on phase recognition using RF models (69-84%).

5.5. Feature importance

The selected RF model allows to interpret the importance of features using the mean decrease in accuracy (MDA) and mean decrease in impurity (MDI) measures. In empirical studies, variable importance measures have shown to provide insight in the discriminative abilities of individual features, also in the case of highly correlated features [88]. However, the feature importance measures are biased towards features with a higher possible amount of values [89]. In the current study, several binary features are used, adopting only two possible values, whereas the indicator for elapsed surgical time has more than ten thousand possible values (Table 3.5). Therefore the reported variable importance measures should be treated without caution.

Whilst taking into the account the possible inflation in the estimated importance of especially the feature `SurgicalTimeL`, the feature importance measures still provide interesting insights. The results show that the bipolar coagulation device, ultrasound coagulation device, grasper/forceps and needle driver are the most important of the tracked instruments. Tracking only a subset of instruments might therefore already be sufficient for achieving valuable phase recognition, which possibly decreases implementation costs. Previous research showed that tracking only the electrosurgical device already provides sufficient information to improve end-time predictions in laparoscopic cholecystectomy procedures [12], which might extend to laraposcopic hysterectomy as it corroborates our findings that the electrosurgery is an important tool in predicting surgical progress.

Stauder et al. (2014) report the MDA of features used in an RF model for phase-detection in laparoscopic cholecystectomy and find the non instrument-related features denoting the intra-abdominal CO^2

pressure, the weight of the suction bag and the degrees of table inclination to be most important [48]. These features are coincidentally also the only features with continuous values and therefore the importance measure is likely to be inflated. Furthermore, the table inclination measurement is corrupted by electrosurgical noise and therefore acts as a proxy for use of the electrosurgical device rather than table inclination itself, again showing the predictive value of electrosurgical devices for surgical progress estimation.

5.6. Data limitations

The instrument usage data used in the current study was generated by the manual analysis of video and audio recordings of laparoscopic hysterectomy, and as such is profoundly different from data that would be acquired using sensors. Obtaining real-time instrument use information automatically is possible using currently available technology, for example using an RFID-based tracking system [11, 90, 91] or computer vision algorithms [49, 92]. As elaborated in Appendix A, the possibilities of RFID-based instrument tracking are promising enough for implementation in clinical practice, although stringent optimization will be necessary to achieve high accuracy tracking in the OR.

Recorded sensor data will inevitably contain measurement noise and needs additional processing before instrument usage can be inferred. On the other hand, a sensor-based tracking system might produce more granular data, for example detecting the exact position of an instrument, rather than just an indicator of use. Furthermore, several instruments in the current data set are grouped together, for example at least four types of graspers and forceps are used during the procedure. With sensor-based tracking, these can be identified individually, possibly resulting in better predictions. Given both the increased noise and increased information of sensor-based recordings, the performance of surgical phase recognition models on such a data set compared to manual annotation is hard to predict.

5.7. Model applications

Surgical process models can be applied in a myriad of ways to assist the surgical team intra-operatively. In the current study, the surgical phase recognition model was evaluated on the tasks of surgical end-time prediction and surgical phase extraction.

5.7.1. End-time prediction task

With an MAE of 16 minutes, the linear regression model predicted the surgical end-time 11 minutes better than a simple estimation using only the mean case duration, with similar results using a model trained on the annotated phases and a model trained on the phases generated by the RF model. Somewhat surprisingly, the RF-based model resulted in better predictions than the model using ground-truth phases, an effect especially observable around 90% into the surgery (Figure 4.13). These results can be explained by the notion that the active phase is not always linearly related to the remaining surgical time. For example, in the annotated surgeries, it is quite common that the surgical team returns to the second phase for a certain time while nearing the end of the procedure, in which case the linear model produces a greatly inflated estimation of remaining time. As these relapses to earlier phases are often not predicted by the RF model, the consequential error in end-time prediction is also lower.

The end-time predictions based on recognized phases could likely be improved by a better model choice. A regression tree or a random regression forest would be suspected to outperform linear regression on this task, given the non-linear relations between phase and remaining end-time. Another option would be to use a segmented regression technique such as MARS, which is also able to fit non-linear relations. Both regression random forest and MARS have been used previously for surgical end-time prediction and shown to outperform linear regression [68]. Another way to improve the forecast with respect to clinical practice could be by applying an asymmetric error function, that penalizes cases that run over time and cases that run under time differently. Although labour costs are probably similar during both over- and under-utilization [93], the consequences for patient and staff satisfaction may differ for cases that run late compared to cases that finish early.

Previous literature predicting end-times using an LR model, reported an MAE of 10 minutes [67] and

20 minutes [14]. Again, there is limited ground for comparison of the current findings to previous literature, due to the large differences in used data and approaches, as these previous results use either pre-operative data [67] or sensor-based recordings [14]. Furthermore, the end-time predictions ideally should be compared using a metric that takes into account the surgery length, such as the MAPE. For example the low error (MAE = 10 min.) reported by Gomes et al. (2012) corresponded to a MAPE of 39% due to the short average surgery duration, which is considerably higher than the MAPE of 13% found in the current study.

Other approaches to end-time prediction used HMM, ontology-based models and SVM [12, 13, 32]. Use of the transition probabilities of an HMM in laparoscopic cholecystectomy procedures, resulted in a MAPE of approximately 21% [32]. An approach using an ontology-based gSPM reported a MAPE of 12% in brain tumor resections and a MAPE of 18% in lumbar discectomy [13]. An approach using SVM resulted in 80% of the predictions falling within an MAE of 10 minutes [12].

The MAPE of 13% observed in the current experiment is at the lower bound of previously reported MAPEs (12%-39%) and although meaningful comparison between research is limited by different study designs and differences in reported performance metrics, this should be considered a good result. Considering the fact that the current prediction error can likely be improved using models incorporating non-linear relations, these results are promising.

5.7.2. Phase extraction task

In the phase extraction task, the largest consecutive run of each phase is annotated automatically based on the predictions of the RF model. The results show that the extraction works well on selected phases that have a low prediction error. Due to the fact that this application has only been suggested in earlier work, but has not been implemented, quantitative results cannot be compared. However, as seven out of ten phases already have an average error smaller than four minutes, the author deems the extraction method already useful for creating a database of training material for these phases.

5.8. Limitations and future work

As highlighted several times in this chapter, the main limitation of this study is the question of transferability of the results to actual clinical practice. The current study uses manually annotated data, which is fundamentally different from sensor data. The change in data structure, quality and granularity might heavily impact the model performance when applied to clinical practice, for better or for worse. However, the question of whether implementation is possible at all can be answered to a certain extent. As shown in preliminary work accompanying this thesis (Appendix A) and previous literature [6, 11, 91], the possibilities of RFID-based instrument tracking are promising enough for implementation in clinical practice. At the time of writing, a complete set of laparoscopic instruments is being equipped with sterilizable RFID-tags, to commence clinical testing within the research OR of the Reinier de Graaf Gasthuis, Delft, The Netherlands.

Following from the fact that the phase recognition system has no been applied in clinical practice, several questions remain. The current study focuses on laparoscopic hysterectomy, but it will be interesting to observe how findings generalize to other surgical procedures. Furthermore, an interesting line of study could be directed at linking clinical outcomes, such as medical complications or hospital length of stay, to the surgical workflow. In this way, the surgical phase recognition could contribute to assessment or improvement in surgical performance. Finally, the model predictions might be improved by adding other sources of intra-operative data, such as video and audio recordings, as well as pre-operative data, especially during the first part of the procedure, when little intra-operative data is available. The pre-operative data could for example entail characteristics of the patient or the surgical team.

5.9. Conclusion

This study demonstrated an intra-operative approach to recognizing surgical phases in 40 laparoscopic hysterectomy cases based on manually annotated instrument usage data. Laparoscopic hysterectomy is well suited for surgical phase recognition because of the considerable case durations, variation within the procedure time and the non-trivial state succession.

With an out-of-sample accuracy of 77%, phases were best recognized by a Random Forest model and the model performance was found to be in line with previous research. The computation time of the RF model is sufficiently low for intra-operative clinical applications and the model poses several other advantages, including noise resistance and suitability for parallel processing. Feature importance analysis suggested that tracking a selection of instruments might already prove sufficient for achieving similar predictive performance. Simulating surgical end-time predictions based on the RF model was shown to be promising, as the MAPE of 13% compared favourably to baseline and ground-truth models and the end-time prediction model showed predictive errors on the lower bound of reported errors in previous research. The model is also found suitable for phase extraction for the generation of training material.

Hence, we conclude that the performance of the Random Forest surgical phase recognition model based on intra-operative data of instrument use, has promising performance and is expected to improve surgical end-time predictions when applied to clinical practice. However, as this study is based on manually annotated data, rather than sensor-based recordings, the degree of transferability of these findings to a real-life OR remains an open question. Further research should therefore specifically be aimed at replicating the simulated findings using sensor-based data in an in-vivo clinical setting.

Bibliography

[1] P. Kougias, V. Tiwari, N. R. Barshes, C. F. Bechara, B. Lowery, G. Pisimisis, and D. H. Berger, *Modeling anesthetic times. predictors and implications for short-term outcomes,* Journal of Surgical Research **180**, 1 (2013).

[2] S. A. Erdogan and B. T. Denton, *Surgery planning and scheduling,* Wiley Encyclopedia of Operations Research and Management Science (2011).

[3] F. Lalys and P. Jannin, *Surgical process modelling: a review,* International journal of computer assisted radiology and surgery **9**, 495 (2013).

[4] K. Cleary, H. Y. Chung, and S. K. Mun, *Or2020 workshop overview: operating room of the future,* in *International Congress Series*, Vol. 1268 (Elsevier, 2004) pp. 847–852.

[5] B. Preim and C. P. Botha, *Computer-assisted surgery,* in *Visual Computing for Medicine: Theory, Algorithms, and Applications* (Newnes, 2013) Chap. 17.

[6] M. Kranzfelder, C. Staub, A. Fiolka, A. Schneider, S. Gillen, D. Wilhelm, H. Friess, A. Knoll, and H. Feussner, *Toward increased autonomy in the surgical or: needs, requests, and expectations,* Surgical endoscopy **27**, 1681 (2013).

[7] D. Katić, C. Julliard, A.-L. Wekerle, H. Kenngott, B. P. Müller-Stich, R. Dillmann, S. Speidel, P. Jannin, and B. Gibaud, *Lapontospm: an ontology for laparoscopic surgeries and its application to surgical phase recognition,* International journal of computer assisted radiology and surgery **10**, 1427 (2015).

[8] D. Katić, P. Spengler, S. Bodenstedt, G. Castrillon-Oberndorfer, R. Seeberger, J. Hoffmann, R. Dillmann, and S. Speidel, *A system for context-aware intraoperative augmented reality in dental implant surgery,* International journal of computer assisted radiology and surgery **10**, 101 (2015).

[9] A. Malpani, C. Lea, C. C. G. Chen, and G. D. Hager, *System events: readily accessible features for surgical phase detection,* International Journal of Computer Assisted Radiology and Surgery , 1 (2016).

[10] K. R. Henken, F. W. Jansen, J. Klein, L. P. Stassen, J. Dankelman, and J. J. van den Dobbelsteen, *Implications of the law on video recording in clinical practice,* Surgical endoscopy **26**, 2909 (2012).

[11] C. Meißner and T. Neumuth, *Rfid-based surgical instrument detection using hidden markov models,* Biomed Tech **57** (2012).

[12] A. C. Guédon, M. Paalvast, F. Meeuwsen, D. Tax, A. van Dijke, L. Wauben, M. van der Elst, J. Dankelman, and J. van den Dobbelsteen, *'it is time to prepare the next patient' real-time prediction of procedure duration in laparoscopic cholecystectomies,* Journal of medical systems **40**, 271 (2016).

[13] S. Franke, J. Meixensberger, and T. Neumuth, *Intervention time prediction from surgical low-level tasks,* Journal of Biomedical Informatics **46**, 152 (2013).

[14] R. Nakamura, T. Aizawa, Y. Muragaki, T. Maruyama, and H. Iseki, *Method for end time prediction of brain tumor resections using analysis of surgical navigation information and tumor size characteristics,* in *World Congress on Medical Physics and Biomedical Engineering May 26-31, 2012, Beijing, China* (Springer, 2013) pp. 1452–1455.

[15] C. E. Reiley, H. C. Lin, D. D. Yuh, and G. D. Hager, *Review of methods for objective surgical skill evaluation,* Surgical endoscopy **25**, 356 (2011).

[16] J. Rosen, J. D. Brown, L. Chang, M. N. Sinanan, and B. Hannaford, *Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete markov model,* IEEE Transactions on Biomedical engineering **53**, 399 (2006).

[17] S.-Y. Ko, J. Kim, W.-J. Lee, and D.-S. Kwon, *Surgery task model for intelligent interaction between surgeon and laparoscopic assistant robot,* International Journal of Assitive Robotics and Mechatronics **8**, 38 (2007).

[18] S. Y. Ko, W.-J. Lee, and D.-S. Kwon, *Intelligent interaction based on a surgery task model for a surgical assistant robot: Awareness of current surgical stages based on a surgical procedure model,* International Journal of Control, Automation and Systems **8**, 782 (2010).

[19] B. Gibaud, C. Penet, and J. Pierre, *Ontospm: a core ontology of surgical procedure models,* (SURGETICA, 2014).

[20] F. van Luyn, *Online estimation of surgical progress,* (Literature Thesis) (2016).

[21] C. Cao, C. L. MacKenzie, and S. Payandeh, *Task and motion analyses in endoscopic surgery,* in *Proceedings ASME Dynamic Systems and Control Division* (Citeseer, 1996) pp. 583–590.

[22] L. MacKenzie, J. Ibbotson, C. Cao, and A. Lomax, *Hierarchical decomposition of laparoscopic surgery: a human factors approach to investigating the operating room environment,* Minimally Invasive Therapy & Allied Technologies **10**, 121 (2001).

[23] K. Den Boer, J. Dankelman, D. Gouma, and H. Stassen, *Peroperative analysis of the surgical procedure,* Surgical endoscopy **16**, 492 (2002).

[24] F. Lalys, L. Riffaud, D. Bouget, and P. Jannin, *A framework for the recognition of high-level surgical tasks from video images for cataract surgeries,* Biomedical Engineering, IEEE Transactions on **59**, 966 (2012).

[25] L. Bouarfa, P. Jonker, and J. Dankelman, *Surgical context discovery by monitoring low-level activities in the or,* in *MICCAI workshop on modeling and monitoring of computer assisted interventions (M2CAI). London, UK* (2009).

[26] T. R. Gruber, *Toward principles for the design of ontologies used for knowledge sharing?* International journal of human-computer studies **43**, 907 (1995).

[27] J. H. Silber, P. R. Rosenbaum, X. Zhang, and O. Even-Shoshan, *Influence of patient and hospital characteristics on anesthesia time in medicare patients undergoing general and orthopedic surgery.* Anesthesiology **106**, 356 (2007).

[28] M. J. Eijkemans, M. van Houdenhoven, T. Nguyen, E. Boersma, E. W. Steyerberg, and G. Kazemier, *Predicting the unpredictable: A new prediction model for operating room times using individual characteristics and the surgeon's estimate,* Anesthesiology **112**, 41 (2010).

[29] J. Zhou, F. Dexter, A. Macario, and D. A. Lubarsky, *Relying solely on historical surgical times to estimate accurately future surgical times is unlikely to reduce the average length of time cases finish late,* Journal of clinical anesthesia **11**, 601 (1999).

[30] S. Arora, N. Sevdalis, D. Nestel, M. Woloshynowych, A. Darzi, and R. Kneebone, *The impact of stress on surgical performance: a systematic review of the literature,* Surgery **147**, 318 (2010).

[31] D. A. Wiegmann, A. W. ElBardissi, J. A. Dearani, R. C. Daly, and T. M. Sundt, *Disruptions in surgical flow and their relationship to surgical errors: an exploratory investigation,* Surgery **142**, 658 (2007).

[32] N. Padoy, T. Blum, H. Feussner, M.-O. Berger, and N. Navab, *On-line recognition of surgical activity for monitoring in the operating room.* in *AAAI* (2008) pp. 1718–1724.

[33] M. Paalvast, F. Meeuwsen, D. Tax, A. van Dijke, L. Wauben, M. van der Elst, J. Dankelman, J. van den Dobbelsteen, *et al.*, *Real-time estimation of surgical procedure duration,* in *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on* (IEEE, 2015) pp. 6–10.

[34] B. Bhatia, T. Oates, Y. Xiao, and P. Hu, *Real-time identification of operating room state from video,* in *AAAI*, Vol. 2 (2007) pp. 1761–1766.

[35] J. J. Leong, M. Nicolaou, L. Atallah, G. P. Mylonas, A. W. Darzi, and G.-Z. Yang, *Hmm assessment of quality of movement trajectory in laparoscopic surgery,* in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2006) pp. 752–759.

[36] L. A. Scherer, M. C. Chang, J. W. Meredith, and F. D. Battistella, *Videotape review leads to rapid and sustained learning,* The American journal of surgery **185**, 516 (2003).

[37] O. Weede, A. Bihlmaier, J. Hutzl, B. P. Müller-Stich, and H. Wörn, *Towards cognitive medical robotics in minimal invasive surgery,* in *Proceedings of Conference on Advances In Robotics* (ACM, 2013) pp. 1–8.

[38] O. Weede, H. Mönnich, B. Müller, and H. Wörn, *An intelligent and autonomous endoscopic guidance system for minimally invasive surgery,* in *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (IEEE, 2011) pp. 5762–5768.

[39] S.-A. Ahmadi, T. Sielhorst, R. Stauder, M. Horn, H. Feussner, and N. Navab, *Recovery of surgical workflow without explicit models,* in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006* (Springer, 2006) pp. 420–428.

[40] B. Ammori, M. Larvin, and M. McMahon, *Elective laparoscopic cholecystectomy: Preoperative prediction of duration of surgery,* Surgical endoscopy **15**, 297 (2001).

[41] T. Blum, N. Padoy, H. Feußner, and N. Navab, *Workflow mining for visualization and analysis of surgeries,* International journal of computer assisted radiology and surgery **3**, 379 (2008).

[42] T. Blum, H. Feußner, and N. Navab, *Modeling and segmentation of surgical workflow from laparoscopic video,* in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010* (Springer, 2010) pp. 400–407.

[43] L. Bouarfa, P. P. Jonker, and J. Dankelman, *Discovery of high-level tasks in the operating room,* Journal of biomedical informatics **44**, 455 (2011).

[44] A. Haji, A. Khan, A. Haq, and B. Ribeiro, *Elective laparoscopic cholecystectomy for surgical trainees: predictive factors of operative time,* The Surgeon **7**, 207 (2009).

[45] A. James, D. Vieira, B. Lo, A. Darzi, and G.-Z. Yang, *Eye-gaze driven surgical workflow segmentation,* in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007* (Springer, 2007) pp. 110–117.

[46] C. Loukas and E. Georgiou, *Surgical workflow analysis with gaussian mixture multivariate autoregressive (gmmar) models: a simulation study,* Computer Aided Surgery **18**, 47 (2013).

[47] N. Padoy, T. Blum, S.-A. Ahmadi, H. Feussner, M.-O. Berger, and N. Navab, *Statistical modeling and recognition of surgical workflow,* Medical image analysis **16**, 632 (2012).

[48] R. Stauder, A. Okur, L. Peter, A. Schneider, M. Kranzfelder, H. Feussner, and N. Navab, *Random forests for phase detection in surgical workflow analysis,* in *Information Processing in Computer-Assisted Interventions* (Springer, 2014) pp. 148–157.

[49] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, *Endonet: A deep architecture for recognition tasks on laparoscopic videos,* arXiv (preprint) (2016).

[50] P. Gambadauro, V. Campo, and S. Campo, *How predictable is the operative time of laparoscopic surgery for ovarian endometrioma?* Minimally invasive surgery **2015** (2015).

[51] W.-C. Hsu, J.-S. Hwang, W.-C. Chang, S.-C. Huang, B.-C. Sheu, and P.-L. Torng, *Prediction of operation time for laparoscopic myomectomy by ultrasound measurements,* Surgical endoscopy **21**, 1600 (2007).

[52] A. M. Alenizi, R. Valdivieso, E. Rajih, M. Meskawi, C. Toarta, M. Bienz, M. Azizi, P. A. Hueber, H. Lavigueur-Blouin, V. Trudeau, *et al.*, *Factors predicting prolonged operative time for individual surgical steps of robot-assisted radical prostatectomy (rarp): A single surgeon's experience,* Canadian Urological Association Journal **9**, E417 (2015).

[53] B. Wehman, E. J. Lehr, K. Lahiji, J. D. Lee, Z. N. Kon, J. Jeudy, B. P. Griffith, and J. Bonatti, *Patient anatomy predicts operative time in robotic totally endoscopic coronary artery bypass surgery,* Interactive cardiovascular and thoracic surgery , ivu226 (2014).

[54] K. N. Wright, G. M. Jonsdottir, S. Jorgensen, N. Shah, and J. I. Einarsson, *Costs and outcomes of abdominal, vaginal, laparoscopic and robotic hysterectomies,* JSLS **16**, 519 (2012).

[55] M. Loring, S. N. Morris, and K. B. Isaacson, *Minimally invasive specialists and rates of laparoscopic hysterectomy,* JSLS: Journal of the Society of Laparoendoscopic Surgeons **19** (2015).

[56] M. F. R. Paraiso, B. Ridgeway, A. J. Park, J. E. Jelovsek, M. D. Barber, T. Falcone, and J. I. Einarsson, *A randomized trial comparing conventional and robotically assisted total laparoscopic hysterectomy,* American journal of obstetrics and gynecology **208**, 368 (2013).

[57] M. K. Whiteman, S. D. Hillis, D. J. Jamieson, B. Morrow, M. N. Podgornik, K. M. Brett, and P. A. Marchbanks, *Inpatient hysterectomy surveillance in the united states, 2000-2004,* American journal of obstetrics and gynecology **198**, 34 (2008).

[58] J. I. Einarsson and Y. Suzuki, *Total laparoscopic hysterectomy: 10 steps toward a successful procedure,* Reviews in Obstetrics and Gynecology **2**, 57 (2009).

[59] D. T. Tran, R. Sakurai, and J.-H. Lee, *An improvement of surgical phase detection using latent dirichlet allocation and hidden markov model,* in *Innovation in Medicine and Healthcare 2015* (Springer, 2016) pp. 249–261.

[60] S. Agarwal, A. Joshi, T. Finin, Y. Yesha, and T. Ganous, *A pervasive computing system for the operating room of the future,* Mobile Networks and Applications **12**, 215 (2007).

[61] S. Franke, J. Meixensberger, and T. Neumuth, *Multi-perspective workflow modeling for online surgical situation models,* Journal of biomedical informatics **54**, 158 (2015).

[62] R. Stauder, A. Okur, and N. Navab, *Detecting and analyzing the surgical workflow to aid human and robotic scrub nurses,* in *The Hamlyn Symposium on Medical Robotics* (2014) p. 91.

[63] M. Maktabi, S. T. Vinz, and T. Neumuth, *Frequency based assessment of surgical activities,* Current Directions in Biomedical Engineering **1**, 152 (2015).

[64] G. Forestier, L. Riffaud, and P. Jannin, *Automatic phase prediction from low-level surgical activities,* International journal of computer assisted radiology and surgery **10**, 833 (2015).

[65] D. M. J. Tax, *One-class classification*, Ph.D. thesis, TU Delft, Delft University of Technology (2001).

[66] S. P. Devi, K. S. Rao, and S. S. Sangeetha, *Prediction of surgery times and scheduling of operation theaters in optholmology department,* Journal of medical systems **36**, 415 (2012).

[67] C. Gomes, B. Almada-Lobo, J. Borges, and C. Soares, *Integrating data mining and optimization techniques on surgery scheduling,* in *Advanced Data Mining and Applications* (Springer, 2012) pp. 589–602.

[68] Z. ShahabiKargar, S. Khanna, N. Good, A. Sattar, J. Lind, and J. O'Dwyer, *Predicting procedure duration to improve scheduling of elective surgery,* in *PRICAI 2014: Trends in Artificial Intelligence* (Springer, 2014) pp. 998–1009.

[69] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees* (CRC press, 1984).

[70] T. M. Therneau, E. J. Atkinson, *et al.*, *An introduction to recursive partitioning using the RPART routines*, Tech. Rep. (Technical report Mayo Foundation, 1997).

[71] L. Breiman, *Random forests,* Machine learning **45**, 5 (2001).

[72] L. Breiman, *Manual on setting up, using, and understanding random forests v3. 1,* Statistics Department University of California Berkeley, CA, USA **1** (2002).

[73] J. H. Friedman, *Multivariate adaptive regression splines,* The annals of statistics , 1 (1991).

[74] M. Rockstroh, M. Wittig, S. Franke, J. Meixensberger, and T. Neumuth, *Video-based detection of device interaction in the operating room,* Biomedical Engineering/Biomedizinische Technik (2015).

[75] G. D. Forney, *The viterbi algorithm,* Proceedings of the IEEE **61**, 268 (1973).

[76] A. Faisal, *CO424 - Machine Learning and Neural Computation,* Imperial College London, Lecture Notes (2015).

[77] S. Theodoridis, *Pattern recognition* (Academic Press, Burlington, MA London, 2009).

[78] M. D. Blikkendaal, S. R. Driessen, S. P. Rodrigues, J. P. Rhemrev, M. J. Smeets, J. Dankelman, J. J. van den Dobbelsteen, and F. W. Jansen, *Surgical flow disturbances in dedicated minimally invasive surgery suites: an observational study to assess its supposed superiority over conventional suites,* Surgical endoscopy **31**, 288 (2017).

[79] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria (2016).

[80] RStudio Team, *RStudio: Integrated Development Environment for R*, RStudio, Inc., Boston, MA (2016).

[81] A. Liaw and M. Wiener, *Classification and regression by randomforest,* R news **2**, 18 (2002).

[82] C. Ferri, J. Hernández-Orallo, and R. Modroiu, *An experimental comparison of performance measures for classification,* Pattern Recognition Letters **30**, 27 (2009).

[83] M. Friedman, *The use of ranks to avoid the assumption of normality implicit in the analysis of variance,* Journal of the american statistical association **32**, 675 (1937).

[84] J. Demšar, *Statistical comparisons of classifiers over multiple data sets,* Journal of Machine learning research **7**, 1 (2006).

[85] S. L. Salzberg, *On comparing classifiers: Pitfalls to avoid and a recommended approach,* Data mining and knowledge discovery **1**, 317 (1997).

[86] T. G. Dietterich, *Approximate statistical tests for comparing supervised classification learning algorithms,* Neural computation **10**, 1895 (1998).

[87] V. Y. Kulkarni and P. K. Sinha, *Random forest classifiers: a survey and future research directions,* Int J Adv Comput **36**, 1144 (2013).

[88] K. J. Archer and R. V. Kimes, *Empirical characterization of random forest variable importance measures,* Computational Statistics & Data Analysis **52**, 2249 (2008).

[89] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, *Bias in random forest variable importance measures: Illustrations, sources and a solution,* BMC bioinformatics **8**, 25 (2007).

[90] M. Kranzfelder, A. Schneider, A. Fiolka, E. Schwan, S. Gillen, D. Wilhelm, R. Schirren, S. Reiser, B. Jensen, and H. Feussner, *Real-time instrument detection in minimally invasive surgery using radiofrequency identification technology,* journal of surgical research **185**, 704 (2013).

[91] F. Miyawaki, T. Tsunoi, H. Namiki, T. Yaginuma, K. Yoshimitsu, D. Hashimoto, and Y. Fukui, *Development of automatic acquisition system of surgical-instrument informantion in endoscopic and laparoscopic surgey,* in *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on* (IEEE, 2009) pp. 3058–3063.

[92] L. Bouarfa, O. Akman, A. Schneider, P. P. Jonker, and J. Dankelman, *In-vivo real-time tracking of surgical instruments in endoscopic video,* Minimally Invasive Therapy & Allied Technologies **21**, 129 (2012).

[93] F. Dexter, R. D. Traub, and F. Qian, *Comparison of statistical methods to predict the time to complete a series of surgical cases,* Journal of clinical monitoring and computing **15**, 45 (1999).

[94] S. Schwaitzberg, *The emergence of radiofrequency identification tags: applications in surgery,* Surgical Endoscopy and Other Interventional Techniques **20**, 1315 (2006).

[95] A. Guédon, L. Wauben, D. de Korne, M. Overvelde, J. Dankelman, and J. van den Dobbelsteen, *A rfid specific participatory design approach to support design and implementation of real-time location systems in the operating room,* Journal of medical systems **39**, 1 (2015).

[96] C. C. Liu, C.-H. Chang, M.-C. Su, H.-T. Chu, S.-H. Hung, J.-M. Wong, and P.-C. Wang, *Rfid-initiated workflow control to facilitate patient safety and utilization efficiency in operation theater,* Computer methods and programs in biomedicine **104**, 435 (2011).

[97] T. Therneau, B. Atkinson, and B. Ripley, *rpart: Recursive Partitioning and Regression Trees* (2015), r package version 4.1-10.

[98] M. K. C. from Jed Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, the R Core Team, M. Benesty, R. Lescarbeau, A. Ziem, L. Scrucca, Y. Tang, C. Candan, and T. Hunt., *caret: Classification and Regression Training* (2016), r package version 6.0-73.

[99] L. Himmelmann, *HMM: Hidden Markov Models* (2010), r package version 1.0.

[100] H. Wickham and R. Francois, *dplyr: A Grammar of Data Manipulation* (2016), r package version 0.5.0.

[101] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis* (Springer-Verlag New York, 2009).

A

RFID-based instrument detection

A.1. Introduction

As shown in this thesis and previous literature, surgical phase recognition provides applications to aid the surgical team during surgery, for example in the areas of predicting of surgical case durations [13, 14, 32], surgical training [15] and automatic generation of post-operative reports [9, 24].

If the goal is to provide assistance during the procedure itself, a surgical phase recognition system must acquire intra-operative data to use as a real-time input for the model. In choosing the right data sources, there are several aspects to consider: the data should have predictive value for the phase of the surgery and needs to be available for measurement. Previous research has seen several sources of intra-operative data, of which video recordings [24, 34, 49, 59] and instrument usage tracking have been most popularly researched [9, 13, 14, 32, 39, 41, 43, 47, 48, 60]. Other sources of intra-operative data include medical device or apparatus use [61], patient monitoring [60] and monitoring surgeon activity, such as following hand movements [45, 46, 60, 64] .

With currently available technology, instrument tracking by means of radio-frequency identification (RFID) is a promising way of acquiring intra-operative data [6, 9]. An RFID system typically consists of a tag, which is placed on the object that needs to be identified, and an RFID reader, to recognize the tag. Both the tag and reader contain, or are connected to, an antenna for communication. RFID tags can be classified according to operating frequency and energy source [94]. Most medical applications operate in the ultra-high frequency (UHF) band and use passive RFID-tags, indicating that the tags do not contain a battery, but are instead powered by interrogation from the RFID-reader.

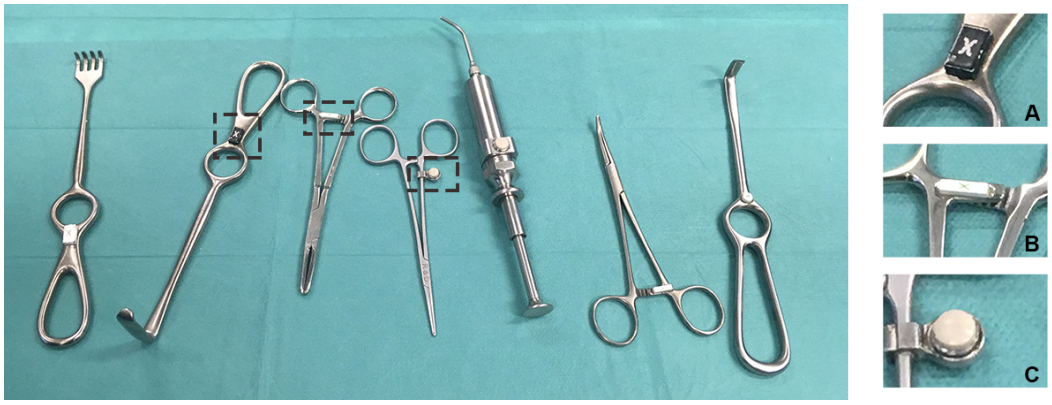


Figure A.1: Several examples of RFID-tagged instruments used in the experimental set-up. The instruments were equipped with several types of XS series sterilizable RFID tags (Xerafy, Hong Kong), which where in some cases glued to the instrument (A, B) and in other cases attached via a proprietary welding and coating technique developed by Van Straten Medical B.V., The Netherlands (C).

Previous research has shown that an RFID approach for online detection and localization of surgical instruments is feasible. Kranzfelder et al. (2013) showed that RFID-based localization of instruments at the Mayo stand showed no significant difference from manual annotation based on video recordings [90]. Meißner et al. (2012) obtained an accuracy of 92% (70%-99%) in tracking the location of RFID-tagged instruments in surgeries simulated by medical students, compared to the annotation of a human observer [11]. Miyawaki et al. (2009) developed a system where the RFID antenna was placed on the cannula of the laparoscopic trocar and the RFID tag was attached to the surgical instrument [91]. Next to tracking of individual instruments, other applications of RFID in the OR have been the detection of the presence and status of necessary medical equipment [95] and identification of patients [96].

This appendix details some preliminary, qualitative work in replicating previous research of detection of RFID-tagged instruments in a lab setting using off-the-shelf RFID technology.

A.2. Materials and methods

The passive RFID-based instrument tracking set-up used a Motorola FX9500 RFID-reader (Motorola Solutions[1], New York, USA) and a Slimline CP A6304 RFID antenna (Times-7, Wellington, New Zealand) with a power of 6 W. A set of various medical instruments was equipped with XS series RFID-tags (Xerafy, Hong Kong) (Figure A.1). This type of RFID tag is designed for use with medical devices and can cope with repeated autoclave sterilization cycles. The RFID tags use ultra-high frequency (UHF) band with a range of 866-868 MHz, following ETSI standards. Some instruments feature a proprietary coating of the RFID tag (Van Straten Medical, Nieuwegein, The Netherlands), for enhanced protection against sterilization cycles and robust attachment of the tag to the instrument.

The RFID-reader was connected to a local computer via a UTP cable where the reader settings where accessible via a static IP address. Further communication, such as the real-time tag readings, was established via a serial port (COM-port) using Matlab software (Mathworks Inc., Massachusetts, USA), which was also used for further processing and visualization. Serial port communication with the RFID reader used a baud rate of 115200 bits/second. The complete set-up was realized at the Department of Biomechanical Engineering at Delft University of Technology, The Netherlands (Figure A.2).

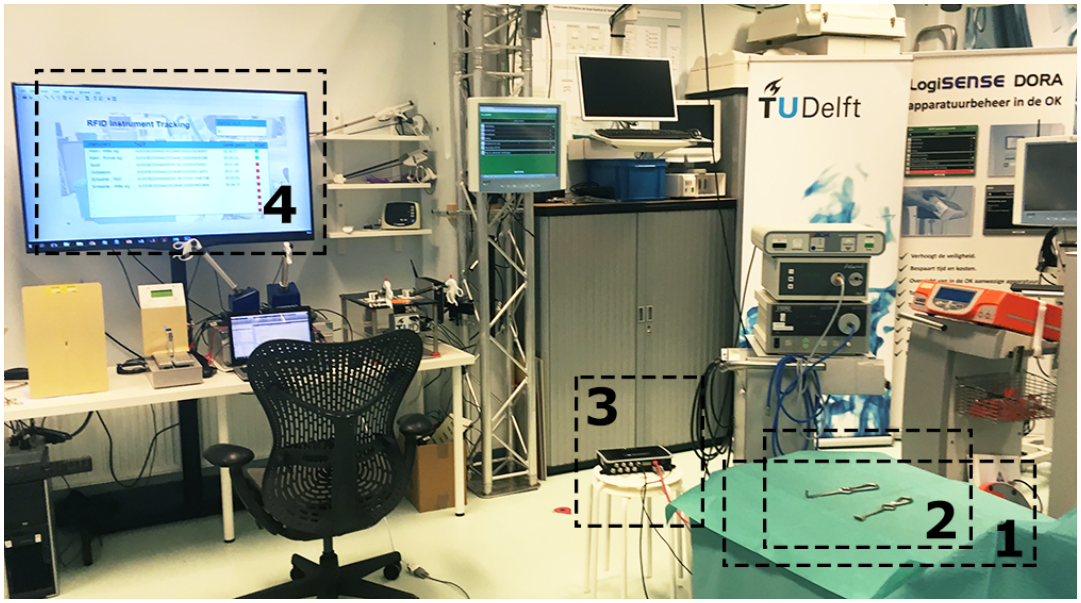


Figure A.2: Experimental set-up of the instrument tracking system. The antenna (1) is placed under surgical drapes to resemble an OR situation. Instruments (2) are placed on the surgical drapes to allow detection by the antenna. The data is sent to the RFID-reader (3) and processed on a computer. The scanned instruments are reported on a screen (4) in a Matlab interface as shown in Figure A.3.

[1]The RFID-technology division of Motorola Solutions has since been acquired by Zebra Technologies (Illinois, USA)

A.3. Results

Although no controlled experiments have been performed with the RFID-based instrument tracking set-up, some qualitative results of the preliminary testing can be shared. All instruments were recognized by the set-up, although in most cases only when the instrument was oriented with the tag facing the antenna. As expected, larger tags performed noticeably better in terms of detection compared to smaller variants and uncoated tags outperformed the coated ones. No reading distances larger than approximately 2 cm were observed for any of the tags. The Matlab implementation showed live updates of the tags within the field on the antenna with an update frequency of 1 Hz maximum (Figure A.3). The latency of approximately 1 second in the live demo were caused mostly by the serial communication with the RFID reader, as this was limited by the baud rate, and implementation in Matlab itself.

A.4. Discussion

The preliminary results described in this section shows that it is indeed possible to create a simple RFID-based instrument detection system in a controlled lab setting, as repeatedly shown in previous literature [11, 90, 91].

The current experimental set-up can be improved in several ways. Since the communication via COM-port has a fairly limited bit rate, communication via the LAN-interface is preferred for real-time detection. The range in which instruments are detected can be improved by use of more powerful antennas and by adding more antennas. The used RFID reader allows for eight simultaneously used antennas, an amount also used in previous research [90].

Transferring the current set-up to actual clinical practice will face several challenges. First and foremost is the aspect of medical safety, including sterilization and cleaning of the instruments. Kranzfelder et al. (2013) used separate RFID-tags of considerable size (2.2 cm diameter), that were separately sterilized and mounted on the instruments in the OR, before the surgery. Related to the extra workload, the scrub nurses rate the solution significantly worse than the surgeon in terms of the RFID tag mounting solution and in terms of the system compromising the course of the surgery [90]. The mounting solution developed by Van Straten Medical (Figure A.1) is designed to be semi-permanent and withstand repeated cycles of sterilization and as such does not add increased workload to the surgical team.

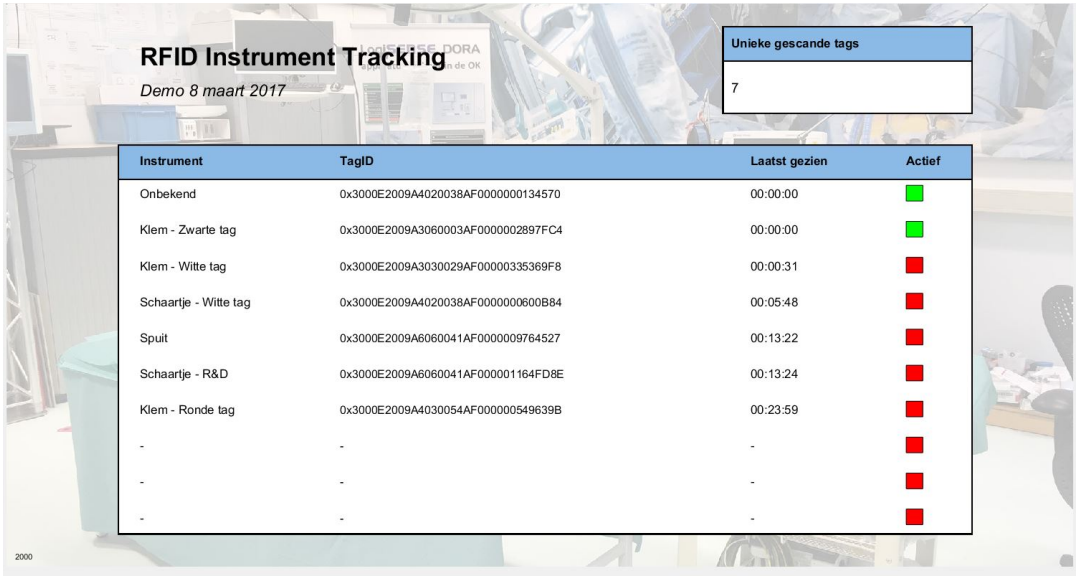


Figure A.3: Prototype of a live software implementation in Matlab listing the instruments detected by the RFID-based tracking set-up. The interface shows the total amount of distinct instruments scanned during this session in the top right corner. Furthermore, it details the ten last observed instruments, by their name as stored in a tag-database, their unique identifier and the time elapsed since the instrument was last observed. The colored indicator that shows whether the instrument was scanned within the last 30 seconds. The current set-up has a refresh rate around 1Hz.

This semi-permanent connection also poses opportunities for application in hospital logistics, as the individual instrument can be tracked during the complete use cycle.
Another limitation in practical implementation is the necessary orientation of the instrument towards the antenna. Our results show that detection of the tag is problematic if the tag is not placed downwards, facing the antenna. Previous research also reports that specific instrument geometry and placing of the tags can cause underperformance of instrument detection [11]. A solution might be found in the placement of more antennas, with increased reading power.

Electrosurgery is a commonly used technique in laparoscopic surgery, however the application of current is shown to distort radio-frequency communication used by the RFID-detection system, resulting in missed readings and increased sensor noise [90, 91]. A solution proposed in previous research is the use of intelligent post-processing algorithms using for example sensor-fusion or Hidden Markov Models [11]. Additionaly it is also possible to track the activity of the electrosurgery device via a current sensor, as already shown in clinical practice during laparoscopic cholecystectomy procedures [12, 33].

A.5. Conclusion

The successful implementation of an RFID-based instrument tracking system balances on the trade-offs between clinical safety and system performance. Small tags with sterilizable and cleanable coatings allow implementation of the tracking system without affecting surgical workflow, however this comes at the cost of decreased readability. This preliminary work has shown that the possibilities of RFID-based instrument tracking are promising enough for implementation in clinical practice, although stringent optimization will be necessary to achieve high accuracy tracking.

B

Model implementation

This appendix provides additional detail into the implementation and resulting models that were generated by the 10-fold cross-validation optimization procedure. All models were implemented in the R programming language (R Foundation for Statistical Computing, Vienna, Austria) using RStudio IDE (RStudio inc., Boston, U.S.A.) [79, 80]. The CART model was realized using the `rpart` package [97] and the RF model using the `randomForest` package [81]. The KNN model was implemented using the `caret` library [98] and the HMM implementation used the `HMM` package [99]. Two other important R packages used in the modelling process were `dplyr` for data transformations and `ggplot2` for visualizing results [100, 101].

The final **CART model** is shown in Figure B.1. Using a complexity parameter of $cp = 0.005$, the structure of the generated decision tree stays rather simple as it features a total of only 18 splits, and the maximum observed tree depth is seven nodes. Also note that phase 9 is never predicted, which results in an accuracy of zero on this specific phase.

The selected **RF model** is an ensemble ($n = 100$) of unpruned CART trees. To assess the performance of the RF model in more detail, the confusion matrix can be inspected (Figure B.2). The confusion matrix shows the phase-wise accuracy on the diagonal, with all off-diagonal terms highlighting misclassifications. Again it can be seen that phase 9 is most problematic, followed by phase 3.

An **HMM** (λ) can be defined by three matrices, the state transition probability matrix (A), the observation probability matrix (B) and the initial state probability matrix (π), which is trivial for the current data set, since the surgery always commences in the first phase. Figure B.4 shows the state transition probability matrix as derived from the forty LH cases, sampled at $t_{step} = 15$ seconds. The diagonal terms of the matrix are related to the expected duration of the phase, with shorter phases having a smaller probability of staying in the same phase and a larger probability of transitioning. It can be seen that phase 3 has the most diverse options, as it can transition to phase 2, 4, 5, 6, 8 or 10 based on the training data. Phase 10 is an absorbing state, as there is zero probability of transitioning out. The A-matrix can also be visualized as a graph, with the nodes representing surgical phases and the edges the transition probabilities (Figure B.3). The B-matrix shows the probabilities of observing a certain combination of instruments used, given that the surgery is in a certain current phase (Figure B.5).

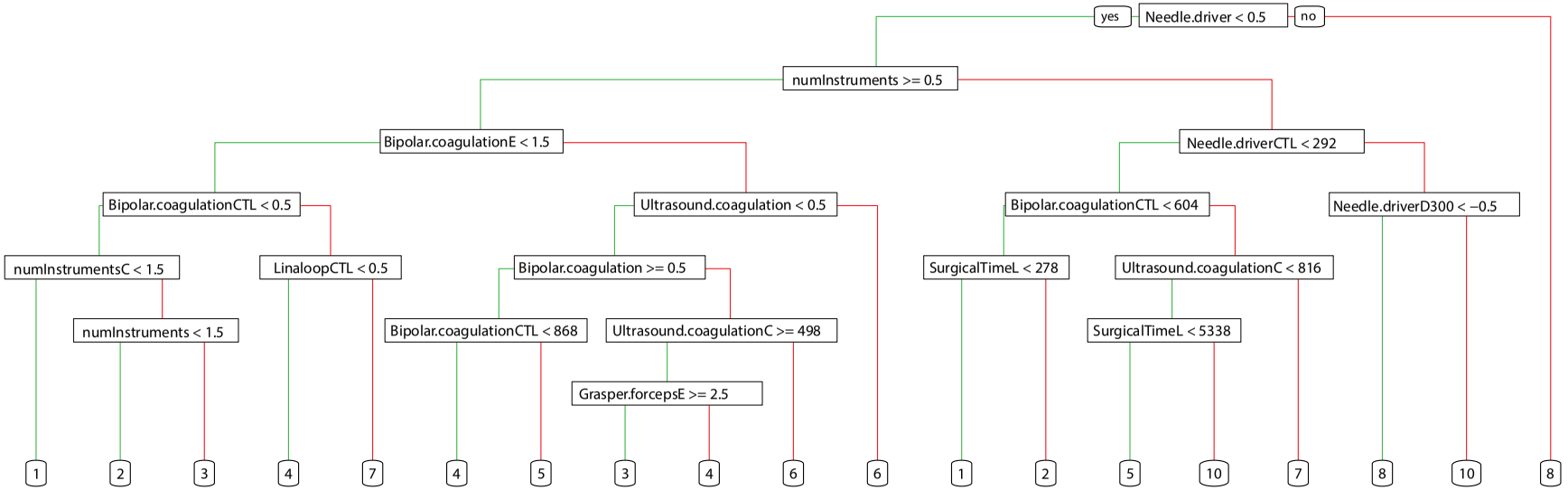


Figure B.1: Optimized CART model ($cp = 0.005$). The shown model has been trained one of the training set folds (n=36 cases). The CART model will differ slightly for each of the ten different folds.

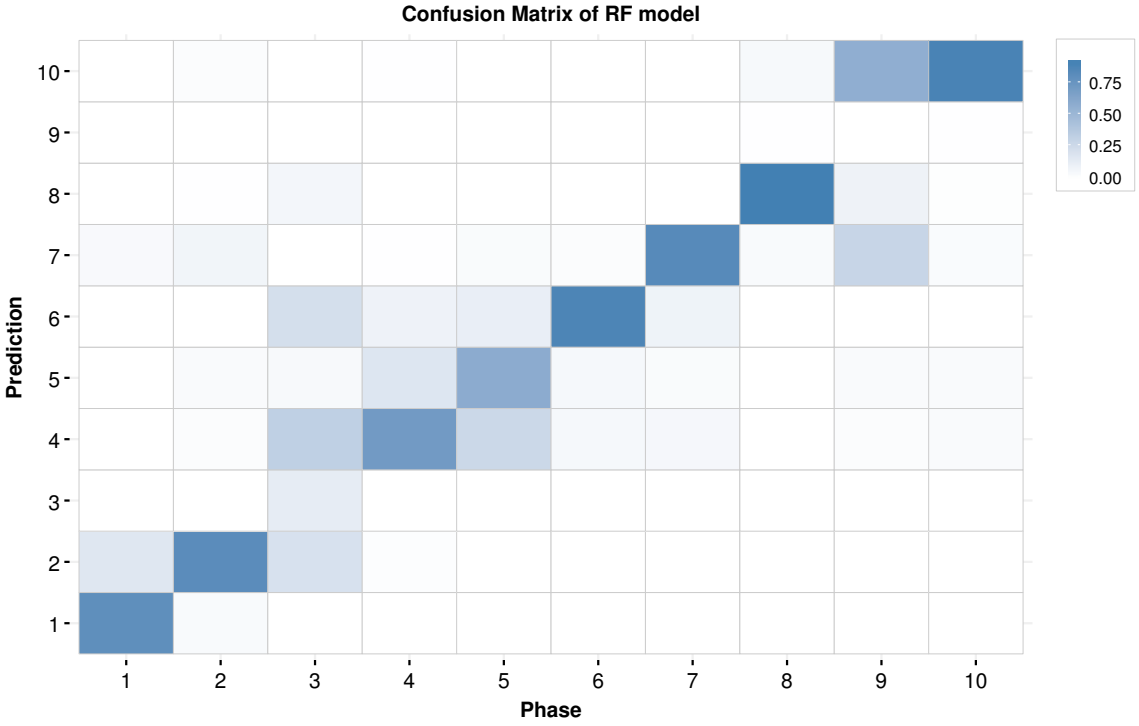


Figure B.2: Confusion matrix of the optimized RF model ($m_{try} = 6, n = 100$). The figure shows the probability of seeing a certain phase prediction, given a certain ground-truth phase. Hence, the diagonal resembles the phase-wise prediction accuracy. and the columns sum up to one.

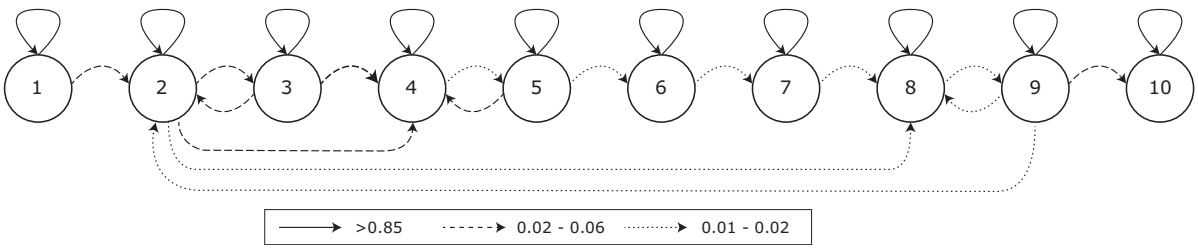


Figure B.3: Visualization of the Hidden Markov Model as a graph. The nodes represent the different surgical phases of the LH, the edges represent the probabilities of changing from one phase to another. Probabilities are estimated using all 40 surgeries with a time step of 15 seconds. For clarity only transitions with a probability higher than 0.01 are shown in the figure. Please see Figure B.4 for the transition probabilities between all hidden states. Note that the edges indicate the prior probabilities, based only on the transitions in the data set. The posterior probabilities are dependent on the observed output (observation symbols).

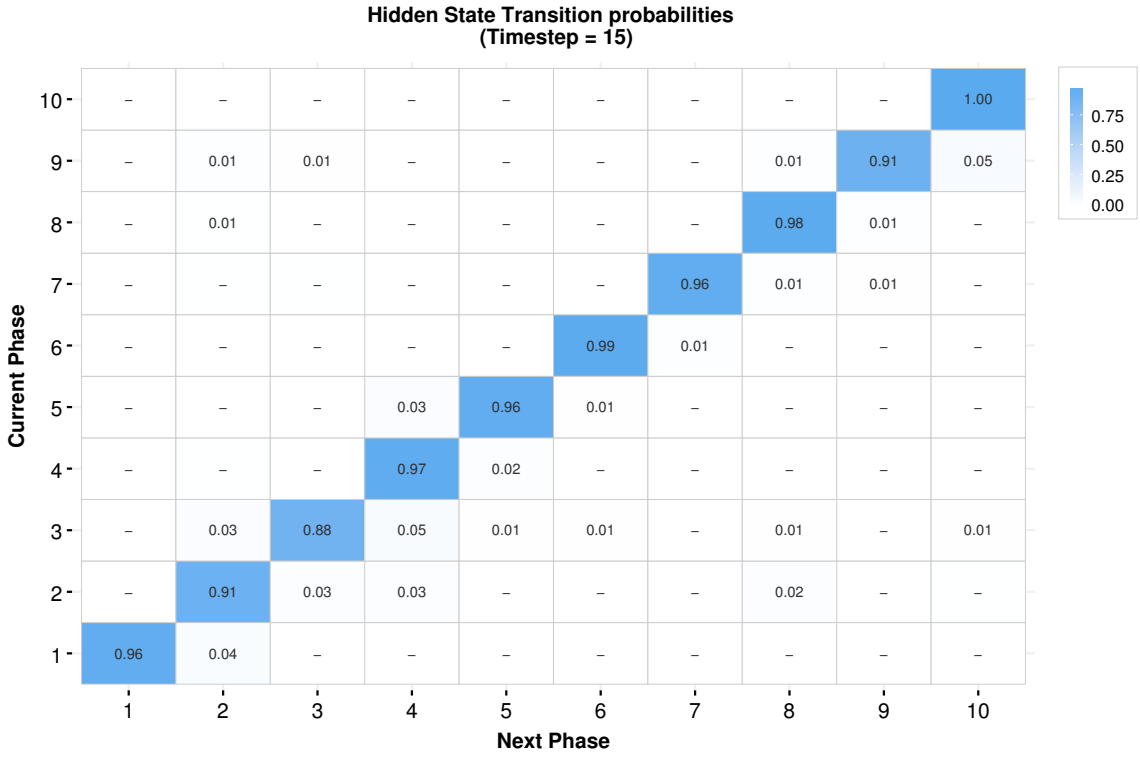


Figure B.4: Hidden state transition probability matrix of an HMM with $t_{step} = 15$. The diagonal terms are related to the expected phase duration based on the training data, where the transition probabilities summary the phase transitions occurring in the 40 LH cases.

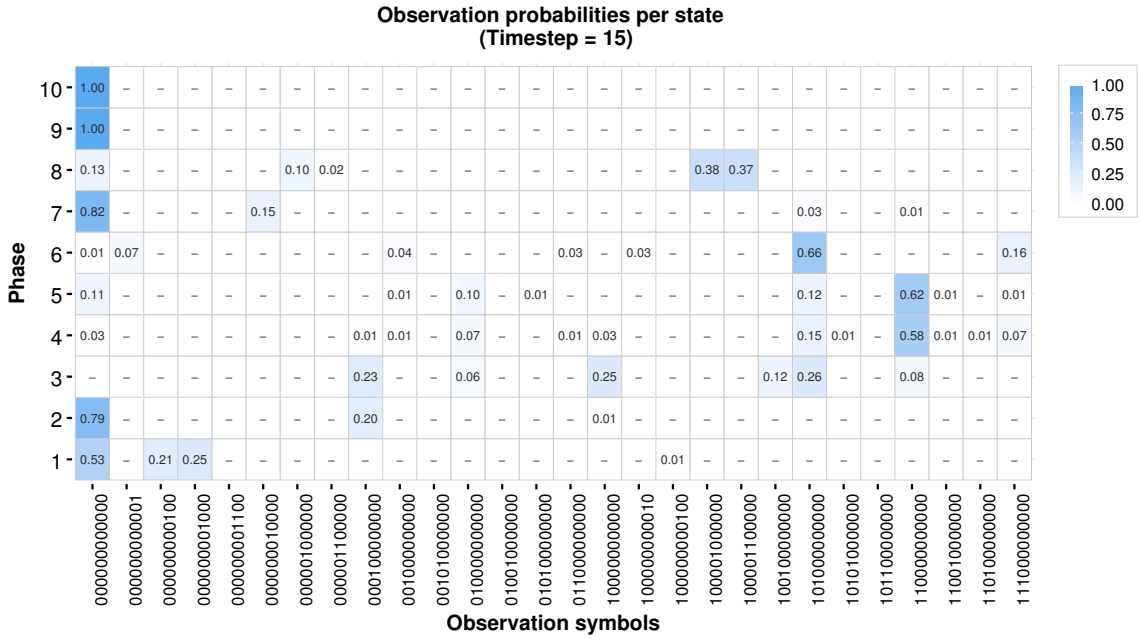


Figure B.5: Observation probability matrix of an HMM with $t_{step} = 15$. Note that the total observation matrix would feature $2^{12} = 4096$ columns, but the overwhelming majority of combinations between used instruments is not observed in the data set. Therefore, all but the 28 shown observations symbols can be omitted as the probability equals zero for all phases.