LUND UNIVERSITY

Deep Learning Applications for Biomedical Data and Natural Language Processing

Medved, Dennis

2018

Document Version:
Publisher's PDF, also known as Version of record

Deep Learning Applications for Biomedical Data and Natural Language Processing

Dennis Medved

Doctoral Dissertation, 2018

Department of Computer Science
Lund University

ISBN 978-91-7753-792-2 (printed version)
ISBN 978-91-7753-793-9 (electronic version)
ISSN 1404-1219
LU-CS-DISS: 2018-03
Dissertation 60, 2018

Department of Computer Science
Lund University
Box 118
SE-221 00 Lund
Sweden

Email: dennis.medved@cs.lth.se
WWW: http://cs.lth.se/dennis_medved

Typeset using LaTeX
Printed in Sweden by Tryckeriet i E-huset, Lund, 2018

Abstract

The human brain can be seen as an ensemble of interconnected neurons, more or less specialized to solve different cognitive and motor tasks. In computer science, the term deep learning is often applied to signify sets of interconnected nodes, where deep means that they have several computational layers. Development of deep learning is essentially a quest to mimic how the human brain, at least partially, operates.

In this thesis, I will use machine learning techniques to tackle two different domain of problems. The first is a problem in natural language processing. We improved classification of relations within images, using text associated with the pictures. The second domain is regarding heart transplant. We created models for pre- and post-transplant survival and simulated a whole transplantation queue, to be able to asses the impact of different allocation policies. We used deep learning models to solve these problems.

As introduction to these problems, I will present the basic concepts of machine learning, how to represent data, how to evaluate prediction results, and how to create different models to predict values from data. Following that, I will also introduce the field of heart transplant and some information about simulation.

Contents

Preface

This thesis consists of two parts. The first part describes the research context, the chosen research methodology, and the findings of the thesis, as well as an agenda for future work. The second part contains the six research papers on which the conclusions in the first part are based.

List of Included Publications

I **Improving the Detection of Relations Between Objects in an Image Using Textual Semantics**
Dennis Medved, Fangyuan Jiang, Peter Exner, Magnus Oskarsson, Pierre Nugues, Kalle Åström
International Conference on Pattern Recognition Applications and Methods, 133–145, 2014

II **Using a RDF Triplestore and Deep Learning to Predict Heart Transplantation Survival**
Dennis Medved, Pierre Nugues, Johan Nilsson
To be submitted, 2018

III **Selection of an optimal feature set to predict heart transplantation outcomes**
Dennis Medved, Pierre Nugues, Johan Nilsson
Engineering in Medicine and Biology Society (EMBC), 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 3290–3293, 2016

IV **Predicting the Outcome for Patients in a Heart Transplantation Queue using Deep Learning**
Dennis Medved, Pierre Nugues, Johan Nilsson
Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 74-77, 2017

V **Improving Prediction of Heart Transplantation Outcome Using Deep Learning Techniques**
Dennis Medved, Mattias Ohlsson, Peter Höglund, Bodil Andersson Pierre Nugues, Johan Nilsson
Nature - Scientific Reports, 2018

VI **Simulating the Outcome of Heart Allocation Policies using Deep Neural Networks**
Dennis Medved, Pierre Nugues, Johan Nilsson
Accepted to the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2018

Contribution Statement

Dennis Medved is the main contributor to all of the included papers, in this doctoral thesis. He was the main designer and implementer of the research experiments, and responsible for most of the writing on the articles.

The supervisors, Prof. Pierre Nugues and Prof. Johan Nilsson, contributed to the design of the experiments, writing on the articles and reviewed the content of the papers.

List of Related Publications

VII **Streamlining a Transplantation Survival Prediction Program with a RDF Triplestore**
Dennis Medved, Johan Nilsson, Pierre Nugues
In Proceedings of the 9th International Conference on Data Integration in the Life Sciences, System papers, 2013

VIII **Using Deep Neural Networks to Simulate Heart Allocation Policies**
Dennis Medved, Pierre Nugues, Johan Nilsson
The Journal of Heart and Lung Transplantation 37.4, 171–172, 2018

IX **Combining Text Semantics and Image Geometry to Improve Scene Interpretation**
Dennis Medved, Fangyuan Jiang, Peter Exner, Magnus Oskarsson, Pierre Nugues, Kalle Åström
In Proceedings of ICPRAM 2014 The 3rd International Conference on Pattern Recognition Applications and Method, 479-486, 2014.

X **Using Syntactic Dependencies to Solve Coreferences**
Marcus Stamborg, Dennis Medved, Peter Exner, Pierre Nugues
Joint Conference on EMNLP and CoNLL 2012 Shared Task, 64–70, 2012.

XI **Image segmentation and labeling using free-form semantic annotation**
Agnes Tegen, Rebecka Weegar, Linus Hammarlund, Magnus Oskarsson, Fangyuan Jiang, Dennis Medved, Pierre Nugues, Kalle Åström
2014 22nd International Conference on Pattern Recognition (ICPR), 2281–2286, 2014.

Comment

Paper VII (Streamlining a Transplantation Survival Prediction Program with a RDF Triplestore) was accepted to the 9th International Conference on Data Integration in the Life Sciences, but it was not published in any archival system.

We, therefore, rewrote it as Paper II (Using a RDF Triplestore and Deep Learning to Predict Heart Transplantation Survival) and it is to be submitted.

Acknowledgements

This research was supported by the Heart Lung Foundation, the Swedish Research Council, and the eSSENCE program.

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Pierre Nugues for the continuous support of my Ph.D. study, for his patience, motivation, and immense knowledge. His guidance helped during the time of research and writing of this thesis. Secondly, I would like to thank Prof. Johan Nilsson for introducing me to the field of heart transplant, his advice and guidance have been invaluable for my research.

I would like to thank M. Eng. Marcus Klang, my roommate, for all of his advice and the interesting discussion we have had.

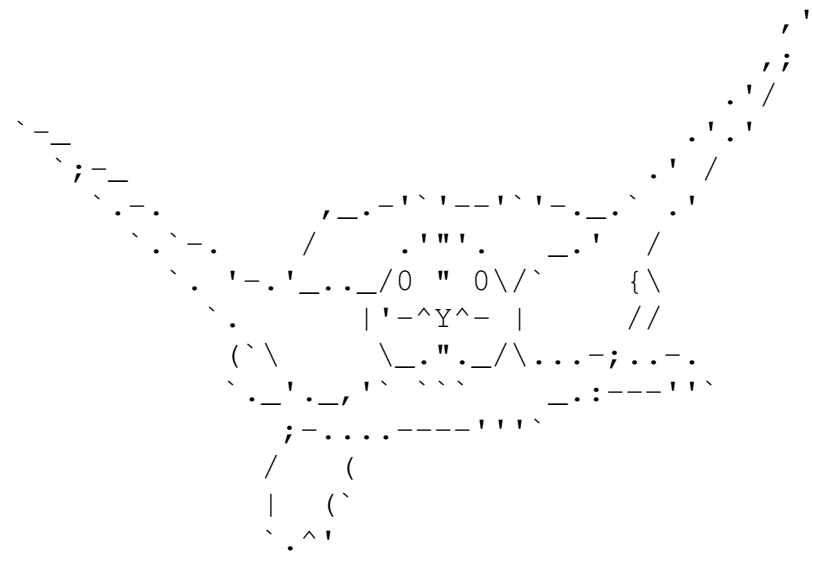
A special thanks to Tekn. Lic. Lars Nilsson for being such a good friend. We have had countless lunches and dinners, seen many motion pictures and travelled to several exotic places together.

Also many thanks to Svenska Frackademien, "Snille och klädsmak!".

I would also like to thank my family, without my parents I probably would not have existed. They are very kind and loving people, and I am fortunate to be raised by them. Many thanks to my brother and sister, for teasing me and being there when it counts. Thanks to my uncle M. Sci. Bengan for getting me interested in technology and computers.

Finally, I would like to thank my girlfriend Malin Andersson for being there — you are a special unicorn! <3

Dennis Medved
Lund, June 2018



Maybe the only significant difference between a really smart simulation and a human being was the noise they made when you punched them[1].

– Terry Pratchett

[1] T. Pratchett, S. Baxter, The Long Earth, 2012.

Popular Science Summary in Swedish

Datorn luskar ut vem som får störst nytta utav hjärtan

Dennis Medved
Institutionen för datavetenskap
Lunds universitet
Lund, Sverige
dennis.medved@cs.lth.se

Människor är duktiga på att lära sig saker. Datorer är duktiga på att utföra många enkla instruktioner på kort tid. Hur får man en dator att lära sig något?

1 Intelligens

1.1 Vad är intelligens?

Detta är en filosofisk fråga, som har många svar. Som delar av intelligens räknas vanligen förmågorna att resonera, planera, lösa problem, associera, tänka abstrakt, förstå idéer och språk, samt förmågan till inlärning. Mer generellt så kan intelligens beskrivas som förmågan att inhämta information, bevara det som kunskap och sedan utnyttja det i en miljö.

Intelligens är egenskaper som ofta brukas tillskrivas människor och djur. Men kan en dator beskrivas som intelligent?

En dator klarar inte av, för tillfället, generell problemlösning, iallafall inte liknande den som människor kan utföra. Den kan uppvisa vissa av egenskaperna som brukar räknas till intelligens och kan utnyttja de här förmågorna till att lösa väldigt specifika problem, dock i många fall bättre än människor. Ett exempel på detta är när Gary Kasparov, den dåvarande världsmästaren, mötte IBM:s superdator Deep Blue, i en serie av schackmatcher 1997 och datorn vann mot Kasparov.

En dator, ifall den skulle ha en fysik manifestation, i form av en robot eller dylikt, skulle antagligen ha problem att laga frukost, knyta skosnörerna, köra till jobbet, prata med kollegor, med mera. Sysslor som de flesta människor kan göra utan allt för stor ansträngning.

Det som en nuvarande dator klarar av att utföra brukar kallas för snäv intelligens, medan det människor uppvisar brukar beskrivas som generell intelligens.

1.2 Vad är maskininlärning?

För att skapa ett datorprogram som uppvisar snäv intelligens, så brukar oftast maskininlärning användas. För många problem så är det svårt att skriva specifika regler som datorn ska följa för att lösa det. Datorseende är ett sådant problem, till exempel ifall man kopplar en kamera till datorn och den skall säga vad den ser för något.

Istället för att ge datorn regler att följa så låter man den i analogi med en människa, träna upp sin förmåga att lösa problemet. Likt en människa som ofta behöver många timmar för att lära sig en ny färdighet, så behöver en dator många träningsexempel för att bli bra på att utföra en syssla.

I exemplet med datorseende så behöver datorn antagligen se många bilder på katter och hundar för att kunna särskilja dem åt och säga “katt” ifall den ser en bild på en katt.

I princip så baseras sig alla maskininlärningsmetoder sig på mer eller mindre avancerade statistiska metoder och kan ha komplicerade namn. Jag tänkte dock presentera en metod som ofta används för att lösa många typer av problem och som har en naturlig analogi, den brukar kallas neuronnät på svenska.

1.3 Vad är ett neuronnät?

Den mänskliga hjärnan består av miljarder av nervceller, eller neuroner med ett annat namn, sammansatta i komplexa nätverk. Den kontrollerar och koordinerar kroppsfunktioner så som hjärtat, blodtryck och vätskebalans, samt mentala funktioner som känslor, minne och inlärning.

Artificiella neuronnät är en maskininlärningsmetod som försöker efterlikna hur den mänskliga hjärnan fungerar, om än i en förenklad modell.

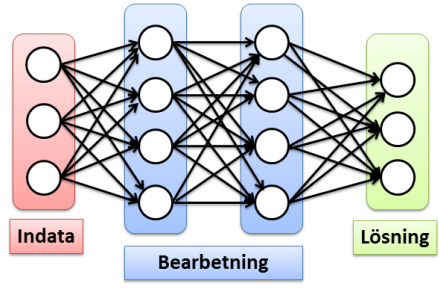
Ett artificiellt neuronnät består av flera lager, ett

Figur 1: Kasparov spelar schack mot datorn Deep blue.

Figur 2: En bild på en katt.

första lager med invärden från problemet som skall lösas, följt av ett eller flera bearbetningslager och sist ett lager bestående av lösning till problemet.

Ifall vi återgår till datorseende-exemplet igen, så består det första lagret av pixlar från en bild, följt av flera lager av neuroner, som behandlar bilden



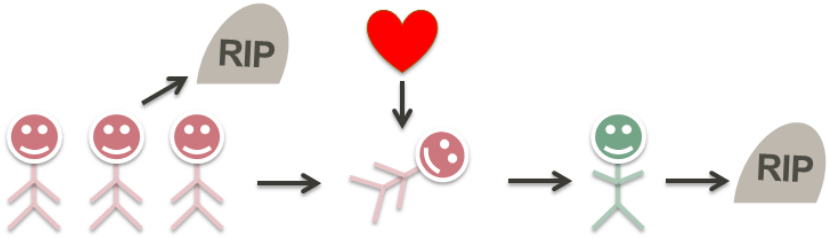
Figur 3: Hur ett neuronnät är uppbyggt.

i allt mer avancerade filter och sist ett lager med sannolikheterna för olika typer av objekt, till exempel katt eller hund.

Ett nätverk som har två eller flera bearbetningslager brukar betecknas som djupinlärning.

2 Hjärttransplantation

Hjärtsvikt är ett tillstånd då hjärtats pumpfunktion inte är tillräcklig för att ge tillräcklig blodförsörjning till kroppens organ. Vid svår



Figur 4: Skiss av hur en väntelista för hjärttransplantation ser ut. Ifall en patient är tillräckligt sjuk så placeras den i kön, personen dör antingen i väntan på ett hjärta, eller så genomgår den en transplantation. Ifall patienten får ett nytt hjärta så lever den 15 år i genomsnitt efter operationen.

hjärtsvikt så är hjärttransplantation en livräddande operation för patienten.

Det utförs ungefär 5000 transplantationer per år i hela världen, en majoritet av dem sker i USA och ungefär ett 60-tal i Sverige. Hjärttransplantation anses vara ett ingrepp som förbättrar livskvalitén för patienter med hjärtsvikt och de lever i genomsnitt 15 år efter operation.

När en läkare bedömer att man har så pass svåra hjärtproblem att en transplantation krävs, så ställs man på en väntelista. Man kan lämna listan på grund av att man blir transplanterad, dör i väntan på organ, eller av andra orsaker, så som att man blir för sjuk för operation.

I USA så står ungefär 4000 patienter på väntelistan för ett nytt hjärta och många dör medan de väntar på en lämplig donator. Det råder en brist på donatorer världen över. Detta gör att man måste prioritera patienter enligt vissa bestämda regler. De här reglerna varierar beroende på ens geografiska plats och även över tid, då de ibland uppdateras.

Tilldelningen av organ behöver balanseras mellan rättvisa, att ge alla en chans till transplantation, med nytta, att utnyttja ett organ till största utsträckning.

3 Användning av maskininlärning för hjärtpatienter

Det finns flera databaser där man sparar information om både mottagare, donatorer och genomförda operationer. Det finns bland annat en stor international databas, en amerikansk och en för Skandinavien. De kan innehålla information så som ålder, vikt, blodgrupp, med mera om både mottagare och donatorer.

För att kunna utnyttja patientdatan så måste den representeras på ett sätt som datorn kan använda den. Vi skapade ett enhetligt gränssnitt mot flera av databaserna så att datorn kan ställa frågor till dem på ett enkelt sätt.

Vi har skapat neuronnät för att kunna förutsäga vad som händer med patienter när de ställs i väntelistan, hur länge de skulle överleva utan operation, samt hur länge de överlever efter operation. Vi har även använt de här modellerna för att se vilka faktorer som påverkar utfallet mest.

Hur länge de överlever efter transplantation bedöms både på data från patient och donator, vilket gör att vi kan simulera potentiella ihopparningar och förutsäga förväntad livslängd på patienten.

Detta kan användas som verktyg av en läkare vid dennes beslutfattningsprocess och kan då till exempel få en rankad lista av möjliga överlevnandstider för personerna på väntlistan för varje donator som inkommer.

Sådana modeller kan även användas för att simulera utfallet för hela väntelistor över längre tider. Vilket gör det möjligt att utvärdera hur pass bra reglerna för tilldelning av hjärtan fungerar.

4 Sammanfattning

Datorer uppvisar en snäv form av intelligens. De klarar av att lösa specifika väldefinierade problem. Ett sätt som datorer kan lära sig att lösa problem, är genom att efterlikna hur den mänskliga hjärnan fungerar, detta kallas neuronnät. Med hjälp av neuronnät så kan man förutsäga överlevnadstiden för patienter både före och efter transplantation. Detta kan användas som hjälp för läkare vid beslut, eller så kan det användas för simulera transplantationsköer och på så sätt bedöma hur pass bra fördelningsregler av hjärtan fungerar.

Introduction

1 Introduction

The human brain can be seen as an ensemble of interconnected neurons, more or less specialized to solve different tasks (Fingelkurts & Fingelkurts, 2004; Fingelkurts, Fingelkurts, & Kähkönen, 2005). A large portion of the brain is dedicated to maintain homeostasis, that is, keeping us alive (Purves, 2011). This part of the brain works on a subconscious level, and affect bodily functions such as breathing, maintaining a constant body temperature, or sustaining circulatory support to organs.

The brain is also responsible for the processing of sensory information (Purves, 2011). This information is produced by stimuli to the different senses that humans have. What constitutes a sense is a somewhat debatable subject. There is no firm agreement as to the number of senses, because of differing definitions.

The five traditional senses are: sight, hearing, taste, smell, and touch. Depending on your definition of a sense, examples of other senses could be balance, temperature, and pain.

The information gathered from the senses are used as input to other parts of the brain, that perform analytical thinking and executive functions. These parts make the decisions, needed to achieve the goals of an autonomous agent, which most humans could be categorized as.

The goals are usually achieved through manipulation of the environment. Humans mainly use the control of their muscles, to be able to change the world around us.

The control loop for any autonomous agent, may it be a human, some other kind of animal, or a robot is similar. It uses its senses to be aware of the surrounding environment, then it analyzes the input, makes a decision based on the information and its current state, and finally uses some kind of actuators to perform an action.

Machine Learning (ML) has the ambition to recreate such actions in the form of processing pipelines. Some kind of data is used as input to the system, this data may need some preprocessing to be useful. The information is then used in a

ML model, which may be a artificial neural network (ANN), or some other kind of model. The selected algorithm analyzes the data and then produces an output based on the available information.

This output can be used to make decisions in automatic system, for example, in a spam filter, where it may tag a mail as unsolicited advertisement and move it to a junk folder. The output may also be used as input to a human, augmenting their decision capacity. For example, a system that can diagnose medical diseases may help a doctor in the treatment of a patient.

The term *deep learning* is often applied to signify deep ANNs, where deep means that it have several computational layers (LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). See Section 5.2 for more information about the ANN model. The development of deep learning is essentially a quest to mimic how the human brain, at least partially, operates.

Deep learning has been successful in solving problems in several different fields, such as computer vision, natural language processing (NLP), bioinformatics, and drug design (Chicco, Sadowski, & Baldi, 2014; Krizhevsky, Sutskever, & Hinton, 2012; Schmidhuber, 2015; Sutskever, Vinyals, & Le, 2014; Wallach, Dzamba, & Heifets, 2015). A big reason for the resurgence of neural networks, as of late, is the increase in computational power and availability of useful data, together with improvements of algorithms themselves.

We have used deep learning techniques to tackle several problems related to heart transplant and this will be described in this thesis. See the following outline for a description of the different chapters in the thesis.

1.1 Outline of Thesis

In this thesis, I will explain the basic elements of a ML pipeline, the terminology, how to represent data, how to evaluate results from a model, and different algorithms that can be used to create models.

I will first describe a problem within NLP and how we solved it.

I will then describe the heart transplant process and the application of machine learning to solve different problems that arise within this domain. This corresponds to the main part of the work.

The outline of this thesis is as follows:

- In Chapter 2, *Machine Learning*, I introduce the concepts of machine learning and its vocabulary.
- Chapter 3, *Representing Data*, describes the different types of data usable for machine learning and how to represent them in a way that a computer may use them.
- Chapter 4, *Evaluation*, explains several different evaluation metrics, because there is a need to objectively compare different models with each other.

- Chapter 5, *Algorithms*, describes the machine learning models: logistic regression and artificial neural networks (ANN). Deep learning often use deep ANNs as the model.
- Chapter 6, *Heart Transplant*, describes briefly the heart transplant process, to serve as background, to the machine learning applications pertaining to heart transplant.
- Chapter 7, *Simulation*, explains important concepts associated with the simulation of a transplant queue. Simulation of the queue is needed to evaluate the utility for the system as a whole.
- Chapter 8, *Application for Natural Language Processing*, describes an application using machine learning together with images and their associated texts, to improve classification of relations found in pictures.
- Chapter 9, *Applications for Heart Transplant*, explains the motivation for using machine learning together with heart transplant data, followed by five different tasks using this data, that I conducted for this thesis.

2 Machine Learning

2.1 Definition

Machine learning explores the construction and study of algorithms that can learn from data and make predictions. It is closely related to the field of statistics, albeit the cultures and backgrounds are different between them. It also has strong ties to mathematical optimization, which is used in the construction of such algorithms.

Machine learning is often used for applications, where an explicit algorithm is infeasible to program, for example in computer vision, where it would be hard to program concrete rules for object recognition.

2.2 Concepts

In this section, we introduce the most important concepts used in machine learning.

Observation. An observation, or a data point, is an example, used either for training or evaluation. It can for example be a Wikipedia page or a heart transplant patient.

Features. An observation has one or many variables associated with it, for example a heart transplantation patient may have features such as: age, blood group, and gender. These variables can be represented as numeric features in a vector. Variables can be of any data type, but most machine learning models require the features to be numeric. The features are either discrete, which are represented by integers, with the special case of binary numbers, or continuous, which are represented by real numbers. See Section 3 for more information about features.

Label. An observation can have one special variable called the label, which is the value that you want to predict, using the features. This label can be either a real number or an integer, depending on what variable you chose as the label. An example of a real valued label is the survival time after transplant, and of a discrete label is if the patient is alive after one-year.

Supervision. In supervised machine learning, see Section 2.4, we want to predict the label of an observation. This label can be either real-valued, which is the case with regression, or it can be discrete, as in the case of classification.

Model. A model is a function that takes the feature vector as the input and uses an algorithm that produce an output, which is the predicted label for that observation. The model can for example use logistic regression, random forests, or neural networks to realize this function. See Section 5 for more information.

Performance measure. To be able to objectively assess a model's performance, we need a metric that is comparable between models, otherwise it would be hard to optimize our model. For examples of different kinds of performance measures, see Section 4.

2.3 Unsupervised Machine Learning

Unsupervised learning creates models from observations that are unlabeled, trying to find hidden structures in this data. Since the examples given to the learner are unlabeled, there is no error or reward signal in order to evaluate a potential solution. This can be a goal in itself or used as a preprocessing step for a supervised algorithm. The two main applications of unsupervised learning are dimensionality reduction and clustering. We have not used unsupervised learning in any of the articles, but have done some experiments with it, for example visualization of the features, see Figure 1.

Dimensionality reduction. The task to transform the data in a higher dimensional space to a space of fewer dimensions, is called dimensionality reduction. This enables us to visualize the data in 2D. See Figure 1 for an example.

Dimensionality reduction can also be used as a preprocessing step for a supervised algorithm. Examples of algorithms that can do this include: *principal component analysis* (PCA), *t-distributed stochastic neighbor embedding* (t-SNE), or *linear discriminant analysis* (LDA) (Izenman, 2013; Maaten & Hinton, 2008; Wold, Esbensen, & Geladi, 1987).

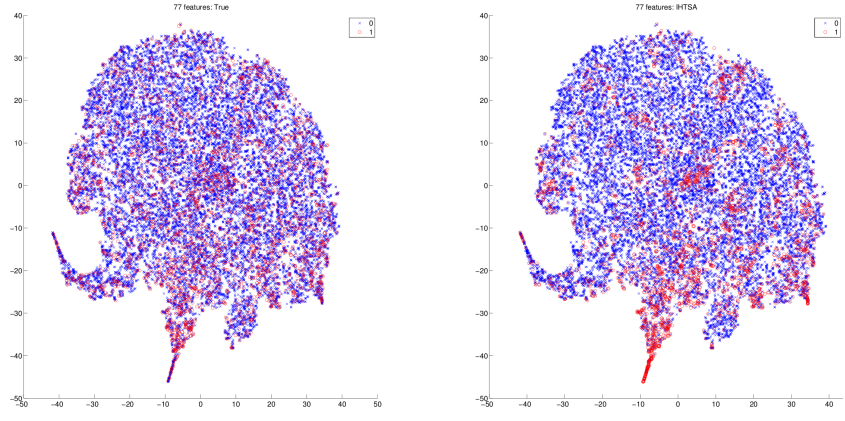


Figure 1: 77 dimensions reduced to 2 using t-Distributed Stochastic Neighbor Embedding algorithm (Maaten & Hinton, 2008). Blue points are transplanted patients that are alive after one year, red corresponds to being dead. The true version is on the left, predicted on the right.

Clustering. This is the task to group observations in clusters. The observations in a cluster are more "similar" to each other than to those in other clusters. Examples include the clustering of different patient types, or Wikipedia categories. Depending on the algorithm, we can specify the number of clusters we want to find, or let the algorithm decide on the cluster size on its own. Some of the usual methods include: K-means clustering, DBSCAN, or affinity propagation(Ester, Kriegel, Sander, Xu, et al., 1996; Forgy, 1965; Frey & Dueck, 2007). See Figure 2 for an example of a clustering.

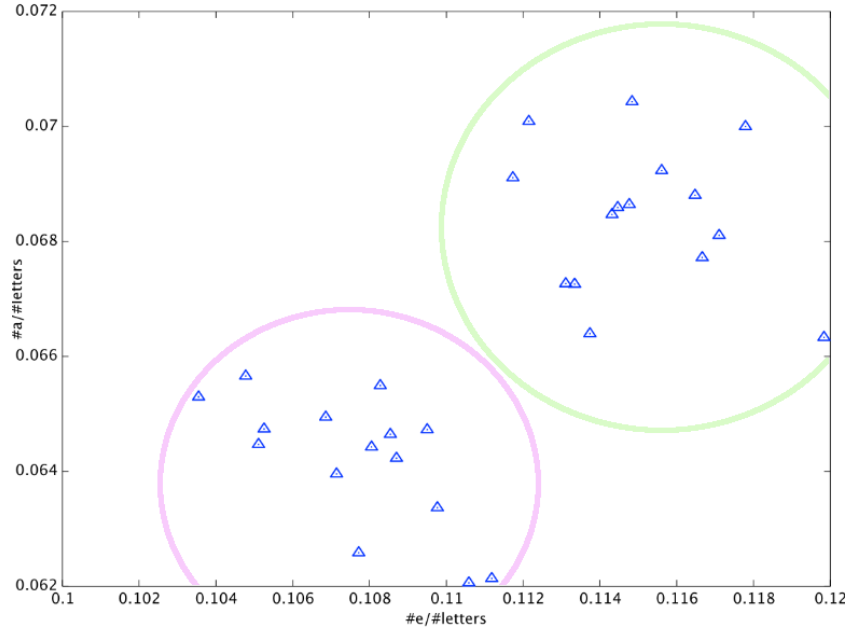


Figure 2: Clustering of Salammbô chapters, using relative frequency between the letter a/e. The circle to the upper right represent English, and the lower represent French.

2.4 Supervised Machine Learning

Supervised methods learn models from observations that are labeled, that is to infer a function from the labeled training data to the desired output. This should be able to generalize to unseen observations. We have used supervised machine learning in all articles but Paper II.

There are two main types of supervised learning algorithms: classification, in which the labels are discrete, that is $l \in \mathbb{Z}$, or regression, where the label is continuous, that is $l \in \mathbb{R}$. Depending on which we want to perform, there exist different algorithms, although some can do both tasks, and there is a difference in

which performance metric we use to evaluate the models. See Section 5 for more information on algorithms.

Regression. Regression is creating a model from the features in order to predict a continuous label for each observation, that is, a real number. Examples of values that could be predicted include: the length of patient, the blood bilirubin value, or the survival time after heart transplantation operation. The algorithms that can be used to create a regression model are for example linear regression, ordinary least squares regression or neural networks (Dismuke & Lindrooth, 2006; Rosenblatt, 1958; Yan & Su, 2009). The *international heart transplant survival algorithm* (IHTSA) model, described in Paper V, utilizes a neural network (Nilsson et al., 2015). The model can be used either for regression, predicting the median survival time of the patients after a heart transplant, or classification, predicting the probability of mortality at certain time points.

Classification. In classification, we construct a model that predicts a discrete label. The class of the observation is represented as an integer. In the special case of binary classification, the label could be represented as a Boolean value, that is either true or false. Gender, blood group, or status in the queue for a patient are examples of values that could be predicted by a classification model. Examples of algorithms that can be used for classification purposes include logistic regression, random forest, or neural networks (Cox, 1958; Liaw & Wiener, 2002; Rosenblatt, 1958). We have performed classification in all of the articles that use machine learning. For example, in Paper I, we classified images of humans and horses as having one of the following relations: riding, leading, or none.

2.5 Hyperparameters

Almost all machine learning algorithms have options or parameters that we can tune, such as the solver or optimizer being used to find the weights of the model. An example of a parameter for logistic regression is the cost function, that is, the evaluation function that we try to minimize, using the solver. Often, however, many of the default values produce decent results, and they can be quite different from the optimal values.

One way to optimize these parameters is to use a grid search, where we specify sets for the values of the hyperparameters that we want to optimize. We then create the Cartesian product of these sets and try each tuple of the parameters. We use each tuple to train and evaluate a model and to choose the model that produces the best performance metric, see Section 4. Cross validation is often used to minimize the overfit of the models, see Section 2.6. The cardinality of the Cartesian product is described in Equation 1. Hence the number of models that needs to be tried increases quite fast with the number of parameters and values. Grid search suffers from the curse of dimensionality, but is often embarrassingly

parallel because we can train and evaluate the models using the hyperparameter settings independently of each other.

For example, if we have two sets of parameter values A and B, we can create the Cartesian product of these sets.

$$\begin{array}{lcl} A & = & \begin{Bmatrix} 1 & 2 & 5 \end{Bmatrix} \\ B & = & \begin{Bmatrix} 0.1 & 0.5 & 1 \end{Bmatrix} \\ A \times B & = & \begin{Bmatrix} (1,0.1) & (1,0.5) & (1,1) \\ (2,0.1) & (2,0.5) & (2,1) \\ (5,0.1) & (5,0.5) & (5,1) \end{Bmatrix} \end{array}$$

$$|S_0 \times \ldots \times S_k| = \prod_{i=0}^{k} |S_i| \tag{1}$$

A more simple approach optimizes one parameter at a time; this requires a lot less evaluation of models, see Equation 2. This often produces a reasonable result, if the parameter interaction is not too large on the model. The order in which the parameters are optimized can influence the result, and usually, some knowledge of the algorithm that is being used, is helpful. Some backtracking may also be required, that is, to go back and reoptimize a parameter.

$$\sum_{i=0}^{k} |S_i| \tag{2}$$

2.6 Overfitting

Using a model that is complex enough, it is possible to get a function that maps the features on the training data to the exact labels. This model will only remember the training data and will not generalize well to unseen examples, which means that the model is overfit with respect to the data. The model will only describe noise, instead of the underlying relationship between the features and the labels.

Overfitting is a problem with almost all machine learning algorithms (Hawkins, 2004). To be able to estimate the possible overfit, two main methods are used, either divide the data set into training, validation and test sets or use cross validation, see the following sections for more information about these techniques. There exist different ways to minimize the overfit, which are often specific to the algorithm of choice, for example regularization for logistic regression and drop-out for neural networks.

Dividing the data set. The data set is often divided into a training, validation and test set. Distribution of the data is arbitrary, but it is preferable that the train

set has the majority of the data, for example dividing the data in a 70%/15%/15% manner, as we did in Paper V. Often a shuffle of the data before dividing it into sets is required, because there could be some systematic pattern in order of the data, for example that the data is ordered after collection date. If we are using cross validation, then we can skip the validation set and only have training and test sets.

We use the training set to train the model, and then the validation set to evaluate the model using some metric. Utilizing this metric, we fine tune the features and hyperparameters to be optimized on the validation set. This will reduce some of the overfit that otherwise will happen if we evaluate on the same data as we train on. The test set is then used to evaluate how good the model will generalize to unseen data.

Cross validation. Instead of using a validation set, one can use cross validation to estimate the model's fit (Kohavi et al., 1995). We divide the data set into k equally sized partitions, for example $k = 5$, which are called folds. We train the model on $k - 1$ parts of the data, and evaluate it on the remaining part using some metric. We iterate this k times, selecting another part as the evaluation data each time, see Figure 3 for an illustration how we divide the data each fold. After the metric for each fold is calculated, we take an average of the evaluations to produce an estimate of the predictive power of the model. This estimate should have a lower bias than training and evaluating on the same data set.

We utilize cross validation for example in Paper III, where we wanted to train and evaluate many models. Here we tried to avoid the possible overfit that can arise from fitting many parameters to data.

Figure 3: Illustration of 5-fold cross validation, the grey squares are the training set and the red square represent the evaluation set.

2.7 Spark framework

Apache Spark is a cluster-computing framework, which provides an API to easily do parallel computing on several computers (Foundation, 2015). It was developed in response to limitations in the MapReduce cluster computing paradigm, which requires a certain linear flow of data when designing a program. One of the reasons why its creators wanted to invent Spark, was to facilitate the use of iterative algorithms, something that is often found in machine learning applications.

We used the Spark framework in Paper III to train and evaluate several thousand models, because the models do not have any dependencies between them, the process is therefore relatively easy to parallelize.

There are several libraries which provide additional functionality beside the Spark core project. One of these libraries is MLlib, which implements several types of machine learning algorithms and tools (Meng et al., 2016).

3 Representing Data

3.1 Basic Types

For an observation, there exist different kind of variables. Below is a list of some of the most usual types:

- Numeric
  - Integer, e.g. age or operation year
  - Real, e.g. blood creatinine value or weight
- Binary, e.g. gender or diabetes
- Categorical, e.g. country of origin or type of ventricular assist device
  - Ordinal, e.g. urgency status or functional status
- Text, e.g. body of an email or a Wikipedia article

Ordinal variables. For ordinal variables, there exists an ordering of the categories, e.g. high > low, but usually no measure of closeness. We can represent the variable as a single numerical feature, for example a variable describing how acutely a patient needs a new heart, with three values: low = 1, medium = 2, and high = 3. If we do this, then we introduce a degree of closeness, which may or may not be desirable. If this is not feasible use one-hot encoding, see Section 3.2.

Categorical variables. For categorical variables, there is no intrinsic ordering or measure of closeness. When using a variable that contains the country of origin of a patient, we can not say that Germany > France. We could represent the variable as a single numerical feature, for example if we only have three countries: Germany = 0, France = 1, Belgium = 2. Then we introduce both an ordering and a degree of closeness, which usually is not desirable. It is usually better to use one-hot encoding, see Section 3.2.

3.2 Representing Categorical Data

One-hot encoding, also called dummy encoding or contrast coding, is a way to represent categorical variables as features (Hardy, 1993). This is done by creating a vector of binary features, where one of them is equal to 1 and the rest is 0, where the index in the vector represents the category value. One-hot encoding does not introduce spurious relationships between the categories, which a single numeric feature could do.

An example of a categorical variable could be the country of origin of a patient. Assume that there are only three countries represented in the group of patients, then we could represent the feature as the following:

Germany = $\begin{bmatrix}1 & 0 & 0\end{bmatrix}$
France = $\begin{bmatrix}0 & 1 & 0\end{bmatrix}$
Belgium = $\begin{bmatrix}0 & 0 & 1\end{bmatrix}$

If there are more values that the variable can assume, we use a longer feature vector. This vector should have the same length as the cardinality of the set of the variable values, which is equal to the number of unique values that the variable can have. For example the variable that describes the functional status of a patient, which is between a range of 1 to 10, will need a vector of length 10 to one-hot encode these features.

3.3 Preprocessing Data

The numerical range of the features has a significant influence on the results. To make them more homogeneous, the idea is to scale, standardize, or normalize the features, so they become more homogeneous. This is a kind of feature transformation, which may improve the prediction result and speed up the training of the model.

The following techniques can be applied on the features independently of each other, but usually either rescaling or standardizing is exclusively chosen. For example in Paper III, we first utilized rescaling followed by normalizing. This made significant difference in both training time, because the model converged much faster, and in the classification result.

This feature vector will be used in the following examples:

$\mathbf{v} = \begin{bmatrix}10 & 15 & 20 & 50 & 70\end{bmatrix}$

Rescaling. Scaling the feature to the interval [0,1], can be written as Equation 3, where min and max are the minimum and maximum values for that feature.

$$x' = \frac{x - \min(x)}{\max x - \min x} \tag{3}$$

Using the vector $\mathbf{v}$: $\min(v) = 10 \quad \max(v) = 70 \quad \max(v) - \min(v) = 60$

$\mathbf{v}' \approx \begin{bmatrix}0 & 0.08 & 0.17 & 0.67 & 1\end{bmatrix}$

Standardizing. Transforming a feature to have zero-mean and unit-variance, see Equation 4, where $\bar{x}$ is the mean of the feature values and σ is the standard deviation.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{4}$$

Using the vector $\mathbf{v}$: $\bar{x} = 33 \quad \sigma \approx 26$

$\mathbf{v}' \approx \begin{bmatrix} -0.88 & -0.69 & -0.5 & 0.65 & 1.4 \end{bmatrix}$

Normalizing. Converting the feature vector to have a unit length in the norm (usually the Euclidean norm, i.e. p = 2), see Equation 5, where $\|\mathbf{v}\|$ is the norm of $\mathbf{v}$.

$$\|\mathbf{v}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p} \qquad \mathbf{v}' = \frac{\mathbf{v}}{\|\mathbf{v}\|_p} \qquad \|\mathbf{v}'\|_p = 1 \tag{5}$$

Using the vector $\mathbf{v}$: $\|\mathbf{v}\|_2 = \sqrt{10^2 + 15^2 + 20^2 + 50^2 + 70^2} \approx 90$

$\mathbf{v}' \approx \begin{bmatrix} 0.11 & 0.17 & 0.22 & 0.55 & 0.78 \end{bmatrix}$

3.4 Representing Knowledge in Natural Language Processing

When dealing with natural language processing, the data is often in a text format. One way of representing the text is using a bag-of-word technique. We have used the following concepts in Paper I and in "Using Syntactic Dependencies to Solve Coreference"(Stamborg, Medved, Exner, & Nugues, 2012), the latter is not part of this thesis.

Bag-of-Words. The bag-of-words representation is a way to code textual data, and it is similar to one-hot encoding (Salton & McGill, 1986). First we create a dictionary of the words, where the index represents a specific word. Instead of binary values, stored at the indices, we save a number corresponding to the frequency of the word in that observation; this is called the *term frequency* (TF).

This usually results in a quite sparse feature vector. The number of unique words in the English language is probably over 1 million, depending on your definition, albeit it is unlikely that a corpus would include most of them. The length of the feature vector is usually considerably smaller than that.

The bag-of-word representation of text, does not preserve the structure of the sentences, hence the bag metaphor.

An example of a bag-of-words encoding:

Texts:
Text 1: John likes to watch movies. Mary likes movies too.
Text 2: John also likes to watch football games.

Dictionary of indices:
John = 0, likes = 1, to = 2, watch = 3, movies = 4, also = 5, football = 6, games = 7, Mary = 8, too = 9

Encoding:
Text 1: $\begin{bmatrix}1 & 2 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1\end{bmatrix}$
Text 2: $\begin{bmatrix}1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0\end{bmatrix}$

TF-IDF. The term frequency is used to measure the importance of the words. It is easy to over-emphasize terms that appear very often, for example *a*, *the* and *of*. If a term appears very often across the corpus, it means that it probably does not carry any special information. The inverse document frequency (IDF) is a numerical measure of how much information a term provides. IDF is defined as the inverse of the number of documents that contain the term, see Equation 6. The TF multiplied by the IDF is often called TF-IDF (Salton & McGill, 1986).

$$IDF(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad , \tag{6}$$

where N is the total number of documents in the corpus: $N = |D|$ and where $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears, that is, the TF is not zero. If a term does not exist in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Continuing on the example in the previous section:

$\log \frac{2}{1} = \log 2 \approx 0.7 \qquad \frac{2}{2} = \log 1 = 0$

IDF = $\begin{bmatrix}0 & 0 & 0 & 0 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7\end{bmatrix}$

TF$\times$IDF$_{Text1}$ =
$\begin{bmatrix}1 & 2 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1\end{bmatrix} *$
$\begin{bmatrix}0 & 0 & 0 & 0 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7\end{bmatrix} =$
$\begin{bmatrix}0 & 0 & 0 & 0 & 1.4 & 0 & 0 & 0 & 0.7 & 0.7\end{bmatrix}$

TF$\times$IDF$_{Text2}$ =
$\begin{bmatrix}1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0\end{bmatrix} *$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0.7 & 0.7 & 0 & 0 \end{bmatrix} =$$

Thus, the words that occurred in both of the texts were weighed down considerably more than the words that only were included in one text.

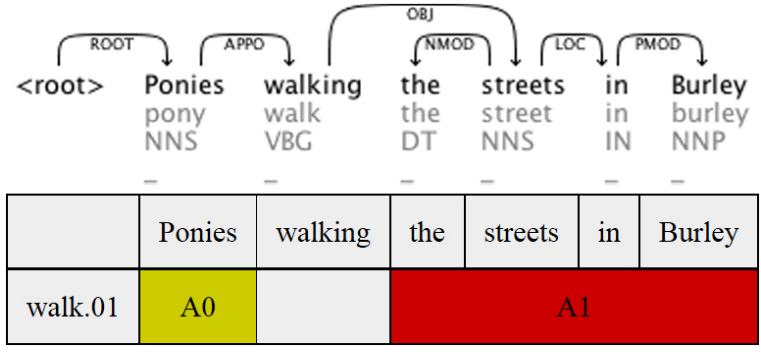


Figure 4: A parsed sentence, showing the predicate and its arguments.

Predicates. Using only the bag-of-word technique does not capture the structure of the sentences. The predicates correspond to actions or relations such as jump, walk or own. They are a way to encode the semantics of a sentence. Each predicate can have one or more senses, that is, specific uses of that predicate. For example, walk.01 is used in the meaning: "be a pedestrian, forward motion, one foot in front of the other", while walk.03 has the definition: "achieve a result through walking". Each sense will correspond to a distinct predicate-argument structure. A predicate has one or more arguments, roughly corresponding to the subject and objects of a verb.

The PropBank nomenclature is often used, where the predicate sense is explicitly shown as a number added after the word (Palmer, Gildea, & Kingsbury, 2005). The sentence in Fig. 3 contains one predicate: walk.01 with its two arguments A0 and A1, where A0 corresponds to the *walker* and A1, to the *path walked*. The PropBank predicates can also have modifying arguments denoted with the prefix AM-. There exist 14 different types of modifiers in PropBank.

Information regarding which predicates that are used and their arguments can be used as features to a machine learning model, for example creating bag-of-word vectors using only these words.

Coreference. If two words or phrases refer to the same entity, then they are coreferent (Hobbs, 1979). For example, in the sentences:

> Anders likes Pokemon. He has got many of them. Birger also likes them.

Anders and *he* refer to the same entity, and *Anders* is the first mention of a new entity in the coreference chain. There exists another chain in the example *Pokemon* and two *them*, which are also coreferent.

The coreference information, together with the part of speech, can be used to substitute words in the documents that are coreferent (Stamborg et al., 2012). The first mention in a coreference chain, that is the first word or phrase linked to a certain entity in the document, usually contains the most information. The reason behind this is that an entity is usually explicitly mentioned first and then implicitly referenced afterwards. Words that corefer can be substituted with the first mention in the chain, although this is mostly useful with pronouns.

3.5 Dealing with Biomedical Data

Working with biomedical data entails working with patient data, which usually is collected in some de-identified form in a database. In such databases, there is usually missing information in the observations, both in the features, and in the labels, the latter is called *censoring*. Missing information in the features can in some sense be remedied by utilizing imputation, see paragraph below.

We have mostly been working with registries containing patients that donate and receive hearts for transplantation. An example of such a database is the one from the *United Network for Organ Sharing* (UNOS), which administers the only organ procurement and transplantation network in the United States of America (UNOS, 2018). The database contains data from October 1, 1987 and onwards. In the database, there is information that encompass recipient, donor and transplant data. It includes almost 500 variables reflecting different attributes of the patients.

Censoring. The term censoring is used in clinical trials to refer to a patient which is lost to follow-up before reaching his/her endpoint. This means that the patient is removed from the study for some reason before the event that is being studied happens (Lagakos, 1979).

Censoring means that only partial information is known of that patient's label. For example if the patient is censored after one-year in a survival study, because he or she moved to a different location, we only know that he or she survived a year. We do not know how long he or she may have additionally survived. This creates some uncertainty in the data, effectively reducing the sample size for the survival curve towards the end.

Hence, censoring is something we have to consider when working with patient data.

Imputation. With real-world data, it is likely that we will encounter some missing feature values (Kenward, 2013). There may be missing values, because in the real world, for various reasons, complete information is not always available.

Some variables may not have been recorded during certain time periods. For example in the UNOS database: the pulmonary artery pressure of the recipient, were not recorded before 1994. It could also be the case that the feature is not applicable to that data point, for example the number of previous pregnancies for a male patient. The variable may also be missing because the doctor or nurse forgot to record it.

There exist a few ways of handling these missing values, the following paragraphs deal with the problem:

Casewise deletion: We delete the observations that have missing values. This effectively reduces the size of the data set. If there are many missing values, we need to delete a considerable amount of the data. If the missing values are random, no bias is introduced in the model. However missing values rarely tend to be completely random.

Mean/mode imputation: We replace the missing value with the mean for real-valued features, and mode for categorical. This has the property that the sample mean for that variable is unchanged. This can severely distort the distribution for this variable, by pulling estimates of the correlation towards zero.

Hot-deck imputation: We replace the missing value from a uniform distribution of the non-missing values. This has the property that the distribution for that variable is unchanged.

Using a model: We replace the missing values based on a machine learning model created using the other variables. Creating a good model for imputation is not trivial, and this may take considerably more effort than the other techniques, depending on what model is chosen. This works well, if the variable is correlated with the other variables.

3.6 Feature Selection

Some models are quite sensitive to the selected feature set, for example, logistic regression, see Section 5.1. The predictive power, using some metric, is heavily dependent on the features that are included in these models. This means that feature engineering is an important part when creating such a model.

Some features are essentially noise compared to the label, which means that they do not contribute to predictive capabilities of the model. Including such features may just confuse the model, because it is trying to optimize weights to an essentially random feature compared to the label, leading to a lowering of the result.

Features may also have interaction with each other, that is, certain combinations of features may improve or worsen the result.

Because a logistic regression model uses a linear combination of the features, it has problems representing non-linear relations between the features and the label. It is possible to capture some of this non-linearity by combining features, using some kind of mathematical function between two or more features. Common functions include Boolean functions such AND or XOR, or polynomial combination of features, see Equation 7, where a_k are constants and x_i are the features.

$$P(\mathbf{x}) = \sum_{k=0}^{n} a_k x_i^k \tag{7}$$

Feature search. To find a globally optimal feature set requires evaluating all possible feature combinations, which require 2^n models where n is the number of features. This is infeasible even for a moderate number of features. For example, with a starting feature set of 100, this number is over 10^{30}.

It is possible to find a locally optimal subset, using considerably less model evaluations. We can for example use forward selection and/or backward elimination to find such a set. This is something we did in Paper III, utilizing the parallelism that the Spark framework together with a local computer cluster could provide.

Backward elimination starts with all the features and removes them one by one from the set. The resulting feature set is then used to produce the classification probabilities. We calculate the chosen performance metric for each of the new feature sets, and remove the feature that produced the best score when excluded. We repeat this process until the stopping criterion is reached, which can be either that the desired amount of features remains or that the result does not improve.

The number of tests for a complete backward elimination, that is every features are removed, is given by Equation 8. When using a feature set of 100, this method takes about 5000 evaluations.

$$B(n) = \sum_{i=0}^{n} (n - i) \tag{8}$$

Forward selection is analogous to backward elimination, but instead we start from the empty set and add the feature for each generation that improves the result the most.

3.7 Resource Description Framework

The *resource description framework* (RDF) is a standard model for data interchange on the Web (WWWC, 2014). It enables programmers to build graph databases, which consist of triples in the form of subject-predicate-object. The

subject and object denote resources, such as patients or feature values. The predicate represents traits or aspects of the resource, and expresses a relationship between the subject and the object. An example of triple could be: Patient1337 - Age - 42. We utilized this kind of database to store the data that we used in Paper III, IV and V.

These triples intrinsically represent a labeled, directed multi-graph, that can be queried. RDF is represented by triples, and it is therefore relatively easy to incorporate data from different sources.

RDF is an abstract model with several serialization formats, for example Turtle or XML. The particular encoding for resources or triples varies from format to format.

SPARQL. SPARQL, which is a recursive acronym for *SPARQL Protocol and RDF Query Language*, is the predominant query language for RDF stores (Prud'hommeaux & Seaborne, 2008). The language is somewhat similar to the *structured query language* (SQL). It shares some of the syntax, such as the keywords "SELECT", "FROM", and "WHERE". The subject-predicate-object structure is utilized to query the graph. See Listing 1 for an example of a simple SPARQL statement.

Listing 1 An example of a SPARQL query, which returns the average age of the patients over 17 years from the UNOS database.

```
SELECT (AVG (?age) AS ?avgAge)
FROM <file://UNOS.ttl>
WHERE{?patient aaot:age ?age .
FILTER (?age > 17)}
```

4 Evaluation

4.1 Confusion Matrix

A confusion matrix, also known as an error matrix, is a table layout that allows visualization of the performance of a classification model (Stehman, 1997). Given a binary classifier and a set of instances, a two-by-two confusion matrix can be constructed. This represents the dispositions of the set of instances, see Table 1.

For an actual instance, a binary classifier, with the predicted positive (p) and negative (n) values, has four possible outcomes:

- If the actual instance is positive and it is predicted as positive, it is a true positive (TP).
- if the actual instance is negative and the predicted outcome is positive, it is a false positive (FP).
- If the actual instance is positive and the predicted outcome is negative, it is a false negative (FN).
- If the actual instance is negative and the predicted outcome is negative, then it is a true negative (TN).

This matrix forms the basis for many common metrics (Fawcett, 2006).

Table 1: A confusion matrix.

		Prediction Outcome		
		p	**n**	**Total**
Actual Value	**p′**	**T**rue **P**ositive	**F**alse **N**egative	P′
	n′	**F**alse **P**ositive	**T**rue **N**egative	N′
	Total	P	N	

4.2 Recall

Recall, also known as sensitivity or *true positive rate* (TPR), is a measure of the ratio of the true observations that the model will classify as true, see Equation 9.

$$Recall = \frac{TP}{P'} = \frac{TP}{TP + FN} \tag{9}$$

Recall answer the question: "Given a positive example, will the classifier detect it?" The recall values go from 0.0, indicating zero true positives, to 1.0, indicating zero false negatives. A low recall indicates many false negatives.

4.3 Precision

Precision, also known as *positive predictive value* (PPV), is a measure of the ratio of the predicted true observations that are actually true, see Equation 10.

$$Precision = \frac{TP}{P} = \frac{TP}{TP + FP} \tag{10}$$

Precision answers the question: "Given a positive prediction from the classifier, how likely is it to be correct?" The precision values go from 0.0, indicating zero true positives, to 1.0, indicating zero false positives. A low precision can indicates a large number of false positives.

4.4 F1

There is often an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other, and therefore we want a measure that balances these two metrics. The F1 score was created to do this. It is defined as the harmonic mean of precision and recall, see Equation 11.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{11}$$

This metric is bounded in the interval 0.0 to 1.0 (Powers, 2007). This score tends to be close to the minimum of both the precision and recall. This means that we need both high recall and precision to get a high F1 score.

4.5 Multiclass Averaging

We want to use the same metrics when we evaluate models using more than two classes. These metrics were created for binary classes and one way to generalize them to more than two classes, is to average the results. This can be done using micro or macro averages.

The micro average method consists of summing up the individual true positives, false positives and false negatives of the system for the different classes, and then calculating the performance measure, see Equation 12.

$$B_{micro} = B(\sum_{i=1}^{q} tp_i, \sum_{i=1}^{q} tn_i, \sum_{i=1}^{q} fp_i, \sum_{i=1}^{q} fn_i) \quad , \tag{12}$$

where i is a label, B is the function for the performance measure, and $\{i_k : k = 1...q\}$ is the set of all labels.

The macro average calculates the average of the performance measure of the system on the different classes, see Equation 13.

$$B_{macro} = \frac{1}{q} \sum_{i=1}^{q} B(tp_i, tn_i, fp_i, fn_i) \tag{13}$$

When the examples are unevenly distributed across the classes, the macro average method is less biased toward the largest class (Van Asch, 2013).

4.6 Area Under the Receiving Operating Characteristic

A receiver operating characteristics (ROC) graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graphs are two-dimensional graphs in which the TPR is plotted on the Y axis and the *false positive rate* (FPR), see Equation 14, is plotted on the X axis.

$$FPR = \frac{FP}{N'} = \frac{FP}{FP + TN} \tag{14}$$

An ROC curve is a two-dimensional graphical visualization of a classifier's performance, see Figure 5. We want to describe the performance of a classifier using a single scalar value. One way to do is using the *area under the curve of the ROC* (AUROC). Because TPR and FPR both are bounded by the interval 0.0 to 1.0, the area is also bounded between zero and one (Fawcett, 2006).

A classifier that outputs a random label should have a AUROC value of 0.5, and therefore no serious classifier should have a lower value than that.

The AUROC has the statistical property that it is equal to the probability that a randomly chosen negative example is ranked lower then a randomly chosen positive example. See Equation 15, where X_1 is the score for a positive instance, predicted from the model, and X_0 is the score for a negative instance.

$$AUROC = P(X_1 > X_0) \tag{15}$$

This is also equivalent to the Wilcoxon test of ranks. It can further be shown that the AUROC is closely related to the Mann-Whitney U (Hanley & McNeil, 1982).

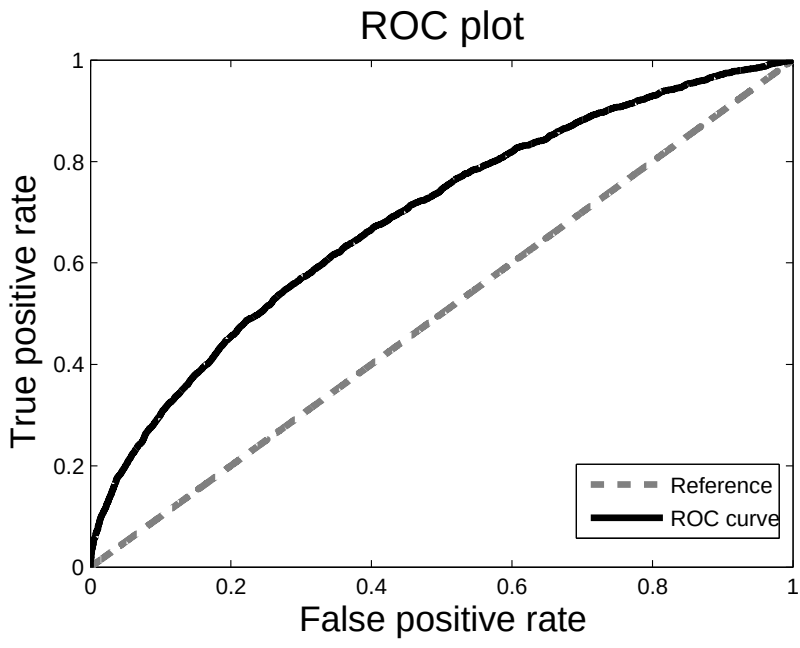


Figure 5: Illustration of a ROC plot. The reference line has an AUROC value of 0.5 and the ROC curve has an AUROC of about 0.7.

4.7 Concordance Index

The *index of concordance* (C-index) is used for validating the predictive ability of a survival model (Harrell, Califf, Pryor, Lee, & Rosati, 1982). It is a generalisation of the AUROC.

The C-index is the fraction of pairs in the data, where the observation with the higher survival time has the higher probability of survival predicted by a model. For a pair of real survival times (T_1, T_2) and $T_1 > T_2$ and the corresponding predicted probabilities for the patients is (P_1, P_2) and $P_1 > P_2$, then the fraction of the pairs where this is true is equal to the C-index.

This survival analysis can be seen as a ranking problem, which is an elegant way of dealing not only with the typically skewed distributions of survival times, but also with the censoring of the data (Steck, Krishnapuram, Dehing-oberije, Lambin, & Raykar, 2008). See Section 3.5 for more information about censoring.

The C-index is the probability of concordance between the predicted and the observed survival. As such it spans between zero and one, and similar to AUROC, a value 0.5 means a random distribution of the predicted values and 1.0 means perfect prediction.

5 Algorithms

5.1 Logistic regression

Logistic regression is a model that is used for binary classification, but it can be extended to do multiclass. The algorithm outputs a probability of the class, which can be useful. Examples of binary labels include: alive/dead, healthy/sick, or pass/fail.

Algorithm. Let t be a linear combination of the features x_i and a set of weights w_i, see Equation 16, where w_0 is the intercept term.

$$t = w_0 + w_1 x_1 + \ldots + w_i x_i \,, \tag{16}$$

where w_i are the regression coefficients, indicating the relative effect of a particular feature on the outcome.

The logistic function σ, is defined by Equation 17 and illustrated in Figure 6. This function is between 0 and 1 for every t. For positive infinity, it is equal to 1 and for negative it is 0. It is therefore interpretable as a probability.

$$\sigma(t) = \frac{1}{1 + e^{-t}} \tag{17}$$

If we substitute t in Equation 17 with Equation 16, we get Equation 18.

$$\sigma(t) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + \ldots + w_i x_i)}} \tag{18}$$

To go from a probability to a binary classification, we use a threshold k, which is often 0.5, see Equation 19.

$$L(t, k) = \begin{cases} 1, & \text{for } \sigma(t) \geq k \\ 0, & \text{for } \sigma(t) \geq k \end{cases} \tag{19}$$

Fitting the model to the data means finding the weights w_i that in some sense are optimal in predicting the observations. It is possible to solve this optimization problem using several different algorithms, using for example *limited-memory Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS) or *stochastic gradient descent* (SGD) (Bottou, 2010; Byrd, Lu, Nocedal, & Zhu, 1995). Often some kind of regularization is used to reduce the possible overfit, see next paragraph for more information.

Regularization. In order to minimize the possible overfit of a logistic regression model, regularization is often used. The basic idea is that an overly complex model often fits noise to the labels, thus will not generalize well to unseen data. To address this issue, a penalty term is added to the loss function, so that when we

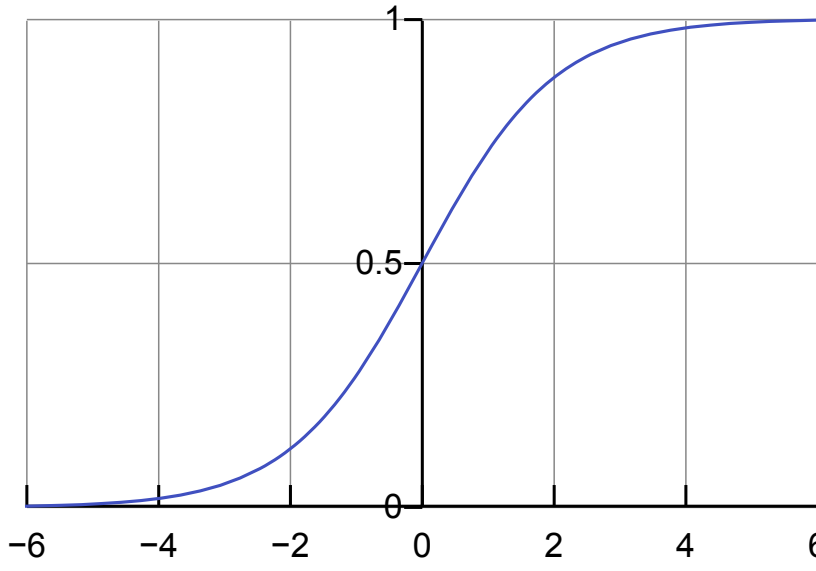


Figure 6: Graph of the logistic function, see Equation 17.

try to minimize the function we also try to minimize the model complexity, see Equation 20.

$$\min_f \sum L(y, f(x)) + \lambda R(f) \quad , \tag{20}$$

where L is a loss function that describes the cost of predicting $f(x)$ when the label is y. λ controls the importance of the regularization term, $R(f)$ which typically is a penalty on the complexity of f, usually a norm of model weights.

5.2 Artificial Neural Networks

Artificial neural networks (ANN) are loosely based on our understanding of how the human brain operates (McCulloch & Pitts, 1943). An ANN consists of a network of nodes, which are the equivalent of the biological neurons. They consist of inputs comparable to dendrites, a summation function similar to the soma, and an output corresponding to the axon.

An artificial neuron consists of a linear combination of the input and its weights on the different connections. This summation is the same as in Equation 16. The sum is then used as input to a non-linear function known as an activation function or transfer function, see Equation 21. If we choose the logistic function as the ac-

tivation function, see Equation 17, we get a function that is equivalent to logistic regression, see Equation 18.

$$y = \varphi\left(\sum_{j=0}^{m} w_i x_i\right) \tag{21}$$

A neural network with the sigmoid as its activation function can therefore be seen as a network of several logistic regression models, connected in parallel and series of each other.

A network consists of three or more layers. The first layer is called the *input layer*, where the features are used as the initial input. The middle layers which can be one or more, are called *hidden layers*. Finally, the last layer, the *output layer*, which has as many nodes as the wanted amount of outputs from the model. Figure 7 illustrates a network with three layers, in which the layers are fully connected.

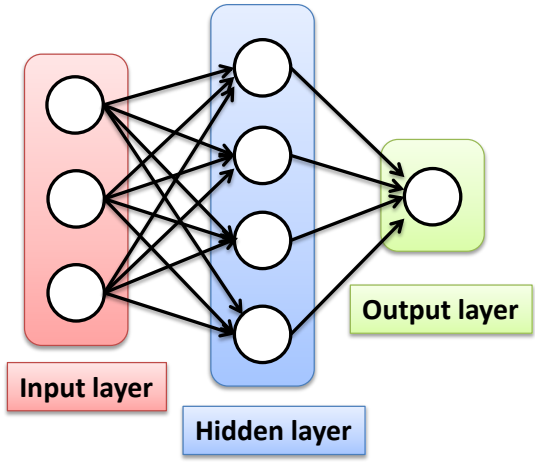


Figure 7: A fully connected neural network with three input nodes, four hidden nodes, and one output node.

A neural network is usually characterized by the following parameters:

- The number of hidden layers.
- The number of nodes in each layer.
- The interconnection pattern between the different layers of nodes.
- The weights of the interconnections, which are updated in the learning process.

- The activation functions, which convert a node's weighted input to its output.

A neural network using a reasonable non-linear function as its activation, such as the logistic, and one hidden layer, can be shown to approximate all continuous functions on compact subsets of $\mathbb{R}^n$ (Gybenko, 1989). An ANN is therefore in some sense a universal approximator.

Backpropagation. Backpropagation is a popular method to train neural networks (Werbos, 1974). It is used together with an optimization method such as gradient descent. The algorithm has two cycles, propagation and weight update. First, a feature vector serves as input to the network, then propagates through the network. The vector passes through the nodes in each layer from the input layer to the output layer; this produces an output from the network. The output is then compared to the label of the example, by using a loss function, and an error value is computed for each of the nodes in the output layer. The backpropagation algorithm utilizes these values to calculate the gradient of the loss function. The gradient is then used by the optimization method to update the weights. The weights are updated so that they try to minimize the loss function.

Dropout. Dropout is a regularization technique for reducing overfitting in neural networks (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). Since a fully connected layer uses most of the parameters, it is prone to overfitting. The idea behind dropout is to randomly drop units, together with their connections, from the neural network during training. The dropout rate controls the probability of a neuron being removed, and a normal value for the rate is 0.5, that is, half of the connections are dropped. By avoiding training all nodes on all of the training data, utilizing dropout usually decreases overfitting in neural networks.

Feature importance. Since neural networks do some feature selection automatically by weighing down the connections of irrelevant features, a feature search as described in Section 3.6 is often unnecessary. It can, however, marginally improve the result, and a reduced feature set can have some other positive impact on the model, for example that a smaller model may suffice to produce the same result or that the collection of the feature data is simplified.

5.3 Deep Learning

Deep learning is also known as deep structured learning or hierarchical learning. Most modern deep learning models are based on an ANN approach. A deep neural network is an ANN with two or more hidden layers between the input and output layers.

In deep learning, each layer level learns to transform its input data into a slightly more abstract representation. For example, in an image recognition application, the raw input may be a matrix of pixels. The first representational layer may abstract the pixels and encode edges, the second layer may compose and encode arrangements of edges, the third layer may encode a nose and eyes, and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn to recognize these feature abstractions on its own (LeCun et al., 2015).

As previously stated, a neural network using only one hidden layer, can be shown to approximate all continuous functions. Therefore only one layer, which may be very wide though, is needed to approximate any function.

The extra hidden layers are useful, because the network can automatically learn features, in an increasing level of abstraction. This is useful, because even though an ANN with one hidden layer theoretically can be used to approximate any function, it may be very hard to practically produce such a model. A deep model may be easier to train to produce good results, compared to a shallow model.

6 Heart Transplant

A heart transplant is an operation in which a failing, diseased heart is replaced with a healthier heart from a suitable donor. Heart transplant is a treatment that is usually reserved for people who have tried medications or other type of surgeries, but their conditions have not sufficiently improved.

In adults, heart failure can be caused by several conditions, for example a weakening of the heart muscle (cardiomyopathy) or a heart problem you are born with (congenital heart defect) (Alraies & Eckman, 2014).

Heart transplantation is a life saving procedure for a patient with end-stage heart failure. While a heart transplant is a major operation, your chance of survival is good, and the median survival time is 12 years after the operation (Lund et al., 2017).

Ventricular Assist Device. For some people who can not have a heart transplant, another option may be a ventricular assist device (VAD). A ventricular assist device is a mechanical pump implanted in the chest that helps pump blood from the lower chambers of the heart to the rest of the body.

VADs are commonly used as a temporary treatment for people waiting for a heart transplant. These devices are increasingly being used as a long-term treatment for people who have heart failure but are not eligible for a heart transplant (Birks et al., 2006). If a VAD is not sufficient enough, doctors may sometimes consider a total artificial heart, a device that replaces the heart, as an alternative short-term treatment while the patient is waiting for a heart transplant.

Heart Transplantation Queue. If a doctor recommends that the patient considers a heart transplant, the patient will likely be referred to a heart transplant center for an evaluation. During the evaluation, the doctors and transplant team will conduct a physical examination, order several tests, and evaluate the patient's mental and emotional health.

If the transplant team determines that the patient is a candidate for a heart transplant, the transplant center will register the patient on a waiting list. There are about 4,000 people in the U.S. waiting for heart transplants. Unfortunately, there is a shortage of potential donors, and not enough hearts for every person in need. A person may wait months for a transplant and more than 25% do not live long enough to get one (Lund et al., 2017).

While the patients is on the waiting list, the medical team will closely monitor the condition of him or her. The transplant team may temporarily remove the patient's name from the waiting list, if he or she develop a significant medical condition. For example, a severe infection or stroke, which makes the patient temporarily unable to have a transplant during recovery.

Donors. Donors for heart transplants are individuals who may have recently died or become brain dead, which means that although their body is being kept alive by machines, the brain has no sign of life. Many times, these donors died as a result of a car accident, severe head injury, or a gunshot wound. Donors give their permission for organ donation before their death.

Allocation. An allocation policy in heart transplantation is used to decide how patients awaiting transplant will be paired with hearts from potential donors. The recipients need to be prioritized because of the lack of potential donor hearts. This means that not everyone that needs a transplant will get it in time.

In the U.S. donor organs are allocated through the United Network for Organ Sharing (UNOS). The current allocation policy they use is to first prioritize on geographical distance between recipient and donor. A heart transplant usually needs to occur within a few hours of organ removal for the donor organ to remain usable. As a result, hearts are offered first to a transplant center close by, then to centers within certain distances of the donor hospital.

Patients at the same location are thereafter ranked after their current status, which corresponds to their present level of acuteness. This is to help the sickest patients first. After the status they are prioritized on their blood group compatibility. This is because incompatible blood group between recipient and donor significantly increases the risk of graft rejection.

For patients within the same priority group, the patient with the longest wait time in the queue is offered the organ.

Operation. Once a donor heart becomes available, a surgeon from the transplant center surgically removes the heart from the donor's body. The heart is cooled and stored in a special solution while being taken to the recipient. The transplant surgery will take place as soon as possible after the donor heart becomes available.

Heart transplant surgery is an open heart surgery that takes several hours. Your surgeons will connect the patient to a heart-lung bypass machine to keep oxygen-rich blood flowing throughout your body.

In this procedure, the surgeon will make an incision in the chest. The surgeon will separate the chest bone and open the rib cage so that he or she can operate on the heart.

The surgeon then removes the diseased heart and sews the donor heart into place. He or she then attaches the major blood vessels to the donor heart. The new heart often starts beating when blood flow is restored, but sometimes an electric shock is needed to make the donor heart beat properly.

Post-transplant. After the transplant, the patient will be monitored for any signs or symptoms of rejection, such as shortness of breath, fever, fatigue, not urinating as much as needed or weight gain.

During the period after the operation, the recipient will have several follow-up appointments at the transplant center, with regular tests, including blood work, echocardiograms, electrocardiograms and heart biopsies. To determine whether the body is rejecting the new heart.

Most people who receive a heart transplant enjoy a high quality of life (Fisher, Lake, Reutzel, & Emery, 1995; Lough, Lindsey, Shinn, & Stotts, 1985). Depending on the recipients' condition, the patients may be able to return to many of the their daily life activities, such as returning to work, participating in hobbies and sports, and exercise.

Recent figures show that 80% of heart transplant patients live at least two years after surgery. The 10-year survival rate is about 56% (Lund et al., 2017). Nearly 85% return to work or other activities they previously enjoyed .

Although heart transplants are not successful for everyone. The new heart may fail because of organ rejection or because of the development of heart valve disease or coronary artery disease.

7 Simulation

7.1 Transplantation Queue

If a physician considers heart transplant a viable treatment option for the patient, the doctor may list the patient in a transplant queue.

Possible ways of leaving the queue is:

- Dying while waiting for an organ.
- Accepting an organ that is offered and being transplanted.
- Removed for other reasons, which includes being to sick or being to healthy, to be considered a candidate for transplantation.

There is a shortage of possible donor hearts (Livi et al., 1994) and an allocation policy is used to prioritize the patients when a suitable donor is found.

If the patient, and the physician in charge, choose to accept the heart from a donor, an operation is performed, where the heart is removed from the donor and grafted into the patient. If this transplant is successful the patient may live ten years or more, with an increased quality of life for the remaining years, see Figure 8 for a rough sketch of the process.

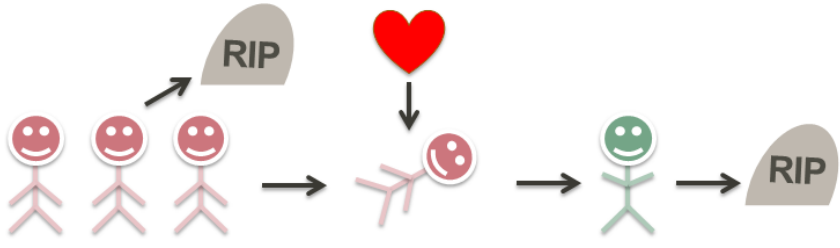


Figure 8: Illustration of the heart transplant process. First the patient are placed in the queue by a doctor. He or she can then leave the queue by dying, or a heart is allocated, or for some other reason, usually being to sick to be operated on. If a heart is allocated to the patient, and he or she accepts it, and undergoes an operation. If it is successful, the median survival time after a graft is about 12 years. With an increased quality of life for the remaining years.

7.2 Optimization Problem

Predictions models are, most of the time, optimized for the prediction of a single patient, and not applicable to larger groups of patients. Using such a prediction model, in a greedy algorithm, results in a locally optimal result. This may or may

not be close to the global optimum, for the group as a whole; measured using some metric, such as survival time. This is one of the reasons why we simulate the whole queue system in a organ allocation process.

Simulating a transplantation queue requires the creation of a model of the queue. This model can thereafter be used to simulate the impact of different policies, on several possible metrics. Examples of potential metrics are: the number of deaths in the waiting list, the mean survival time after transplant, and the end size of the waiting list.

The selection of the best allocation policy can be formulated as an optimization problem, where we try to maximize or minimize the selected metrics, depending on how the problem is defined, by selecting an appropriate policy.

7.3 System Model

Often some kind of discrete event model is used to simulate the allocation process (Cassandras & Lafortune, 2009). Mainly because the nature of the problem lends itself to be described with such a model.

Figure 9 shows a block diagram on how a simulation model of transplant queue is constructed. The different components of such a model are described in this chapter.

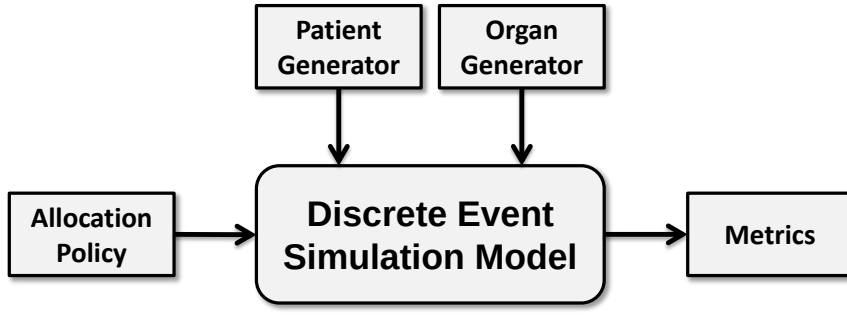


Figure 9: The basic structure of an organ allocation simulation system.

7.4 Allocation Policies

An allocation policy is used to decide how to prioritize the patients in the waiting list, with regard to the organs coming from the donors.

An allocation policy may for example be that the patient must match in blood type as prerequisite and then patients are prioritized by geographical location and secondly the acuteness of the patient using some metric.

It is possible to use a machine learning algorithm as the allocation policy. To our knowledge, it is not currently used in any of the national transplant systems.

Using machine learning as an allocation policy may possibly be implemented by creating a model to predict the survival time after transplant, ranking the patients after survival time for each potential donor, and selecting the patient with the highest rank to receive the organ. Such an algorithm would be greedy, and would maximize the predicted survival locally for each potential donor heart.

7.5 Metrics

Metrics are used to measure some property of the allocation system. These are usually divided into two main types: utility and equity, corresponding to making the best use of a scarce resource, and giving everyone an equal chance for a transplant.

Examples of utility measures are pre-transplant deaths, patients removed for other reasons, and survival time after transplant. The total number of transplants, differences in waiting time and probability of transplants are examples of equity measures.

7.6 Patient and Organ Generation

It is possible to use real datasets as the basis for simulating the flow of patients and organs. There exist at least four different ways of generating patients and donor organs:

- The generation of patients and donors may use exact historical data, that is, replicating the waiting list and using the real dates of the incoming patients and donor organs.
- It may use some stochastic process to select which real patients or organs arrive at certain time points, utilizing for example a Poisson process to simulate the arrivals, see next paragraph.
- It is also possible to create synthetic examples of patients and organs, using some stochastic process, drawing from a distribution of real variables for the patients.
- Another way of creating synthetic examples could be by creating a model of the patient values and generating new patients from it.

Poisson Point Process. A Poisson point process, or Poisson process, is a type of random mathematical object that consists of points randomly located on a mathematical space. The point process has convenient mathematical properties, which has led to its use as a mathematical model for seemingly random processes in various disciplines (Kingman, 1992).

The Poisson point process is often defined on the real number line, where it can be considered as a stochastic process. In this setting, it is used, for example, in queuing theory to model random events, such as the arrival of customers at a store or phone calls at an exchange, distributed in time.

The Poisson point process is related to the Poisson distribution. The Poisson distribution implies that the probability of a Poisson random variable N being equal to k, is given by Equation 22. Which means that the probability of observing k events in an interval. Where λ denotes the rate or intensity, that is the average number of events per interval, and $k!$ is the factorial of k.

$$P(N = k) = e^{-\lambda}\frac{\lambda^k}{k!} \tag{22}$$

The homogeneous Poisson point process, when considered on the positive half-line, can be defined as a counting process, a type of stochastic process. It can be denoted by $\{N(t), t \geq 0\}$.

A counting process represents the total number of occurrences or events that have happened up to and including time t. The number of events in any interval of length t is a Poisson random variable with parameter (or mean) λt. The probability of the random variable $N(t)$ being equal to k is given by Equation 23.

$$P(N(t) = k) = e^{-\lambda t}\frac{\lambda t^k}{k!} \tag{23}$$

The time differences between the events of the counting process are known as interarrival times. The Poisson counting process can also be defined by stating that the interarrival time are exponential variables with mean $1/\lambda$.

A Poisson point process can be used to simulate the arrival times of both patients to the queue and donor hearts (Zenios, 1999).

7.7 Discrete Event Simulation Model

Our allocation simulation model begins at a certain date with a starting wait list of patients, which may be either from real historic data or generated using some stochastic process.

Then date is stepped forward in a discrete manner and the status of the patients in the waiting list are updated. The status update uses a model that includes mortality in the wait list and delisting for other reasons.

Following the status update, the patient generator simulates the addition of new patients to the waiting list and the organ generator produce the arrival of new transplantable organs.

An allocation policy is then used to prioritize the patients in the waiting list. The highest prioritized patient in the list always accepts the offered organ.

A model is then used to predict the post-graft survival time of the receiving patient.

This process is then repeated for each date from the starting point to the end. When the simulation is finished it calculates and outputs the desired metrics.

See Algorithm 1 for pseudo code of our simulation process.

Algorithm 1 Pseudo code for a discrete event simulation model

```
1:  procedure SIMULATEALLOCATION
2:      curDay ← 0
3:      waitList ← GENERATESTARTLIST
4:      dead ← [ ]
5:      transplanted ← [ ]
6:      while curDay < endDate do
7:          dead ←UPDATEPATIENTS(waitList)
8:          waitList ← waitList+ GENERATEPATIENTS
9:          donorOrgans ← GENERATEDONORORGANS
10:         transplanted ←ALLOCATE(waitList, donorOrgans)
11:         CALCULATESURVIVAL(transplanted)
12:         day ← day + 1
13:     CALCULATEMETRICS(waitList, dead, transplanted)
```

8 Application for Natural Language Processing

8.1 Motivation

The machine learning (ML) models used in natural language processing (NLP) are often applicable to other fields. The concept of a pipeline to process the information from raw data to solve a useful problem is often constructed in a similar way, regardless of the domain that the ML is applied to. See Figure 11 for an example of such a pipeline.

Thus, the techniques and the construction of a ML pipeline, could be transferred to other problems, such as processing data that corresponds to heart transplant patients.

A large percentage of queries to retrieve images relate to people and objects as well as relations between them, the 'story' within the image. Although the automatic recognition, detection and segmentation of objects in images has reached remarkable levels of accuracy, the identification of relations is still a territory that is yet largely unexplored. The identification of these relations, though, would enable users to search images illustrating two or more objects more accurately.

Relations between objects within images are often ambiguous and captions are intended to help us in their interpretation. As human beings, we often have to read the caption or the surrounding text to understand what happened and the nature of the relations between the entities. This combined use of text and images has been explored in automatic interpretation.

8.2 Classifying Relations in Images

Problem We wanted to see if combining visual information from images with text associated with the pictures, could improve the quality of classifying relations in the images.

To this end, we designed an experiment where we extracted images from Wikipedia and the articles associated with them. To restrict the problem in to something manageable, we only used images with horses and humans in them.

We further restricted the problem to only include three potential relations between horse-human pairs. The relations we chose were: *Ride*, *Lead*, or *None*, where the ride relation corresponds to human sitting on the horse, lead relation is when a human is standing next to the horse holding its reins, and the none relation is the complementary relation of the other two, that is, every other relation that is not either ride or lead. Examples of none relations could be a human just standing next to the horse, taking a photo of the horse or feeding the horse.

We processed the images visually and annotated them with bounding boxes that contains the objects, which were either a human or horse, see Figure 10 for an example. A bounding box is represented by the length and width of the box and its position.

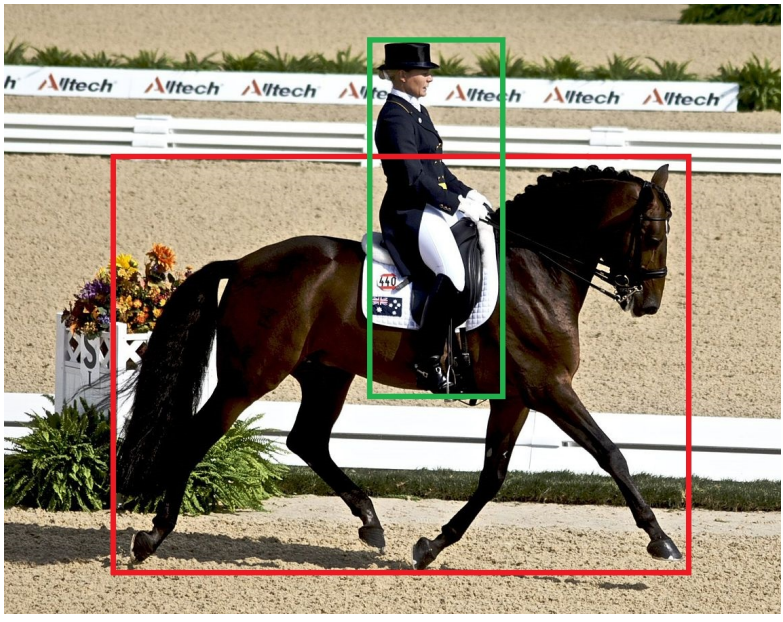


Figure 10: The bounding boxes illustrated for human-horse pair with the relationship riding. A woman is competing in dressage with her horse.

An image can contain more than a single human or horse, and for each possible human-horse pair in the images, we wanted to classify the relation between the possible pair.

Method As our focus was to investigate to what extent the use of combinations of text and visual cues could improve the interpretation or categorization precision, we set aside the automatic detection of objects in the images. We manually identified the objects, within the images extracted from Wikipedia, by creating bounding boxes around horses and humans. It resulted in 2,235 possible human-horse pairs in the images, but the distribution of relations was quite heavily skewed towards the none relation.

The visual parsing annotation provided us with a set of objects within the images and their bounding boxes defined by the coordinates of the center of each box, its width, and height.

To implement the baseline, we derived a larger set of visual features from the bounding boxes, such as the overlapping area, the relative positions, etc, and combinations of them. We ran an automatic generation of feature combinations and we applied a feature selection process to derive our visual feature set. We evaluated the results using cross-validation.

We extracted the semantic features from the Wikipedia articles. We implemented a selector to choose the size of the input between: complete articles, partial articles (the paragraph that is the closest to an image), captions, and file names.

A bag-of-word (BoW) feature was created for each of the four different inputs. The BoW features have a filter that can exclude words that are either too common, or not common enough, based on their frequency, controlled by a threshold. We used a $TF \cdot IDF$ weighting on the included words.

Instead of using all of the words in a document, we used information derived from the predicate–argument structure to filter out more relevant terms. We created a feature that only used the predicate names and their arguments as input. The words that are not predicates, or arguments to the predicates, are removed as input to the feature. The arguments can be filtered depending on their type, for example A0, A1, or AM-TMP. We can either consider all of the words of the arguments, or only the heads.

To classify the relations, we used a logistic regression model. This model outputs probabilities for each of the classes. The easiest way to classify a horse-human pair is to take the corresponding probability vector and pick the class with the highest probability. But sometimes the probabilities are quite equal and there is no clear class to chose. We selected a threshold using cross-validation. If the maximum probability in the vector is not higher than the threshold, the pair is classified as *None*. We observed that because *None* represents a collection of actions and nonaction, it is more likely to be the true class when *Ride* and *Lead* have low probabilities.

Figure 11 shows an overview of the system architecture that we used.

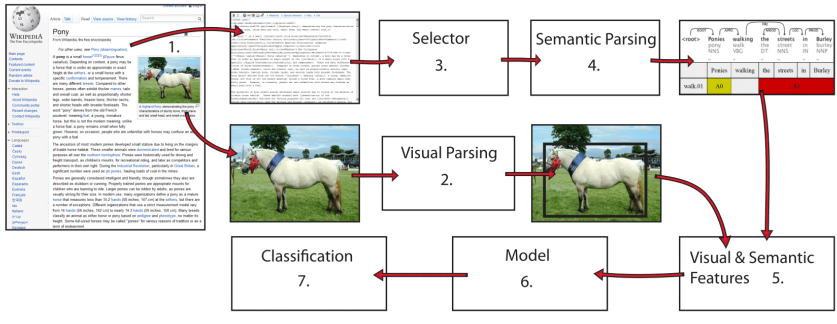


Figure 11: An overview of the system design.

Results Table 2 shows an overview of the results. The baseline which corresponds to the geometrical features; we obtained a mean F_1 of 0.67 using them.

Predicate corresponds to the baseline features combined with the predicate features. Predicate features on the whole article text gave better results than combining different portions of the text and produced the best result a F_1 of 0.73, this is a relative error reduction of 18.6% compared to the baseline.

		Mean of F_1	Difference (pp)	Relative error reduction (%)
Baseline		0.6706	0.00	0.00
BoW	Articles	0.6779	0.73	2.22
	Partial articles	0.6818	1.12	3.40
	Captions	0.6829	1.23	3.73
	Filenames	0.6802	0.96	2.91
	Combination	0.7132	4.26	12.9
Predicate	Articles	**0.7318**	**6.12**	**18.6**
	Partial articles	0.6933	2.27	6.89
	Captions	0.6791	0.85	2.58
	Articles + Words	0.6830	1.24	3.76
	Articles + Coref	0.7280	5.74	17.4

Table 2: An overview of the results, with their mean F_1-value, difference in percentage points and relative error reduction from the baseline mean F_1-value.

Article The article corresponding to this task is Paper I: “Improving the Detection of Relations Between Objects in an Image Using Textual Semantics”.

9 Applications for Heart Transplant

In Section 6, I described the heart transplant process. In this chapter, I summarize the tasks in this thesis related to heart transplantation.

9.1 Motivation

Heart transplantation is a life saving operation for patients with end-stage heart failure. After a successful graft of a heart from the donor to the receiving patient, the patient's median survival time is about 12 years post-transplant. The patient's quality of life will probably improve considerably, compared to living with a failing heart. He or she may resume many of their previous life activities such as starting to work again, or continuing with hobbies, or exercising.

There exist a shortage of potential heart donors, and more than 25% of the patients that are placed in the transplantation queue, die while awaiting a suitable heart. A patient may be in the queue for months before a heart is offered to him or her. Because of the shortage, not everyone in need of heart will receive it, and the doctors have to prioritize the incoming hearts to the patients waiting in the queue.

The rules for prioritization of donor organs are usually formulated on a national level, and is often referred to as an allocation policy. An allocation policy is formulated with regards of both utility and equity. These terms correspond to make the best use of a scarce resource, and give everyone an equal chance for a transplant.

The substantial improvement in survival time and quality of life for patients with heart failure, makes this an interesting problem to work with. It has real life implications, compared to for example solving some abstract problem in computer science.

I wanted to maximize the potential use of the available organs. To make this possible, I wanted to answer some questions that arose while working on this problem. Some of these questions were the following:

- What variables affect patient survival?
- Is it possible to predict survival time in the queue and after transplant?
- How does different allocation policies change the outcome for the patients?
- Is it possible to use machine learning as an allocation policy?

I designed and executed several experiments trying to answer these questions. In the following sections I will explain the different tasks, corresponding to the papers included in this thesis. The tasks build upon each other and follow the flow seen in Figure 12.

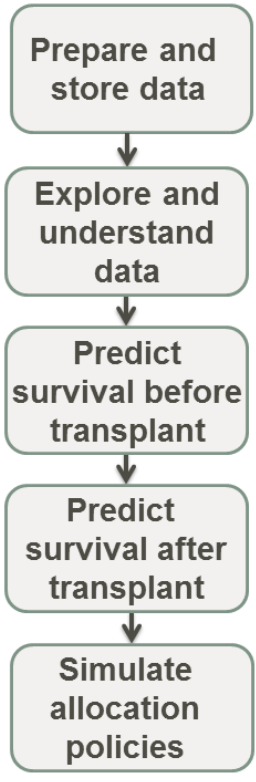


Figure 12: The flow of the different tasks. These tasks were undertaken to answer some of the questions we had, regarding to heart transplant.

9.2 Prepare and Store the Data

We start with the first task in the chain, to preprocess and represent the data in a suitable format.

Problem To be able to use machine learning on problems related to heart transplant, we needed data as input to the models. The data that we wanted to use comes from three different sources, an American database that is handled by the United Network for Organ Sharing (UNOS), a Scandinavian registry called Scandia Transplant, and an international database handled by the The International Society for Heart & Lung Transplantation (ISHLT).

In total, ISHLT contains about 100,000 recorded heart transplantations. Although ISHLT could be seen as a superset of all the included databases, in regards to the patients recorded, it only incorporates a subset of the variables that are

contained within the different registries. ISHLT is restricted to variables that are frequently recorded by the different regional registries.

The three data sources we considered: ISHLT, UNOS, and Scandiatransplant, have different structures, a different number of variables, use different variable names, and may use different units or encoding of the data.

The variables contained in these databases pertain to both recipient, donor and the operation itself. It can for example be the age, weight, gender, or blood group of the patients.

UNOS contains the largest number of variables, about 500. ISHLT, for example, does not feature the variable *crossmatch_done*, that is available in UNOS. This is a test before a blood transfusion to determine if the donor's blood is compatible with the blood of an intended recipient.

We wanted a unified interface to make it easy to access all of the databases.

Method The ISHLT, UNOS, and Scandiatransplant data sets are normally distributed to the researchers as SAS or CSV files. We started from the CSV files and we converted them to an RDF format.

The CSV files represent the transplants as rows, where each column is a variable for the transplant. In the RDF conversion, we mapped each row to a head node and we created leaf nodes for the selected variables.

The data sets use different names to denote the same variables. For example, the most recent blood creatinine value for the recipient patient is *Most rec. Creat.* in Scandiatransplant, *creat* in ISHLT, and *creat_trr* in UNOS, see Figure 13.

Figure 13: A unification of the variable representing the most recent creatinine level of the recipient

We created unified names for about 140 of the variables, such as *aaot:creatinine* for the creatinine value, where the *aaot* prefix stands for *Algorithms and Applications for Organ Transplantation*.

We cleaned the data and removed unreasonable outliers from variables, such as having height below 30 cm or above 300 cm. These outliers can, for example, be produced by a nurse using the wrong unit, when filling in the forms for a patient. Maybe using meters instead of centimeter, when recording the height of a patient.

We had to encode the data in a unified way between the databases, for example binary variables were both recorded as Y/N and 1/0, and categorical variables often used different codes to encode the data between the registries.

As previously mentioned, UNOS has more variables than ISHLT and Scandiatransplant. We used the UNOS variable names, when they had no counterpart in the other two registries.

We also added metadata about the variables containing the original variable name, as well the new one, the description of the variable, the source form of the data, the unit where it is applicable, as well as comments, and start and possibly end date of the recording of the variable

Results The creation of the RDF representation has simplified the use of the three registries. It enabled us to utilize a unified interface to query the data using SPARQL, which made it easier to handle the patient variables.

Article The article corresponding to this task is Paper II: "Using a RDF Triplestore and Deep Learning to Predict Heart Transplantation Survival".

9.3 Explore and Understand the Data

We continue with the next task, to analyze and get an understanding of the data.

Problem We wanted to find optimal feature sets to predict the survival of patients after one, five and ten year time period after heart transplant. This is a binary classification problem, where the patient are either alive or dead after respective time period.

We also wanted to rank the features after their importance, for each period, to find out which features had the largest impact on the prediction.

To find a globally optimal feature set requires 2^n tests, where n is the number of features. This is infeasible even for a moderate number of features. Using the 482 features we had available, this would require $2^{482} \approx 1.25 \times 10^{145}$ tests.

Method We applied a greedy forward selection and a greedy backward elimination that enabled us to find a locally optimal subset. This a much more computationally feasible feature set to find then a global optimal.

Table 3: The best validation set AUROC values for 1, 5, and 10 years, found using a search with 482 possible features.

Years	AUROC
1	0.6990
5	0.6892
10	0.7509

The greedy forward selection starts from a subset of the features, which can be empty, and adds one feature from the remaining set to the current subset. The selection procedure uses the new subset to produce the classification probabilities. These probabilities are then used to calculate an evaluation metric. The feature which improves the performance the most is then added to the current feature set for the next generation. The procedure is repeated if it improves the score of the preceding subset over a certain threshold Δ. If there is no improvement, we use the current feature set for a backward elimination instead.

The backward elimination removes the features one by one from the starting set and the resulting feature set is used to produce the classification probabilities. If the score improves on the preceding generation, then the process is repeated with the resulting feature set.

If two following forward selections and backward eliminations do not improve the score, the process is stopped and the resulting feature set corresponds to a local optimum.

Starting from the empty feature set and doing a full forward search of the 482 features, meaning that every feature is added, would result in about 100,000 models being tested. A number somewhat smaller than 1.25×10^{145}, but it is only locally optimal though.

We used logistic regression as the machine learning model for the search procedure, mainly because it has a short training time for each model, and few hyperparameters to tune.

For each generation in a forward and backward search there is no dependence between the models. This makes it quite easy to parallelize. We parallelized the search using the Spark framework, to distribute the workload on a local cluster.

To assess the importance of the variables, we did a forward search from an empty feature set and recorded the order in which they were added. This roughly corresponds to the most important features for each time period.

Results We found locally optimal feature sets, utilizing the available 482 features, for each time period, using our logistic regression model. The best results for these feature sets, using area under the receiver operating characteristic curve (AUROC) as the metric, is presented in Table 3.

Table 4: The ten first features added for a forward search for the 1, 5, and 10 year time periods. Functional status (F.s.); Ventricular assist (Vent. ass.); Research immunosuppressive (Res. immuno.); Donor coronary (Don. cor.)

Rank	1 year	5 years	10 years
1	Anti viral	Ethnicity: white	Days in status: 1
2	Creatinine	Creatinine clearance	Days in status: 2
3	Height	Func. status: very sick	Days in status: 1b
4	Donor age	Donor age	Don. cor. angiogram: no
5	Ventricular assist	Ventricular assist	Func. status: very sick
6	Vent. ass. type: none	Donor ischemic time	Res. immuno. medication
7	Serum bilirubin	F.s.: cares for self	F.s.: cares for self
8	Donor ischemic time	F.s.: occasional assistance	Diabetes
9	Other therapies	F.s.: normal activity with effort	Anti viral
10	Dialysis	F.s.: considerable assistance	F.s.: considerable assistance

The validation AUROC scores that are about the same for 1 and 5 years, but approximately 8 percentage points higher for 10 years. This is somewhat unintuitive and we tried without success to find confounding factors to explain these results. A possible explanation is that there are much more positive examples, that is, dead patients, for 10 years compared with 1 and 5 years. Another bias is that many patients of this cohort are censored compared to 1 and 5 years: About 50% of the patients are censored after 10 year time period.

We listed the most important features, found using a forward search, in Table 4.

Article The article corresponding to this task is Paper III: “Selection of an optimal feature set to predict heart transplantation outcomes”.

9.4 Predict the Survival Before Transplant

The next task was to predict the outcome of patients placed in a heart transplantation queue.

Problem Estimating the probability of dying in the waiting list given a waiting time could support the decision of surgeons on the priority of a transplantation. In addition, knowing the probability for a patient to be transplanted within a certain time frame would help plan operation resources and inform the patient. Extending the models to predict the amount of days a patient may survive in the queue, could be used in a queue simulation system, see the following task described in Section 9.6.

Table 5: The F1 values for 180, 365, and 730 days obtained on the test set

Days	F1 (micro)	F1 (macro)
180	0.750	0.675
365	0.760	0.680
730	0.888	0.680

We carried out the prediction at three different time points: 180 days, 365 days, and 730 days, and we categorized the patient status with three possible outcomes: still waiting, transplanted, or dead in the waiting list.

We chose to use these time periods, because a patient should have survival time less than a year, predicted by a physician, to be placed on the waiting list. Although a small fraction of the patients may survive several years on the wait list.

There are other outcomes for patient standing in a heart transplantation queue, such as being to sick to be operated on, but most patients either is transplanted or die while waiting for an organ.

Method We created a neural network with two hidden layers and 128 nodes in each layer. The hidden layers used the rectified linear unit as activation function and the final output layer uses a softmax activation. We used categorical cross entropy as the loss function and adamax as the optimizer. Dropout was used as a regularization technique, to reduce potential overfitting. We used the Keras framework to represent this model.

In our model, we included 87 variables as input, describing the patients in the queue that were available at the time of listing. Example of such features are age, sex, weight, and blood group.

We wanted to know which features contributed the most to the result of the classification. We utilized backward elimination to find these features.

Results Table 5 shows the best obtained F1 values for 180, 365, and 730 days, respectively. Because there is more than two classes, the F1 score needs to be averaged. It was calculated using both micro and macro averaging.

The macro average takes the average of the precision and recall of the system on the different classes. When the examples are unevenly distributed across the classes, the macro average method is less biased toward the largest class.

The ten most contributing features found through a complete backward elimination, also known as an ablation study, for each time period is presented in Table 6. Using only the ten most important features resulted in a decrease of

Table 6: The ten most contributing features for each time period in order of importance, found using an ablation study.

Rank	180 days	365 days	730 days
1	Urgency status 2	BMI	BMI
2	Weight	Weight	Weight
3	BMI	Height	Height
4	Height	Urgency status 2	Urgency status 2
5	Inotropes	Creatine clearance	Creatinine
6	Blood group: AB	Inotropes	Functional status
7	Life support	Blood group: A	Pulmonary Vascular Resistance
8	Blood group: B	Life support	Educational level: none
9	Inotropic support	Blood group: AB	Ventricular assist type: LVAD + RVAD
10	Ethnicity: black	Blood group: B	Educational level: grade school

only about 2% (absolute difference) from the F1 macro score with all the features. This means that most of the predictive power from the ANN comes from a few features.

The features shared by all of the three sets are: urgency status 2, weight, height and BMI. BMI can be considered a feature transformation of weight and height as BMI = weight $\times$ height^2, but it provided extra predictive information over the constituent variables. A sufficiently complex neural network could probably approximate this transformation and therefore BMI would probably not be needed.

Article The article corresponding to this task is Paper IV: "Predicting the Outcome for Patients in a Heart Transplantation Queue using Deep Learning".

9.5 Predict the Survival After Transplant

The following task was to predict the survival of patients after a heart transplant.

Problem One of the most limiting factors of the number of heart transplants performed is the lack of donor organs and a conservative allocation policy that results in the loss of about half of the organs being offered. An improved prediction of the outcome would augment the confidence in the post-transplantation performance and make it possible to optimize the allocation of organs. Furthermore, it would enable practitioners to determine the risk of early and late graft dysfunction more accurately and improve donor and recipient management.

Although there exist several survival models within cardiac surgery, currently there is no accepted tool for estimating the outcome after heart transplantation. In recent years, some risk score algorithms designed to predict post-transplantation performance have been developed. One of the most notable was the Index for Mortality Prediction After Cardiac Transplantation (IMPACT).

IMPACT was created with a data set of heart transplant patients between 1997 to 2008 that were collected from the UNOS database. IMPACT only utilizes recipient variables. By apportioning points according to the relative importance of the variables for the one-year mortality, a risk index was created. The points are after that converted to a predicted probability of one-year mortality by a formula derived from logistic regression.

The International Heart Transplantation Survival Algorithm (IHTSA) was developed on the ISHLT registry, with patients who were transplanted between 1994 and 2010. IHTSA utilizes both recipient and donor variables. The survival model consists of a flexible nonlinear generalization of the standard Cox proportional hazard model. Instead of using a single prediction model, this model integrates ensembles of artificial neural networks. In addition, its prediction capability is not limited to one year.

We wanted to determine the most suitable risk stratification model for heart transplantation by comparing the IMPACT and IHTSA algorithms.

Method We included all the adult heart transplant patients (>17 years) from January 1997 to December 2011, from the UNOS database. The data set was divided into two temporal cohorts: transplantation done before 2009 (derivation cohort) and after or during 2009 (test cohort). These time periods were chosen because both IMPACT and IHTSA were developed on patients between 1997-2008 and we wanted disjoint sets (derivation and test) to evaluate the prediction performance.

We used the cohorts as input to both algorithms and then evaluated the performance for both methods.

The discriminatory power for one-year mortality was assessed by calculating the AUROC. We compared the statistical significance of the difference between the AUROC of the two models using the non-parametric DeLongs test. To evaluate the discrimination for long-term survival of the patients, we utilized the Harrells concordance index (C-index). We used a z-score test to compare the C-indexes.

Table 7: The AUROC values for one-year mortality with the different cohorts using IMPACT and IHTSA respectively.

	AUROC (95% CI)		
Time period	**IMPACT**	**IHTSA**	**P-Value**
1997-2008	0.61 (0.59-0.62)	0.69 (0.68-0.70)	0.001
2009-2011	0.61 (0.58-0.63)	0.65 (0.63-0.68)	0.001

Table 8: The Harrells C-index for survival for the different cohorts using IMPACT and IHTSA respectively.

	C-index (95% CI)		
Time period	**IMPACT**	**IHTSA**	**P-Value**
1997-2008	0.56 (0.56-0.56)	0.62 (0.61-0.62)	0.001
2009-2011	0.58 (0.56-0.61)	0.63 (0.61-0.65)	0.001

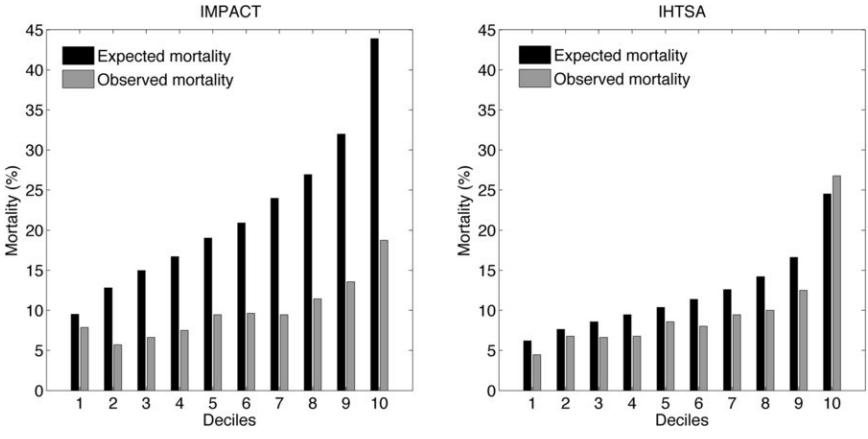


Figure 14: The observed (gray bars) and expected mortality (black bars), in percent, for each decile, for the IMPACT and IHTSA models, in the test cohort (2009-2011). The patients are divided into deciles according to their expected mortality, and the observed mortality was derived for each decile.

Results As shown in Table 7, the IHTSA model has a significantly higher discrimination compared with the IMPACT model for one-year mortality, P=0.001, corresponding to an error reduction of 11.7%. Harrells C-index for the recalibrated IHTSA compared with IMPACT was substantially larger, as shown in Table 8, with about a 4% absolute difference for the later time era.

The calibration plot, Figure 14, shows that the predictive mortality compared with actual mortality was more consistent over all deciles for the ITHSA model, compared with the IMPACT model.

We have shown that a flexible nonlinear artificial neural network model (IHTSA), utilizing deep learning techniques, exhibits better discrimination and accuracy than a more traditional risk score model (IMPACT) for predicting one-year mortality. We made public the results of the IHTSA model in the form of a web-based batch calculator, that could be used as a virtual recipient-donor matching tool.

Article The article corresponding to this task is Paper V: "Improving Prediction of Heart Transplantation Outcome Using Deep Learning Techniques".

9.6 Simulate Impact of Allocation Policies

Problem Allocation policies in heart transplantation are used to decide how patients awaiting transplant will be paired with hearts from donors. There is a trade-off between medical justice, giving everyone an equal chance for a transplant, and medical utility, which aims at making the best use of a scarce resource.

Predictions models are, most of the time, optimized for the prediction of a single patient, and not applicable to a larger group of patients. This is the reason why the simulation of the whole queue system in an organ allocation process better fits the goal of selecting a policy that maximizes the benefit over all the patients.

Simulating a transplantation queue requires the creation of a model of the queue. This model can thereafter be used to simulate the impact of different policies, on several possible metrics. Examples of potential metrics are the number of deaths in the waiting list, the mean survival time after transplant, and the end size of the waiting list.

The selection of the best allocation policy can be seen as an optimization problem, where you try to maximize predefined metrics by selecting an appropriate policy.

Method We used a discrete event model to simulate the allocation process, see Section 7.7.

We chose a Poisson process to simulate the arrival of recipients and donors. This is achieved by selecting patients, without replacement, from the all of the real patients from that specific year.

We created two prediction models; one to simulate the removal of patients from the wait list, mainly caused by death, and the other to predict the survival after heart transplant. A similar model architecture is used. The main difference is the input features. The pre-transplant prediction uses 87 features, while the post-transplant utilizes 267 features. We have called this model: Lund Deep Learning Transplant Algorithm (LuDeLTA).

In addition to our own model LuDeLTA, we also used the IHTSA model, see Section 9.5, to predict the post-graft survival of the patients. We evaluated the different allocation methods with both models.

We selected four allocation policies we wanted to evaluate. The policies were the following: wait time, clinical rules, and neural networks in two different versions. Wait time prioritized the patients with the longest wait time in the queue. Clinical rules ranked the patients based on simple rules based on weight, gender, age and blood group. Allocation based on neural networks ordered the patients

after predicted survival time and chose the patients with the longest predicted survival time for each donor, using either the IHTSA or LuDeLTA as the survival model.

Table 9: Performance metrics of the LuDeLTA models

Metric	Pre-transplant	Post-transplant
AUROC 1 year	0.89	0.66
C-index	0.80	0.61

Table 10: Results from simulating heart allocation policies.

Allocation policy	Mean survival IHTSA (days)	Mean survival LuDeLTA (days)
Wait time	4,285	4,309
Clinical rules	4,349	4,309
IHTSA	4,976	4,719
LuDeLTA	4,541	5,668

Table 11: Results from simulating heart allocation policies.

Allocation policy	Number transplanted	Number dead wait list	Number alive wait list	Mean wait time (days)
Wait time	9,469	5,485	444	139
Clinical rules	9,345	5,481	572	150
IHTSA	9,469	4,801	1128	150
LuDeLTA	9,469	4,993	936	110

Results We evaluated the LuDeLTA models using the AUROC for the one year mortality, and the long time survival using The Harrells C-index on the validation set. Results are shown in Table 9. The predicted mean survival on the wait list without transplant was 447 days using our pre-transplant survival model.

The results for the different allocation policies can be found in the Table 10 and Table 11. The predicted mean survival using LuDeLTA, for allocating according to wait time was about 4,300 days, clinical rules 4,300 days and using IHTSA 4,700 days.

The transplant policies based on the neural network models or wait time utilize all of the available organs, while using clinical rules lead to a discard of 124 hearts.

We have shown that an organ transplant queue can be simulated by utilizing neural networks to predict survival, both pre- and post-transplant. Additionally we have shown that using neural networks as the allocation policy, could possibly result in longer survival post-transplant for the patients.

Article The article corresponding to this task is Paper VI: "Simulating the Outcome of Heart Allocation Policies using Deep Neural Networks".

10 Conclusion

In this thesis, I have presented the basic concepts of machine learning (ML), how to represent data, how to use different metrics to evaluate prediction results, how to create different models to predict values from data, and finally some applications of these techniques to problems in the field of natural language processing (NLP) and biomedical data.

In the domain of NLP, we have shown that using textual information associated with the images could improve classification of relations in images. Even though we chose to do it on a quite small subset of potential objects and relations. The objects we chose were humans and horses, with the possible relations ride, lead, or none.

The general structure of a ML pipeline looks the same, independently of the domain it is being applied to. This means that knowledge learned from the NLP task could be transferred to the heart transplant field. The main part of the work in this thesis was done on tasks related to heart transplant, see Section 9.

First, we created a unified RDF representation of the three databases, UNOS, ISHLT, and Scandia transplant. It made it easier for us to query the data using the same format. This representation of the data was used as the basis for the next tasks.

The following task was to explore and understand the data. We designed an experiment in which tried to find locally optimal feature sets for one, five, and ten year survival of heart transplant patient, with logistic regression. The AUROC values for the periods were of 0.70, 0.69, and 0.75, respectively. We also ranked the features after their importance in predicting the outcome.

The next task was to predict pre-transplant outcome for patients. We did this for the 180, 365, and 730 day periods. We created ANN models to predict the outcome as either dead, transplanted, or queuing, for each period. The F1 score with macro averaging for each of these models were about 0.68 for all periods. We also performed an ablation study to find the most contributing features for each period.

The fourth task was to predict survival post-transplant. We wanted to evaluate a model created here in Lund, IHTSA, with an older, but often referenced model from the USA, IMPACT. We retrained the ANN model used in IHTSA on the same time period as IMPACT was trained on (1997-2008) and evaluated the result on a later period 2009-2011. This was to have disjoint train and test sets, to avoid overfitting on the data. For the later time period, we got a difference of 4 percentage points on the AUROC score and a 5 point difference in C-index.

The final task was to simulate a whole transplant queue system, to be able to assess the impact of different allocation policies. We created an algorithm which we called LuDeLTA, based on the work from the previous tasks. These models were used to predict both pre- and post-transplant survival times. The models were used in conjunction with a discrete simulation system. We used this simulation to

evaluate some simple allocation policies and using ANNs to prioritize the patients. In our simulation, the predicted mean survival time utilizing LuDeLTA, was about 400 days longer using IHTSA as the allocation policy, compared to using wait time.

In this thesis, I have shown that several different kinds of problems were solvable using machine learning techniques, especially utilizing deep learning models.

10.1 Future Work

An extension of the work done in this thesis is to train prediction models for both pre- and post-transplant survival, based on the features available in Scandia transplant. The amount of features that can be used is less than in UNOS and ISHLT registries. The amount of recorded patients in the Scandia database is also considerably less than the other two registries.

It is possible extend the training set, using a concatenation of data from Scandia with the UNOS or ISHLT, including only the overlapping features between them. If the patient outcomes, depending on the features, are not wildly different between registries, this should help the accuracy of the model.

The models produced by this work could then be introduced as components in a tool that the doctors can used, which are involved in the transplant process, in Norway, Sweden, and Finland. Such a tool could consist of web page where the physicians could enter potential donors, where the tool predicts the survival for each patient in the waiting list and each recipient-donor pair after transplant, and may combine these two metrics and rank the patient after predicted survival. This could be used to augment the doctors decision process, to help with the allocation of a potential heart donor.

Something else that may be interesting is to combine NLP with a biomedical application. An idea we had was to use the database of recorded journals and operation descriptions to see if we could improve the outcome prediction of the transplants based on the written text in these journals. It would be possible to reuse some of the techniques discussed in the NLP part of this thesis to create a model from the texts annotated by the surgeons to predict the outcome of the surgery.

The operation descriptions seem to be, in large part, automatically generated, but the physician has the possibility to add additional information, about potential problems they may encounter during the operation and so forth. This information may be used a features for a model to predict the outcome of the patients post-transplant.

Another interesting problem to explore would be to use machine learning on genetic data of patients to see if the outcome could be predicted depending on their DNA sequence. The genetic makeup of a person is probably not the only factors for a successful operation and long survival afterwards. There are presumably many environmental factors that can affect the outcome of a transplant, such as

lifestyle of the patient and the efficacy of the surgical team. Hopefully there is some correlation between genetics and the end result for a patient.

DNA sequencing is still a quite expensive and cumbersome process, even though prices are decreasing over time. This limits the amount of potential genomes that we could potentially analyze.

Any of the problems above could be interesting to explore in the future.

Bibliography

Alraies, M. C., & Eckman, P. (2014). Adult heart transplant: indications and outcomes. *Journal of thoracic disease*, *6*(8), 1120.

Birks, E. J., Tansley, P. D., Hardy, J., George, R. S., Bowles, C. T., Burke, M., ... Yacoub, M. H. (2006). Left ventricular assist device and drug therapy for the reversal of heart failure. *New England Journal of Medicine*, *355*(18), 1873–1884.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of compstat'2010* (pp. 177–186). Springer.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, *16*(5), 1190–1208.

Cassandras, C. G., & Lafortune, S. (2009). *Introduction to discrete event systems*. Springer Science & Business Media.

Chicco, D., Sadowski, P., & Baldi, P. (2014). Deep autoencoder neural networks for gene ontology annotation predictions. In *Proceedings of the 5th acm conference on bioinformatics, computational biology, and health informatics* (pp. 533–540).

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 215–242.

Dismuke, C., & Lindrooth, R. (2006). Ordinary least squares. *Methods and Designs for Outcomes Research*, *93*, 93–104.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226–231).

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

Fingelkurts, A. A., & Fingelkurts, A. A. (2004). Making complexity simpler: multivariability and metastability in the brain. *International Journal of Neuroscience*, *114*(7), 843–862.

Fingelkurts, A. A., Fingelkurts, A. A., & Kähkönen, S. (2005). Functional connectivity in the brainis it an elusive concept? *Neuroscience & Biobehavioral Reviews*, *28*(8), 827–836.

Fisher, D. C., Lake, K. D., Reutzel, T. J., & Emery, R. W. (1995). Changes in health-related quality of life and depression in heart transplant recipients.

The Journal of heart and lung transplantation: the official publication of the International Society for Heart Transplantation, *14*(2), 373–381.

Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, *21*, 768–769.

Foundation, T. A. S. (2015). *Apache spark.* Retrieved 2015-04-07, from `https://spark.apache.org/`

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *science*, *315*(5814), 972–976.

Gybenko, G. (1989). Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, *143*(1), 29–36.

Hardy, M. A. (1993). *Regression with dummy variables* (No. 91-93). Sage.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., & Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama*, *247*(18), 2543–2546.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, *44*(1), 1-12. Retrieved from `http://dx.doi.org/10.1021/ci0342472` (PMID: 14741005) doi: 10.1021/ci0342472

Hobbs, J. R. (1979). Coherence and coreference. *Cognitive science*, *3*(1), 67–90.

Izenman, A. J. (2013). Linear discriminant analysis. In *Modern multivariate statistical techniques* (pp. 237–280). Springer.

Kenward, M. G. (2013). The handling of missing data in clinical trials. *Clinical Investigation*, *3*(3), 241–250.

Kingman, J. F. C. (1992). *Poisson processes* (Vol. 3). Clarendon Press.

Kohavi, R., et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, pp. 1137–1145).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lagakos, S. (1979). General right censoring and its impact on the analysis of survival data. *Biometrics*, 139–156.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553),

436.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R news*, *2*(3), 18–22.

Livi, U., Bortolotti, U., Luciani, G. B., Boffa, G. M., Milano, A., Thiene, G., & Casarotto, D. (1994). Donor shortage in heart transplantation: Is extension of donor age limits justified? *The Journal of thoracic and cardiovascular surgery*, *107*(5), 1346–1355.

Lough, M., Lindsey, A., Shinn, J., & Stotts, N. (1985). Life satisfaction following heart transplantation. *The Journal of heart transplantation*, *4*(4), 446–449.

Lund, L. H., Khush, K. K., Cherikh, W. S., Goldfarb, S., Kucheryavaya, A. Y., Levvey, B. J., ... others (2017). The registry of the international society for heart and lung transplantation: thirty-fourth adult heart transplantation report2017; focus theme: Allograft ischemic time. *The Journal of Heart and Lung Transplantation*, *36*(10), 1037–1046.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., ... others (2016). Mllib: Machine learning in apache spark. *Journal of Machine Learning Research*, *17*(34), 1–7.

Nilsson, J., Ohlsson, M., Höglund, P., Ekmehag, B., Koul, B., & Andersson, B. (2015). The international heart transplant survival algorithm (ihtsa): A new model to improve organ sharing and survival. *PLoS ONE*, *10*(3), e0118644.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, *31*(1), 71–106.

Powers, D. (2007). Evaluation: From precision, recall and f factor to roc, informedness, markedness & correaltion. *School of Informatics and Engineering, Flinders University of South Australia Adelaide*.

Prud'hommeaux, E., & Seaborne, A. (2008). Sparql query language for rdf.

Purves, D. (2011). *Neuroscience* (5th ed.). Sunderland, Mass.: Sinauer.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85–117.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929-1958. Retrieved from `http://jmlr.org/papers/v15/srivastava14a.html`

Stamborg, M., Medved, D., Exner, P., & Nugues, P. (2012). Using syntactic dependencies to solve coreferences. In *Joint conference on emnlp and conll-shared task* (pp. 64–70).

Steck, H., Krishnapuram, B., Dehing-oberije, C., Lambin, P., & Raykar, V. C. (2008). On ranking in survival analysis: Bounds on the concordance index. In *Advances in neural information processing systems* (pp. 1209–1216).

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, *62*(1), 77–89.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

UNOS. (2018). *Organ procurement and transplantation network data.* Retrieved 2018-07-10, from `https://optn.transplant.hrsa.gov/data/`

Van Asch, V. (2013). *Macro-and micro-averaged evaluation measures* (Tech. Rep.). University of Antwerp.

Wallach, I., Dzamba, M., & Heifets, A. (2015). Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*.

Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37–52.

WWWC. (2014). Rdf 1.1 concepts and abstract syntax. Retrieved from `https://www.w3.org/TR/rdf11-concepts/`

Yan, X., & Su, X. (2009). *Linear regression analysis: theory and computing*. World Scientific.

Zenios, S. A. (1999). Modeling the transplant waiting list: A queueing model with reneging. *Queueing systems*, *31*(3-4), 239–251.

Improving the Detection of Relations Between Objects in an Image Using Textual Semantics

PAPER I

Improving the Detection of Relations Between Objects in an Image Using Textual Semantics

Dennis Medved[1], Fangyuan Jiang[2], Peter Exner[1], Magnus Oskarsson[2], Pierre Nugues[1(✉)], and Kalle Åström[2]

[1] Department of Computer Science, Lund University, Lund, Sweden
{dennis.medved,peter.exner,pierre.nugues}@cs.lth.se
[2] Department of Mathematics, Lund University, Lund, Sweden
{fangyuan,magnuso,kalle}@maths.lth.se

Abstract. In this article, we describe a system that classifies relations between entities extracted from an image. We started from the idea that we could utilize lexical and semantic information from text associated with the image, such as captions or surrounding text, rather than just the geometric and visual characteristics of the entities found in the image.

We collected a corpus of images from Wikipedia together with their corresponding articles. In our experimental setup, we extracted two kinds of entities from the images, human beings and horses, and we defined three relations that could exist between them: *Ride*, *Lead*, or *None*. We used geometric features as a baseline to identify the relations between the entities and we describe the improvements brought by the addition of bag-of-word features and predicate–argument structures that we extracted from the text. The best semantic model resulted in a relative error reduction of more than 18 % over the baseline.

Keywords: Semantic parsing · Relation extraction from images · Machine learning

1 Introduction

A large percentage of queries to retrieve images relate to people and objects [12,20] as well as relations between them: the 'story' within the image [8]. Although the automatic recognition, detection and segmentation of objects in images has reached relatively high levels of accuracy, reflected by the Pascal VOC Challenge evaluation [1,6,10], the identification of relations is still a territory that is yet largely unexplored. Notable exceptions include [2,15]. The identification of these relations would result in a richer model of the image content and would enable users to search images illustrating two or more objects more accurately.

Relations between objects within images are often ambiguous and captions are intended to help us in their interpretation. As human beings, we often have to read the caption or the surrounding text to understand what happened in a

A. Fred et al. (Eds.): ICPRAM 2014, LNCS 9443, pp. 133–145, 2015.
DOI: 10.1007/978-3-319-25530-9_9

scene and the nature of the relations between the entities. This combined use of text and images has been explored in automatic interpretation mostly in the form of bag of words, see Sect. 2. This approach might be inadequate however, as bags of words do not take the word or sentence context into account. This model inadequacy formed the starting idea of this project: As we focused on relations in images, we tried to model their counterparts in the text and reflect them not only with bags of words but also in the form of predicate–argument structures.

2 Related Work

To the best of our knowledge, no work has been done to identify relations in images using a combined analysis of image and text data. There are related works however:

Reference [16] combined image segmentation with a text-based classifier using image captions as input. They used bags of words and applied a $TF \cdot IDF$ weighting on the text. The goal was to label the images as either taken indoor or outdoor. They improved the results by using both text and image information together, compared to using only one of the classifiers.

Reference [3] used a set of 100 image-text pairs from *Yahoo! News* and automatically annotated the images utilizing the associated text. The goal was to detect the presence of specific humans, but also more general objects. They analyzed the image captions to find named entities. They also derived information from discourse segmentation, which was used to determine the saliency of entities.

Reference [14] used a large corpus of French news articles, composed of a text, images, and image captions. They combined an image detector to recognize human faces and logos, with a named entity detection in the text. The goal was to correctly annotate the faces and logos found in the images. The images were not annotated by humans, instead named entities in the captions were used as the ground truth, and the classification was based on the articles.

Reference [19] used a large collection of images from *Flickr* that users had annotated by supplying keywords and short descriptions. The goal was to categorize the images, utilizing a combination of features derived from image analysis, together with relevant image labels extracted from the text associated with the images.

Reference [13] used a semantic network and image labels to integrate prior knowledge of inter-class relationships in the learning step of a classifier to achieve better classification results. All of these works combined text and image analysis for classification purposes, but they did not identify relations in the images. Another area of related work is the generation of natural language descriptions of an image scene, see [7,9].

3 Data Set and Experimental Setup

The internet provides plenty of combined sources of images and text including news articles, blogs, and social media. Wikipedia is one of such sources that, in addition to a large number of articles, is a substantial repository of images

Fig. 1. The upper row shows: a ford mustang, the 3rd light horse regiment hat badge, and a snuff bottle. The lower row shows: a human riding a horse, one human leading the horse and one bystander, and seven riders and two bystanders. Bounding boxes are displayed.

illustrating the articles. As of today, the English version has over 4 million articles and about 2 million images [21]. It is not unusual for editors to use an image for more than one article, and an image can therefore have more than one article or caption associated with it. The images used in the articles are stored in Wikimedia Commons, which is a database of freely reusable media files.

We gathered a subset of images and articles from Wikipedia restricted to two object categories: *Horse* and *Human*. We extracted the articles containing the keywords *Horse* or *Pony* and we selected their associated images. This resulted in 901 images, where 788 could be used. Some images were duplicates and some did not have a valid article associated with them.

An image connected to the articles with the words *Horse* or *Pony* does not necessarily depict a real horse. It can show something associated with the words for example: a car, a statue, or a painting. Some of the images also include humans, either interacting with the horse or just being part of the background, see Fig. 1 for examples. An image can therefore have none or multiple horses, and none or multiple humans.

We manually annotated the horses and humans in the images with a set of possible relations: *Ride*, *Lead*, and *None*. *Ride* and *Lead* are when a human is riding or leading a horse and *None* is an action that is not *Ride* or *Lead* including no action at all. The annotation gave us the number of respective humans and horses, their sizes and their locations in the image.

We processed the articles with a semantic parser [4], where the output for each word is its lemma and part of speech, and for each sentence, the dependency graph and predicate-argument structures it contains. We finally applied a coreference solver to each article.

Table 1. The number of different objects in the source material.

Item	Count
Extracted images	901
Usable images	788
Human-horse pairs	2,235
Relation: *None*	1,935
Relation: *Ride*	233
Relation: *Lead*	67

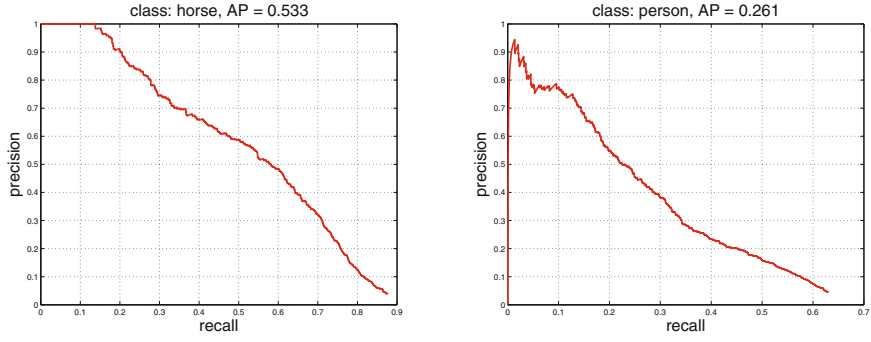


Fig. 2. The precision-recall curves on our image test set using [6]'s detector and generically trained models for the horse (left) and person (right) categories.

4 Visual Parsing

As our focus was to investigate to what extent the use of combinations of text and visual cues could improve the interpretation or categorization precision, we set aside the automatic detection of objects in the images. We manually identified the objects within the images by creating bounding boxes around horses and humans. We then labeled the interaction between the human-horse pair if the interaction corresponded to *Lead* or *Ride*. The *None* relationships were left implicit. It resulted in 2,235 possible human-horse pairs in the images, but the distribution of relations was quite heavily skewed towards the None relation. The Lead relation had significantly fewer examples; see Table 1.

The generation of the bounding boxes could be produced automatically by an object detection algorithm trained on the relevant categories (in our case people and horses) such as e.g. the deformable part-based model described in [6]. Figure 2 shows the precision-recall curve using this detector with generically trained models for the horse (left) and person (right) categories. Such a detection step would have enabled us to skip the manual annotation. Nonetheless, in the experiment we report here, we focused on the semantic aspects and we used manually created bounding boxes.

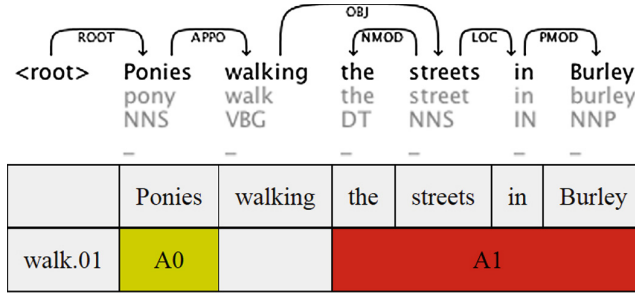


Fig. 3. A representation of a parsed sentence: the upper part shows the syntactic dependency graph and the lower part shows the predicate, *walk*, and its two arguments the parser has extracted: *Ponies* and *the streets in Burley*.

5 Semantic Parsing

We used the Athena parsing framework [4] in conjunction with a coreference solver [18] to parse the Wikipedia articles. For each word, the parser outputs its lemma and part of speech (POS). The lemma is the dictionary form of the word, for example the lemma of *running* is *run*. The POS is the word category. We used the Penn Treebank tag set [11], where, for example, JJ denotes an adjective and NNS, a plural noun. In addition, the parser produces a dependency graph with labeled edges for each sentence, corresponding to grammatical functions, as well as the predicates the sentence contains and their arguments. For each Wikipedia article, we also identify the words or phrases that refer to a same entity i.e. words or phrases that are coreferent.

Figure 3 shows the dependency graph and the predicate–argument structure of the caption: *Ponies walking the streets in Burley*[1].

5.1 Predicates

The predicates correspond to actions or relations such as *jump*, *walk*, or *own*. Each predicate can have one or more senses, where each sense will correspond to a distinct predicate–argument structure. The semantic parser uses the PropBank [17] nomenclature, where the predicate sense is explicitly shown as a number added after the word. The sentence in Fig. 3 contains one predicate: walk.01 with its two arguments A0 and A1, where A0 corresponds to the walker and A1, the path walked.

PropBank predicates can also have modifying arguments denoted with the prefix "AM-". There exist 14 different types of modifiers in PropBank such as:

AM-DIR: shows motion along some path,
AM-LOC: indicates where the action took place, and
AM-TMP: shows when the action took place.

[1] http://en.wikipedia.org/wiki/New_Forest, retrieved November 9, 2012.

5.2 Coreferences

We applied a coreference resolution to create sets of coreferring mentions as with *the rider* and the two *he* in this caption:

> If *the rider* has a refusal at the direct route *he* may jump the other B element without additional penalty than what *he* incurred for the refusal.[2]

The phrase *the rider* is the first mention of an entity in the coreference chain. It usually contains most information in the chain. We use it together with part-of-speech information and we substitute coreferent words with this mention in a document, although this is mostly useful with pronouns. The modified documents can thereafter be used with different lexical features.

6 Feature Extraction

We used classifiers with visual and semantic features to identify the relations. The visual features formed a baseline system. We then added semantic features to investigate the improvement over the baseline.

6.1 Visual Features

The visual parsing annotation provided us with a set of objects within the images and their bounding boxes defined by the coordinates of the center of each box, its width, and height.

To implement the baseline, we derived a larger set of visual features from the bounding boxes, such as the overlapping area, the relative positions, etc., and combinations of them. We ran an automatic generation of feature combinations and we applied a feature selection process to derive our visual feature set. We evaluated the results using cross-validation. However, as the possible number of combinations was very large, we had to discard manually a large part of them. Once stabilized, the baseline feature set remained unchanged while developing and testing lexical features. It contains the following features:

F_Overlap Boolean feature describing whether the two bounding boxes overlap or not.
F_Distance numerical feature containing the normalized length between the centers of the bounding boxes.
F_Direction(8) nominal feature containing the direction of the human relative the horse, discretized into eight directions.
F_Angle numerical feature containing the angle between the centers of the boxes.
F_OverlapArea numerical feature containing the size of the overlapping area of the boxes.

[2] http://en.wikipedia.org/wiki/Eventing, retrieved November 9, 2012.

Table 2. Precision, recall and F_1 for visual features.

	Precision	Recall	F_1
None	0.9472	0.9648	0.9559
Ride	0.7685	0.7553	0.7619
Lead	0.4285	0.2239	0.2941
Mean			0.6706

Table 3. The confusion matrix for visual features.

		Predicted class		
		None	*Ride*	*Lead*
Actual class	*None*	1867	49	19
	Ride	56	176	1
	Lead	48	4	15

F_MinDistanceSide numerical feature containing the minimum distance between the sides of the boxes.
F_AreaDifference numerical feature containing the quotient of the areas.

We used logistic regression and to cope with nonlinearities, we used pairs of features to emulate a quadratic function. The three following features are pairs involving a numerical and a Boolean features, creating a numerical feature. The Boolean feature is used as a step function: if it is false, the output is a constant; if it is true, the output is the value of the numeric feature.

F_Distance+F_LowAngle(7) numerical feature, F_LowAngle is true if the difference in angle is less than 7°.
F_Angle+F_LowAngle(7) numerical feature.
F_Angle+F_BelowDistance(100) numerical feature, F_BelowDistance(100) is true if the distance is less than 100.

Without these feature pairs, the classifier could not correctly identify the *Lead* relation and the F_1 value for it was 0. With these features, F_1 increased to 0.29. Table 2 shows the recall, precision, and F_1 for the three relations using visual features. Table 3, shows the corresponding confusion matrix.

6.2 Semantic Features

We extracted the semantic features from the Wikipedia articles. We implemented a selector to choose the size of the input between: complete articles, partial articles (the paragraph that is the closest to an image), captions, and file names. The most specific information pertaining to an image is found in the caption and the file name, followed by the partial article, and finally, the whole article.

Bag-of-Words Features. A bag-of-word (BoW) feature was created for each of the four different inputs. A BoW feature is represented by a vector of weighted word frequencies. The different versions have separate settings and dictionaries. We also used a combined bag-of-word feature vector consisting of the concatenation of the partial article, caption, and filename feature vectors.

The features have a filter that can exclude words that are either too common, or not common enough, based on their frequency, controlled by a threshold. We used a $TF \cdot IDF$ weighting on the included words.

We used file names as one of the inputs, as it is common to have a long descriptive names of the images in Wikipedia. However, they are not as standardized as the captions. Some images have very long descriptive titles; others were less informative, for example: "DMZ1.jpg". The file names were not semantically parsed, but we defined a heuristic algorithm, which was used to break down the file name strings into individual words.

Predicate Features. Instead of using all of the words in a document, we used information derived from the predicate–argument structure to filter out more relevant terms. We created a feature that only used the predicate names and their arguments as input. The words that are not predicates, or arguments to the predicates, are removed as input to the feature. The arguments can be filtered depending on their type, for example A0, A1, or AM-TMP. We can either consider all of the words of the arguments, or only the heads.

As for the BoW, we created predicate features with articles, partial articles, and captions as input. We never used the file names, because we could not carry a semantic analysis on them. We also created a version of the predicate-based features that we could filter further on the basis of a list of predicate names, including only predicates present in a predefined list, specified by regular expressions.

7 Classification

To classify the relations, we used the LIBLINEAR [5] package and the output probabilities over all the classes. The easiest way to classify a horse-human pair is to take the corresponding probability vector and pick the class with the highest probability. But sometimes the probabilities are almost equal and there is no clear class to chose. We selected a threshold using cross-validation. If the maximum probability in the vector is not higher than the threshold, the pair is classified as *None*. We observed that because *None* represents a collection of actions and nonaction, it is more likely to be the true class when *Ride* and *Lead* have low probabilities.

Even with the threshold, this scheme can classify two or more humans as riding or leading the same horse. Although possible, it is more likely that only one person is riding or leading the horse at a time. Therefore we added constraints to the classification: a horse can only have zero or one rider, and zero or one leader. For each class, only the most probable human is chosen, and only if it is higher than the threshold.

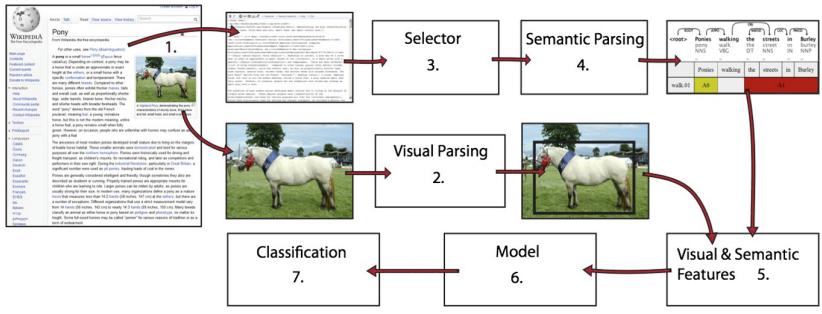


Fig. 4. An overview of the system design, see Sect. 8 for description.

For each human-horse pair, the predicted class is compared to the actual class. The information derived from this can be used to calculate the precision, recall, and F_1 for each class. The arithmetic mean of the three F_1 values is calculated, and can be used as a comparison value. We also computed the number of correct classifications and a confusion matrix.

8 System Architecture

Figure 4 summarizes the architecture of the whole system:

1. Wikipedia is the source of the images and the articles. The text annotation uses the Wiki markup language.
2. Image analysis: placement of bounding boxes, classification of objects and actions. This was done manually, but could be replaced by an automatic system.
3. Text selector between: the whole articles, paragraphs that are the closest to the images, filenames, or captions.
4. Semantic parsing of the text, see Sect. 5.
5. Extraction of feature vectors based on the bounding boxes and the semantic information.
6. Model training using logistic regression from the LIBLINEAR package. This enables us to predict probabilities for the different relations.
7. Relation classification using probabilities and constraints.

9 Results

We used the L2-regularized logistic regression (primal) solver from the LIBLINEAR package and we evaluated the results of the classification with the different feature sets starting from the baseline geometric features and adding lexical features of increasing complexity. We carried out a 5-fold crossvalidation.

We evaluated permutations of features and settings and we report the set of combined BoW features that yielded the best result. Table 4 shows an overview of the results:

Table 4. An overview of the results, with their mean F_1-value, difference and relative error reduction from the baseline mean F_1-value.

		Mean of F_1	Difference (pp)	Relative error reduction (%)
Baseline		0.6706	0.00	0.00
BoW	Articles	0.6779	0.73	2.22
	Partial articles	0.6818	1.12	3.40
	Captions	0.6829	1.23	3.73
	Filenames	0.6802	0.96	2.91
	Combination	0.7132	4.26	12.9
Predicate	Articles	**0.7318**	**6.12**	**18.6**
	Partial articles	0.6933	2.27	6.89
	Captions	0.6791	0.85	2.58
	Articles + Words	0.6830	1.24	3.76
	Articles + Coref	0.7280	5.74	17.4

- The baseline corresponds to the geometrical features; we obtained a mean F_1 of 0.67 with them;
- BoW corresponds to the baseline features and the bag-of-word features described in Sect. 6.2; whatever the type of text we used as input, we observed an improvement. We obtained the best results with a concatenation of the partial article, caption, and filename (combination, $F_1 = 0.71$);
- predicate corresponds to the baseline features and the predicate feature vector described in Sect. 6.2. Predicate features using only one lexical feature vector from the article text gave better results than combining different portions of the text ($F_1 = 0.73$).

Our best feature set is the predicate features utilizing whole articles as input. It achieves a relative error reduction of 18.6 percent compared to baseline.

Tables 2 and 3 show the detailed results of the baseline with the geometric features only. Tables 5 and 6 show the results of the best BoW feature combination: a concatenation of the feature vectors from the inputs: partial articles, captions, and filenames. Tables 7 and 8 show the result of the best predicate features.

Table 5. Precision, recall, and F_1 for the concatenation of BoW features with the inputs: partial articles, captions and filenames.

	Precision	Recall	F_1
None	0.9638	0.9638	0.9638
Ride	0.7642	0.8626	0.8104
Lead	0.5135	0.2835	0.3653
Mean			0.7132

Table 6. The confusion matrix for BoW for the concatenation of BoW features with the inputs: partial articles, captions and filenames.

		Predicted class		
		None	*Ride*	*Lead*
Actual class	*None*	1865	57	13
	Ride	27	201	5
	Lead	43	5	19

Table 7. Precision, recall and F_1 for predicate feature on articles.

	Precision	Recall	F_1
None	0.9745	0.9498	0.9620
Ride	0.7301	0.9055	0.8084
Lead	0.4500	0.4029	0.4251
Mean			0.7318

Table 8. The confusion matrix for predicate feature on articles.

		Predicted class		
		None	*Ride*	*Lead*
Actual class	*None*	1838	70	27
	Ride	16	211	6
	Lead	32	8	27

10 Discussion

Classifying the *Lead* relation with geometric features with only bounding boxes as the input revealed quite difficult. There is indeed very little visual difference between standing next to a horse and leading it. We were not able to classify any *Lead* correctly until we added the combination features.

For single BoW features, the captions gave the best result, followed by partial articles, filenames, and lastly articles. The order of the results was what we expected, based on how specific information the features had about the images. But for the predicate features, the order was reversed: articles produced the best result, followed by partial articles, and captions.

Using a specific list of predicates did not produce good results although, depending on the list, results vary greatly. Using a list with the words: *ride*, *lead*, *pull*, and *race*, with articles as input, gave the best result, but Table 4 shows a relative drop of 4.88 compared to no filtering. The negative results could possibly be explained by the fact that it is not common to explicitly describe the relations in the images, and only utilizing keywords such as *ride* is of little help.

Applying coreference resolution on the documents lowered the results. Table 4 shows a relative drop of 0.38 if applied on the predicate feature based on articles.

Despite these negative results, we still believe that solving coreferences could improve the results. The solver was designed to be used with another set of semantic information. To be able to use the solver, we altered its source code and possibly made it less accurate. We checked manually coreference chains and we could observe a significant number of faulty examples, leading us to believe that the output quality of the solver left to be desired.

11 Conclusions and Future Work

We designed a supervised classifier to identify relations between pairs of objects in an image. As input to the classifier, we used geometric, bag-of-words, and semantic features. The results we obtained show that semantic information, in combination with geometric features, proved useful to improve the classification of relations in the images. Table 4 shows that the relative error reduction is 12.9 percent by utilizing a combination of bag-of-words features. An even greater improvement is made using predicate information with an relative error reduction of 18.6 percent compared to baseline.

Coreference resolution lowered the performance, but the interface between the semantic parser and the coreference solver was less than optimal. There is room for improvement regarding this solver, either with the interface to the semantic parser or with to another solver. It could also be interesting to try other types of classifiers, not just logistic regression, and see how they perform.

Using automatically annotated images as input to the program could be relatively easily implemented and would automate all the steps in the system. A natural continuation of the work is to expand the number of objects and relations. [6], for example, use 20 different classifiers for common objects: cars, bottles, birds, etc. All, or a subset of it, could be chosen as the objects, together with some common predicates between the objects as the relations.

It would also be interesting to try other sources of images and text than Wikipedia: either using other resources available online or creating a new database with images captioned with text descriptions. Another interesting expansion of the work would be to map entities found in the text with objects found in the image. For example, if a caption includes the name of a person, one could create a link between the image and information about the entity.

Acknowledgements. This research was supported by Vetenskapsrådet, the Swedish research council, under grant 621-2010-4800 and *Det digitaliserade samhället* thematic grant, and the Swedish e-science program: eSSENCE.

References

1. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: IEEE International Conference on Computer Vision and Pattern Recognition (2010)
2. Chen, N., Zhou, Q.-Y., and Prasanna, V.: Understanding web images by object relation network. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012, pp. 291–300. ACM, New York (2012)

3. Deschacht, K., Moens, M.-F.: Text analysis for automatic image annotation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 1000–1007, Prague (2007)
4. Exner, P., Nugues, P.: Constructing large proposition databases. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012), Istanbul (2012)
5. Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
7. Gupta, A., Verma, Y., Jawahar, C.: Choosing linguistics over vision to describe images. In : Proceeding of the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
8. Jörgensen, C.: Attributes of images in describing tasks. Inf. Process. Manage. **34** (2–3), 161–174 (1998)
9. Kulkarni, G., Premraj, V., Dhar, S., Siming, L., Choi, Y., Berg, A., Berg, T.: Baby talk: understanding and generating image descriptions. In: Proceedings of Conference Computer Vision and Pattern Recognition (2011)
10. Torr, P.H.S., Torr, P.H.S., Ladicky, L., Ladicky, L., Kohli, P., Kohli, P., Russell, C., Russell, C.: Graph cut based inference with co-occurrence statistics. In: Maragos, P., Maragos, P., Daniilidis, K., Daniilidis, K., Paragios, N., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 239–253. Springer, Heidelberg (2010)
11. Marcus, M., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the penn treebank. Comput. Linguis. **19**(2), 313–330 (1993)
12. Markkula, M., Sormunen, E.: End-user searching challenges indexing practices in the digital newspaper photo archive. Inf. Retrieval **1**(4), 259–285 (2000)
13. Marszalek, M., Schmid, C.: Semantic hierarchies for visual object recognition. In: Proceeding of Conference Computer Vision and Pattern Recognition (2007)
14. Moscato, V., Picariello, A., Persia, F., Penta, A.: A system for automatic image categorization. In: IEEE International Conference on Semantic Computing, ICSC 2009, pp. 624–629. IEEE (2009)
15. Myeong, H., Chang, J.Y., Lee, K.M.: Learning object relationships via graph-based context model. In: CVPR, pp. 2727–2734 (2012)
16. Paek, S., Sable, C., Hatzivassiloglou, V., Jaimes, A., Schiffman, B., Chang, S., Mckeown, K.: Integration of visual and text-based approaches for the content labeling and classification of photographs. In: ACM SIGIR, vol. 99 (1999)
17. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. Comput. Linguis. **31**(1), 71–105 (2005)
18. Stamborg, M., Medved, D., Exner, P., Nugues, P.: Using syntactic dependencies to solve coreferences. In: Joint Conference on EMNLP and CoNLL - Shared Task, pp. 64–70. Association for Computational Linguistics, Jeju Island, Korea (2012)
19. Tirilly, P., Claveau, V., Gros, P., et al.: News image annotation on a large parallel text-image corpus. In: 7th Language Resources and Evaluation Conference, LREC 2010 (2010)
20. Westman, S., Oittinen, P.: Image retrieval by end-users and intermediaries in a journalistic work context. In: Proceedings of the 1st International Conference on Information Interaction in context, pp. 102–110. ACM (2006)
21. Wikipedia. Wikipedia statistics English (2012). http://stats.wikimedia.org/EN/TablesWikipediaEN.htm

Using a RDF Triplestore and Deep Learning to Predict Heart Transplantation Survival

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

SEMANTiCS 2018 – 14th International Conference on Semantic Systems

Using a RDF Triplestore and Deep Learning to Predict Heart Transplantation Survival

Dennis Medved[a,*], Johan Nilsson[b], Pierre Nugues[a]

[a]*Department of Computer Science, Lund University, Lund, Sweden*
[b]*Department of Clinical Sciences, Cardiothoracic Surgery, Lund University and Skåne University Hospital, Lund, Sweden*

Abstract

In this paper, we describe the conversion of three different heart transplantation data sets to an RDF representation and how it can be utilized to train deep learning models. These models were used to predict the outcome of patients both pre- and post-transplant and to calculate their survival time.

The International Society for Heart & Lung Transplantation (ISHLT) maintains a registry of heart transplantations that it gathers from grafts performed worldwide. The American organization United Network for Organ Sharing (UNOS) and the Scandinavian Scandiatransplant are contributors to this registry, although they use different data models.

We designed a unified graph representation covering these three data sets and we converted the databases into RDF triples. We used the resulting triplestore as input to several machine learning models trained to predict different aspects of heart transplantation patients.

Recipient and donor properties are essential to predict the outcome of heart transplantation patients. In contrast with the manual techniques we used to extract data from the tabulated files, the RDF triplestore together with SPARQL, enables us to experiment quickly and automatically with different combinations of features sets, to predict the survival.

Keywords: Deep learning; heart transplantation; RDF; SPARQL

1. Introduction

Heart transplantations are life saving procedures that made it possible to extend the median survival time to 12 years for patients with end-stage heart diseases. Unfortunately, patients have to wait a relatively long time before being transplanted, because of a limited donor supply that forces the surgeons to prioritize the recipients.

* Corresponding author. Tel.: +46-46-222-71-00;
E-mail address: dennis.medved@cs.lth.se

The understanding of factors that predict mortality could help the doctors with the prioritization task and improve the post-operation care. With an improved outcome prediction, surgeons could be more confident in the transplantation performance. In addition, a better allocation of organs would make it possible to increase the survival as well as the number of organs that can be used.

The availability of medical databases which have been created during the last two decades, and the application of machine-learning methods, such as deep learning, have led to the development of advanced models of survival prediction.

Patient and donor factors are essential to predict the mortality of heart transplantations [10, 8]. Domingos [3] provides an eloquent advocacy of the importance of such factors, or features, in the success of machine-learning projects.

We wanted to mine the feature sets from the patient variables and integrate data from all our sources. We designed a unified, extendable, RDF representation of the variables. Our goal was to make the data extraction easier, using the different registries that were available to us, and simplify the feature engineering for the machine learning models.

The usage of RDF to store the patient data, helped us streamline the development process of such survival models.

2. Medical Registries

The International Society for Heart & Lung Transplantation (ISHLT) maintains a registry of heart transplantations it collects from national or regional organizations across the world. ISHLT aggregates the data submitted by the contributing organizations. The American organization United Network for Organ Sharing (UNOS) and the Scandinavian Scandiatransplant are two such contributing institutions. In total, ISHLT contains about 100,000 recorded heart transplantations.

Although ISHLT could be seen as a superset of all the included databases, in regards to the patients recorded, but it only contains a subset of the variables that is contained within the different registries. ISHLT is restricted to variables that are frequently recorded by the different regional registries.

The three data sources we considered: ISHLT, UNOS, and Scandiatransplant, have different structures, a different number of variables, use different variable names, and may use different units or encoding of the data.

The variables contained in these databases pertain to both recipient, donor and the operation itself. It can for example be the age, weight, gender, or blood group of the patients.

UNOS contains the largest number of variables, about 500. ISHLT, for example, does not feature the variable *crossmatch_done*, a patient compatibility test, that is available in UNOS.

3. Representing the Data in RDF

The ISHLT, UNOS, and Scandiatransplant data sets are normally distributed to the researchers as SAS or CSV files. We started from the CSV files and we converted them to an RDF format.

The CSV files represent the transplants as rows, where each column is a variable for the transplant. In the RDF conversion, we mapped each row to a head node and we created leaf nodes for the selected variables.

The data sets use different names to denote the same variables. For example, the most recent blood creatinine value for the recipient patient is *Most rec. Creat.* in Scandiatransplant, *creat* in ISHLT, and *creat_trr* in UNOS, see Figure 1.

We created unified names for about 140 of the variables, such as *aaot:creatinine* for the creatinine value, where the *aaot* prefix stands for *Algorithms and Applications for Organ Transplantation*.

We had to encode the data in a unified way between the databases, for example binary variables were both recorded as Y/N and 1/0, and categorical variables often used different codes to encode the data between the registries.

As previously mentioned, UNOS has more variables than ISHLT and Scandiatransplant. We used the UNOS variable names, when they had no counterpart in the other two registries.

We also added metadata about the variables containing the original variable name, as well the new one, the description of the variable, the source form of the data, the unit where it is applicable, as well as comments, and start and possibly end date of the recording of the variable. Figure 2 shows the metadata on *aaot:creatinine*.

Fig. 1. An unification of the variable representing the most recent creatinine level of the recipient.

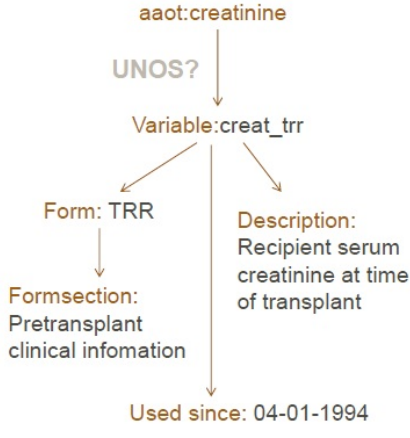


Fig. 2. *aaot:creatinine* metadata for the UNOS part of the database.

4. Querying the Database

We created a SPARQL endpoint to be able to query the data. Compared with the tedious copy-and-paste techniques we used to previously create the data sets and to test our survival prediction programs, SPARQL offers an easier way to extract relevant data samples. The extraction of the survival duration for transplants matching the conditions:

- The recipient is a male older than 17 with blood group A;
- The donor is female with blood group A;
- From Scandiatransplant and ISHLT registries.

is concisely expressed using the SPARQL query:

```
SELECT ?transplant ?survival_time
```

```
FROM <file://Scandia.ttl>
FROM <file://ISHLT.ttl>
WHERE {
    ?transplant aaot:gender "M" .
    ?transplant aaot:age ?age .
    ?transplant aaot:ABO "A" .
    ?transplant aaot:gender_donor "F" .
    ?transplant aaot:ABO_donor "A" .
    ?transplant aaot:survival_time ?survival_time .
    FILTER (?age > 17)
}
```

Although the RDF database can be used to do statistics and exploratory analysis, our major use of the database is as input to the machine learning algorithms. We want to answer questions such as: What variables are important for heart transplantation survival and how do they affect the outcome?

5. Deep Learning Models

Artificial neural networks are models inspired by the human brain that approximate functions used in machine learning, such as classification or regression. It consists of a network of neurons that emulate the properties of their real counterparts.

The neurons propagate signals depending on the weight of their connections. These connection strengths are tuned during the training step from observations when the network learns what it should output for a certain set of inputs.

A feed forward network consists of three or more layers. The first layer is called the input layer, where the features are used as the initial input. The middle layers which can be one or more, are called hidden layers. Finally, the last layer, the output layer, which has as many nodes as the wanted amount of outputs from the model. A neural network with two or more hidden layers is usually referred to as a deep learning model. Figure 3 illustrates a network with four layers, in which the layers are fully connected.

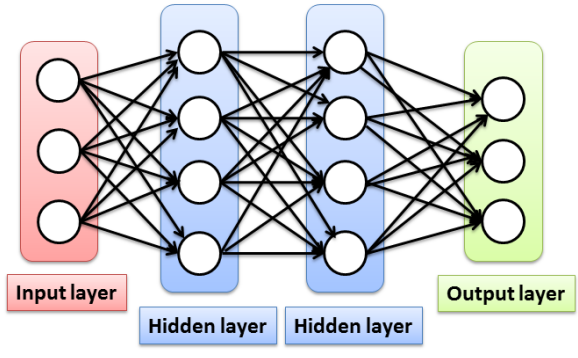


Fig. 3. The topology of a fully connected neural network with three input nodes, two hidden layers with four hidden nodes, and three output nodes.

These models have many practical applications, for example in computer vision, spam filtering, or medicine [1]. They have shown a superior predictive ability over more conventional models such as risk scores created by classical statistical methods [2].

6. Applications

We have used the RDF representation of the patients in Sect. 3 to create different survival models for heart transplantation.

1. We first carried out an analysis on the features that had the largest impact on the post-transplant survival of the patient and to find locally optimal feature sets for different survival time periods [4].
2. Patients enter a waiting queue before they are transplanted and they may die in this queue if no appropriate organ becomes available for transplant. We designed a model to predict the outcome of patients awaiting heart transplant and we explored which features were the most predictive in assessing the result for the patients [5].
3. After a patient is being transplanted, the registries record his/her survival time. We trained a post-transplant model based on neural networks and we evaluated its performance against a more simple, point based model [9, 7]. We used data from UNOS instead of ISHLT in this experiment. This model is available via a web application (ihtsa.cs.lth.se), where a user can input a patient's data and the server returns the predicted survival. The application shows the survival prediction as a probability curve depending on the years after transplant.
4. And finally, we trained a pre- and post-transplant algorithm and we used it together with a discrete simulation model, to simulate a queue system for heart transplantation. This algorithm, the Lund Deep Learning Transplant Algorithm (LuDeLTA), enables analysts to evaluate the impact of different allocation policies on patient survival [6].

7. Conclusion

The creation of the RDF representation has simplified the use of the three registries. It enabled us to utilize a unified interface to query the data using SPARQL, which made it easier to handle the patient variables.

We have successfully created several deep learning models using this patient data. Prediction using these models have produced results that were comparable to state-of-the-art systems.

Acknowledgements

This work is based on OPTN data as of October 1, 2013 and was supported in part by the Health Resources and Services Administration contract 234-2005-370011C. The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government. This research was supported by Heart Lung Foundation, The Swedish Research Council, and the eSSENCE program.

References

[1] Baxt, W.G., 1995. Application of artificial neural networks to clinical medicine. The lancet 346, 1135–1138.

[2] Cucchetti, A., Vivarelli, M., Heaton, N.D., Phillips, S., Piscaglia, F., Bolondi, L., La Barba, G., Foxton, M.R., Rela, M., O'Grady, J., 2007. Artificial neural network is superior to meld in predicting mortality of patients with end-stage liver disease. Gut 56, 253–258.

[3] Domingos, P., 2012. A few useful things to know about machine learning. Communications of the ACM 55, 78–87. URL: http://doi.acm.org/10.1145/2347736.2347755, doi:10.1145/2347736.2347755.

[4] Medved, D., Nugues, P., Nilsson, J., 2016. Selection of an optimal feature set to predict heart transplantation outcomes, in: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 3290–3293.

[5] Medved, D., Nugues, P., Nilsson, J., 2017. Predicting the outcome for patients in a heart transplantation queue using deep learning, in: Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE, IEEE. pp. 74–77.

[6] Medved, D., Nugues, P., Nilsson, J., 2018a. Simulating the outcome of heart allocation policies using deep neural networks, in: Engineering in Medicine and Biology Society (EMBC), 2018 40th Annual International Conference of the IEEE, IEEE. p. to appear.

[7] Medved, D., Ohlsson, M., Höglund, P., Andersson, B., Nugues, P., Nilsson, J., 2018b. Improving prediction of heart transplantation outcome using deep learning techniques. Scientific reports 8, 3613.

[8] Nilsson, J., Ohlsson, M., Höglund, P., Ekmehag, B., Koul, B., Andersson, B., 2011. Artificial neural networks - relative importance of different recipient-donor characteristic combinations on survival after heart transplantation. The Journal of Heart and Lung Transplantation 30, S68.

[9] Nilsson, J., Ohlsson, M., Höglund, P., Ekmehag, B., Koul, B., Andersson, B., 2015. The international heart transplant survival algorithm (ihtsa): a new model to improve organ sharing and survival. PloS one 10, e0118644.

[10] Weiss, E.S., Allen, J.G., Arnaoutakis, G.J., George, T.J., Russell, S.D., Shah, A.S., Conte, J.V., 2011. Creation of a quantitative recipient risk index for mortality prediction after cardiac transplantation (impact). The Annals of Thoracic Surgery 92, 914 – 922. URL: http://www.sciencedirect.com/science/article/pii/S0003497511009350, doi:10.1016/j.athoracsur.2011.04.030.

Selection of an optimal feature set to predict heart transplantation outcomes

Selection of an Optimal Feature Set to Predict Heart Transplantation Outcomes

Dennis Medved[1], Pierre Nugues[1], and Johan Nilsson[2]

Abstract— **Heart transplantation (HT) is a life saving procedure, but a limited donor supply forces the surgeons to prioritize the recipients. The understanding of factors that predict mortality could help the doctors with this task. The objective of this study is to find locally optimal feature sets to predict survival of HT patients for different time periods. To this end, we applied logistic regression together with a greedy forward and backward search. As data source, we used the United Network for Organ Sharing (UNOS) registry, where we extracted adult patients (>17 years) from January 1997 to December 2008. As methods to predict survival, we used the Index for Mortality Prediction After Cardiac Transplantation (IMPACT) and the International Heart Transplant Survival Algorithm (IHTSA). We used the LIBLINEAR library together with the Apache Spark cluster computing framework to carry out the computation and we found feature sets for 1, 5, and 10 year survival for which we obtained area under the ROC curves (AUROC) of 68%, 68%, and 76%, respectively.**

I. Introduction

Heart transplantation (HT) has been the gold standard for treating patients with end-stage heart disease. Unfortunately, this operation can not be offered to all the potential patients, because of a limited donor supply. This organ scarcity makes the allocation of donated hearts a tricky task [5]. One criterion to consider for the potential recipients is their expected survival after transplantation. In this paper, we describe a procedure to extract features predicting the one, five, and ten year survival of patients. Such features aim at providing the doctors with more information to support their decision for the allocation of organs.

II. Materials and Methods

A. Data Source

The data set used was obtained from the UNOS database containing HT patients. UNOS is a non-profit organization that administers the only Organ Procurement and Transplantation Network (OPTN) in the United States of America [8]. The database contains data from October 1, 1987 onwards and includes almost 500 variables that encompass recipient, donor, and transplant information. The Ethics Committee for Clinical Research at Lund University, Sweden approved the study protocol. The data was de-identified prior to analyzing it and the institutional review board waived the need for written informed consent from the participants.

*This research was supported by Heart Lung Fondation, The Swedish Research Council, and the eSSENCE program.

[1]Department of Computer Science, Lund University, Lund, Sweden {dennis.medved, pierre.nugues}@cs.lth.se

[2]Department of Clinical Sciences Lund, Cardiothoracic Surgery, Lund University and Skåne University Hospital, Lund, Sweden johan.nilsson@med.lu.se

B. Study population

We applied the same criteria for inclusion in the data set as in [12], which means we examined all the primary, adult (> 17 years) HT patients from January 1997 to December 2008. We used this data set to create three different cohorts, where the patients were not censored after 1, 5, and 10 years, respectively. "Not censored" means that the patients follow-up time either was over the number of years chosen or that the patient died before that time. Each of these three cohorts was further divided into two subcohorts: one derivation cohort or training set containing 4/5 of the patients and one validation set containing 1/5 of the patients; see Table I for a summary.

TABLE I
THE THREE COHORTS AND THEIR DIVISION INTO TRAINING AND VALIDATION SUBCOHORTS.

Years	Total patients	Train. set	Validation	Dead (%)
1	22,195	17,756	4,439	13
5	17,101	13,681	3,420	32
10	11,134	8,908	2,226	68

The first cohort has a total of 22,195 patients, where 13% died before the end of the first year after the operation; the second one has 17,756 patients, where 32% died before five years; and finally, the third cohort has 11,134 patients, where 68% died before 10 years after the operation.

C. Imputation of missing data

As with all large patient registries, there is missing data. Excluding the patients with missing data fields from the cohorts would have reduced the data set to almost nothing, because no patient has a complete information record. For this reason, we chose to impute the missing data, where we utilized a probabilistic approach. We followed [10] and we imputed the missing values in a particular variable by choosing a random value from a discrete uniform distribution of the non-missing values in that variable.

D. Feature Extraction

We considered 482 features, such as the age, gender, and blood group. For each patient, we extracted them from UNOS and we converted categorical features into binary features, utilizing one binary feature for each category value. As a result, each feature pertains to one of two categories: binary, either 0 or 1, or real valued.

We tried two different sets of features: The 482 features previously mentioned and a superset consisting of 2,462 features. We created this larger set by computing the Cartesian product of every real-valued feature, excluding the product of the feature by itself. As the original set has 45 real-valued features, we obtained a total of $482 + 45^2 - 45 = 2462$ features. We decided to restrict the interaction terms to the real-valued features to keep the number of features down to a manageable level.

E. Scaling and Normalizing

We first scaled the real-valued features between 0 and 1 using Eq. 1. We then used the Euclidean norm and Eq. 2 to normalize the feature vectors of the individual instances to 1.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

$$\hat{u} = \frac{u}{\|u\|} \overset{using\ell^2}{=} \frac{u}{\sqrt{x_1^2 + \cdots + x_n^2}} \tag{2}$$

F. Forward and Backward Search

To find a globally optimal feature set requires 2^n tests, where n is the number of features. This is infeasible even for a moderate number of features: Our 482 features would require $2^{482} \approx 1.25 \times 10^{145}$ tests. Instead, we applied a greedy forward selection and a greedy backward elimination that enabled us to find a locally optimal subset. The greedy forward selection starts from a subset of the features, which can be empty, and adds one feature from the remaining set to the current subset. The selection procedure uses the new subset to produce the classification probabilities: The probability that the patient is dead after a certain time period. These probabilities are then used to calculate an evaluation metric, see Sect. II-G. The feature which produces the best figure is then added it to the current feature set for the next generation. The procedure is repeated if it improves the score of the preceding subset over a certain threshold Δ, set to $\Delta = 0.0001$. If there is no improvement, we use the current feature set for a backward elimination instead.

The backward elimination removes the features one by one from the starting set and the resulting feature set is used to produce the classification probabilities. The same metric is utilized to create a score. If the score improves on the preceding generation by a threshold Δ then the process is repeated with the new starting feature set. If two following forward selections and backward eliminations do not improve the score, the process is stopped and the resulting feature set corresponds to a local optimum.

Eq. 3 gives the number of tests done in one forward selection, where n is the number of features, x is the number of features in the starting set, and y is the number of features found by the forward selection. Eq. 4 gives a worst case for 482 features, where we start from the empty set and every feature is found to improve the score: $n = 482$, $x = 0$, and $y = 482$. This results is 116,403 tests, a number somewhat smaller than 1.25×10^{145}.

$$\sum_{i=0}^{n-x}(n-i) - \sum_{i=0}^{n-x-y}(n-i) \tag{3}$$

$$\sum_{i=0}^{482-0}(n-i) - \sum_{i=0}^{482-0-482}(n-i) = \sum_{i=0}^{482}(n-i) = 116,403 \tag{4}$$

Eq. 5 gives the number of tests for one backward selection, where z is the number of features in the current feature set and y is the number of features removed until the score is not improved anymore. The worst case for 482 features is when you start from all the features in the current set and every feature is eliminated, that is $z = 482$ and $y = 482$. The equation can be simplified to the same number as in Eq. 4, which is equal to 116,403 tests.

$$\sum_{i=0}^{z}(n-i) - \sum_{i=0}^{z-y}(n-i) \tag{5}$$

G. Evaluation

1) F1 score: The F1 score is the harmonic mean of precision and recall, see Eq. 6 and is bounded in the interval [0.0,1.0] [9]. This score tends to be close to the minimum of both the precision and recall.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6}$$

2) Receiver Operating Characteristic: A receiver operating characteristic (ROC) graph is a technique for visualizing, organizing, and selecting classifiers based on their performance. ROC graphs are two-dimensional graphs in which the true positive rate (TPR) is plotted on the y axis and the false positive rate (FPR) is plotted on the x axis. AUROC describes the performance of a classifier using a single scalar value. Because both TPR and FPR are bounded in the interval [0.0,1.0], the area is also bounded between [0.0,1.0] [3].

A classifier that outputs a random label should have an AUROC value of 0.5, and therefore no functional classifier should have a lower value than that. The AUROC has the statistical property that it is equal to the probability that a randomly chosen negative example is ranked lower than a randomly chosen positive example.

H. Implementation Details

We implemented the program in Java utilizing the LIBLINEAR library [2], Roc library [1], Sesame library [11], and the Spark framework [4]. We chose not to use the linear regression implementation available in the Spark library MLlib, because it is designed for larger data sets and the overhead is larger than using LIBLINEAR.

The first step in the program is to load the data from a RDF database [6] and handle missing values. We carry this out using the SPARQL query language and its OPTIONAL clause. We then impute the missing values using a discrete uniform random distribution derived from the non-missing values. The data is then saved as a CSV file for a faster subsequent access.

TABLE II

THE BEST VALIDATION SET AUROC VALUES FOR 1, 5, AND 10 YEARS, FOUND USING A SEARCH WITH 482 POSSIBLE FEATURES.

Years	AUROC Train set	AUROC Val. set	Precision Val. set	Recall Val. set	F1 Val. set
1	0.6990	0.6835	0.7174	0.0520	0.0969
5	0.6892	0.6795	0.6594	0.2952	0.4078
10	0.7509	0.7626	0.7597	0.8373	0.7966

TABLE III

THE BEST VALIDATION SET AUROC VALUES FOR 1, 5, AND 10 YEARS, FOUND USING A SEARCH WITH 2,462 POSSIBLE FEATURES.

Years	AUROC Train set	AUROC Val. set	Precision Val. set	Recall Val. set	F1 Val. set
1	0.6973	0.6765	0.6591	0.0457	0.0854
5	0.6935	0.6711	0.6646	0.2934	0.4071
10	0.7754	0.7521	0.7462	0.8485	0.7941

We define the starting features using an array of integers, which is empty when starting the search from scratch. Then a search method is called, which implements the forward and backward search described in Sect. II-F. A single search (forward or backward) consists of a parallel Spark method which distributes the tasks to the different nodes in the computer cluster. The tasks apply a function to the data set; the function itself uses LIBLINEAR to train a model and evaluate the data on that model. The evaluation metrics are then calculated by either the Roc library or by a method that produces the F1 metric. The feature with the highest improvement of the metric is then added or removed from the feature set if it is an improvement over the threshold Δ. The search stops if the following forward and backward searches fail to increase the figures. The final feature set then represents a locally optimal set.

III. RESULTS

Table II shows the best validation AUROC values for 1, 5, and 10 years, respectively, found using a search with the set of 482 possible features, and Table III, the set of 2,462 possible features.

The configuration for a forward and backward search included the following parameters: the number of years of survival: 1, 5, or 10 (years), the validation cost which is the regularization cost used in the LIBLINEAR solver, used only for the evaluation and not the training (cost), the evaluation metric to optimize on (metric), number of cross validations to do (cross), and the set of starting features (start features). Where "number of cross validations" mean that the k-fold cross validation is done k times, with different partitions, and the metric, that is used for evaluation, is averaged between the number of cross validations. See Table V for the best configuration, for each year, corresponding to the results as in Table II.

For each time period, we started from the empty set, did a forward search, and we recorded the first ten features added, i.e. the ten most influential features; see Table IV.

TABLE V

THE BEST CONFIGURATIONS FOR VALIDATION SET AUROC VALUES.

Years	Cost	Metric	Cross	Start Features
1	75	AUROC	2	Subset
5	50	AUROC	2	$\emptyset$
10	75	F1	2	All

TABLE VI

THE AUROC VALUES FOR THE WHOLE DATASET USING IMPACT AND IHTSA, RESPECTIVELY.

Years	IMPACT (AUROC)	IHTSA (AUROC)
1	0.6064	0.6593
5	0.5690	0.6033
10	0.5592	0.5423

IV. COMPARISON TO IMPACT AND IHTSA

We compared our results with two other methods to predict the survival of HT patients: IMPACT [12] and IHTSA [7].

IMPACT was trained on the same data set from UNOS as this study, see Sect. II-A, to predict one-year survival. It only uses recipient-specific features. Even though it was not designed for it, this score can also be used as a predictor for five and ten year mortality. Table VI shows AUROC values on the whole data set using IMPACT and IHTSA. These AUROC values are not directly comparable to the results in Table II, because we only used 20% of the set for our validation, but they should have similar distributions of the feature values as we picked our validation set as a uniformly random subset of the whole data set.

The IMPACT AUROC value is calculated on the union of the training and validation set and therefore should have a somewhat higher score than evaluated on a validation set only. In spite of this, the IMPACT AUROC score was still, in absolute difference, about 8% lower for one-year survival, 11% lower for five years, and 21% lower for ten years. (The exact content of the IMPACT validation set was unknown to us.)

IHTSA was trained using a nonlinear artificial neural network (ANN) on another data set: The International Society for Heart & Lung Transplantation (ISHLT). The AUROC value for one year survival is about, in absolute difference, 3%, 8% lower for five year survival and 22% lower for ten year survival II.

V. DISCUSSION

Table II shows validation AUROC values that are about the same for 1 and 5 years, but approximately 8 percentage points higher for 10 years. This is somewhat unintuitive and we tried without success to find confounding factors to explain these results. A possible explanation is that there are much more positive examples, i.e. dead patients, for 10 years compared with 1 and 5 years, see Table I. Another bias is that many patients of this cohort are censored compared to 1 and 5 years: Only about 50% are not censored after 10 years.

TABLE IV
THE TEN FIRST FEATURES ADDED FOR 1, 5, AND 10 YEARS.

Order	1 year	5 years	10 years
1	Anti viral	Ethnicity: white	Days in status: 1
2	Creatinine	Creatinine clearance	Days in status: 2
3	Height	Functional status: very sick	Days in status: 1b
4	Donor age	Donor age	Donor coronary angiogram: No
5	Ventricular assist	Ventricular assist	Functional status: very sick
6	Ventricular assist type: None	Donor ischemic time	Research Immunosuppressive medication
7	Serum bilirubin	Functional status: cares for self	Functional status: cares for self
8	Donor ischemic time	Functional status: occasional assistance	Diabetes
9	Other therapies	Functional status: normal activity with effort	Anti viral
10	Dialysis	Functional status: considerable assistance	Functional status: considerable assistance

The ten most influential features for 5 and 10 years do not seem to correspond to the features found in previously published articles and should be taken with some care.

The cause of the inferior results obtained by the superset of 2,462 possible features compared with the 482 original features is probably due to the interaction features that create an overfit to the data.

The lower performance of IMPACT for 5 and 10 year can be explained by the fact that it was not designed for the prediction of these time periods, and for the 1-year case, a part of the explanation could be that the model did not include any donor features. For IHTSA, the lower scores can be explained by the different sources of the training set, and that the model is not optimized for a specific time period. ISHLT has more patients, but a lot less potential features for each patient. This hints at potential better figures for IHTSA if its ANN was trained on the UNOS feature set.

Future work to improve the results could include the evaluation of other machine learning algorithms, for example support vector machines (SVM) or random forests, or maybe combine several models into an ensemble. Nonetheless, a learning algorithm such as SVM can be a lot slower to train, which means that it can be prohibitively computationally complex to do a forward and backward search on all the features. We could also try to use all the interaction or polynomial combinations of the features, which for the interaction features means a product of feature x $\times$ feature y, that is equal to $482^2 = 232324$ potential features. This could take over a month to run on the cluster that we are currently using, if it can handle the memory requirements.

ACKNOWLEDGMENT

This work is based on OPTN data as of January 1, 2012 and was supported in part by the Health Resources and Services Administration contract 234-2005-370011C. The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government. This research was supported by Heart Lung Foundation, The Swedish Research Council, and the eSSENCE program.

REFERENCES

[1] Kendrick Boyd. *All About Roc*. 2012. URL: http://kboyd.github.io/Roc/ (visited on 03/06/2015).

[2] Rong-En Fan et al. "LIBLINEAR: A Library for Large Linear Classification". In: *Journal of Machine Learning Research* 9 (2008), pp. 1871–1874.

[3] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861 –874.

[4] The Apache Software Foundation. *Apache Spark*. 2015. URL: https://spark.apache.org/ (visited on 04/07/2015).

[5] AS Klein et al. "Organ donation and utilization in the United States, 1999–2008". In: *American Journal of Transplantation* 10.4p2 (2010), pp. 973–986.

[6] Dennis Medved, Johan Nilsson, and Pierre Nugues. "Streamlining a Transplantation Survival Prediction Program with a RDF Triplestore". In: *9th International Conference on Data Integration in the Life Sciences*. 2013.

[7] Johan Nilsson et al. "The International Heart Transplant Survival Algorithm (IHTSA): a new model to improve organ sharing and survival". In: *PloS one* 10.3 (2015), e0118644.

[8] United Network for Organ Sharing. *Organ Procurement and Transplantation Network Data*. 2015. URL: http://optn.transplant.hrsa.gov/converge/data/default.asp (visited on 11/19/2015).

[9] DM Powers. "Evaluation: From Precision, Recall and F Factor to ROC, Informedness, Markedness & Correaltion". In: *School of Informatics and Engineering, Flinders University of South Australia Adelaide* (2007).

[10] Michael Schemper and Georg Heinze. "Probability imputation revisited for prognostic factor studies". In: *Statistics in medicine* 16.1 (1997), pp. 73–80.

[11] Sesame. *Sesame*. 2015. URL: http://rdf4j.org/ (visited on 04/07/2015).

[12] Eric S. Weiss et al. "Creation of a quantitative recipient risk index for mortality prediction after cardiac transplantation (IMPACT)". In: *The Annals of thoracic surgery* 92.3 (2011), pp. 914–922.

Predicting the Outcome for Patients in a Heart Transplantation Queue using Deep Learning

Predicting the Outcome for Patients in a Heart Transplantation Queue using Deep Learning

Dennis Medved[1], Pierre Nugues[1], and Johan Nilsson[2]

Abstract— **Heart transplantations have made it possible to extend the median survival time to 12 years for patients with end-stage heart diseases. This operation is unfortunately limited by the availability of donor organs and patients have to wait on average about 200 days in a waiting list before being operated. This waiting time varies considerably across the patients. In this paper, we studied the outcome for patients entering a transplantation waiting list using deep learning techniques. We implemented a model in the form of two-layer neural networks and we predicted the outcome as still waiting, transplanted or dead in the waiting list, at three different time points: 180 days, 365 days, and 730 days. As data source, we used the United Network for Organ Sharing (UNOS) registry, where we extracted adult patients (>17 years) from January 2000 to December 2011. We trained our model using the Keras framework, and we report F1 macro scores of respectively 0.674, 0.680, and 0.680 compared to a baseline of 0.271. We also applied a backward elimination procedure, using our neural network, to extract the 10 most significant parameters predicting the patient status for the three different time points.**

I. Introduction

Heart transplantations have made it possible to extend the median survival time to 12 years for patients with end-stage heart diseases. Unfortunately, the need for donated hearts greatly exceeds supply and many candidates die awaiting transplantation. Estimating the probability of dying in the waiting list for a specific time period, could support the decision of surgeons on the priority of a transplantation. In addition, knowing the probability for a patient to be transplanted within a certain time frame would help plan operation resources and inform the patient.

In this study, we have used neural network models to predict the outcome for patients entering a heart transplantation waiting list. We carried out the prediction at three different time points: 180 days, 365 days and 730 days. We categorized the patient status with three possible outcomes: still waiting, transplanted, or dead in the waiting list.

II. Previous Work

A few studies investigated waiting times of allografts. They include heart transplants [10, 4], liver [1], and kidney [3], that all revealed increased waiting times for group O recipients. Other studies proposed models to predict the outcome in heart failures and outlined lists of predictors. [9] is a review of 64 such models, where the possible outcomes were death, hospitalization, and death or hospitalization, depending on the model. The authors could distill a list of 10 consistently used predictors: age, renal function, blood pressure, blood sodium level, etc. Other papers provide models to estimate the survival time after a heart transplantation such as [16, 6], while [5] describe a procedure to extract features predicting the one, five, and ten year survival of patients.

III. Materials and Methods

A. Data Source

UNOS administers the only Organ Procurement and Transplantation Network in the United States of America [7], and is a non-profit organization. The patient data that we used was obtained from the UNOS database. The database contains data from October 1, 1987 and onwards. In the database, there is information that encompass recipient, donor and transplant data. It includes almost 500 variables reflecting different attributes of the patients.

The Ethics Committee for Clinical Research at Lund University, Sweden approved the study protocol. The data was de-identified prior to analyzing it and the institutional review board waived the need for written informed consent from the participants.

B. Study population

We included adult (> 17 years) heart transplantation (HT) patients from January 2000 to December 2011, that either died in the queue, got transplanted, or were still waiting in the queue. We did not include patients, who were removed from the list for other reasons, such as being too sick to be operated. We excluded these patients because they could potentially confuse the model. We assumed that the features predicting the death of a patient would probably be correlated with a removal from the queue.

We used this data set to create three different temporal cohorts, where we recorded the patients' outcome after 180, 365, and 730 days. Table I shows the distribution of outcomes in the different time periods.

The total number of patients included in our data set was of 27,444. We randomly divided the data in train/validation/test in sets of 70%/15%/15% which translates to 19210/4117/4117 patients, respectively.

As features, we included 87 variables describing the patients in the queue that were available at the time of listing such as: age, sex, weight, and blood group.

*This research was supported by Heart Lung Fondation, The Swedish Research Council, and the eSSENCE program.

[1]Department of Computer Science, Lund University, Lund, Sweden {dennis.medved, pierre.nugues}@cs.lth.se

[2]Department of Clinical Sciences Lund, Cardiothoracic Surgery, Lund University and Skåne University Hospital, Lund, Sweden johan.nilsson@med.lu.se

TABLE I
THE THREE TEMPORAL COHORTS AND OUTCOME DISTRIBUTION

Days	Dead (%)	Transplanted (%)	Queueing (%)
180	9.7	57.0	33.4
365	11.6	69.1	19.3
730	13.4	77.3	9.3

C. Imputation of Missing Data

As with all large registries, there is missing patient data. No patient has a complete information record and excluding the patients with missing data fields from the cohorts would have reduced the data set to almost nothing. To mitigate this, we chose to impute the missing data, where we applied a probabilistic approach. For each variable, we replaced the missing values with a random value from a discrete uniform distribution of the non-missing values in this variable, following the method in [11].

D. Evaluation

To evaluate the models, we used the F1 score, which is the harmonic mean of precision and recall [8]. These metrics were created for binary classes and to generalize them to more than two classes, we averaged the results using micro and macro averages.

The micro average method consists of summing up the individual true positives, false positives, and false negatives of the system for the different classes and then calculating the average. The macro average takes the average of the precision and recall of the system on the different classes. When the examples are unevenly distributed across the classes, the macro average method is less biased toward the largest class [15].

We also computed a confusion matrix, where each column of the matrix represents the instances of a predicted class, while each row represents the actual class. The diagonal then represents the correctly classified outcomes. Confusion matrices make it easier to visualize the classification errors that a model produces [13].

E. Implementation Details

We used the Keras framework to train the model [2]. It utilizes Python as a programming interface and enables the user to easily create and configure artificial neural networks (ANN) of different architectures. It serves as a high level abstraction, that utilizes Theano as the back-end [14].

We created a network with two hidden layers and 128 nodes in each layer. The hidden layers used the rectified linear unit as activation function and the final output layer used a softmax activation. We selected categorical cross entropy as the loss function and adamax as the optimizer with 30 epochs.

Dropout is a regularization technique for reducing overfitting in neural networks [12]. The idea behind dropout is to randomly drop units, together with their connections, from the neural network during training. The dropout rate controls the probability of a neuron being removed. We chose to use a dropout rate of 0.5.

F. Feature Significance

We wanted to know which features contributed the most to the result of the classification. We utilized backward elimination to find these features.

Backward elimination starts with all the features and removes them one by one from the set. The resulting feature set is then used to produce the classification probabilities. We calculate the F1 macro metric for each of the new feature sets and remove the feature that produced the best score when excluded. We repeat this process until the desired amount of features remain.

IV. RESULTS

We optimized the hyperparameters on the validation set. Using these parameters, Table II shows the precision and recall values we obtained on the test set, while Table III shows the F1 values for 180, 365, and 730 days, respectively. We included a baseline model in the table that always classifies the most frequent class, in this case: the patient was transplanted. The best macro averaged F1 was achieved for 365 days: 0.680. Figure 1 shows the precision-recall curve for this time period.

TABLE II
THE PRECISION AND RECALL VALUES FOR 180, 365, AND 730 DAYS OBTAINED ON THE TEST SET

Days	Class	Precision	Recall	F1
180	Dead	0.680	0.644	0.664
	Transplanted	0.764	0.887	0.820
	Queuing	0.654	0.485	0.557
365	Dead	0.782	0.684	0.705
	Transplanted	0.842	0.967	0.900
	Queuing	0.605	0.314	0.413
730	Dead	0.770	0.747	0.759
	Transplanted	0.918	0.992	0.954
	Queuing	0.606	0.226	0.329
Baseline 180	Dead	0.000	0.000	0.000
	Transplanted	0.567	1.000	0.724
	Queuing	0.000	0.000	0.000
Baseline 365	Dead	0.000	0.000	0.000
	Transplanted	0.77	1.000	0.869
	Queuing	0.000	0.000	0.000
Baseline 730	Dead	0.000	0.000	0.000
	Transplanted	0.685	1.000	0.813
	Queuing	0.000	0.000	0.000

Using the neural network and backward elimination, we extracted the ten most important features, shown in Table IV. The features are ranked within the sets, according to their removal order. We evaluated these feature sets and Table V shows the results. Figure 2 shows the confusion matrix for the 365 days time period that reveals that the most misclassified outcome is queueing as transplanted.

We wanted to look at the distributions of outcomes depending on the patient having blood group O or not, mostly because previous studies had shown that it was a predictor. In addition, it was also implicitly included in the ten most predictive features for 365 days. Table VI shows that there is a 17% absolute difference between the number of transplanted.

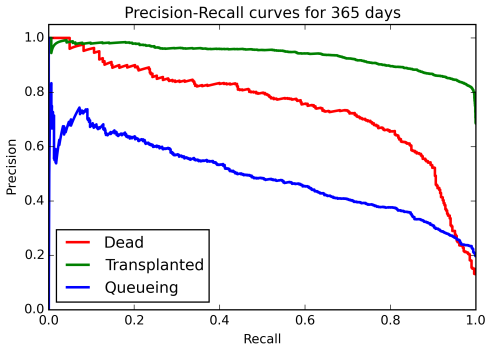


Fig. 1. Precision-recall curves for 365 days

TABLE III
THE F1 VALUES FOR 180, 365, AND 730 DAYS OBTAINED ON THE TEST SET

Days	F1 (micro)	F1 (macro)
180	0.750	0.675
365	0.760	0.680
730	0.888	0.680
Baseline 180	0.567	0.241
Baseline 365	0.685	0.271
Baseline 730	0.769	0.290

V. DISCUSSION

The distribution of patient outcomes within the cohorts is quite imbalanced, where transplanted is the outcome for 57-77% of the patients, during the chosen time periods. We tried a simple baseline, where we classified all the patient outcomes as the most frequent, see Table III for the results. It produced quite good micro averaged values, mostly because these metrics are biased towards the largest class, but comparatively bad macro values.

The largest misclassification error in Figure 2 corresponds to queueing as transplanted. This is probably because it is hard to differentiate between the patients that were transplanted at a certain time point versus those that are still waiting in the queue, based on the available features.

We carried out a backward elimination using our neural network and the ten most contributing features is shown in Table IV. This results in a decrease of only about 2% (absolute difference) from the F1 macro score with all the features, see Table V. This means that most of the predictive power from the ANN comes from a few features. Neural networks do a kind of feature selection naturally as part of the model, weighing up more predictive features and weighing down the less predictive. Because of this, feature search for neural networks is usually not needed. But considering it is hard to interpret the matrices produced by the ANN model directly, we carried out a backward elimination to approximate the features importance.

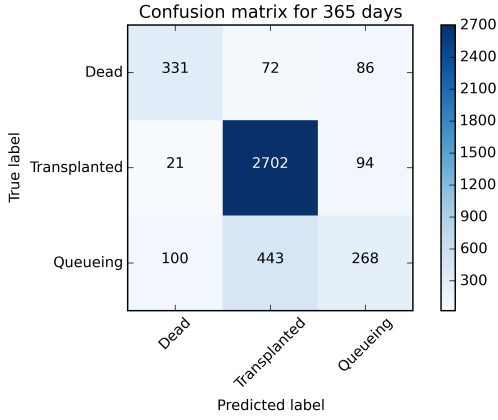


Fig. 2. Confusion matrix for 365 days time period.

The features shared by all of the three sets are: urgency status 2, weight, height and body mass index (BMI). BMI can be considered a feature transformation of weight and height as BMI = weight $\times$ height^2, but it provided extra predictive information over the constituent variables. A sufficiently complex neural network could probably approximate this transformation and therefore BMI would probably not be needed.

Table VI shows some discrepancy between the number of transplanted patients depending on having blood group O. This can probably be explained by the fact that only patients that are blood-group compatible with the donor are transplanted. Even though type O is quite common, patients of this group can only receive from donors from the same blood group and can give to all other types.

A. Future Work

We did not have time to fully optimize the hyperparameters of the neural network and there are some variables that are available that we did not include, both which could produce better results.

We also plan to build a more advanced model based on networks similar to those we described in this paper to be able to estimate the probability the patient would die or would be transplanted depending on the time s/he spent in the waiting list.

ACKNOWLEDGMENT

This work is based on OPTN data as of October 1, 2013 and was supported in part by the Health Resources and Services Administration contract 234-2005-370011C. The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services. This research was supported by Heart Lung Foundation, The Swedish Research Council, and the eSSENCE program.

TABLE IV

THE TEN MOST CONTRIBUTING FEATURES FOR EACH TIME PERIOD USING BACKWARD ELIMINATION, IN ORDER OF IMPORTANCE.

Rank	180 days	365 days	730 days
1	Urgency status 2	BMI	BMI
2	Weight	Weight	Weight
3	BMI	Height	Height
4	Height	Urgency status 2	Urgency status 2
5	Inotropes	Creatine clearance	Creatinine
6	Blood group: AB	Inotropes	Functional status
7	Life support	Blood group: A	Pulmonary Vascular Resistance
8	Blood group: B	Life support	Educational level: none
9	Inotropic support	Blood group: AB	Ventricular assist type: LVAD + RVAD
10	Ethnicity: black	Blood group: B	Educational level: grade school

TABLE V

EVALUATION ON THE TEST WITH THE 10 BEST FEATURES FOUND FOR EACH TIME PERIOD.

Days	F1 (micro)	F1 (macro)
180	0.710	0.657
365	0.714	0.655
730	0.889	0.660

TABLE VI

THE DISTRIBUTION OF OUTCOMES DEPENDING ON BLOOD GROUP FOR 365 DAYS

Blood group	Dead (%)	Transplanted (%)	Queueing (%)
O	14.2	59.3	26.5
not O	9.7	76.4	13.8

REFERENCES

[1] Michele Barone et al. "ABO blood group-related waiting list disparities in liver transplant candidates: effect of the MELD adoption." In: *Transplantation* 85.6 (2008), pp. 844–849.

[2] François Chollet. *Keras*. `https://github.com/fchollet/keras`. 2015.

[3] Petra Glander et al. "The 'blood group O problem' in kidney transplantation—time to change?" In: *Nephrology Dialysis Transplantation* 25.6 (2010), p. 1998.

[4] J.C. Hussey, J. Parameshwar, and N.R. Banner. "Influence of Blood Group on Mortality and Waiting Time Before Heart Transplantation in the United Kingdom: Implications for Equity of Access". In: *The Journal of Heart and Lung Transplantation* 26.1 (2007), pp. 30–33. ISSN: 1053-2498.

[5] D. Medved, P. Nugues, and J. Nilsson. "Selection of an optimal feature set to predict heart transplantation outcomes". In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Aug. 2016, pp. 3290–3293.

[6] Johan Nilsson et al. "The International Heart Transplant Survival Algorithm (IHTSA): A New Model to Improve Organ Sharing and Survival". In: *PLoS ONE* 10.3 (2015), e0118644.

[7] United Network for Organ Sharing. *Organ Procurement and Transplantation Network Data*. 2015. URL: `http://optn.transplant.hrsa.gov/converge/data/default.asp` (visited on 11/19/2015).

[8] DM Powers. "Evaluation: From Precision, Recall and F Factor to ROC, Informedness, Markedness & Correaltion". In: *School of Informatics and Engineering, Flinders University of South Australia Adelaide* (2007).

[9] Kazem Rahimi et al. "Risk Prediction in Patients With Heart Failure". In: *JACC: Heart Failure* 2.5 (2014), pp. 440–446. ISSN: 2213-1779.

[10] Helena Rexius, Folke Nilsson, and Anders Jeppsson. "On the Allocation of Cardiac Allografts from Blood Group-O Donors". In: *Scandinavian Cardiovascular Journal* 36.6 (2002), pp. 342–344.

[11] Michael Schemper and Georg Heinze. "Probability imputation revisited for prognostic factor studies". In: *Statistics in medicine* 16.1 (1997), pp. 73–80.

[12] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: `http://jmlr.org/papers/v15/srivastava14a.html`.

[13] Adi L Tarca et al. "Machine learning and its applications to biology". In: *PLOS Computational Biology* 3.6 (2007), e116.

[14] Theano Development Team. "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: `http://arxiv.org/abs/1605.02688`.

[15] Vincent Van Asch. *Macro-and micro-averaged evaluation measures*. Tech. rep. University of Antwerp, 2013.

[16] Eric S. Weiss et al. "Creation of a Quantitative Recipient Risk Index for Mortality Prediction After Cardiac Transplantation (IMPACT)". In: *The Annals of Thoracic Surgery* 92.3 (2011), pp. 914–922.

Improving Prediction of Heart Transplantation Outcome Using Deep Learning Techniques

SCIENTIFIC REPORTS

OPEN

Improving prediction of heart transplantation outcome using deep learning techniques

Received: 10 October 2017
Accepted: 1 February 2018
Published online: 26 February 2018

Dennis Medved[1], Mattias Ohlsson[2], Peter Höglund[3], Bodil Andersson[4], Pierre Nugues[1] & Johan Nilsson[5]

The primary objective of this study is to compare the accuracy of two risk models, International Heart Transplantation Survival Algorithm (IHTSA), developed using deep learning technique, and Index for Mortality Prediction After Cardiac Transplantation (IMPACT), to predict survival after heart transplantation. Data from adult heart transplanted patients between January 1997 to December 2011 were collected from the UNOS registry. The study included 27,860 heart transplantations, corresponding to 27,705 patients. The study cohorts were divided into patients transplanted before 2009 (derivation cohort) and from 2009 (test cohort). The receiver operating characteristic (ROC) values, for the validation cohort, computed for one-year mortality, were 0.654 (95% CI: 0.629–0.679) for IHTSA and 0.608 (0.583–0.634) for the IMPACT model. The discrimination reached a C-index for long-term survival of 0.627 (0.608–0.646) for IHTSA, compared with 0.584 (0.564–0.605) for the IMPACT model. These figures correspond to an error reduction of 12% for ROC and 10% for C-index by using deep learning technique. The predicted one-year mortality rates for were 12% and 22% for IHTSA and IMPACT, respectively, versus an actual mortality rate of 10%. The IHTSA model showed superior discriminatory power to predict one-year mortality and survival over time after heart transplantation compared to the IMPACT model.

Heart transplantation (HT) is a life-saving operation for patients with end-stage heart disease. Despite this reality, the transplantation number does not increase over the years. One of the most limiting factors is the lack of donor organs and a conservative allocation policy that results in the loss of about half of the organs being offered[1]. An improved prediction of the outcome would augment the confidence in the post-transplantation performance and make it possible to optimise the allocation of organs. Furthermore, it would enable practitioners to determine the risk of early and late graft dysfunction more accurately and improve donor and recipient management.

Although there exist several survival models within cardiac surgery, currently there is no accepted tool for estimating the outcome after heart transplantation. In recent years, some risk score algorithms designed to predict post-transplantation performance have been developed, which almost all have been derivate on the single national, multi institutional United Network for Organ Sharing (UNOS) registry[2]. The most notable ones are: Donor Risk Index (DRI), Risk Stratification Score (RSS), and Index for Mortality Prediction After Cardiac Transplantation (IMPACT)[3–5]. The IMPACT model has additionally been validated on the International Society of Heart and Lung Transplantation (ISHLT) registry and showed an acceptable accuracy in predicting mortality. Recently a multinational model, the International Heart Transplantation Survival Algorithm (IHTSA), developed on the ISHLT registry was published[6]. This model was designed to predict both short-term and long-term mortality and, in contrast to previous models, it utilises deep learning techniques. The results it obtained showed an improved discrimination compared with the DRI, RSS, and IMPACT models. However, the validation was performed on the ISHLT registry, which was also used for the development of the model[6].

Even if the validation cohort was separated from the derivation cohort, the IHTSA model might be biased towards this registry.

[1]Department of Computer Science, Lund University, Lund, Sweden. [2]Department of Astronomy and Theoretical Physics, Computational Biology and Biological Physics, Lund University, Lund, Sweden. [3]Department of Laboratory Medicine Lund, Clinical Chemistry and Pharmacology, Lund University, Lund, Sweden. [4]Department of Clinical Sciences Lund, Surgery, Lund University and Skåne University Hospital, Lund, Sweden. [5]Department of Clinical Sciences Lund, Cardiothoracic Surgery, Lund University and Skåne University Hospital, Lund, Sweden. Correspondence and requests for materials should be addressed to J.N. (email: johan.nilsson@med.lu.se)

The aim of this study was to determine the most suitable risk stratification model for heart transplantation by applying the IMPACT and IHTSA algorithms to the UNOS registry.

Results

Characteristics of the Study Population. The preoperative characteristics of the recipients are listed in Table 1 and for the donors in Table 2. The number of adult HT with a follow-up time of at least one year, from January 1997 to December 2011, was of 27,860, corresponding to 27,705 patients. Over the time span, the cumulative sum of follow-up years was of 165,206. The median survival time was 12 years (Interquartile Range [IQR]: 5–16). The one-year mortality was of 13% (n = 3,561). The average age of the recipients was 52 ± 13 years, with a range from 18 to 78 years. Most of the recipients were males 76% (n = 21,151). Multi-organ transplants were marginal (2.5%). The number of transplants contained in the derivation cohort was of 22,263, and the number of transplants in the test cohort was of 5,597.

IMPACT versus IHTSA. The IHTSA model includes 32 recipient risk variables, while the IMPACT model has 18 variables; five of these variables are shared between the models: female gender, diagnosis: ischemic cardiomyopathy, diagnosis: congenital, infection within two weeks, and mechanical ventilation. Additionally, IHTSA also has 11 donor variables, while IMPACT has no donor variables.

We evaluated the original IHTSA model in the test cohort (2009–2011) for one-year mortality; it had an area under receiver operating characteristic (AUROC) of 0.643 (95% CI: 0.619–0.667), while IMPACT had an AUROC of 0.608 (0.583–0.634), $P = 0.004$, see Table 3. As shown in Fig. 1 and Table 3, the recalibrated IHTSA model has a significantly higher discrimination compared with the IMPACT model for one-year mortality, $P = 0.001$, corresponding to an error reduction of 11.7%. Harrell's C-index for the recalibrated IHTSA compared with IMPACT was substantially larger, as shown in Table 4, with about a 4% absolute difference for the later time era. This corresponds to an error reduction of 10.3%. On the time era 1997–2008, on which the models were trained using 5-fold cross-validation technique, the recalibrated IHTSA had an AUROC of 0.688 (0.678–0.699), and IMPACT had 0.606 (0.595–0.617) for one-year mortality, $P = 0.001$, Table 3. The absolute difference in C-index was 5% higher for the IHTSA model compared with the IMPACT model, $P < 0.001$, Table 4.

We analysed the sensitivity of both models relatively to the deceased patients after one year at the levels of 25%, 50%, and 75%. Out of the transplants in the test cohort (N = 5,597), the numbers of correctly classified patients after one year were 4,812, 3,890, and 2,582 patients respectively for IHTSA, and 4,539, 3,396, and 2,140 patients respectively for IMPACT. See Fig. 2 for a graph of the difference in correctly classified patients.

We furthermore compared the predicted one-year mortality rate for IMPACT and IHTSA, with the true mortality rate. The predicted one-year mortality for the second time-era (test cohort) was 12% and 22% for the recalibrated IHTSA and IMPACT, respectively, versus an actual mortality rate of 10%. The Hosmer-Lemeshow (HL) chi-square for one-year, using ten groups, was of 40 in the IHTSA model and 101 for the IMPACT model, both with a P-value less than 0.05. As shown in the calibration plot, Fig. 3, the predictive mortality compared with actual mortality was more consistent over all deciles for the ITHSA model compared with the IMPACT model.

To evaluate difference in methodology approach (deep learning versus logistic regression), we performed two additional experiments. We quantify the difference between the deep learning technique used by the IHTSA model and the more traditional logistic regression approach used by the IMPACT model, by letting the two systems use identical features. The second experiment was to assess the difference between a model that include and exclude donor variables.

As shown in Tables 5 and 6, a recalibrated IHTSA model including only the same risk variables as the IMPACT model still showed a substantial improvement in the AUROC (about 2%) and C-index in the test cohort compared with the IMPACT model. The recalibrated IHTSA model excluding the donor variables showed a decrease in discrimination compared with the original IHTSA model, however the difference was minor, producing nearly the same AUROC.

Discussion

The purpose of this study was to compare the IMPACT and IHTSA models with regards to the prediction accuracy of one-year mortality on the UNOS database. There exist some biases in both models when used on the UNOS data set for the time era 1997–2008. Because IMPACT was developed on these data and IHTSA on the ISHLT dataset, which consists in part of the same UNOS data, the models may be subjected to a non-negligible overfit to the data, skewing the result towards a more positive value. Therefore, we chose to validate the models on a later time era, which has no overlapping patients with the training set.

The results show that the IHTSA model exhibited improved performance and accuracy compared to the IMPACT model. Even though IMPACT was designed to predict one-year mortality and IHTSA was created for long-term survival, IHTSA shows better discrimination on one-year mortality.

This study could also prove the benefits of using deep learning modelling techniques. Such techniques are inspired by the human brain. They consist of a network of "neurons" that emulate the properties of their real counterparts. Using multiple processing layers makes it possible to learn representations of data with multiple levels of abstraction[7]. These methods have improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains[8].

Our results show that the IHTSA model can be applied to predict short-term mortality with greater accuracy than a more traditional risk-based model based on logistic regression. Although the comparison of ROC curves to evaluate models in a statistically valid manner is controversial, the ROC curve is currently the most developed statistical tool for describing performance[9,10]. The improvements seen can be explained by the difference in the variable selection, such as the absence of donor risk factors in the IMPACT model, but also by the the neural network's ability to handle interactions between variables and nonlinearities. An increased donor age has in previous

Feature	N	Time era 1997–2008 (n = 22,263)	Time era 2009–2011 (n = 5,597)	p-Value	IMPACT	IHTSA
Demographic data						
Age (years)	27,860	52 ± 13	53 ± 13	0.001		✓
Age >60 years	27,860	5,707 (26%)	1,809 (32%)	0.001	✓	
Female gender	27,860	5,298 (24%)	1,411 (25%)	0.029	✓	✓
Height (cm)	27,740	174 ± 10	174 ± 10	0.835		✓
Weight (kg)	27,760	80 ± 17	82 ± 17	0.001		✓
Race: African American	27,860	3,324 (15%)	1,103 (20%)	0.001	✓	
Diagnosis						
Ischemic cardiomyopathy	27,859	9,976 (45%)	2,793 (50%)	0.001	✓	✓
Non-ischemic cardiomyopathy	27,859	10,247 (46%)	2,119 (38%)	0.001		✓
Congenital	27,859	518 (2%)	149 (3%)	0.159	✓	✓
Other	27,859	852 (3%)	247 (4%)	0.001	✓	
Graft failure	27,859	669 (3%)	197 (4%)	0.058		✓
Diabetes mellitus[#]	27,597	4,735 (22%)	1,500 (27%)	0.001		✓
Hypertension[†]	17,876	7,108 (40%)	—			✓
Infection within two weeks[‡]	26,543	2,333 (11%)	594 (11%)	0.550	✓	✓
Antiarrhythmic drugs prior transplant	17,266	6,371 (37%)	—			✓
Amiodarone prior to transplant	17,530	4,726 (27%)	—			✓
Dialysis prior to transplant	27,002	706 (3%)	185 (3%)	0.510	✓	
Previous blood transfusion	15,221	5,285 (35%)	27 (29%)	0.247		✓
Previously transplanted[*]	27,860	680 (3%)	199 (4%)	0.067		✓
Previous cardiac surgery	14,069	1,866 (22%)	1,483 (27%)	0.001		✓
ICU	27,860	7,991 (36%)	1,493 (27%)	0.001		✓
Mechanical ventilation	27,860	625 (3%)	166 (3%)	0.532	✓	✓
ECMO	27,860	90 (0.04%)	48 (1%)	0.001		✓
IABP	27,860	1193 (5%)	263 (5%)	0.039	✓	✓
Ventricular assist device	24,357	4,665 (25%)	2,191 (39%)	0.001		✓
Early generation[a]	6,856	911 (20%)	114 (5%)	0.001	✓	
Late generation[b]	6,856	536 (11%)	1,610 (74%)	0.001	✓	
Other/Unknown	6,856	3,218 (69%)	467 (21%)	0.001		
Temporary circulatory support[c]	27,860	209 (1%)	113 (2%)	0.001	✓	
Transplant era						
1996–2000	27,860	7781 (35%)	—			✓
2001–2005	27,860	8981 (40%)	—			✓
>2005	27,860	5501 (25%)	5,598 (100%)	0.001		✓
Hemodynamic status						
PVR (wood units)	21,782	2.5 ± 1.8	2.4 ± 1.8	0.205		✓
SPP (mmHg)	25,100	43 ± 14	42 ± 14	0.001		✓
Laboratory values						
Creatinine (mg/dl)	27,027 1	1.4 ± 0.8	1.3 ± 0.8	0.038		✓
Creatinine clearance (mL/min)						
30–49	27,054	2,964 (14%)	698 (12%)	0.008	✓	
<30	27,054	674 (3%)	189 (3%)	0.376	✓	
Serum bilirubin (mg/dl)	26,224	1.3 ± 2	1.2 ± 2	0.001		✓
1.00–1.99	26,224	6,117 (30%)	1,562 (28%)	0.102	✓	
2.00–3.99	26,224	1261 (6%)	300 (5%)	0.070	✓	
≥4	26,224	1314 (6%)	297 (5%)	0.007	✓	
Immunology status						
PRA > 10%	18,351	1,113 (8%)	1,114 (20%)	0.001		✓
HLA-DR, 2 mismatch	23,858	10,289 (55%)	2,746 (55%)	0.906		✓
Recipient blood group						
A	27,860	9,543 (43%)	2,313 (41%)	0.036		✓
B	27,860	3,040 (14%)	795 (14%)	0.343		✓
AB	27,860	1,143 (5%)	295 (5%)	0.597		✓
O	27,860	8,549 (38%)	2,198 (39%)	0.092		✓

Table 1. The recipient features used in the IMPACT and IHTSA Models. N, number of transplants with non-missing values. n, total number of transplants. Qualitative data are expressed as n (%), and quantitative data as mean ± SD. [#]Drug or insulin treated diabetes mellitus. [†]Drug treated systemic hypertension. [‡]Infection requiring intravenous antibiotic therapy within two weeks prior to transplant. *Previous transplant—previous kidney, liver, pancreas, pancreas islet cells, heart, lung, intestine and/or bone marrow transplant. [a]Early generation includes para and intracorporeal pulsatile VADs: Abiomed AB5000, Heartmate I, XE, and XVE, ThortecIVAD, Toyobo, Medos and LionHeart. [b]Later generation continuous VADs including Heartmate II, Jarvik, Micromed, Debakey, and VentrAssist. [c]Includes ECMO and [or] extracorporeal VADs: Abiomed BVS5000, Bio-Medicus, TandemHeart, and Levitronix/Centrimag. ECMO, extracorporeal membrane oxygenation; ICU, intensive care unit; IHTSA, international heart transplantation survival algorithm; IMPACT, index for mortality prediction after cardiac transplantation; HLA, human leukocyte antigen; PRA, panel reactive antibody; PVR, pulmonary vascular resistance; SD, standard deviation; SPP, systolic pulmonary pressure. The t-test and chi-squared test was used for continuous respectively categorical values.

Feature	N	Time era 1997–2008 (n = 22,263)	Time era 2009–2011 (n = 5,597)	p-Value	IMPACT	IHTSA
Demographic data						
Age (years)	27,075	32 ± 12	32 ± 12	0.515		✓
Female gender	27,860	6,546 (29%)	1,645 (29%)	0.979		✓
Weight (kg)	27,838	79 ± 19	82 ± 19	0.001		✓
Duration of ischemia (min)	26,029	189 ± 63	194 ± 10	0.001		✓
CODD: Head Trauma	27,825	13,733 (62%)	3,068 (55%)	0.001		✓
CODD: Cerebrovascular event	27,825	5,894 (27%)	1,297 (23%)	0.001		✓
Donor blood group						
A	27,859	8,232 (37%)	1,983(35%)	0.030		✓
B	27,859	2,284 (10%)	617 (11%)	0.102		✓
AB	27,859	477 (2%)	125 (2%)	0.682		✓
O	27,859	11269 (40%)	2,873 (51%)	0.001		✓
Recipient-donor weight ratio	27,739	1.03 ± 0.22	1.02 ± 0.20	0.001		✓
Recipient-donor height ratio	27,660	0.998 ± 0.06	0.999 ± 0.06	0.068		✓

Table 2. The donor features used in the IHTSA model. N, number of transplants with non-missing values. n, total number of transplants. Qualitative data are expressed as n (%), and quantitative data as mean ± SD. CODD, cause of donor death; IHTSA, international heart transplantation survival algorithm; IMPACT, index for mortality prediction after cardiac transplantation. The t-test and chi-squared test was used for continuous respectively categorical values.

Time era	AUROC (95% CI)				
	IMPACT	IHTSA	*P*-Value	IHTSA cal.	*P*-Value
1997–2008	0.61 (0.59–0.62)	0.66 (0.64–0.67)	0.001	0.69 (0.68–0.70)	0.001
2009–2011	0.61 (0.58–0.63)	0.64 (0.62–0.67)	0.004	0.65 (0.63–0.68)	0.001

Table 3. The AUROC for one-year mortality for the different cohorts using IMPACT and IHTSA respectively. AUROC, area under the receiver-operating curve; CI, confidence interval; IHTSA, international heart transplantation survival algorithm; cal, the recalibrated version; IMPACT, index for mortality prediction after cardiac transplantation.; *P*, probability that the result is the same as IMPACT.

reports been shown to have a negative influence on short-term survival[6,11]. To examine this, we compared the difference of the deep learning model and the logistic regression model using the same variables. Here, we show a substantial improvement when using the deep learning approach compared with the traditional approach. Furthermore, we could show that the predictive availability for the deep learning model was less dependent on the variables included compared with a standard model. Donor variables showed to be of less importance than expected. A possible explanation for that may be the deep learning technology has an increased ability to identify new patterns with the data it has available. It is interesting to note that the two models do not show a considerable overlap of features. Only five features are shared by the two models out of 18 for IMPACT and 43 for IHTSA. If we compare the overlapping variables with the seven most important variables for IHTSA, we find that three of them are shared: age, diagnosis, and mechanical ventilation[6].

One disadvantage of the deep learning technique is that it yields a black box model with a limited ability to explicitly identify possible causal relationships. Logistic regression, on the contrary, makes it feasible to determine the strongly predictive variables based on the size of the coefficients. To cope with the lack of a well-established

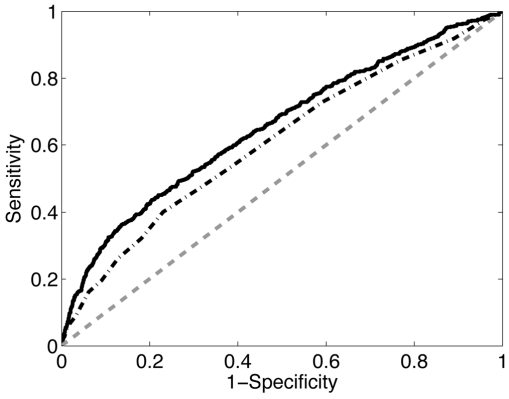


Figure 1. The ROC curves show the sensitivity of prediction of one-year mortality vs. 1-specificity for the IMPACT (short-long dashed line) and the recalibrated IHTSA (solid line) risk algorithms is plotted on the test cohort (2009–2011). The gray dashed line represents the absence of discrimination.

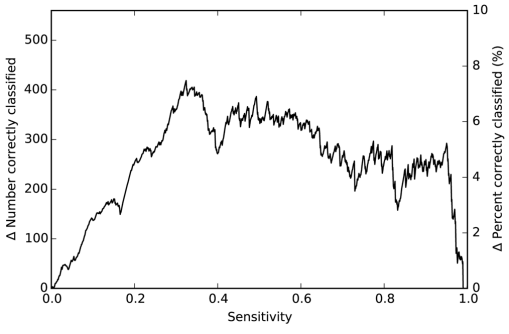


Figure 2. The sensitivity of prediction of one-year mortality versus the total number of additional correctly classified patients by IHTSA compared with IMPACT, both in absolute numbers and percentage, plotted on the test cohort (2009–2011).

	C-index (95% CI)				
Time era	IMPACT	IHTSA	*P*-Value	IHTSA cal.	*P*-Value
1997–2008	0.56 (0.56–0.56)	0.59 (0.59–0.60)	0.001	0.62 (0.61–0.62)	0.001
2009–2011	0.58 (0.56–0.61)	0.61 (0.59–0.63)	0.002	0.63 (0.61–0.65)	0.001

Table 4. The Harrells C-index for survival for the different cohorts using IMPACT and IHTSA respectively. CI, confidence interval; IHTSA, international heart transplantation survival algorithm; cal, the recalibrated version; IMPACT, index for mortality prediction after cardiac transplantation; *P*, probability that the result is the same as IMPACT.

method for interpreting the weights of a connection matrix in a neural network, the developers of the IHTSA algorithm used a classification and regression tree (CART), fitted to the predicted median survival time, to assess the relative importance of the features[6]. Furthermore, the web-based calculator (http://ihtsa.cs.lth.se) makes it possible to estimate the survival on a computer or mobile device.

During 2011, approximately 17,000 donors were reported[12]. Unfortunately, not more than one-third of all donors could be utilised for heart transplantation. One explanation for this may be the uncertainty in the risk of early and late graft dysfunction, which means that some suitable donors are not accepted. Although there are many donor predictors of allograft discard in the current era, these characteristics seem to have little effect on recipient outcomes when the hearts are transplanted, which also is confirmed in this study[13]. A more liberal use of

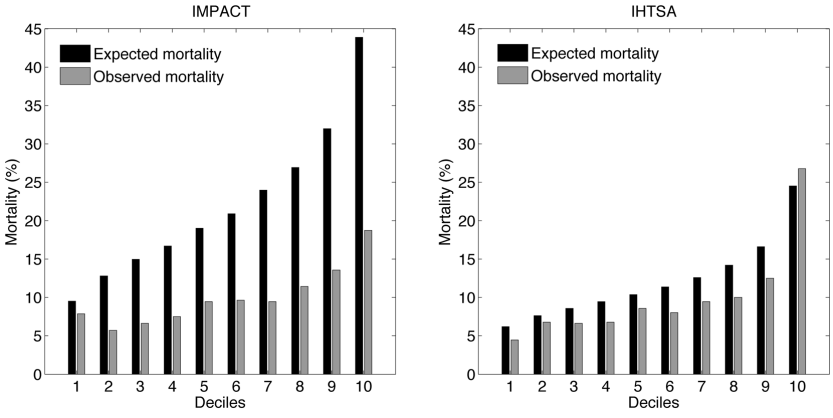


Figure 3. The observed (gray bars) and expected mortality (black bars), in percent, for each decile, for the IMPACT and IHTSA models, in the test cohort (2009–2011). The patients are divided into deciles according to their expected mortality, and the observed mortality was derived for each decile.

	AUROC (95% CI)				
Time era	IMPACT	ANN I	*P*-Value	ANN II	*P*-Value
2009–2011	0.61 (0.58–0.63)	0.63 (0.60–0.65)	0.027	0.65 (0.63–0.68)	0.001

Table 5. The AUROC for one-year mortality for the test cohort (2009–2011) using an artificial neural network model derived on the derivation cohort (1997–2008) with IMPACT features only (ANN I) and with IHTSA recipient features only (ANN II). AUROC, area under the receiver-operating curve; CI, confidence interval; IHTSA, international heart transplantation survival algorithm; IMPACT, index for mortality prediction after cardiac transplantation.; *P*, probability that the result is the same as IMPACT.

	C-index (95% CI)				
Time era	IMPACT	ANN I	*P*-Value	ANN II	*P*-Value
2009–2011	0.58 (0.56–0.61)	0.60 (0.58–0.62)	0.002	0.62 (0.60–0.64)	0.001

Table 6. The Harrells C-index for one-year mortality for the test cohort (2009–2011) using an artificial neural network model derived on the derivation cohort (1997–2008) with IMPACT features only (ANN I) and with IHTSA recipient features only (ANN II). CI, confidence interval; IHTSA, international heart transplantation survival algorithm; IMPACT, index for mortality prediction after cardiac transplantation.; *P*, probability that the result is the same as IMPACT.

cardiac allografts with relative contraindications may be warranted. A calculator would allow us to conveniently perform batch estimation of survival for multiple patients at the same time. This would allow the IHTSA model to be used as a virtual recipient-donor matching tool that models survival for potential recipients on a waiting list when there is a donor heart available. This could potentially increase the number of organs that could be used compared with a traditional criterion-based model[6]. Additionally, it will make it easier for other research groups to validate the model.

The results of this study carry limitations associated with the retrospective analysis of a registry database, the quality of the source data, the number of missing data, and the lack of standardization associated with multi-center studies (such as different immunosuppressive regimens and different matching criteria). However, those limitations are the same for both models. Even if a comparison of risk models remains controversial, the C-index is probably the best statistical tool for describing performance. A C-index of <0.7 may seem low, but it should be kept in mind that the IHTSA model predicts long term survival, and to the best of our knowledge, it is higher than previously reported studies.

Conclusions

In this study, we have shown that a flexible nonlinear artificial neural network model (IHTSA), utilising deep learning techniques, exhibits better discrimination and accuracy than a more traditional risk score model (IMPACT) for predicting one-year mortality. We made public the results of this model in the form of a web-based

batch calculator that could be used as a virtual recipient-donor matching tool. This is a first step in the implementation of a deep learning architecture for transplantation data that, we hope, will pave the way for further improvements and an even more accurate model.

Materials and Methods

Data Source. The data set of heart transplant patients was obtained from the UNOS database. UNOS is a non-profit organisation that administers the only Organ Procurement and Transplantation Network (OPTN) in the United States of America[14]. The database contains data from October 1, 1987, onwards and includes almost 500 variables that encompass recipient, donor, and transplant information. It consists of both deceased- and living-recipient transplants. The Ethics Committee for Clinical Research at Lund University, Sweden approved the study protocol. The data was anonymized and de-identified prior to analysis and the institutional review board waived the need for written informed consent from the participants.

Study Population. We included all the adult HT patients (>17 years) from January 1997 to December 2011. The latest annual follow-up was on September 30, 2013. The data set was divided into two temporal cohorts: transplantation done before 2009 (derivation cohort) and after or during 2009 (test cohort). These time periods were chosen because both IMPACT and IHTSA were developed on patients between 1997–2008 and we wanted disjoint sets (derivation and test) to evaluate the prediction performance. The number of variables extracted from the database was 56 in total, where IHTSA uses 43 of them and IMPACT 18. The primary endpoint was one-year mortality and the second endpoint was all-cause cumulative mortality during the study period.

Storing the Data. We converted the complete UNOS database containing heart transplants until 2011, except a few variables, into a Resource Description Framework (RDF) database following the procedure outlined in a previously published report[15]. This enabled us to use the SPARQL language to query the data and easily retrieve the variables used by both the IMPACT and IHTSA model to predict the mortality of the transplants[16].

Statistical Analysis. We performed the statistical analyses using the Stata MP statistical package version 13 (2013) (StataCorp LP, College Station, TX), and with RStudio Desktop 0.99.441 (RStudio, Boston, MA) using R version 3.3.1. Data are presented as means with standard deviation (SD), and frequency as appropriate. The Anderson-Darling test was used to assess the normality of the variables[17]. We used the t-test and chi-squared test for continuous, respectively categorical values, to test if the data was significantly different from each other. As with all patient registries, the dataset contains missing values. We applied a probability imputation technique by creating a list for each variable in the data set, containing the non-missing values for that variable, and then we imputed each missing value with a value from the list, chosen from a uniform distribution[18]. In consequence, the distribution of the imputed values should follow that of the non-missing ones.

The discriminatory power for one-year mortality was assessed by calculating the AUROC[19]. We compared the statistical significance of the difference between the AUROC of the two models using the non-parametric DeLong's test[20]. To evaluate the discrimination for long-term survival of the patients, we utilised the Harrell's concordance index (C-index)[21]. We used a z-score test to compare the C-indexes[22]. The AUROC and C-index values are both presented with 95% confidence limits. The predictive accuracy of the models was assessed by comparing the observed and expected mortality for equal-sized quantiles of risk by using the Hosmer–Lemeshow goodness-of-fit test[23].

The IMPACT model. IMPACT was created with a data set of heart transplant patients between 1997 to 2008 that were collected from the UNOS database. IMPACT only utilises recipient variables. Creatinine clearance was not directly available from the data set and had to be calculated using the Cockcroft-Gault equation[24]. By apportioning points according to the relative importance of the variables for the one-year mortality, a risk index was created. The minimum number of scoring points a patient can have is 0 and the maximum is 50. The points are after that converted to a predicted probability of one-year mortality by a formula derived from logistic regression[5].

The IHTSA model. The data set used in developing IHTSA was extracted from the ISHLT containing HT patients who were transplanted between 1994 and 2010. IHTSA utilises both recipient and donor variables. The survival model consists of a flexible nonlinear generalisation of the standard Cox proportional hazard model. Instead of using a single prediction model, this model integrates ensembles of artificial neural networks (ANNs). In addition, its prediction capability is not limited to one year[6].

However, the variables hypertension and antiarrhythmic drugs are not recorded in the UNOS database from 2007 and onward. To handle this problem, we first imputed them with random values taken from the earlier time era. Secondly, we excluded these two variables, and retrained (calibrated) the neural network, utilizing a 5-fold cross validation of the patients between 1997 and 2008 in UNOS. The same training procedure was used as described in the original IHTSA article, but we did not carry out any new variable selection[6]. We called this model the recalibrated IHTSA model.

Web-Based IHTSA Calculator. The IHTSA model is available via a web application (ihtsa.cs.lth.se), where a user can either input a single patient's data or submit a file of multiple patients in a batch calculator. To compute the results, the user then selects one of the two prediction models developed either on UNOS or IHSLT data, corresponding to American or international patients respectively. The submitted file should consist of comma-separated values (CSV) reflecting the patient data in a table format. The batch calculator uses this data to predict one-, five-, and ten-year survival respectively and median survival time. Once processed, the result

consisting of relevant survival and mortality numbers is either emailed back to the user in a CSV format, in the case of the batch calculator, or presented directly in the web interface.

The applications were implemented as a Java program, for the graphical user interface part and a Matlab (version 2010A and 2015b) application for running the survival models.

Data availability. The data that support the findings of this study are available from UNOS but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available.

References

1. Klein, A. S. *et al*. Organ donation and utilization in the United States, 1999–2008. *Am J Transplant* **10**, 973–986, https://doi.org/10.1111/j.1600-6143.2009.03008.x (2010).
2. Nilsson, J., Algotsson, L., Höglund, P., Lührs, C. & Brandt, J. Comparison of 19 pre-operative risk stratification models in open-heart surgery. *Eur Heart J* **27**, 867–874, https://doi.org/10.1093/eurheartj/ehi720 (2006).
3. Weiss, E. S. *et al*. Development of a quantitative donor risk index to predict short-term mortality in orthotopic heart transplantation. *The Journal of Heart and Lung Transplantation* **31**, 266–273 (2012).
4. Hong, K. N. *et al*. Who is the high-risk recipient? Predicting mortality after heart transplant using pretransplant donor and recipient risk factors. *The Annals of thoracic surgery* **92**, 520–527 (2011).
5. Weiss, E. S. *et al*. Creation of a Quantitative Recipient Risk Index for Mortality Prediction After Cardiac Transplantation (IMPACT). *The Annals of Thoracic Surgery* **92**, 914–922 (2011).
6. Nilsson, J. *et al*. The International Heart Transplant Survival Algorithm (IHTSA): A New Model to Improve Organ Sharing and Survival. *PloS one* **10**, e0118644 (2015).
7. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444, https://doi.org/10.1038/nature14539 (2015).
8. Cucchetti, A. *et al*. Artificial neural network is superior to MELD in predicting mortality of patients with end-stage liver disease. *Gut* **56**, 253–258 (2007).
9. Bradley, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* **30**, 1145–1159 (1997).
10. Kumar, R. & Indrayan, A. Receiver operating characteristic (ROC) curve for medical researchers. *Indian pediatrics* **48**, 277–287 (2011).
11. Ammirati, E. *et al*. A prospective comparison of mid-term outcomes in patients treated with heart transplantation with advanced age donors versus left ventricular assist device implantation. *Interact Cardiovasc Thorac Surg* **23**, 584–592, https://doi.org/10.1093/icvts/ivw164 (2016).
12. Matesanz, R. International Figures on Donation and Transplantation −2012. 74 pages (Council of Europe European committee on organ transplantation, Global Observatory on Donation & Transplantation, 2013).
13. Khush, K. K., Menza, R., Nguyen, J., Zaroff, J. G. & Goldstein, B. A. Donor Predictors of Allograft Use and Recipient Outcomes After HeartTransplantation. *Circ-Heart Fai* **l6**, 300–309, https://doi.org/10.1161/Circheartfailure.112.000165 (2013).
14. *Organ Procurement and Transplantation Network*, https://optn.transplant.hrsa.gov/data/ (2015).
15. Medved, D., Nilsson, J. & Nugues, P. Streamlining a Transplantation Survival Prediction Program with a RDF Triplestore. Paper presented at *9th International Conference on Data Integration in the Life Sciences* https://doi.org/10.1007/978-3-642-39437-9 (2013).
16. Prud, E., Seaborne, A. & others. Sparql query language for rdf. (2006).
17. Anderson, T. W. & Darling, D. A. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, 193–212 (1952).
18. Schemper, M. & Heinze, G. Probability imputation revisited for prognostic factor studies. *Statistics in medicine* **16**, 73–80 (1997).
19. Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters* **27**, 861–874 (2006).
20. DeLong, E. R., DeLong, D. M. & Clarke-Pearson, D. L. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 837–845 (1988).
21. Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L. & Rosati, R. A. Evaluating the yield of medical tests. *Jama* **247**, 2543–2546 (1982).
22. Kang, L., Chen, W., Petrick, N. A. & Gallas, B. D. Comparing two correlated C indices with right-censored survival outcome: a one-shot nonparametric approach. *Statistics in medicine* **34**, 685–703 (2015).
23. Hosmer, D. W. Jr & Lemeshow, S. *Applied logistic regression*. (John Wiley & Sons, 2004).
24. Cockcroft, D. W. & Gault, M. H. Prediction of creatinine clearance from serum creatinine. *Nephron* **16**, 31–41 (1976).

Acknowledgements

This work is based on OPTN data as of October 1, 2013 and was supported in part by the Health Resources and Services Administration contract 234-2005-370011C. The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organisations imply endorsement by the U.S. Government. This research was supported by Swedish National Infrastructure for Computing, Swedish Heart-Lung Foundation, Swedish Society of Medicine, Government grant for clinical research, Region Skåne Research Funds, Donation Funds of Skane University Hospital, Anna-Lisa and Sven Eric Lundgrens Foundation, the Crafoord Foundation, the Swedish Research Council, and the eSSENCE program. The supporting sources had no involvement in the study.

Author Contributions

All authors contributed to the study design and data interpretation. D.M., P.H., P.N. and J.N., undertook the analysis and validation of the data. D.M., P.N. and M.O. performed the computer programming. D.M., P.N. and J.N. drafted the initial report, and all authors contributed to the final draft. Data was provided from the UNOS registry by staff at the US, United Network for Organ Sharing, and compiled by B.A. and J.N.

Additional Information

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Simulating the Outcome of Heart Allocation Policies using Deep Neural Networks

Simulating the Outcome of Heart Allocation Policies Using Deep Neural Networks

Dennis Medved[1], Pierre Nugues[1], and Johan Nilsson[2]

***Abstract*— We created a system to simulate the heart allocation process in a transplant queue, using a discrete event model and a neural network algorithm, which we named the Lund Deep Learning Transplant Algorithm (LuDeLTA). LuDeLTA is utilized to predict the survival of the patients both in the queue and after transplant. We tried four different allocation policies: wait time, clinical rules and allocating the patients using either LuDeLTA or The International Heart Transplant Survival Algorithm (IHTSA) model. Both IHTSA and LuDeLTA were used to evaluate the results. The predicted mean survival for allocating according to wait time was about 4,300 days, clinical rules 4,300 days and using neural networks 4,700 days.**

I. Introduction

Allocation policies in heart transplantation are used to decide how patients awaiting transplant will be paired with hearts from donors. There is a trade-off between medical justice, giving everyone an equal chance for a transplant, and medical utility, which aims at making the best use of a scarce resource [5].

Predictions models are, most of the time, optimized for the prediction of a single patient, and not applicable to a larger group of patients. This is the reason why the simulation of the whole queue system in an organ allocation process better fits the goal of selecting a policy that maximizes the benefit over all the patients.

Simulating a transplantation queue requires the creation of a model of the queue. This model can thereafter be used to simulate the impact of different policies, on several possible metrics. Examples of potential metrics are the number of deaths in the waiting list, the mean survival time after transplant, and the end size of the waiting list.

The selection of the best allocation policy can be seen as an optimization problem, where you try to maximize predefined metrics by selecting an appropriate policy.

II. Previous Work

There are several papers that detail organ allocation simulation for different organs, for example liver or lung [10, 18, 20, 14, 15, 7], but only a few that model heart allocation [9, 16]. All of the papers describe the use of a discrete event model as their main simulation model [2].

*This research was supported by Heart Lung Fondation, The Swedish Research Council, and the eSSENCE program.

[1]Department of Computer Science, Lund University, Lund, Sweden {dennis.medved, pierre.nugues}@cs.lth.se

[2]Department of Clinical Sciences Lund, Cardiothoracic Surgery, Lund University and Skåne University Hospital, Lund, Sweden johan.nilsson@med.lu.se

The main difference between our simulation model and those described in these articles is that they have used traditional statistical models such as Cox regression or sampling from a probability distribution, and predefined rules to select the patients, while our system uses machine learning to predict the status of the patients in the queue and the post-graft survival instead. We also used a survival model as an allocation policy to prioritize the patients with the highest predicted survival instead of rules.

III. Materials and Methods

A. Data Source

UNOS administers the only Organ Procurement and Transplantation Network in the United States of America [13], and is a non-profit organization. The patient data that we used was obtained from the UNOS database. The database contains data from October 1, 1987 and onwards. In the database, there is information that encompass recipient, donor, and transplant data. It includes almost 500 variables reflecting different attributes of the patients.

The Ethics Committee for Clinical Research at Lund University, Sweden approved the study protocol. The data was de-identified prior to analyzing it and the institutional review board waived the need for written informed consent from the participants.

B. Study population

We included all adult ($>$ 17 years) patients that were entered to the wait list and donors from a ten year period, corresponding to January 2003 until December 2012. The number of potential recipients was 30,584 and donors was 18,982 during this time period.

We split this data set in half, using the first half of the patients to develop the models, and the second half to simulate the allocation process and to be used as validation set.

We imputed the missing values for a particular variable by choosing a random value from a discrete uniform distribution of the non-missing values in that variable, following the method in [17].

C. System Model

We used a discrete event model to simulate the allocation process. Mainly because the nature of the problem lends itself to be described with such a model. See Figure 1, for a block diagram on how an allocation model is constructed.

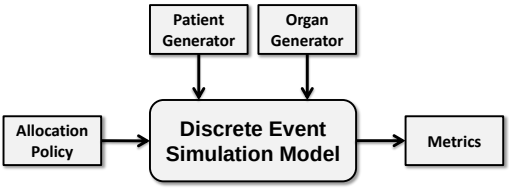


Fig. 1. The basic structure of an organ allocation simulation system.

D. Allocation Policies

An allocation policy is used to decide how to prioritize the patients in the waiting list, with regard to the organs coming from the donors. A policy, for example, may require, as prerequisite, that the patient and the organ match in blood type and then prioritize patients, first by geographical location, and secondly by acuteness of the patient condition using some metric.

We evaluated allocation policies that we describe below. All these policies have the requirement that the patient must be blood group compatible with the organ, otherwise the risk of graft rejection is too high [4].

Longest Wait: The patients in the waiting list are ranked according to their wait time. The patient who has waited the longest is selected for each donor candidate that is generated.

Clinical Rules: Patients within a weight difference of 20% from the donor are eligible for transplant and no donation from female to male is allowed. Among those, the recipients are prioritized in the following order:

1) Identical blood group and both recipient and donor age $\leq$ 35 years;
2) Identical blood group and a donor age $<$ recipient age +15 years.

Neural Networks: We use a neural network that can predict the survival time of a recipient-donor pair. Then, given a specific transplantation day, we create the Cartesian product of the donors generated by the simulator for that day with the current wait list. We apply the neural network model to all the possible pairs resulting from the product. The predicted survival times are sorted in descending order. A greedy algorithm selects the patient with the highest survival time after transplant, for each donor.

We used two neural network models: the IHTSA model [12] and the Lund Deep Learning Transplant Algorithm (LuDeLTA or LuΔ) described in Sect. IV.

E. Metrics

Metrics are used to measure some property of the allocation system. These are usually divided into two main types: utility and equity. This corresponds to making the best use of a scarce resource, and giving everyone an equal chance for a transplant.

Examples of utility measures are pre-transplant deaths, patients removed for other reasons, and survival time after transplant. The total number of transplants, differences in waiting times, and probability of transplants are examples of equity measures.

F. Patient and Organ Generation

We used the data set for validation as the basis for simulating the flow of patients and organs, by utilizing a stochastic process to select which of the real patients or organs arrive at certain time points. We chose a Poisson process to simulate the arrival of recipients and donors [11]. This is achieved by selecting patients, without replacement, from the all of the real patients from that specific year.

G. Discrete Event Simulation Model

The allocation simulation model begins at a certain date with a starting wait list of patients, who are selected from the year preceding the start date.

The date is then incremented in a discrete manner and the status of the patients in the waiting list is updated. The status update uses a neural model, which estimates the mortality in the wait list. The patients who are predicted as dead are removed from the wait list.

Following the status update, the patient generator simulates the addition of new patients to the waiting list and the organ generator produces the arrival of new transplantable hearts.

An allocation policy is then used to prioritize the patients in the waiting list. The highest priority patient in the waiting list always accepts the organ from a generated heart donor.

A neural model is then used to simulate the length of the post-graft survival time of the patients that were transplanted.

This process is then repeated for each date from the starting point to the end. When the simulation is finished, it calculates and outputs the metrics.

Algorithm 1 shows the pseudo-code of the simulation process.

IV. Implementation Details

A. Pre and Post-transplant Models

We created two models; one to simulate the removal of patients from the wait list, mainly caused by death, and the other to predict the survival after heart transplant. Similar model architecture is used. The main difference are the input features. The pre-transplant prediction uses 87 features, while the post-transplant utilizes 267 features. We have called this model: Lund Deep Learning Transplant Algorithm (LuDeLTA).

We used the Partial Logistic Artificial Neural Networks (PLANN) for modelling patients with censored survival data [1].The model architecture consists of 20 neural networks, each of these networks predicting the probability of mortality of the patients at certain time points. These time points were chosen to divide the patients in 20 equally sized groups. The area under the graph of the probabilities is then used to calculate the median survival of the patients.

For the pre-transplant model we used 20 networks with four hidden layers and 128 nodes in each layer. For the post-transplant we instead utilized 20 networks each with two hidden layers with 32 nodes in each layer.

Algorithm 1 Pseudo code for a discrete event simulation model

```
procedure SIMULATEALLOCATION
    curDay ← 0
    waitList ← GENERATESTARTLIST
    dead ← [ ]
    transplanted ← [ ]
    while curDay < endDate do
        dead ←UPDATEPATIENTS(waitList)
        waitList ← waitList+ GENERATEPATIENTS
        donorOrgans ← GENERATEDONORORGANS
        transplanted ←ALLOCATE(waitList, donorOrgans)
        CALCULATESURVIVAL(transplanted)
        day ← day + 1
    CALCULATEMETRICS(waitList, dead, transplanted)
```

The hidden layers used the scaled exponential linear unit as activation function and the final output layer uses a sigmoid activation. We used binary cross entropy as the loss function and adagrad as the optimizer.

Dropout is a regularization technique for reducing overfitting in neural networks [19]. The idea behind dropout is to randomly drop units, together with their connections, from the neural network during training. The dropout rate controls the probability of a neuron being removed. We chose to use a dropout rate of 0.48 for each of the layers in LuDeLTA and no dropout in the pre-transplant model.

We used the Keras framework to create these machine learning models [3]. Keras enables the user to create and configure easily artificial neural networks (ANN) of different architectures. It serves as a high level abstraction that can use Theano, TensorFlow, or Microsoft Cognitive Toolkit as its back end. It utilizes Python as a programming interface.

B. Evaluation Procedure

In addition to our own model LuDeLTA, we used a second model to predict the post-graft survival of the patients: The International Heart Transplant Survival Algorithm (IHTSA) [12]. We evaluated the different allocation methods with both models, where we prioritized the patients with one neural allocation model, and evaluated the survival with the other model. Table I shows the four possible combinations.

TABLE I
PREDICATION/EVALUATION COMBINATIONS

Combination	Prediction	Evaluation
1	LuDeLTA	IHTSA
2	LuDeLTA	LuDeLTA
3	IHTSA	LuDeLTA
4	IHTSA	IHTSA

V. RESULTS

We evaluated the LuDeLTA models using the Area Under the Receiver-Operating Curve (AUROC) for the one year mortality [6], and the long time survival using The Harrells C-index [8] on the validation set. Results are shown in Table II. The predicted mean survival on the wait list without transplant was 447 days using our pre-transplant survival model. The results for the different allocation policies can be found in the Table III. The mean survival days after transplant policies based on the neural network models or wait time utilize all of the available organs, while using clinical rules lead to a discard of 124 hearts.

TABLE II
PERFORMANCE METRICS OF THE LUDELTA MODELS

Metric	Pre-transplant	Post-transplant
AUROC 1 year	0.89	0.66
C-index	0.80	0.61

VI. DISCUSSION

We used two predictive models to mitigate the bias introduced by using the same predictive model to both prioritize the patients during allocation and then to evaluate the survival for the same set of patients.

The hyperparameters of the LuDeLTA models such as topology, activation function and drop out, were chosen by empirical testing of the models using 5-fold cross-validation on the training data, to maximize the performance metrics.

We chose to only use half of the patients from the time period in the simulation, this is to minimize the bias of using the same patients in both training and validation of the models. This means that all the results obtained in Table III represent a queue with half of the patient, in contrast to the real historic UNOS queue. This could influence metrics such a mean wait time or mean survival time using a neural network as the allocation method. The latter because the potential number of recipient-donor pairs is lower to maximize the predicted survival on.

The reason that only the clinical rules discarded hearts for transplant, was that the only requirement for a transplant to occur for the others was blood group compatibility. The waiting list was sufficiently large to always have a compatible recipient for the donors.

In this paper, we have shown that an organ transplant queue can be simulated by utilizing neural networks to

TABLE III
RESULTS FROM SIMULATING HEART ALLOCATION POLICIES.

Allocation policy	Number transplanted	Number dead wait list	Number alive wait list	Mean survival IHTSA (days)	Mean survival LuDeLTA (days)	Mean wait time (days)
Wait time	9,469	5,485	444	4,285	4,309	139
Clinical rules	9,345	5,481	572	4,349	4,309	150
Neural network (IHTSA)	9,469	4,801	1128	4,976	4,719	150
Neural network (LuDeLTA)	9,469	4,993	936	4,541	5,668	110

predict survival, both pre- and post-transplant. Additionally we have shown that using neural networks as the allocation policy, could possibly result in longer survival post-transplant for the patients.

ACKNOWLEDGMENT

This work is based on OPTN data as of October 1, 2013 and was supported in part by the Health Resources and Services Administration contract 234-2005-370011C. The content is the responsibility of the authors alone and does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government. This research was supported by Heart Lung Foundation, The Swedish Research Council, and the eSSENCE program.

REFERENCES

[1] Elia Biganzoli et al. "Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach". In: *Statistics in medicine* 17.10 (1998), pp. 1169–1186.

[2] Christos G Cassandras and Stephane Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[3] François Chollet. *Keras*. `https://github.com/fchollet/keras`. 2015.

[4] DKe Cooper. "Clinical survey of heart transplantation between ABO blood group-incompatible recipients and donors." In: *The Journal of heart transplantation* 9.4 (1990), pp. 376–381.

[5] Walter Graham. "The UNOS statement of principles and objectives of equitable organ allocation". In: *Seminars in Anesthesia, Perioperative Medicine and Pain*. Vol. 14. 2. Elsevier. 1995, pp. 142–166.

[6] James A. Hanley and Barbara J. McNeil. "The meaning and use of the area under a receiver operating characteristic (ROC) curve." In: *Radiology* 143.1 (1982), pp. 29–36.

[7] Ann M Harper et al. "Organ transplantation policies: an update on a successful simulation project: the Unos Liver Allocation Model". In: *Proceedings of the 32nd conference on Winter simulation*. Society for Computer Simulation International. 2000, pp. 1955–1962.

[8] Frank E Harrell et al. "Evaluating the yield of medical tests". In: *Jama* 247.18 (1982), pp. 2543–2546.

[9] Wilbert B van den Hout et al. "The heart-allocation simulation model: a tool for comparison of transplantation allocation policies1". In: *Transplantation* 76.10 (2003), pp. 1492–1497.

[10] Jennifer Kreke et al. "Methods for special applications: incorporating biology into discrete event simulation models of organ allocation". In: *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*. Winter Simulation Conference. 2002, pp. 532–536.

[11] Averill M Law, W David Kelton, and W David Kelton. *Simulation modeling and analysis*. Vol. 2. McGraw-Hill New York, 1991.

[12] Johan Nilsson et al. "The International Heart Transplant Survival Algorithm (IHTSA): a new model to improve organ sharing and survival". In: *PloS one* 10.3 (2015), e0118644.

[13] United Network for Organ Sharing. *Organ Procurement and Transplantation Network Data*. 2017. URL: `https://www.unos.org/data/transplant-trends/#waitlists_by_organ` (visited on 08/23/2017).

[14] JP Ouwens et al. "Simulated waiting list prioritization for equitable allocation of donor lungs". In: *The Journal of heart and lung transplantation* 21.7 (2002), pp. 797–803.

[15] A Alan B Pritsker. "Organ transplantation allocation policy analysis". In: *OR/MS Today* 25.4 (1998).

[16] Douglas E Schaubel et al. "Analytical approaches for transplant research, 2004". In: *American journal of transplantation* 5.4p2 (2005), pp. 950–957.

[17] Michael Schemper and Georg Heinze. "Probability imputation revisited for prognostic factor studies". In: *Statistics in medicine* 16.1 (1997), pp. 73–80.

[18] Steven M Shechter et al. "A clinically based discrete-event simulation of end-stage liver disease and the organ allocation process". In: *Medical Decision Making* 25.2 (2005), pp. 199–209.

[19] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: `http://jmlr.org/papers/v15/srivastava14a.html`.

[20] David Thompson et al. "Simulating the allocation of organs for transplantation". In: *Health Care Management Science* 7.4 (2004), pp. 331–338.