A Review of Machine Learning Applied to Medical Time Series

CUED/F-INFENG/TR.702

September 2016

Jos van der Westhuizen and Joan Lasenby

0951-9211

Abstract

The field of medicine has witnessed numerous improvements in recent decades, yet many of the biological processes underlying the medical problems faced today remain a mystery. Recent advances in machine learning enable the extraction of features that would be nearly impossible for experts to find in the massive medical datasets. The techniques not only offer improved continuous patient monitoring but also provide novel empirical methods for solving the latest medical problems when combined with medical practitioners. This report is focussed on the physiological signals stemming from intensive care units. We argue that the ability of Long Short-term Memory (LSTM) Recurrent Neural Networks (RNNs) to infer long and short-term dependencies in time series make their combination with high-resolution physiological signals the best solution for predicting patient status. A brief background of the machine learning techniques that are often used for medical time series analysis is provided. Moreover, previous studies that applied machine learning to vital signs are reviewed and compared for both the neonatal and general intensive care unit settings.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

A pivotal challenge in science and technology is the assurance of making the optimal decision in any given situation. In recent developments, machine learning techniques have been able to produce compelling advances toward making the optimal decision given real world data, much of which is temporal in nature. Standard machine learning techniques are sufficient when data are independent over time. However, when it cannot be assumed that the data in different time steps are independent, as in most cases of human decision making, sequential machine learning techniques, modelling dependencies across time, become the logical approach.

Sequential machine learning models have been largely adopted in domains such as image captioning, language processing, and handwriting recognition [Lipton, 2015; Singh et al., 2012; Sutskever et al., 2014], but there have been relatively few applications in healthcare [Alagoz et al., 2010; Lipton et al., 2015; Schaefer et al., 2005]. McGlynn et al. [2003] has shown that patients receive correct diagnoses and treatment less than 50% of the time (at first pass). There has also been distinct evidence of a 13-to-17-year gap between research and use clinical practice [Bauer, 2002]. The combination of the high cost of training human doctors in the complexity of modern healthcare ($\pm$10 years) and out-of-date clinical care, means that improvement in healthcare is bound to plateau [Bennett and Hauser, 2013]. Whilst still in its infancy, it has been argued that machine learning holds substantial promise for the future of medicine, and for our ability to tailor care to the particular physiology of the patient [Abston et al., 1997; Clifton et al., 2015; Cooper et al., 1997; Lapuerta et al., 1995; Mani et al., 1999; Morik et al., 2000; Ohmann et al., 1996; Sboner and Aliferis, 2005; Svátek et al., 2003; Vairavan et al., 2012].

Clinicians typically determine the course of treatment given the current health status of the patient as well as some internal estimate of the outcome of possible future treatments. The effect of treatments for a given patient is non-deterministic (uncertain), and predicting the effect of a series of treatments over time compounds the uncertainty [Bennett and Hauser, 2013]. There is a growing body of evidence that complex medical treatment decisions are better handled with the aid of modelling, compared to intuition alone [Meehl, 1986; Schaefer et al., 2005]. Intensive care units (ICUs) are responsible for much of the increase in the healthcare budget [Halpern and Pastores, 2010]. As such, ICUs are a major target in our drive to limit healthcare costs. Evolving systematic processes, such as haemorrhage, sepsis, and acute lung injury are mostly disguised by the body's own defensive mechanisms. As a result, serious internal pathological processes are often hidden from the causal observer. Machine learning holds great potential for improving critical care outcomes: (i) providing aid to clinicians through accurate and timely detection of pathological processes; (ii) efficient patient stratification according to risk, combined with optimal resource allocation; and (iii) rapid identification of proper treatments, as well as verification that provided treatments are functional.

Generally, the management of acutely ill patients is arduous, even more so when patients are preterm infants. Physiological time-series data from the neonatal ICU (NICU), constitutes an abundant and largely untapped source of medical insights. The vast majority of studies on clinical applications for machine learning have been based on ICU for adults [Braga et al., 2014; Clifton et al., 2015; Gultepe et al., 2014, 2012; Hravnak et al., 2011; Kim et al., 2011; Meyfroidt et al., 2009; Pirracchio et al., 2015; Ramon et al., 2007; Tarassenko et al., 2011; Tu and Guerriere, 1993; Zhai et al., 2014; Zhang et al., 2007]. Fewer studies have been based in the NICU because the data is often more noisy and sparse compared to that in the ICU. In the NICU, cot-side monitors are used to display continuous physiological signals of the patients to the tending clinicians. Alarms are sounded to indicate a change in the patient's state when one or more of the patient's vital signs, such as heart rate (HR), blood pressure (BP), or temperature, breach predetermined thresholds. The intention of these alarms is to allow for timely intervention by clinical staff. It is reported that current technology often provides false alarms and clinicians ignore the alarms when signals are observed to look healthy even though the signals might exceed certain 'normal' thresholds. This results in a muddling of alarms with clinical activities [Da Costa, 2016]. There is a need for models that analyse the dynamic waveforms of signals over time to allow for serendipitous discovery of important features, in lieu of combining domain knowledge and hand-engineered features. The Holy Grail of neonatology has been to stratify the infants according to risk. Many prediction methods attest to the difficulty of the task, ranging from the well-established Apgar score to modern techniques such as CRIB, SNAP, and SNAPPE scores [Saria et al., 2010]. Current challenges encompass identification of compelling clinical problem areas, construction of powerful predictive models based on quality data and the prudent validation of these models.

Most research on the application of machine learning in medicine has made use of low-resolution (data sampled at frequencies lower than 1 Hz) physiological data [Choi et al., 2015; Colopy et al., 2015; Ghassemi et al., 2015; Güiza et al., 2013; Ongenae et al., 2013; Ryan et al., 2013; Stanculescu et al., 2014]. Critical information is embedded in the high-resolution waveforms of physiological signals from the human body. Whilst long-term trends could indicate significant correlations, our work argues that the combination of long-term trends and high-resolution waveform dynamics paves the way to achieving unparalleled diagnosis accuracies with machine learning. To elucidate the amount of information lost when using low-resolution data, Figure 1.1 illustrates an ECG signal sampled at 240 Hz and 1 Hz. Studies have made use of trends of HR values averaged over an hour or more to predict patient outcomes [Caballero Barajas and Akella, 2015; Lipton et al., 2015], where high-resolution changes would remain unnoticed in these models. Research indicates that complex datasets, such as those containing high-resolution physiological signals, contain information that is neither extractable with conventional methods of analysis nor visually apparent [Goldberger et al., 2000; Ivanov et al., 1999; Vikman et al., 1999]. Such datasets promise to be of clinical value in forecasting sudden death or cardiopulmonary catastrophes during medical procedures such as critical care or surgery. Goldberger [1996] suggests that the embedded information may relate to basic mechanisms in physiology and molecular biology.

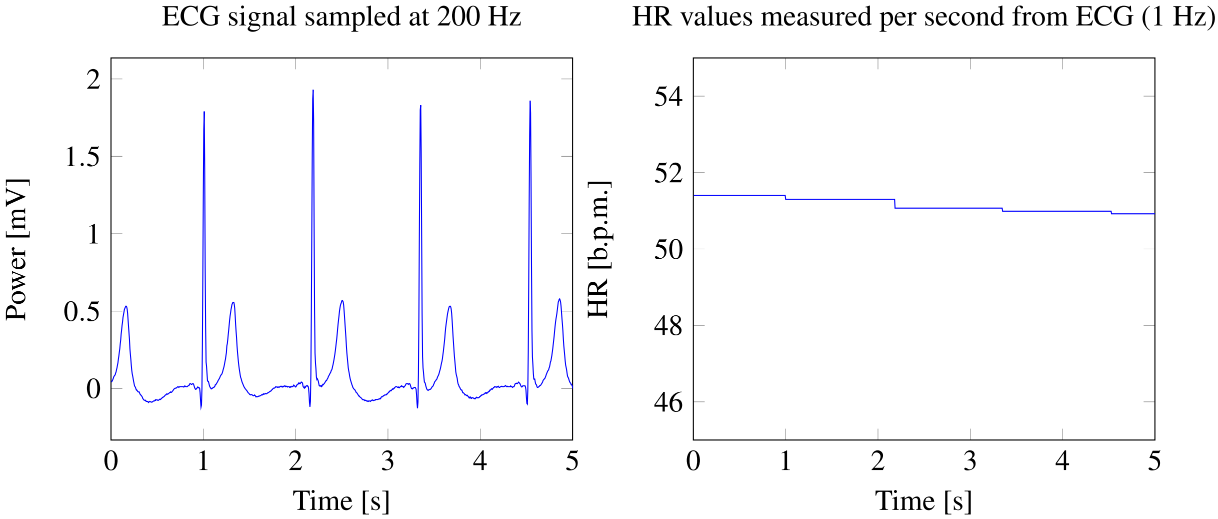


Fig. 1.1 Comparison of high-resolution waveforms (Left) to low-resolution per second heart rate (Right) for an ECG signal. The graphs highlight the amount of information lost when downsampling high-resolution waveforms.

Feature extraction is a promising technique for reducing the resolution of data, whilst retaining the embedded information. Studies such as those by Lipton et al. [2015] and Temko et al. [2011] have successfully implemented feature extraction methods combined with classification algorithms (Naïve Bayes (NB) or Support Vector Machines (SVMs)) on medical time series. Classification algorithms such as NB and SVMs 'cannot directly deal with temporal data' [Ongenae et al., 2013]. As mentioned before, feature extraction requires domain knowledge, and the research reported here takes a more non-parametric approach, in order to have the models learn the important features heuristically. Gaussian Processes (GPs), Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), and Recurrent Neural Networks (RNNs) are machine learning techniques that are well suited for sequential data [Baccouche et al., 2011; Bishop, 2006; Hill et al., 1996; Lafferty et al., 2001]. These techniques provide the means to model and analyse the underlying features of temporal data. When applied to medicine the task often requires prediction or classification of multiple categories over the continuous time axis and is, therefore, similar to sequential learning tasks such as those in natural language processing, where there have been major recent research successes [De Mulder et al., 2015]. Two main classes of sequential techniques have been proposed in medicine: (i) intensity-based point process modelling techniques such as Hawkes processes [Liniger, 2009; Zhu, 2013]; and (ii) continuous-time Markov chain-based models [Johnson and Willsky, 2013; Lange et al., 2015; Nodelman et al., 2002], but both are expensive to generalise to multilabel and nonlinear settings [Choi et al., 2015]. Moreover, these techniques often make strong assumptions about the data generation process, which might not be valid in ICU monitored datasets.

The key aim of this research is to learn an accurate representation of patient status over time and to leverage the representation to predict future clinical events of the patient. This study proposes the implementation of RNNs to learn such representations based on the time series data collected in the ICU. Most of these time series are of different lengths, and RNNs have been shown to be successful in dealing with this sort of sequential data [Graves, 2013; Graves and Jaitly, 2014; Sutskever et al., 2014; Zaremba et al., 2014]. The focus will be specifically on the long short-term memory (LSTM) variation of RNNs, which enables the accurate modelling of long and short term dependencies, abundant in high-resolution physiological signals. HMMs and their combination with GPs will also be explored and compared to the LSTM architecture in this work.

RNNs are recurrent versions of Neural Networks (NNs); but while they have received much attention in the medical research field, any superior performance has yet to be proven [Meyfroidt et al., 2009; Sargent, 2001; Tong et al., 2002]. Moreover, the black-box nature of NNs results in a lack of interpretability, which is especially unfavoured in the medical community. Recent decades have witnessed a change in the spectrum of diseases, from infectious conditions to tumours and cardiovascular diseases.This has hindered the translation of bench research into clinical efficacy. Scientific knowledge has lead to a reduction of infectious diseases and malnutrition, but evidence-based research has yet to find the underlying mechanisms of tumours or cardiovascular diseases. There is a possibility to develop a diagnostic algorithm and treatment strategy by learning from millions of examples. Much of medical science is non-deterministic and cannot be predicted accurately via formulae – the random errors in medicine stem from the largely unknown underlying molecular pathways [Zhang, 2016]. As the biological or pathological reasonings behind many treatments are often vague, a black-box machine learning approach would not be a far stretch from these current approaches. It might even provide a more relevant solution for aiding doctors in their diagnoses.

Chapter 2

Literature Review

The aim of this chapter is to provide a concise overview of research in the application of machine learning in medicine and an overview of the literature on the machine learning techniques employed for sequence modelling in this study.

2.1 Machine Learning

This section serves as a background on the machine learning techniques and performance metrics often referred to in this study.

2.1.1 Naïve Bayes Classifiers

The Naïve Bayes (NB) classifier is a probabilistic statistical classifier based on Bayes' theorem

$$P(h|e) = \frac{P(e|h)P(h)}{P(e)} \tag{2.1}$$

and it assumes conditional independence among the attributes (see Figure 2.1). Here h refers to the hypothesis and e is the evidence. The NB classifier assigns a class label $\hat{y} = C_k$ as follows:

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{2.2}$$

where K is the number of classes and n is the number of different features. The conditional independence assumption is unrealistic in most medical cases, as patient physiology and symptoms are often very highly correlated, for example, HR and respiratory rate (RR). Despite this drawback, the NB algorithm has yielded excellent results in medical machine learning, and as a result, is widely used [Kononenko, 2001]. The good performance could be attributed to the data used in most studies having a low-resolution, which leads to a reduction or removal of dependence between signals. The advantage of this classifier is that it requires a small training dataset and the training is rapid, owing to the simplicity of the model.

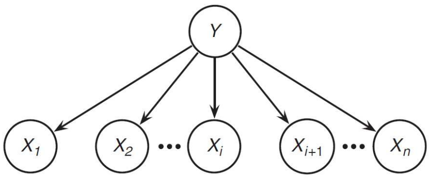


Fig. 2.1 Naïve Bayes classifier example, illustrating the independence assumption. X_1 to X_n are feature vectors that are independent of each other (no connecting edges), given the label Y. (Adapted from: Zheng and Webb [2010])

2.1.2 Bayesian Networks

Bayesian Networks (BN), a form of supervised learning [Meyfroidt et al., 2009], are probabilistic graphical models that specify a joint probability distribution on a set of random variables. The two components of a Bayesian network are a graph structure G, illustrating dependencies, and a probability distribution θ. G is a directed acyclic graph with nodes representing each variable. The conditional dependence of variable X_i on variable X_j is encoded by an edge in G directed from node i to j. X_i is called the parent of X_j. A variable X_i is independent of its non-descendants given all its parents $\mathbf{Pa}(X_i)$ in G [Wang et al., 2011]. Therefore the joint probability distribution over $\mathbf{X}$, a vector of variables $X_1,...,X_n$, can be decomposed by the chain rule

$$p(\mathbf{X}) = \prod_{i=1}^{n} p(X_i|\mathbf{Pa}(X_i)) \tag{2.3}$$

The parameter set $\theta = \{\theta_i\}_{i=1....n}$ specifies the parameters of each conditional distribution $p(X_i|\mathbf{Pa}(X_i))$ in Equation 2.3. The form of the distribution determines how θ_i is interpreted. If the distribution is Gaussian, θ_i may contain the values of the variance and mean; when the distribution is multinomial, θ_i is a conditional probability table. A network that best matches the training dataset is found by a scoring function that evaluates each network with respect to the training data. The resulting network returns the label c given the variables $X_1,...,X_n$, that maximizes the posterior probability $P(c|X_1,...,X_n)$ [Friedman et al., 1997].

2.1.3 Decision Trees

Decision tree (DT) classifiers construct a flowchart-like tree structure in a top-down, recursive, divide and conquer manner as shown in Figure 2.2. Intuitively, this is the closest to what physicians do in the ICU. Attributes (nodes of the DT) are selected in such a way as to partition the data records in the purest way by class in terms of the class values. The class-focussed visualisation of the data allows users to readily understand the overall structure of the model. When too many parameters cause the model to become too complex, an approach called *pruning* is used to remove the least important nodes and in doing so, makes the model less susceptible to overfitting and more interpretable. DTs where the target variable takes a continuous value are called regression trees.

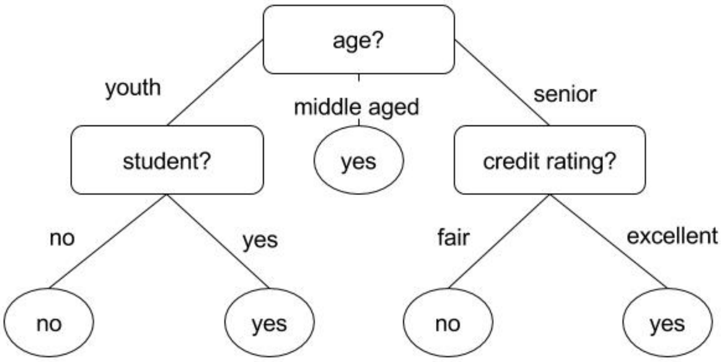


Fig. 2.2 A simple decision tree example for predicting whether a customer will buy a given kind of computer. Each decision node (rectangles) represents a choice between a number of alternatives called branches (edges). The leaf nodes (ovals) represent a classification or *decision*.

An ensemble technique called bagging, short for ‘bootstrap aggregating’ [Breiman, 1996], can be used to create another tree-based technique called Random Forests (RF), where several different trees are trained on different subsets of the data and outputs are averaged [Murphy, 2012]. Whereas bagging is parallel, boosting is the sequential application of the DT to weighted versions of the data, where more weight is given to samples that were misclassified in previous rounds [Schapire, 1990; Yoo et al., 2011; Zhang and Szolovits, 2008]. Ensemble techniques, such as bagging and boosting, have demonstrated improved classification performance in healthcare [Meyfroidt et al., 2009; Moon et al., 2007; Ramon et al., 2007; Santos-Garcıa et al., 2004; Zhou et al., 2002].

2.1.4 Support Vector Machines

Support Vector Machines (SVMs) find a (linear) optimal separating hyperplane in a high dimensional feature space of that dataset. The hyperplanes are decision boundaries between different sets of objects, where the hyperplane with the maximal margin is determined via the use of support vectors (see Figure 2.3). Compared to 16 classification methods including Neural Networks, and 9 regression methods, Meyer et al. [2003] found that SVMs provide good classification accuracies, but are not always top ranked. Murphy [2012] argues that the popularity of SVMs are not due to their superior performance, but to ignorance and lack of high quality software implementations of alternatives such as GPs and Relevance Vector Machines. The use of SVMs is sensible under circumstances where the output is structured and likelihood-based methods could be slow. Relevance Vector Machines are a Bayesian approach to SVMs, which operates over distributions and provides probabilistic classification instead of point estimates, making them more suitable when speed is important. SVMs are particularly efficient in making predictions, but training on large datasets can quickly become intractable, due to memory limitations and the complexity involved in inference [Scalzo and Hu, 2013]. SVMs are designed for two-class classification problems and a common modification for multi-class classification is to reduce the problem into multiple binary problems (one-vs-one or one-vs-all). Figure 2.3 illustrates an example of SVM classification between two classes, for simplicity, only two dimensions are shown.

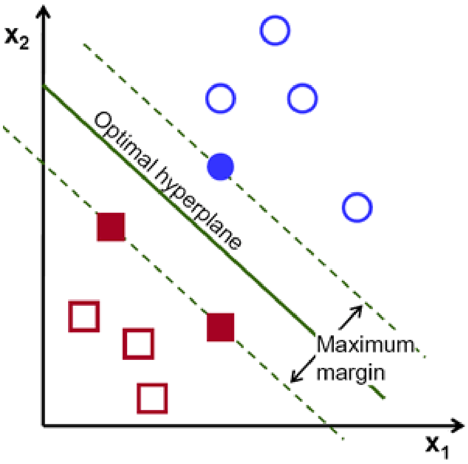


Fig. 2.3 SVM example, showing the maximum margin between support vectors (filled shapes). The circles and squares are data vectors from two different classes. (Adapted from: OpenCV [2014])

2.1.5 Gaussian Processes

A Gaussian Process (GP) is fully specified by its covariance function $k(x,x')$ and mean function $m(x)$ [Rasmussen and Williams, 2006]. The mean and covariance are a vector and a matrix, respectively, for this natural generalisation of the Gaussian distribution. The GP is defined over functions, where the Gaussian distribution is over vectors. The function f distributed as a GP with covariance function k and mean function m is written as

$$f \sim GP(m,k) \tag{2.4}$$

Essentially, GPs dispense with the parametric model and instead define a prior probability distribution over functions directly. Although the function space is infinite, in practice the working space is finite, because only the values of the function at a discrete set of input values, corresponding to the training data, have to be considered. Thus GPs can be defined as a probability distribution over functions $y(\mathbf{x})$ such that the set of values of $y(\mathbf{x})$ evaluated at an arbitrary set of points $\mathbf{x}_1,...,\mathbf{x}_N$ jointly have a Gaussian distribution [Bishop, 2006]. The family of functions, such as the choice of mean and covariance functions, remain infinite and have to be specified based on prior knowledge.

Similar to SVMs, GPs are kernel methods. They have a small number of tunable parameters, allowing for multi-dimensional input and provide full predictive distributions as opposed to point predictions normal to other methods. GPs are increasingly being adopted in probabilistic health informatics as they offer a principled manner of coping with noisy or incomplete data [Clifton et al., 2015]. The ability to learn complex non-linear decision boundaries allows GPs to outperform traditional methods such as logistic regression (Section 2.1.6), and motivates implementation in the ICU, especially the NICU.

There are two main advantages in the use of Gaussian Processes (GPs), which allow a Bayesian use of kernels for learning. First, the Bayesian evidence framework (updating the probability of a hypothesis as more data becomes available) applied to GPs allows learning of the parameters of the kernel. Second, GPs provide fully probabilistic predictive distributions, including the estimates of the uncertainty of the predictions. The undesirable behaviour of Bayesian Linear Regression, of being very confident in its predictions when extrapolating outside the region occupied by the basis functions [Bishop, 2006], is thus mitigated by GPs. A disadvantage is that the naïve implementation of GPs requires computation of $\mathscr{O}(n^3)$, where n is the number of training samples [Chu and Ghahramani, 2009]. Consequently, the simple implementation on current desktop machines can handle problems with at most a few thousand training samples. Recent studies [Snelson and Ghahramani, 2005; Titsias, 2009] have proposed approximation techniques to allow for scalable applications of GPs.

The techniques can broadly be classified into two classes: i) those relying on approximate matrix-vector multiplication conjugate gradient methods, and ii) techniques based on *sparse* methods, where the full posterior is approximated by expressions involving matrices of lower rank $m \ll n$ [Quinonero-Candela et al., 2007]. Where well-calibrated probabilistic output matters and speed has a lower priority (e.g., active learning or control problems), GPs provide well-suited solutions [Murphy, 2012].

2.1.6 Logistic Regression

Logistic Regression (LR) is one of the most widely-used methods in medicine [Bagley et al., 2001]. The technique measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities as described by

$$p(y|\mathbf{x},\mathbf{w}) = \mathrm{Ber}(y|\mathrm{sigm}(\mathbf{w}^T\mathbf{x})) \tag{2.5}$$

which takes inputs **x** and weights **w**. The label y has a Bernoulli distribution, and the *sigmoid* function, also known as the *logistic* or *logit* function, is defined as

$$\mathrm{sigm}(\mathbf{w}^T\mathbf{x}) \triangleq \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})} \tag{2.6}$$

The sigmoid function maps the whole real line to [0,1], which is necessary for the output to be interpreted as a probability. It should be noted that LR is a form of classification and not regression [Murphy, 2012]. The sensitivity of LR models to data of poor quality depends on the method used for fitting the model, for example, the maximum likelihood approach is sensitive to poor quality data [Pregibon, 1981]. The model's prediction y_{pred} is the class whose probability is maximal:

$$y_{pred} = \mathrm{argmax}_i P(y = i|\mathbf{x},\mathbf{w}) \tag{2.7}$$

2.1.7 Neural Networks

NNs consist of computational nodes that emulate the functions of neurons in the brain. The nodes referred to as neurons, are connected via adjustable weights which are modified during the training of the model. Each neuron has an activation function transforming all its inputs into a specific output according to the activation function generally defined as

$$y = f(\phi(\mathbf{x}, \mathbf{w})) \tag{2.8}$$

where ϕ is usually a linear function (such as a sum) of the inputs $\mathbf{x}$ and the weights $\mathbf{w}$. Function f is chosen from a small selection of functions, including the popular *sigmoidal nonlinearity* given in Equation 2.6, producing output between 0 and 1, and the *tanh* function producing outputs between -1 and 1 [Cheng and Titterington, 1994]. The neurons are classified into three types of layers (input, output, and hidden), where the input layer is not considered as it merely passes the input values to the next layer. A popular output function is the *softmax* function σ defined for a specific class j as

$$\sigma(\mathbf{xw})_j = \frac{e^{\mathbf{xw}_j}}{\sum_{k=1}^{K} e^{\mathbf{xw}_k}}, \tag{2.9}$$

where subscripts of the weight matrix $\mathbf{w}$ indicate weight matrices for each specific class, and K is the number of classes.

The most widely used NN is the multi-layer perceptron with back-propagation because its performance is considered superior to other NN algorithms [Delen et al., 2005; Yoo et al., 2011]. Back-propagation is a method used for calculating the gradient of a loss function with respect to all the weights in the network. The gradients are then used to train the network by updating the weights through some function, such as the commonly used stochastic gradient descent (SGD):

$$w_t = w_{t-1} - \eta \frac{\partial E}{\partial w} \tag{2.10}$$

where E is the specified loss function, η is the specified learning rate, and t denotes the current weight update [Orr and Müller, 2003]. NNs were considered to be the best classification algorithms prior to the introduction of DTs and SVMs. Although NNs can handle noisy training data and yield decent classification performance on unseen data, there remain disadvantages in their use. NNs have a large number of parameters, resulting in training that is computationally expensive and time-consuming. The performance of NNs greatly depends on the hyperparameters (number of layers and neurons) and parameter optimisation. Furthermore, NNs lack the transparency or exploratory power of, for example, DTs. This makes it difficult for medical domain experts to properly understand the classification decisions reached, which has lead to difficulties with their implementation in practice. Regularisation is a technique that introduces additional information to a model, usually to penalise complexity, in order to prevent overfitting. For NNs this is mostly done via *dropout*, which is a recently developed technique that effectively approximates an entire ensemble of neural networks. Dropout temporarily removes a random selection of neurons in each iteration during training [Pham et al., 2014; Srivastava et al., 2014; Zaremba et al., 2014], and as a result, prevents overfitting to the training data and improves generalisation.

2.1.8 Performance Evaluation

The testing process in classification is usually computationally inexpensive compared to the complex training process. The approaches for validation of the model performances include cross-validation [Efron, 1983] and data splitting [Picard and Berk, 1990] into *training*, *testing*, and occasionally *validation* sets. Cross-validation involves randomly splitting the data into n folds (subsets of the data), and $n-1$ folds are then used to train the classifier and the remaining fold is used to test the performance. An unbiased estimate of the classifier performance is then obtained by rotating or 'shuffling' the data and repeating the above n times [Greene et al., 2007].

Performance metrics are useful for comparing the quality of classifications and predictions across systems and studies. Measures (define below) often used in binary classification include accuracy, precision, recall (sensitivity), F-score, specificity, and AUC. For multi-class classification, average accuracy, error rate, micro-averaged precision, micro-averaged F-score, and macro-averaged F-score are often used for performance evaluation [Sokolova and Lapalme, 2009]. In healthcare the preferred measures of performance are sensitivity and specificity. *Sensitivity* is defined as the percentage of correctly identified true positive events, for example, the percentage of patients with sepsis, correctly classified as having the condition.

$$\text{Sensitivity} = \frac{\text{True positve events}}{\text{True positive events + False negative events}} \tag{2.11}$$

Specificity is defined as the percentage of correctly classified true negative events, for example, the percentage of healthy patients who are correctly identified as being healthy.

$$\text{Specificity} = \frac{\text{True negative events}}{\text{True negative events + False positive events}} \tag{2.12}$$

The Receiver Operating Characteristic (ROC) curve (see Figure 2.4) illustrates the classifier performance as the discrimination threshold is varied. The curve is a plot of the sensitivity (true positive rate) against 1 - specificity (false positive rate). In the clinical setting, it is imperative to maintain high sensitivity. The area under the curve (AUC) of the ROC, popular in medical research, provides a measure of the performance of a classifier over all discrimination thresholds of sensitivity and specificity. For a randomly chosen pair of positive and negative samples, the AUC can be interpreted as the probability that the classifier will assign a higher score to the positive sample [Scalzo and Hu, 2013]. However, AUC is not an ideal indicator in the clinical setting, because a high AUC value could arise from a high specificity and a relatively low sensitivity. AUC was chosen as the comparative measure for literature reviewed in this study because it was the most consistently reported metric. A perfect classification model, which correctly classifies all positive and negative examples, would have an AUC value of 1 [Braga et al., 2014]. It should be noted that currently there is no well-developed multi-class ROC analysis [Lachiche and Flach, 2003; Sokolova and Lapalme, 2009] and that most of the studies reported here involve binary classification.

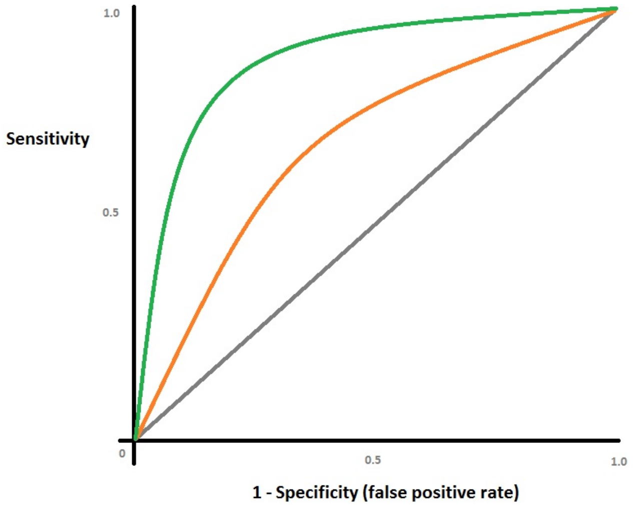


Fig. 2.4 Receiver Operating Characteristic curve example. The diagonal line with an AUC of 0.5 indicates random classification, and the green line represents good classification with an AUC of roughly 0.85. (Adapted from: Morgan [2015])

2.2 Machine Learning in Medicine

A large variety of machine learning techniques have been applied to various medical problems. In the interest of brevity, we only report studies that are the most important to our work. Data and therefore outcomes in the NICU are different to those in general medicine. Consequently, reported studies are grouped by medical domain to avoid confusion in technique comparisons.

2.2.1 General Medical Data

In this section, we report studies that involve physiological signals that are less noisy than those stemming from NICUs.

Support Vector Machines

Scalzo and Hu [2013] compared kernel spectral regression (SR-KDA) and SVMs for the reduction of false alarms for intracranial pressure (ICP) using a semi-supervised approach. The study aimed to determine whether semi-supervised learning methods, that can naturally integrate unlabelled data samples in the model, could improve accuracy. This is important to our study, because, as with most medical datasets, our datasets had insufficient annotations and the annotations are not always accurately timed. The ICP signals used in the study were recorded at 240 Hz for 108 neurosurgical patients and filtered with a 40 Hz low-pass filter. The Bayesian tracking algorithm [Scalzo et al., 2012] was used to extract 4 features from the 20-minute segments in which the signals were analysed. Three-fold cross-validation was used for performance analysis. With a sensitivity of 0.99, the specificity improved from 0.03 (supervised) to 0.16 (semi-supervised) for SVMs. The study clarified that more labelled data is always beneficial and that the optimal proportion of labelled and unlabelled samples is application dependent. Given a model with effective regularisation, the supervised approach would be preferred.

Logistic Regression

LR has been used to predict the need for paediatric ICU transfer within the first 24 hours of admission [Zhai et al., 2014]. The dataset consisted of temporal, narrative and nominal measurements collected from the EHRs of 7,298 patients. The performance analysis was done by means of 10-fold cross-validation. The model yielded a 0.91 AUC value - outperforming two published Early Warning Scores (EWS).

APACHE III is one of the currently used methods for acute physiology and chronic health evaluation, and it is based on LR [Kim et al., 2011; Knaus et al., 1991]. The system analyses 12 low-resolution routinely measured variables, along with age and chronic health points, during the first 24 hours of admission. The analysis is compared to approximately 18,000 cases in its memory to reach a prognosis with 95% accuracy. The two types of outputs provided by the system is a morbidity score and a predictive equation. Although exact performance measures are unclear, the system does prove that LR is effective in low-resolution medical data applications.

Vital signs, laboratory values, and demographic variables were used in a discrete-time survival analysis to predict the outcome of death, cardiac arrest, or ICU transfer [Churpek et al., 2016]. Two LR models, one using restricted cubic splines and the other using linear predictor terms, were compared to various other machine learning methods. Ten variables originating from 5 different hospitals were collected from EPIC, Verona, and WI electronic health records (EHRs). Data were separated into discrete eight-hour intervals, due to the frequency of physiological data collection in previous work [Churpek et al., 2014]. The balance of the training dataset was improved by matching windows where events occurred with non-event windows. Dropouts were replaced by values from the previous interval and the median was imputed if no previous values were present. The models compared were tree-based models, K-Nearest Neighbours (KNN), SVMs, and NNs, and had to predict whether an event occurred in the next window. Models were optimised using 10-fold cross-validation [Churpek et al., 2016].

In this study, the commonly cited Modified Early Warning score yielded the worst AUC of 0.7. The ensemble tree-based methods achieved the best AUC values, outperforming SVMs, NNs, and LR. Interestingly, the RF yielded an AUC of 0.8 and the gradient boosted machine an AUC of 0.79. The gradient boosted machine is where the consecutive trees are derived using random samples of the training data to predict the residuals of the previous models. The result is a combination of trees that weight the 'difficult to predict' events to a greater degree. This utilises the boosting ensemble approach and is expected to outperform the bagging techniques used in RFs [Breiman, 1998]. For the RF, the principal components were found to be RR, HR, age, and systolic blood pressure (BP) [Churpek et al., 2016]. This supports our focus on using ECG and BP as input variables for most of the experiments. Compared with our work, this study has a much larger cohort of patients (269,999 patient admissions), but the inputs to the models are of low resolution (8-hour intervals).

Neural Networks

A study by Clermont et al. [2001] compared two NN models (categorised and uncategorised input data) to LR (APACHE III). The worst value of 16 acute physiological variables for the first 24 hours in ICU were used along with age for the 1,647 patients in the dataset. The results indicated that the performance of NNs was generally comparable to LR techniques. Similarly, a study by Kim et al. [2011] compared NNs, SVMs, and DTs to the conventional LR model for ICU mortality prediction. Fifteen non-temporal variables were selected for data analysis in the first 24 hours of admission for 23,446 patients. DTs performed the best (0.98 AUC) followed by SVMs (0.876 AUC) and NNs (0.874 AUC), compared to the APACHE III (0.871 AUC). Conversely, a study by Zhang and Szolovits [2008] found NNs (0.99 accuracy) to outperform DTs (0.97 accuracy) in classifying eleven patients as stable or deteriorating, based on 1 Hz 8-channel data. Whether NNs perform better than other techniques remains unclear, but these studies do indicate that NNs perform better when the data has a higher resolution. Some studies have suggested that DTs can be applied to uncover the hidden layers of NNs, which could be valuable in our applications of LSTMs as well.

A feedforward multilayer perceptron (MLP) was used for cardiovascular disease detection [Oresko et al., 2010]. Cardiovascular disease, encompassing a variety of cardiac conditions, such as hypertension and heart attack, is the single leading cause of death. The MLP, with a 51-30-5 structure, was used for QRS complex (Figure 2.5) beat classification. The dataset consisted of 48 ambulatory ECG recordings of 30 minutes each, from the MIT-BIH database. A one-vs-all classification approach was taken with an average accuracy of 93.3% over all the classes. The Pan-Tompkins QRS-detection algorithm [Pan and Tompkins, 1985], consisting of a bandpass filtering stage using a set of cascaded filters, was used for feature extraction. The study demonstrates that classification of cardiovascular disease using ECG is possible. However, the dataset used in this study is small compared to other clinical machine learning application studies. Important to the continuation of our study, this study supports the reasoning that high-resolution signals contain features related to pathologies.

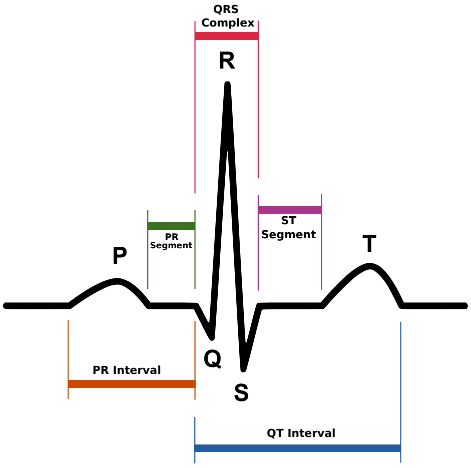


Fig. 2.5 Single ECG pulse with interest points labelled. (Adapted from: Atkielski [2007])

NNs were used to predict ICU patient mortality (binary) from the Physionet/Computing in Cardiology Challenge 2012 in a study by Ryan et al. [2013]. The dataset consisted of 12,000 patients with 4 static variables and 37 dynamic variables that could be measured once, more than once, or not at all for a given patient. The performance of the models in this study is reported as score achieved for the Physionet challenge, which is not relevant to report in our review. The importance of this study was the use of deep Boltzmann machines (DBMs), a generative model, to pre-train the feed-forward NNs and increase the performance of the final discriminative model. Our study could investigate the use of simpler generative models to pre-train the deep LSTM networks, similar to what has been done in this study.

A relatively out-dated review comparing studies of NN and logistic regression found that both methods provide inadequate performance [Sargent, 2001]. In contrast, a more recent study by Meyfroidt et al. [2009] found that NNs provide increased classification accuracy in the ICU compared to LR. However, NNs have also been found to manifest inferior classification results compared to DT and SVM algorithms, with SVMs achieving the highest accuracies, followed by DT [Yoo et al., 2011].

Gaussian Processes

GPs have demonstrated outstanding performance in various ICU applications; accurate predictions of the length of patient stay based on a patient's specific characteristics [Guiza Grandas et al., 2006], successful prediction of patient core temperatures several hours in advance [Güiza et al., 2006], outperforming conventional methods for EEG seizure detection [Faul et al., 2007]. Moreover, GPs have been used for regression in the ICU, due to their multi-parameter input and good predictive characteristics [Meyfroidt et al., 2009]. Our study sought to use GPs in smoothing signals before fitting HMMs to the data. The success of GPs in the reviewed literature supports the decision in our study.

An interesting and novel approach with GPs was the study by Ghassemi et al. [2015], in which multi-task GPs were used to assess and forecast patient acuity. Whereas other studies made use of the predictive mean of GPs for preprocessing and smoothing, this study used the inferred hyperparameters for supervised learning. The correlation between and within multiple time series was used to estimate parameters, instead of considering time series separately. The approach was applied to two problems. The first was to estimate and forecast the cerebrovascular autoregulation index by using the ICP and ABP signals. Data from 35 patients was collected over more than 24 hours at 0.1 Hz. The data was analysed in 10 minute windows and the result was a 0.09 root mean squared error between the estimated and true correlation coefficients. The second problem was to predict patient mortality using scores and notes from the EMRs of 10202 patients. The data split used for performance analysis was 70:30 (training:testing), and the AUC value achieved for predicting hospital mortality was 0.81. Although both applications made use of low-resolution data, the main limitation in the studies was computational cost, which was experienced in our study as well.

Colopy et al. [2015] made use of GP regression for prediction of HR time series. Five signals were measured for 333 patients in the surgical trauma step down unit and downsampled to 1-minute averages. Seven different sum combinations of covariance functions were explored in the study, which was important in our covariance function considerations for the GP applications. The models were trained sequentially on 5 windows of an hour long each, and then tested on the following 2 windows of an hour long each. The windows were moved through the entire dataset of one patient, to allow for several performance evaluations. It was found that using the mean of the hyperparameters yielded superior results compared to the use of the Maximum a posterior (MAP) of the hyperparameters. The study suggests exploration of Markov Chain Monte Carlo (MCMC) techniques to determine the most accurate posterior values of the hyperparameters. The covariance functions yielding the best performance are

$$k = k_{SE_1} + k_{SE_2} + k_{WN} \tag{2.13}$$

and

$$k = k_{Mat(\frac{5}{2})} + k_{SE_2} + k_{SE_3} + k_{WN} \tag{2.14}$$

Where k_{SE} is the squared exponential function (numeric subscripts indicate functions with different hyperparameters), k_{WN} is the white-noise covariance function, and $k_{Mat(\frac{5}{2})}$ is the Matern$_{(\frac{5}{2})}$ covariance function (for a definition of the covariance functions see [Rasmussen and Williams, 2006]). This study was very similar to our implementation of GPs for smoothing the signals, and provided valuable guidance.

Multivariate LR and GPs were used to predict intracranial pressure episodes 30 minutes in advance in a study by Güiza et al. [2013]. The patient cohort consisted of 264 traumatic brain injury (TBI) patients with their ICP and MAP measured every minute. The Glasgow Outcome Score (GOS) [Jennett and Bond, 1975] at 6 months was used to label patients into a binary set (good or bad outcomes). A sigmoid function was applied to the GP (with squared exponential covariance function) inferred mean and standard deviation values to produce a probabilistic output. For the data split of 178:61 an AUC value of 0.87 was achieved for 30-minute predictions. The models including dynamic data outperformed those based solely on static data, which indicates that the analysis of dynamic features holds promise.

Hidden Markov Models

Gultepe et al. [2014] made use of HMMs, Gaussian Mixture Models (GMMs), and NB classifiers to predict patient lactate levels. The dataset consisted of 7 EHR collected variables from 741 patients. The GMMs yielded the best results with an AUC of 1, and the HMMs followed yielding and AUC of 0.9. This study indicates that HMMs and GMMs are effective when applied to low-resolution datasets.

A study by Pimentel et al. [2015] fused features derived from ECG and ABP into a hidden semi-Markov model (HSMM) (see Section 2.3.1), in order to detect heart beats. A two-state heart beat HMM is a case of a ‘non-ergodic’ HMM. As shown in Figure 2.6, a non-QRS period has to precede a QRS complex; a QRS complex cannot occur directly after another QRS complex without a period of non-QRS activity. The mean delay between ABP and ECG was dealt with by removing the first 250ms of the ABP feature vectors and zero padding the end. Whereas this study mitigated the delay manually, our argument is that the non-parametric nature of LSTMs would be able to handle the delays serendipitously. The wavelet transform, signal gradient, and signal quality were extracted and fed into the HSMM framework. The features were downsampled to 50 Hz to increase the speed of computation. An interesting aspect of this study is that the data analysed remained high-resolution even though features were extracted. The reported performance was an average sensitivity of 0.9.

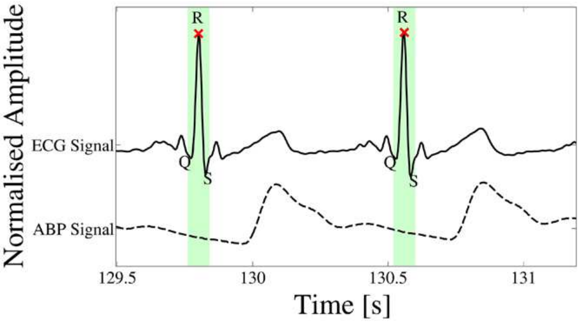


Fig. 2.6 ABP and ECG signals for two example heart beats. The beats in the ABP signal lag the ECG signal due to the pulse transit time. Shaded areas resemble the QRS period, 'state 1' in an HMM and non-shaded areas are labelled as 'state 2'. (Adapted from: Pimentel et al. [2015])

Recurrent Neural Networks

Ongenae et al. [2013] compared the efficacy of RNNs to SVMs and NB in predicting the need for dialysis between the fifth and tenth day after ICU admission. The dataset consisted of roughly three days of low-resolution creatinine and diuresis measurements for each of the 830 patients of which only 82 required dialysis. The NB and SVM models were combined with feature extraction methods, because, unlike RNNs, they 'can not directly deal with temporal data'. Echo state networks (ESNs) (Section 2.3.2), a recent development to improve RNN training, was employed in this study. The RNNs (0.85 AUC) performed second best, trumped by the NB (0.87 AUC) classifiers, which is understandable given that RNNs are not well suited for long-term memory.

A study by Choi et al. [2015] used Gated Recurrent Units (GRUs) (Section 2.3.2) to jointly forecast medication prescription and disease diagnosis along with their timing. Low-resolution data collected from the EHRs of 263,706 patients was used in this study. The models aimed to predict future diagnoses, medication and durations until the next event. Dropout was used between the GRU and predictive layer as a regularizer. The predictive layer was a softmax layer for predicting future codes and a rectified linear unit to predict the duration until the next event. Performance evaluation was done by means of cross-validation on an 85:15 split. The GRU model outperformed the NN and LR models it was compared to, yielding an accuracy of 80% for prediction of diagnoses and medication codes, and a 72.3%

accuracy for predicting the duration, disease and medication codes. The predictions were made on a patient-independent basis, proving that acceptable generalisation could be made across patients. The outcome is important, because GRUs and LSTMs are similar and the better suited of the two for medical data, has yet to be determined.

Lipton et al. [2014] made use of LSTMs to classify patients in the paediatric ICU as having one of a selection (128) of medical conditions, given multivariate physiological time series. Multilabel classification was used since diagnoses are not mutually exclusive. Thirteen hourly averaged variables were collected from an EHR system containing 10,401 medical episodes that vary from 12 hours to several months. Variables without values were replaced by 'normal' values as defined by domain experts, with the motivation being that clinicians decide not to measure variables if they believe it to be normal. Each LSTM was trained for 100 epochs using stochastic gradient descent (SGD) with momentum. For comparisons, baseline models were chosen to be a multilayer perceptron (MLP) and LR classifiers trained for each diagnosis. The MLP was trained for 1000 epochs and with 300 neurons in each layer. Both raw time series and hand-engineered features were used as inputs to the MLP. For the raw data, it was found that the first and last 6 hours of data yielded better results than using the last 12 hours. Using data earlier in the patient's stay was also considered in our study, as the biological processes are unnatural during later stages of a patient's stay, due to clinician intervention.

The data split used for performance evaluation was 80:10:10, training:validation:testing. Among the many performance metrics used in the study are micro- and macro-averaged AUC values. The micro-averaged AUC value is the average AUC value for the entire dataset, whereas the macro-averaged value weights the average AUC value according to the number of samples in each class. The best performing LSTM model made use of dropout and target replication (Section 2.3.2), and yielded a 0.856 micro-AUC, outperforming the strongest baseline. A combination of the strongest baseline and the best LSTM model improved the micro-AUC to 0.864. This study lies very close to the research in our study. However, the key differentiator is our argument that the majority of the information is embedded in the high-resolution data.

Other Approaches

Lugovaya [2005] made use of Linear Discriminant Analysis and a Majority Vote Classifier to perform biometric human identification. The ECG signals of 90 patients were analysed, and a correct classification rate of 0.96 was achieved. The argument presented is that shapes of ECG waveforms depend on the anatomic features of the human heart and body, and can thus be used as a human biometric characteristic [Biel et al., 2001; Yi et al., 2003]. ECG signals are governed by multiple individual factors, in particular, the presence and nature of pathologies, and the shape and position of the heart. ECG is known to be variable, for example, taking medication may temporarily change the configuration of the cardiac cycle and some pathologies gradually change the form of the cardiac cycle. This is important to consider when our study would attempt to find similarities in pathology induced waveform features and generalise across certain physical induced features. An ensemble of filters was used to preprocess the ECG signals. Wavelet drift correction was used for baseline drift correction, and an adaptive bandstop filter, low-pass filter, and smoothing was used to remove both power-line and high-frequency noise. The study alludes to ECG not being unique enough for identification in large groups, but that is an application for smaller groups. Similarly, work by Shen et al. [2002] combined template matching and a decision-based NN to classify 20 subjects with 100% accuracy. However, results from such a small dataset should be interpreted with caution.

NB classifiers (0.99 accuracy) outperformed DTs (0.97 accuracy) in predicting a patient's readmission to ICU [Braga et al., 2014]. Six variables were used in total, of which none were high resolution, and the study concluded that the inclusion of static variables, such as age and gender, is beneficial. The results sit well with a study by [Ramon et al., 2007], in which NB outperformed DTs and RFs in mortality prediction.

Caballero Barajas and Akella [2015] made use of generalised linear dynamic models to model the probability of mortality as a latent state that evolves over time. In addition to the usual clinical data, the models made use of text features, based on statistical topical models and medical noun phase extraction, to provide the context of the patient and in doing so, improve the model accuracy. Essentially, the model is similar to standard LR. However, the patient features are aggregated into a patient state that evolves over time, and a weighting vector β made the model effective for a patient-independent approach, by allowing the incorporation of static variables relating to patients groups (such as age).

Static and dynamic variables recorded every 3 hours, were obtained from the EMR data of 15,000 patients. Dropouts were replaced using a regularised Expectation Maximization (EM). Performance evaluation was done by means of 5-fold cross-validation, and the model yielded an AUC of 0.87, outperforming RFs, LR, and NB classifiers [Caballero Barajas and Akella, 2015]. Extensive work was done on text extraction, which could be a beneficial process for improving classification based on the observations of clinicians and allowing earlier deployment of models when there is an abundance of text descriptions and a shortage of time series data. This provides the means for clinicians to work with machines and will form part of future work in our study.

Comparison

A summary of the studies reviewed in general medicine is provided in Table 2.1. The table indicates the variability in the number of patients used for the different studies, and also the variety of medical conditions addressed, with mortality receiving the most attention. Table 2.2 compares the different studies based on the main technique investigated. From the table, it is evident that most of the research efforts were based on NNs, SVMs, and LR. It could be argued that the best performance was obtained by the DT employed by Kim et al. [2011] for mortality prediction over many patients. Another study showing high performance is the implementation of HMMs by Gultepe et al. [2014], which also yielded a high AUC value. The studies by Lipton et al. [2015] and Choi et al. [2015], which are similar to our work, achieved a micro-AUC (average AUC value for all classes) of 0.856 and an accuracy of 0.8 respectively on datasets containing orders of magnitude more patients than our datasets. Moreover, it is interesting to note that the studies by Lugovaya [2005] and Oresko et al. [2010] achieved similar accuracies on a similar number of patients for applications with almost opposite targets, the prior to differentiate between patients, and the latter to find similarities among patients (classes).

2.2.2 Neonatal Intensive Care Unit

Work reported in this section identified studies on the application of machine learning in the NICU. The studies were grouped according to the medical condition addressed because multiple machine learning techniques would often be implemented in a single study.

Seizure Detection

Seizure is common in premature infants [Gangadharan, 2013]. Faul et al. [2007] made use of windowed GPs to model EEG data for the detection of seizures in neonates. In total, 51 hours of 12-channel EEG data from 4 patients was used in this study. In the interest of computation time, the signals were decimated to 66 Hz. The sliding window was chosen to be 1s long, with an overlap of $\frac{1}{6}$s, because GPs have been known to obtain good results with small datasets [Gregorcic and Lightbody, 2005], and the computation time greatly increased with an increase in the number of training points. The overlap ensures that the GP output is smooth. The variance inferred by the GP was used as the indicator for seizures. It was found that seizure EEG signals are much more deterministic than non-seizure EEG signals, and as a result, a small variance inferred by the GP indicates the onset of a seizure. The model yielded an accuracy of 0.83, outperforming a Wavelet energy method and an autoregressive

Table 2.1 Summary of general medical machine learning studies

Study	Targets	Inputs	Methods	Performance	Measure	n[a]
Ramon et al. [2007]	Mortality	Demographic and daily measured variables	RF[b], DT, NB, TAN[c]	0.82, 0.79, 0.88, 0.86	AUC	1,548
Churpek et al. [2016]	Mortality	age, num-ICU-stays, admission time, routinely collected lab values	SVM, RF, NN, LR, DT	0.786, 0.801, 0.782, 0.77, 0.734	AUC	269,999
Kim et al. [2011]	Mortality	15 non-temporal variables	SVM, LR, NN, DT	0.876, 0.871, 0.874, 0.98	AUC	23,446
Clermont et al. [2001]	Mortality	Temp, HR, BP, RR, SpO_2[d], acid-base status, sodium, nitrogen, creatinine, albumin, bilirubin, glucose, WBC[e] count, GCS[f], urine, haematocrit	LR, NN	0.839, 0.836	AUC	1,647
Ghassemi et al. [2015]	Mortality	EMR[g] data, clinical notes	GP	0.81	AUC	10,202
Güiza et al. [2013]	ICP[h]	ICP[i], MAP[j]	GP	0.87	AUC	264
Scalzo and Hu [2013]	Artefacts	ICP	SVM	0.16	Specificity at 0.99 sensitivity	108
Ongenae et al. [2013]	Need for Dialysis	diuresis, creatinine	SVM, RNN, NB	0.84, 0.85, 0.87	AUC	830
Lipton et al. [2015]	Diagnosis	Temp, HR, BP[k], capillary refil rate, CO_2, GCS, pH, RR, SpO_2, glucose	SVM, LR, RNN	0.83, 0.855, 0.856	micro-AUC	10,401
Choi et al. [2015]	Diagnosis, prescription, and timing	codes from EHR[l]	RNN	0.8	Accuracy	263,706
Zhai et al. [2014]	ICU transfer	continuous, narrative, and nominal measurements	LR	0.91	AUC	7,298
Lugovaya [2005]	Biometric identification	ECG	LDA	0.96	Accuracy	90
Shen et al. [2002]	Biometric identification	ECG	NN	0.8	Accuracy	20
Oresko et al. [2010]	Cardiovascular disease detection	ECG	NN	0.93	Accuracy	48
Tu and Guerriere [1993]	Risk stratification	15 nominal, narrative, and discrete variables	NN	0.7	AUC	1,682
Zhang and Szolovits [2008]	Patient deterioration	HR, pulse rate, BP, RR, SpO_2	NN, RF	0.99, 0.97	Accuracy	11
Gultepe et al. [2014]	Risk of hyperlactemia	Temp, RR, WBC count, MAP, lactate levels, mortality, sepsis occurrence	HMM	0.9	AUC	741
Pimentel et al. [2015]	Detect heart beats	ECG, ABP	HSMM	0.9	Average Sensitivity	410

[a] Number of subjects in study
[b] Random Forest
[c] Tree Augmented Networks
[d] Oxygen saturation
[e] White Blood Cell
[f] Galsgow Coma Scale
[g] Electronic Medical Record
[h] Intracranial pressure
[i] Intracranial Pressure
[j] Mean Arterial Pressure
[k] Blood Pressure
[l] Electronic Health Record

Table 2.2 Machine learning techniques applied in general medicine (AUC values)

Study	SVM	LR	NN	GP	HMM[a]	RNN	RF	DT	Other	n[b]	Validation method
Scalzo and Hu [2013]	0.16[c]									108	3-fold cv[d]
Zhai et al. [2014]		0.91								7,298	10-fold cv
Kim et al. [2011]	0.876	0.871	0.874					0.98		23,446	first 24h for training
Clermont et al. [2001]		0.839	0.836							1,647	1200:447 split
Shen et al. [2002]			0.8[e]							20	20 b.p.p.[f] for training
Oresko et al. [2010]			0.93[g]							48	3-fold cv
Tu and Guerriere [1993]			0.7							1,682	713:969 split
Zhang and Szolovits [2008][h]			0.99				0.97			11	10-fold cv
Ghassemi et al. [2015]				0.81						10202	70:30 split
Güiza et al. [2013]				0.87						264	178:61 split
Gultepe et al. [2014]					0.9					741	10-fold cv
Pimentel et al. [2015]					0.9[i]					410	200:210 split
Ongenae et al. [2013]	0.84					0.85			0.87(NB)	830	30-fold cv
Lipton et al. [2015][j]	0.83	0.855				0.856				10,401	80:10:10 split
Choi et al. [2015]						0.8[k]				263,706	85:15 split
Ramon et al. [2007]							0.82	0.79	0.88(NB)	1548	10-fold cv
Churpek et al. [2016]							0.8	0.79		269,999	60:40 split
Lugovaya [2005]									0.96[l](LDA)	90	65:35 split

[a] Variations of

[b] Number of patients in study

[c] Specificity at 0.99 sensitivity

[d] Cross-validation

[e] Accuracy (not AUC)

[f] beats per patient

[g] Accuracy (not AUC)

[h] Accuracy (not AUC)

[i] Average sensitivity (not AUC)

[j] micro-AUC

[k] Accuracy (not AUC)

[l] Accuracy (not AUC)

model. The performance evaluation used in the study remains unclear, thus the study was omitted from the comparison.

Greene et al. [2007] achieved an AUC of 0.73 for classifying the ECG epochs of 7 term neonates as ‘seizure’ or ‘non-seizure’ with a linear discriminant classifier. For a similar classification, a sensitivity of 100% at an operating point of 4 false detections per hour, was reported by Temko et al. [2011] when using SVMs on EEG channels. The data from 17 full-term newborns in the NICU was used, amounting to a total of 1,920 seizures epochs. The leave-one-out [Rangayyan, 2004] approach was used for performance analysis. Fifty-five features were extracted from the EEG and preprocessing consisted of Fast Fourier Transform, anti-aliasing, and downsampling.

Ansari et al. [2015] made use of a multi-stage classifier for detection of neonatal seizures from EEG signals. The first stage is a heuristic (if-then) model, which mimics a human expert. The second is a pre-trained SVM which reduces the number of false alarms. During the first stage, the EEG signal was decomposed using discrete wavelet transform (DFT) to characterise the low-frequency activities in EEG. This was an important implementation because DFT was a consideration in the early stages of our study. Seventeen scalp electrodes were filtered between 1 and 20 Hz for the 35 patients in the dataset. The reported performance was a sensitivity of 72% and a false positive rate (specificity) of 1.5 per hour. Both this and the study by Temko et al. [2011] are important for our study in determining the efficacy and relevance of preprocessing methods. The datasets used had a similar amount of patients as ours and also made use of high-frequency data, but the models made classifications based on only one variable, where our model would ideally detect interdependencies between multiple signals.

Artefact Detection

Williams et al. [2006] made use of a Factorial Switching Kalman Filter (FSKF) to model NICU artefactual and physiological data patterns. The developed technique is able to detect artefacts in data and provide the type of artefact among a predetermined selection (transcutaneous probe recalibration, disruption from sampling arterial blood, probe disconnection, bradycardia, and open incubator). The data consisted of 9 signals collected at 1 Hz for 8 infants. Four randomly chosen patients were used for training and the rest for testing. Our implementation selects random time series segments from all patients in order to improve model generalisation. The mean AUC value for classifying artefacts was 0.89, with the highest AUC achieved being 0.99 for the detection of blood samples being taken. The FSKF outperformed the Factorial HMM (0.83 mean AUC), due to the latter not having knowledge of dynamics and changes in baseline physiological levels over time and between patients.

The Factorial HMM has no hidden continuous state, where the FSKF does [Ghahramani, 2001]. Although the data has a lower resolution than that in ours, this study alludes that HMMs should not by default be considered the best models for time series.

To reduce the number of false alarms, DTs have been employed for the detection of artefacts in physiological signals [Tsien et al., 2000]. Roughly 3 hours of 4-signal data were collected at 1-minute intervals for 67 patients. Each signal was modelled separately with both a DT and an LR model as the baseline. Data preprocessing included abstraction of the raw data signals into time series features (moving mean, median, best fit linear regression, slope, absolute value of best fit linear regression slope, standard deviation, maximum value, minimum value, and range). The DTs outperformed LR with an average AUC of 0.94. The time series features extracted present appropriate examples for our study.

Patient Deterioration

Kernel Density Estimators (KDE), Kernel Principal Component Analysis (KPCA), and One-Class SVMs (OCSVMs) were compared for novelty detection [Gangadharan, 2013]. The dataset consisted of roughly 17 days of continuous data for each of the 9 patients. Six variables were collected with the Intensive Care Monitoring Plus software package (ICM+) [Smielewski, 2011], at 200 Hz and downsampled to 1 Hz. Noise and artefacts from momentary movements were reduced by means of a 5s moving average filter. Short (<60s) dropouts were replaced by the last recorded value, and the mean and standard deviation of the training data was used to replace longer dropouts. Artefact rejection limits were manually set through observations of histograms for each parameter, and by flagging signal values that are impossible. A dwell period of 100s was used, meaning the signal had to sustain a value exceeding a threshold for more than 100s for a positive to be registered. This mitigates momentary rises in the signal, probably due to artefacts [Townsend et al., 2004]. A refractory period of 3 minutes yielded the best classification performance. The refractory period is the time a detector becomes inactive after detection because successive positive or false positive detections are assumed to be part of the first. For each method compared in this study, the first hour of each dataset was used as a reference dataset for a stable patient condition, and the rest was classified as stable or abnormal.

The best performing model was the OCSVM, with a partial AUC of 0.71. The partial AUC of the ROC was calculated for a sensitivity greater than 0.85. To account for the longitudinal variation of physiological parameters within the first week of life, the model was retrained every 24 hours, with the first model serving as a reference map to provide a consistent novelty output. However, this did not improve the original model. Another interesting finding is that weighting the inputs did not improve the models. The optimum results were obtained when the data was preprocessed to extract 3 principal components with PCA and 120 clusters with k-means clustering. The hand-engineered approach of this study, such as those used for dropouts, artefact rejection, and dwell period are important to our study in providing a scope for parameters that can be specified or learned. The classification performance achieved in this study is not adequate for medical use and we aim to prove that high-resolution data combined with deep learning techniques could provide acceptable indicators in practice.

Late-onset Sepsis

Late-onset sepsis is a bloodstream infection, usually bacterial, that occurs 72 hours after birth. Additionally, sepsis is the number one cause of death in ICUs [Ongenae et al., 2013]. Detection of reduced variability and transient deceleration in HR have proved effective for early stage diagnosis of sepsis in neonates [Shavdia, 2007]. A study by Mani et al. [2014] found that DTs (0.65 AUC) outperformed SVMs, NB, LR, and RFs in the detection of late-onset sepsis, which contradicts comparisons done by Ramon et al. [2007] (ICU based) and Churpek et al. [2016], where DTs had the worst performance.

Stanculescu et al. [2014] made use of an autoregressive HMM (AR-HMM) to predict late-onset sepsis. The AR-HMM enhances the HMM architecture (Section 2.3.1) by introducing direct stochastic dependence between observations. It explicitly models the (possible long range) correlations in sequential data. Samples drawn from an AR-HMM are smoother than samples drawn from an HMM, making them a better generative model for time series problems. Roughly 30 hours of 4-channel data were collected at 1 Hz for 38 patients, of which half had sepsis. The model yielded an AUC of 0.8 and made extensive use of domain knowledge to facilitate inference and learning.

Morbidity Prediction

Saria et al. [2010] used a regularised LR for predicting infant morbidity based on the first 3 hours after admission for 138 patients. The probabilities of morbidity, stemming from the probability of each vital sign for a patient-specific class, were aggregated into a single score [Clifton et al., 2015]. Ten hand-engineered features were used as input to the model, which yielded an AUC of 0.92 for prediction of morbidity, and AUC values of 0.97 and 0.98 for determining the risk of infection and cardiopulmonary disease respectively. The model had superior discrimination of neonatal morbidities compared to commonly used severity scores (Apgar, SNAP II, SNPPE II, and CRIB). Although the data is low-resolution and the technique does not model sequences, the study yielded good results.

Predicting Ventilation Duration

Tong et al. [2002] made use of NNs to classify patients according to ventilation duration. This was an extension of a project based in the adult ICU. Data from 1,321 infants were used to predict ventilation duration being less than/equal to, or more than 8h ($\leq$ or $>$). Similar to the adult study, 6 variables were used, and the training and testing datasets consisted of 881 and 440 randomly selected patients respectively. The maximum correct classification rate obtained was 0.93, which was similar to the adult study (0.91). The results are interesting because it is expected that the noisier data from the NICU would result in worse classification performances compared to the classification based on the cleaner data from adult ICUs. Our study was based on datasets from both the NICU and ICU, and this study indicates that results from both should be similar.

Comparison

A summary of the NICU based studies reviewed is provided in Table 2.3. The results indicate that SVMs are better suited for temporal data, and DTs and NB classifiers are well suited when there is a combination of temporal and non-temporal data. The datasets used in the studies had a similar number of patients to the datasets used in our study. The findings sit well with that of the adult ICU based studies. Zhai et al. [2014] made use of LR to predict the need for paediatric ICU transfer within the first 24 hours of admission, achieving an AUC of 0.91, similar to that of [Saria et al., 2010]. Studies such as those by Ansari et al. [2015]; Temko et al. [2011]; Tsien et al. [2000] made extensive use of hand-engineered features that provided a wide range of examples to consider in our study.

Table 2.4 compares the performance of the machine learning techniques applied in the NICU. The LR and SVM algorithms were the most popular among the studies. DTs which are widely used in the ICU, due to their visual and interpretable representations of data, were not as popular as expected in the NICU. Little attention was received by NNs, which are often in medicine seen as black-box techniques. It is also evident that no research has been done on deep learning techniques applied to NICU data. The table elucidates the difficulty in comparing the models and studies. The best performance could be attributed to the DTs in the study by Tsien et al. [2000], because it contained a good number of patients, compared to the other studies, and achieved a high AUC value. Another decent performing model would be LR in the study by [Saria et al., 2010] because it also achieved a high AUC value with more patients than the aforementioned study, and an arguably more complex classification problem. However, the study evaluated performance with the leave-one-out technique, which would

Table 2.3 Summary of NICU machine learning studies

Study	Targets	Inputs	Methods	Performance	Measure	n[a]
Le Compte et al. [2010]	Insulin sensitivity (pdf[b])	Insulin and nutrition records, blood glucose	2D KDE[c]	0.934	Accuracy at 90% confidence interval	21
Gangadharan [2013]	Patient deterioration (binary)	HR, sysBP[d], diaBP[e], temp, RR, SaO_2	KDE, KPCA[f], OCSVM[g]	0.77, 0.8, 0.84	Specificity at 90% sensitivity	9
Temko et al. [2011]	Seizure	EEG	SVM	1	Sensitivity with 4 FD/h	17
Ansari et al. [2015]	Seizure	EEG	SVM	0.72	Sensitivity with 1.5 FD/h	35
Greene et al. [2007]	Seizure	ECG	LD	0.73	AUC	7
Mani et al. [2014]	Late-onset sepsis	101 temporal and non-temporal variables from EMR	SVM, NB, RF[h], KNN[i], LBR[j], TAN, LR, DT	0.61, 0.64, 0.57, 0.54, 0.62, 0.59, 0.61, 0.65	AUC	299
Stanculescu et al. [2014]	Late-onset sepsis	SpO_2[k], HR, CT[l], PT[m]	AR-HMM	0.8	AUC	38
Tsien et al. [2000]	Artefacts	HR, mean BP, $PaCO_2$, PaO_2	DT, LR	0.94, 0.15	AUC	67
Williams et al. [2006]	Artefacts	HR, sysBP, diaBP, $TcPO_2$[n], $TcPCO_2$[o], SpO_2, CT, incubator temp.	FSKF, FHMM[p]	0.89, 0.83	AUC	8
Saria et al. [2010]	Morbidity (score)	HR, SpO_2, RR, birth weight, gestational age	LR	0.92	AUC	138
Tong et al. [2002]	Ventilation duration (binary)	low BP, low temp, low serum pH, presence of seizures, urine, PaO_2/FiO_2[q] ratio	NN	0.93	CCR	1321

[a] Number of subjects in study
[b] probability density function
[c] Kernel Density Estimator
[d] Systolic blood pressure
[e] Diastolic blood pressure
[f] Kernel Principal Component Analysis
[g] One-Class SVM
[h] Random Forest
[i] K-Nearest Neighbours
[j] Linear Bayesian Regression
[k] Oxygen saturation
[l] Core temperature
[m] Peripheral temperature
[n] Transcutaneous oxygen tension
[o] Transcutaneous carbon dioxide tension
[p] Factorial Hidden Markov Model
[q] Fraction of inspired oxygen

always return higher performance compared to the larger data splits used for evaluation in the other studies.

Table 2.4 Machine learning techniques applied in the NICU (AUC values)

Study	LD[a]	KDE[b]	SVM	DT	NN	HMM[c]	LR	NB	TAN[d]	n[e]	Validation method
Greene et al. [2007]	0.73									7	10-fold cv[f]
Williams et al. [2006]						0.83				8	50% split
Stanculescu et al. [2014]						0.8				38	leave-1-out
Le Compte et al. [2010]		0.93[g]								21	5-fold cv
Gangadharan [2013][h]		0.63	0.71							9	First hour training data
Temko et al. [2011]			0.9[i]							17	5-fold cv
Ansari et al. [2015]			0.72[j]							35	17:18 split
Mani et al. [2014]			0.61	0.65			0.61	0.64	0.59	299	5-fold cv
Tsien et al. [2000]				0.94			0.15			67	(70:9:21)% split
Saria et al. [2010]							0.92			138	leave-1-out
Tong et al. [2002]					0.93[k]					1321	881:440 split

[a] Linear Discriminant Classifier
[b] Kernel Density Estimator
[c] Variations of
[d] Tree Augmented Networks
[e] Number of patients in study
[f] Cross-validation
[g] Accuracy (not AUC)
[h] Values of pAUC at 0.85 sensitivity
[i] Sensitivity (not AUC)
[j] Sensitivity (not AUC)
[k] Correct classification rate

2.3 Machine Learning for Sequential Data

Sequential patterns are valuable, because they can be exploited to improve the prediction accuracy of classifiers. In this section we discuss machine learning techniques suited for sequential data.

2.3.1 Hidden Markov Models

An HMM is a stochastic process with a hidden (not observable) underlying stochastic process. This underlying stochastic process can only be observed through another set of stochastic processes that produce the sequence of observed symbols. An HMM has a finite number of hidden states in the model, and within a state, the signals possess some measurable distinctive properties. At each clock time, t, a new state (which could be the same state) is entered based on a transition probability distribution, which depends on the previous state $t-1$ (the Markovian property). After each transition, an observation output symbol is produced according to the probability distributions which depend on the current state [Rabiner and

Juang, 1986]. The HMM λ is defined as

$$\lambda = (A, B, \pi) \tag{2.15}$$

where $A = \{a_{ij}\}$ is the transition matrix, defining the probability of moving from state i at time t to state j at time $t+1$. The emission distribution, $B = \{b_j(\mathbf{O}_t)\}$, defines the probability that state j generates the observations $\mathbf{O}_t$ at time t. $\pi = \{\pi_i\}$ is the initial state distribution, defining the probability of being in state i at the first time point. The probabilistic description provided by HMMs can be used to provide a statistical *confidence measure* in its analysis of a given signal [Hughes and Tarassenko, 2006]. Furthermore, HMMs can be viewed as a simple form of dynamic bayesian networks [Dagum and Galper, 1993; Dagum et al., 1991; Murphy, 2012], which is a BN that relates variables over adjacent time steps. A variant of the HMM, called a hidden semi-Markov model (HSMM), and used in the aforementioned study by Pimentel et al. [2015], is a consideration for our future work, in order to model patient states more effectively. An HSMM is defined as $\lambda = (A, B, \pi, p)$ where $p = \{p_i(d)\}$ is the explicitly defined probability of remaining in state i for duration d. Here only the transition between different states is Markovian.

HMMs provide a sound and elegant methodology, but they suffer from a fundamental drawback: the structure of the HMM is often a poor model of the true process producing the data [Dietterich, 2002]. The Markov property is responsible for part of the problem. The relationship between two time separated values (e.g., y_1 and y_4) must be communicated through the intervening values (e.g., y_2 and y_3). A first-order Markov model (i.e., where $P(y_t)$ only depends on y_{t-1}) cannot in general capture these kinds of relationships. Additionally, the performance of HMMs depends on the specified number of hidden states. Murphy [2012] suggests finding the optimal number of states by means of a grid search over a specific range, by using reversible jump MCMC, or by using variational Bayes to remove unwanted components. Infinite HMMs (iHMMs) attempt to mitigate this limitation via a non-parametric approach, based on the hierarchical Dirichlet process [Beal et al., 2001].

Chatzis and Tsechpenakis [2010] derived an efficient variational Bayesian inference algorithm for the infinite Hidden Markov Random Field (iHMRF) model. Methods using likelihood- or entropy-based criteria, and reversible MCMC methods, impose heavy computation requirements and yield noisy model size estimates. When the number of clusters is unknown *a priori*, Dirichlet process (DP) mixture models are well suited non-parametric Bayesian statistical methods. Infinite mixture models based on the DP have demonstrated promising results in image segmentation, and the variational Bayes approach has improved scalability in terms of computational cost compared to Monte Carlo techniques. The iHMRF yielded better results (mean probabilistic rand index [Unnikrishnan et al., 2005] of 72) than the DP mixture and the Markov Random Field method.

Another limitation of HMMs is the assumption of statistical independence between successive observations within a state [Hughes and Tarassenko, 2006]. Physiological signals clearly exhibit strong correlations over time. The transition stationary assumption also limits HMMs in our application. The assumption is concerned with the *time invariant* nature of the state transition probabilities. For any two times t_1 and t_2, the assumption is that

$$P(s_{t_1+1} = j|s_{t_1} = i) = P(s_{t_2+1} = j|s_{t_2} = i) \tag{2.16}$$

The individual state durations follow a geometric distribution as a consequence of the time-invariant transition probabilities. The distribution can lead to physiologically implausible segmentations and can be a poor match to the true state duration distributions of many real-world signals. Several alternatives have been explored to try to overcome the limitations of the HMM, one of which is conditional random fields (CRFs).

CRFs are discriminative models, for which the principal advantage over generative models is being better suited to include rich, overlapping features [Sutton and McCallum, 2010]. The classic label bias problem is solved by CRFs, leading to superior performance compared to HMMs, when the true distribution has higher-order dependencies than the model, which is often the case in practice. A disadvantage of discriminative models, especially for medical data, often lacking annotations, is that supervised applications are less natural than that for generative models. Results from a part-of-speech tagging experiment by Lafferty et al. [2001] did not show a significant difference in HMM and CRF performance. Since LSTMs are the focus of our study, HMMs, which have received more attention in the literature compared to CRFs, were used as our baseline model.

2.3.2 Recurrent Neural Networks

The characteristic that makes RNNs stand out from other machine learning methods is their ability to learn and carry out complicated transformations of data over extended periods of time [Graves and Jaitly, 2014]. Lipton [2015] provides a thorough review of RNN's. Here it is argued that since the earliest conception of artificial intelligence, such as Alan Turing's "imitation game" [Turing, 1950], researchers have sought to build systems that interact with humans in time, hence the motivation for models of sequential behaviour.

Standard operations become infeasible with an HMM when the set of possible hidden states grows large, because the Viterbi algorithm used to perform efficient inference with HMMs, scales in time as $\mathcal{O}(N^2T)$ [Viterbi, 1967], where N is the number of states, and T is the sequence length. The transition matrix, with time-adjacent state transition probabilities, is of size N^2. To avoid the Markovian limitation, the models can account for larger context windows by creating a new state space equal to the cross product of the possible states at each time in the window. The new state space becomes a combination of the current and previous states, resulting in each transition being dependent on more than one previous state. However, Markov models are rendered computationally impractical for modelling long-range dependencies, because the state space grows exponentially with the size of the larger context windows [Graves et al., 2014]. In RNNs, each hidden state at any time step can contain information from a nearly arbitrarily long context window, because the number of distinct states that can be represented by a hidden layer of nodes grows exponentially with the number of nodes in the layer.

RNNs have been known to be difficult to train because the errors are backpropagated across many time steps. The derivative of the error with respect to the inputs, can be decomposed into terms that involve the product of Jacobians, which tend to vanish or explode [Bengio et al., 2013, 1994; Lipton, 2015; Ongenae et al., 2013; Yao et al., 2015]. As shown in Figure 2.7, RNNs allow connections from one neuron in a hidden layer, to all the neurons in the hidden layer of the next time step. Echo state networks (ESN) is a recently developed method that is optimised to handle time series data and improve the training of RNNs. Figure 2.8 illustrates the general layout of an ESN, consisting of l output nodes, k input nodes and n reservoir nodes. Each node is a perceptron with a sigmoid activation function. At a given time, the state of a node is the weighted sum of the last fed inputs, namely

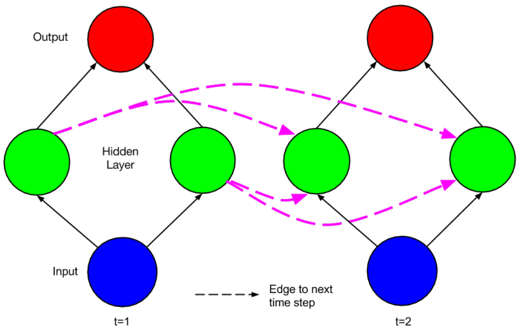


Fig. 2.7 A recurrent neural network unfolded across time steps. Adapted from: [Lipton, 2015]

$$\mathbf{x}[t+1] = (1-\mu)\mathbf{x}[t] + \mu f(\mathbf{W}\mathbf{x}[t] + \mathbf{W}^{in}\mathbf{u}[t]) \tag{2.17}$$

where $\mathbf{u}$ is the input matrix and $\mathbf{x}[t]$ denotes the network state at time t. The $k \times n$ matrix $\mathbf{W}^{in}$ contains the weights between input and reservoir nodes and the $n \times n$ matrix $\mathbf{W}$ contains the recurrent weights between the reservoir nodes. $\mu \in (0,1]$ is the leaking rate, which can be regarded as the speed of the reservoir update dynamics discretized in time, and f is a sigmoid wrapper.

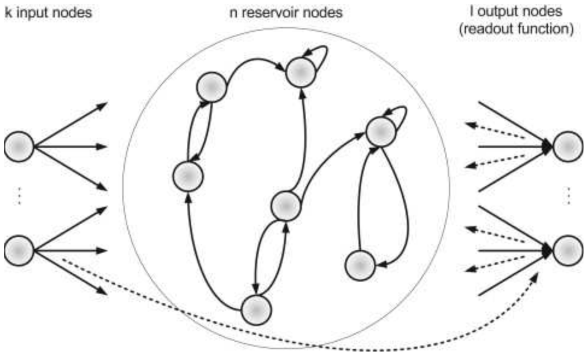


Fig. 2.8 Echo state network general layout. Input, reservoir and output nodes are represented by circles. Optional connections are denoted by dotted arrows. Arrows represent non-zero weighted connections. (Adapted from: Ongenae et al. [2013])

A caveat for the use of deep learning techniques such as RNNs is the computational expense. Sutskever et al. [2014] made use of 8 GPUs to train a language translation model, which took 10 days to train. Four GPUs were used to train each of the 4 hidden layers containing 1,000 nodes. The other GPUs were used to calculate the softmax, and the implementation was coded in C++. The input vocabulary contained 160,000 words and the output vocabulary contained 80,000 words. This dataset has more samples than the dataset in our study, but the sequences are an order of magnitude shorter, which implies that our study might require even more computation.

A study by Graves [2013] demonstrated that RNNs can be used to generate complex sequences with long-range structure, simply by predicting one data point at a time, and using the new output as input for the next prediction. The study made use of a 'deep' RNN, composed of stacked LSTM layers. The problem, common to most conditional generative models, is that the models have little opportunity to recover from past mistakes if predictions are based on few inputs, which have themselves been predicted by the model. A stabilising effect is created by a longer memory. This is where the long short-term memory (LSTM) architecture is introduced.

Long Short-Term Memory

RNNs based on Long Short-Term Memory (LSTM) units were originally introduced in [Hochreiter and Schmidhuber, 1997] and have since yielded state-of-the-art results for problems in genomic analysis, image captioning, natural language processing, and handwriting recognition [Auli et al., 2013; Graves et al., 2009; Sutskever et al., 2014; Vinyals et al., 2015; Xu et al., 2007]. The key benefit being that LSTMs can capture nonlinear dynamics and long-range dependencies. Kalman filters, CRFs, and Markov models are sequential models that are ill-equipped to learn long-range dependencies. Results from speech recognition have shown that LSTMs outperform other models using raw features, which minimises the need for preprocessing and feature engineering [Lipton et al., 2015]. A similar variant of RNNs is the gated recurrent unit (GRU) proposed by Cho et al. [2014], where each recurrent unit adaptively captures dependencies of different time scales. The performance of GRUs was found to be comparable to LSTMs in an empirical evaluation of RNNs on sequence modelling [Chung et al., 2014].

We introduce the LSTM architecture based on the description in Graves [2013]. Each j-th LSTM unit maintains a memory c_t^j at time t, unlike the recurrent unit, which simply computes a weighted sum of the input signals and applies a nonlinear function. Both the LSTM unit and GRU have gating units, that modulate the flow of information inside the unit. It is a *gate* in the sense that when the value is zero, the flow from the source node is cut off. Moreover, both share additive components of their update from t to $t-1$, which is missing in the traditional recurrent unit. The $j^{\mathbf{th}}$-layer activation h_t^j, or the output of the LSTM unit is then

$$h_t^j = o_t^j \tanh(c_t^j), \tag{2.18}$$

where o_t^j is an *output gate* that modulates the amount of memory content exposure. The output gate is computed by

$$o_t^j = \sigma(W_o\mathbf{x}_t + U_o\mathbf{h}_{t-1} + V_o\mathbf{c}_t)^j, \tag{2.19}$$

where W and U are the input weight matrices and hidden weight matrices respectively, and V are diagonal matrices. σ is a logistic function and the memory cell c_t^j is updated by partially forgetting the existing memory and adding a new memory content $\tilde{c}_t^j$:

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j, \tag{2.20}$$

where the new memory content is

$$\tilde{c}_t^j = \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1})^j. \tag{2.21}$$

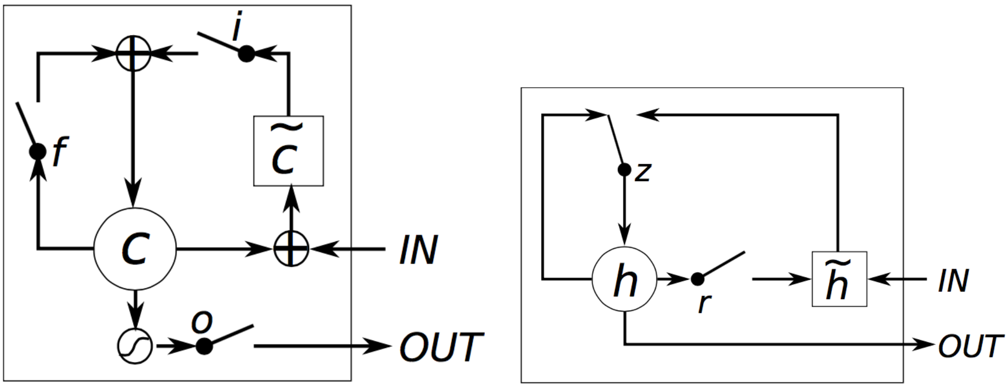
A *forget gate* f_t^j modulates the extent to which existing memory is forgotten, and an *input gate* i_t^j modulates the degree to which new memory content is added to the memory cell. If the forget gate is set to allow almost zero decay, the learning of long-range dependencies is enhanced. Gates are computed by

$$f_t^j = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + V_f \mathbf{c}_{t-1})^j, \tag{2.22}$$

$$i_t^j = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + V_i \mathbf{c}_{t-1})^j. \tag{2.23}$$

An LSTM unit is able to decide whether or not to keep the existing memory via the introduced gates, unlike the traditional recurrent unit, which overwrites its content at each time-step. Intuitively, the LSTM unit detects important features from an input sequence during early time steps and is able to carry the existence of the feature over the entire sequence, hence, capturing potential long-term dependencies. Figure 2.9 illustrates schematics of the LSTM and GRU. A feature of the LSTM that is missing from the GRU is the controlled exposure of the memory content (output gate). The GRU exposes its full content without any control, and it is this increased control of the LSTM that motivated our investigation of their application to the often noisy medical data.

Gers and Schmidhuber [2000] proposed peephole connections that pass from the internal state directly to the input and output gates of the same nodes, without having to first be modulated by the output gate. These connections have been found to improve performance on timing tasks where the network is required to learn to measure precise intervals between events. This could be important in medicine when for example, attempting to determine the time between onset of physiological symptoms and the underlying change in vital signs. Lipton [2015] uses the following example to describe the peephole connection: consider a network which must learn to count objects and emit a desired output when n objects have been seen. The network might learn to increase the internal state by some fixed amount of activation after seeing each object. When the n^{th} object is seen, the network needs to know to let out content from the internal state so that it can affect the output. The peephole allows the output gate o_c to know the content of the internal state s_c, which in this scenario acts as an input to o_c.



(a) (b)

Fig. 2.9 Illustration of (a) LSTM and (b) Gated Recurrent units. (a) c and $\tilde{c}$ denote the memory cell and the new memory cell content. i, f and o are the input, forget and output gates, respectively. (b) z and r are the update and reset gates, $\tilde{h}$ and h are the candidate activation and the activation. (Adapted from: Chung et al. [2014])

Another variant, Bidirectional RNNs, have yielded good results in language processing, but it is not an appropriate algorithm for online applications because it requires sequence elements after the sequence element being predicted or classified. A bidirectional LSTM model (0.82 accuracy) has outperformed an HMM model (0.7 accuracy) in the task of handwriting recognition [Lipton, 2015].

LSTMs and GRUs were compared in [Chung et al., 2014] by means of a polyphonic music dataset from Boulanger-Lewandowski et al. [2012] and a speech modelling dataset from Ubisoft[1]. Similar to our study, RMSprop was used to optimise the model. The norm of the gradient was clipped after each update, to remain smaller or equal to one and prevent exploding gradients. The validation dataset was used for early stopping during training. The study demonstrated the clear advantage of using gated units in RNNs, but the results for the comparison of LSTMs and RNNs remained inconclusive. The study indicates that the type of gated recurrent unit may greatly depend on the dataset and the corresponding task.

A study by Lee et al. [2015] compared the performance of rectified linear units [Le et al., 2015], LSTM units, and GRUs in RNNs to model DNA sequences and to detect splice junctions thereon. The GRU and LSTM models had a 4-60-30-3 structure and an increase in the number of layers or neurons did not improve performance. The LSTM model yielded the best performance with an F1-score of 0.94, and all the RNN models outperformed SVM and dynamic Bayesian network methods.

[1] http://www.ubi.com/

Yao et al. [2015] introduced the use of a depth gate, which connects the memory cells of adjacent layers in an LSTM RNN. The result is a linear dependence between lower and upper layer recurrent units. The depth gate is a function of the memory cell in the lower layer, the input and the previous memory cell of the current layer. The depth-gated LSTM, standard LSTM, and the GRU architectures were compared in machine translation applications. All the models made use of 200 nodes in each hidden layer, and experimentation was done with depths of 3, 5, and 10. The depth-gated LSTM is considered a simple variant of Grid LSTM [Kalchbrenner et al., 2015], that has a gate function, depending on depth and time, only on memory cells. For machine translation, the depth-gated LSTM outperformed the GRU and standard LSTM architectures, with the GRU performing the best with 3 layers, and LSTM performing the best with 5 and 10 layers.

The largest study to date (5,400 experiment simulations) on LSTMs was conducted by Greff et al. [2015]. The models were trained using stochastic gradient descent with Nesterov-style momentum [Sutskever et al., 2013], with updates after each sequence. The learning rate was rescaled by a factor (1-momentum). Training was stopped after 150 epochs or after 15 epochs of no improvement in the validation set. The variants of the standard LSTM were:

1. No Input Gate (NIG)
2. No Forget Gate (NFG)
3. No Output Gate (NOG)
4. No Input Activation Function (NIAF)
5. No Output Activation Function (NOAF)
6. No Peepholes (NP)
7. Coupled Input and Forget Gate (CIFG)
8. Full Gate Recurrence (FGR)

The first five variants are self-explanatory. The CIFG variant uses one gate for both the input and the cell recurrent self-connection (GRU variant). The FGR variant adds recurrent connections between all the gates as in the original formulation of the LSTM [Hochreiter and Schmidhuber, 1997]. This significantly increases the number of parameters and computational expense, due to 9 additional recurrent weight matrices.

Interestingly, this study found that clipping gradients to a range of [-1,1] reduced model performance. The results indicate that none of the LSTM variants significantly improve the standard model. This is important to our study, both as a guideline for our own modifications to the LSTM and in omitting experiments with variants in our study. Without peepholes (NP), the model complexity is reduced, but the study by Greff et al. [2015] found no significant change in performance, which meant our architecture would omit peephole connections.

There also exists a broader range of RNN variants, such as ESNs, clockwork RNNs, and variational RNNs, which would make for an interesting comparison.

The functional Analysis of Variance (fANOVA) method [Hooker, 2012] was used to determine which hyperparameters have the largest influence on model performance. The principal hyperparameter was found to be the learning rate, responsible for more than two-thirds of the variance in model performance. The second most important hyperparameter was the number of units in the hidden layer (more is better), followed by the input noise and leaving momentum with less than 1% of the variance. It should be noted that momentum may play a more important role in batch training, where the gradients are less noisy. Additionally, the analysis suggests that finding the optimal learning rate depends on the dataset, and could, therefore, be optimised using a smaller network before training a large model.

Clockwork Recurrent Neural Network

Koutnik et al. [2014] introduced a simpler variation of the standard RNN architecture, the *Clockwork* RNN (CW-RNN). The hidden layer is partitioned into separate modules, each processing inputs at its own temporal granularity, in turn, making computations at its prescribed clock rate. The long-term dependency issue is solved by having different parts of the hidden layer process at different clock speeds. Consequently, a smaller number of weights is needed, because slower modules are not connected to faster ones, and the networks are executed faster because not all the modules are executed at every time step. Intuitively, this architecture seems optimal, but fewer parameters could lead to less flexibility.

The clock period for each module is arbitrarily chosen. For example, the study by Koutnik et al. [2014] made use of an exponential series of periods; module i has clock period $T_i = 2^{i-1}$. The key differentiator is that during each CW-RNN time step t, only the output of modules i that satisfy $(t \mod T_i) = 0$ are executed. The result is that the low-clock-rate modules process, retain and output the long-term information, where the high-speed modules focus on local information, having context provided by the low-speed modules available.

CW-RNNs, RNNs, and LSTM networks were compared with one hidden layer in each and the same amount of parameters (clock-periods of the CW-RNN were included as parameters) [Koutnik et al., 2014]. The preliminary experiments, with spoken word classification and audio signal generation, demonstrate that the model outperforms RNNs and LSTMs. The results should be considered with caution because it is difficult to compare different models on equal grounds. For example, experimentation was done with 100, 250, 500, and 1000 parameters, but at 1000 parameters, the LSTM had only 15 hidden units where the CW-RNN had 40. Additionally, the SGD with Nesterov-style momentum [Sutskever et al., 2013] was used to train the models, and it could be argued that the RMSprop optimiser would be better suited to the more complicated LSTM architecture.

Neural Turing Machines

It is known that RNNs are Turing-complete [Siegelmann and Sontag, 1995] and thus have the capacity to simulate arbitrary procedures, *if* wired properly. In work by Graves et al. [2014], the RNN architecture was enriched by a large, addressable memory, which by analogy to Turing's enrichment of finite-state machines by an infinite memory, was called a Neural Turing Machine (NTM). NTMs are differentiable and can, therefore, be trained using gradient descent. Figure 2.10 illustrates a high-level representation of the NTM architecture.

The read and write operations are 'blurry', in that they interact in a greater or lesser extent with all elements in memory, rather than a single element. The degree of blurriness is determined by an attentional mechanism constricting each read and write operation to interact with a small portion of the memory. Similar to LSTMs, each write in the NTM has two parts, an *erase* followed by an *add*. The erase and write vectors are accompanied by a weight vector, indicating which areas in the memory bank receive attention for the given operation. The addressing weights are generated by two methods used concurrently. The first method is content-based addressing [Hopfield, 1982], which focusses attention on content that is similar. The second, which improved generalisation, is location-based addressing, where data is addressed based on its location in the memory bank. All elements in the memory matrix lie in the range (0,1). An important free parameter to be chosen for the NTM is the type of NN used as the controller. A recurrent controller such as LSTM has its own internal memory, which can complement the larger memory in the matrix [Graves et al., 2014]. Using a feedforward NN as the controller results in the NTM acting similar to an RNN, with the benefit of increased transparency compared to RNNs. A limitation of the NN controller is that the number of read and write heads imposes a bottleneck on the type of computation the NTM can perform.

Compared to NTMs, LSTMs without external memory do not generalise well to longer inputs. Graves et al. [2014] compared the performance of LSTMs, NTMs with an LSTM controller, and NTMs with a feedforward controller. The experiment required the networks to copy sequences of 8-bit random vectors, where the sequence lengths were randomised between 1 and 20. RMSprop with a momentum of 0.9 and a learning rate of 10^{-4} or 3×10^{-5} was used to train the models. The LSTM model had 3 stacked hidden layers and the number of parameters to be trained far exceeded the number required for the NTMs. This is due to the number of LSTM parameters growing quadratically with the number of hidden units, because of the recurrent connections in the hidden layers. The NTMs made use of a memory

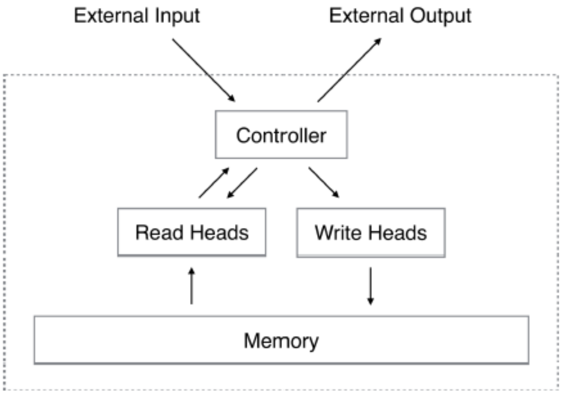


Fig. 2.10 Neural Turing Machine Architecture. The controller network receives inputs from an external environment and emits outputs in response, during each update cycle. The controller also writes and reads from a memory matrix via a set of read and write heads. All components within the dashed line are part of the NTM circuit. (Adapted from: Graves et al. [2014])

bank with size 128x20, and the controllers had one hidden layer. During the backward pass of the implemented models, the gradient components were clipped to a range (-10,10). The training curves are shown in Figure 2.11. The NTMs were found to significantly outperform LSTMs, learning much quicker, retaining longer memory segments, and converging to much lower costs. The ability to retain longer memory segments could be an invaluable benefit in modelling high-resolution physiological signals, where sequences often have many time steps (3,000 or more).

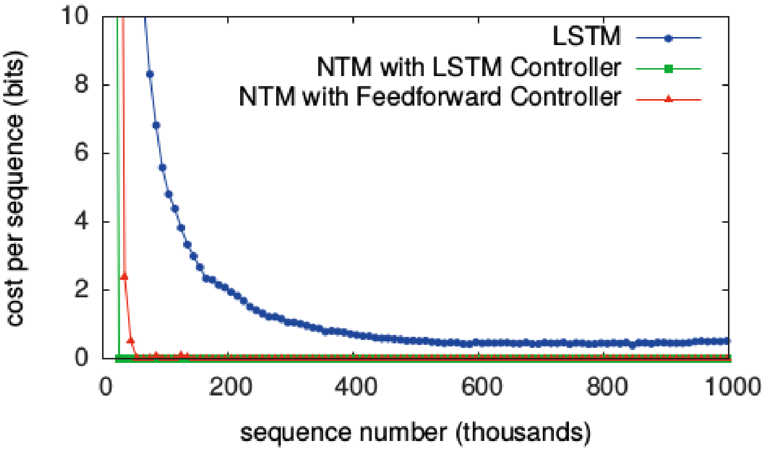


Fig. 2.11 Copy Learning Curves. (Adapted from: Graves et al. [2014])

References

Abston, K. C., Pryor, T. A., Haug, P., and Anderson, J. (1997). Inducing practice guidelines from a hospital database. In *Proceedings of the AMIA Annual Fall Symposium*, page 168. American Medical Informatics Association.

Alagoz, O., Hsu, H., Schaefer, A. J., and Roberts, M. S. (2010). Markov decision processes: a tool for sequential decision making under uncertainty. *Medical Decision Making*, 30(4).

Ansari, A. H., Matic, V., De Vos, M., Naulaers, G., Cherian, P., and Van Huffel, S. (2015). Improvement of an automated neonatal seizure detector using a post-processing technique. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 5859–5862. IEEE.

Atkielski, A. (2007). 12-lead electrocardiogram (ecg) to diagnose cardiac arrhythmias. http://www.washingtonhra.com/ekg-monitoring/12-lead-electrocardiogram-ekg.php. (Accessed on 07/29/2016).

Auli, M., Galley, M., Quirk, C., and Zweig, G. (2013). Joint language and translation modeling with recurrent neural networks. In *EMNLP*, volume 3, page 0.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. (2011). Sequential deep learning for human action recognition. In *Human Behavior Understanding*, pages 29–39. Springer.

Bagley, S. C., White, H., and Golomb, B. A. (2001). Logistic regression in the medical literature:: Standards for use and reporting, with particular attention to one medical domain. *Journal of clinical epidemiology*, 54(10):979–985.

Bauer, M. S. (2002). A review of quantitative studies of adherence to mental health clinical practice guidelines. *Harvard review of psychiatry*, 10(3):138–153.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2001). The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584.

Bengio, Y., Boulanger-Lewandowski, N., and Pascanu, R. (2013). Advances in optimizing recurrent networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8624–8628. IEEE.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.*, 5(2):157–166.

Bennett, C. C. and Hauser, K. (2013). Artificial intelligence framework for simulating clinical decision-making: A Markov decision process approach. *Artificial Intelligence in Medicine*, 57(1):9–19.

Biel, L., Pettersson, O., Philipson, L., and Wide, P. (2001). Ecg analysis: a new approach in human identification. *Instrumentation and Measurement, IEEE Transactions on*, 50(3):808–812.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012). Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. *arXiv:1206.6392 [cs, stat]*.

Braga, P., Portela, F., Santos, M. F., and Rua, F. (2014). Data mining models to predict patient's readmission in intensive care units.

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

Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). *Ann. Statist.*, 26(3):801–849.

Caballero Barajas, K. L. and Akella, R. (2015). Dynamically Modeling Patient's Health State from Electronic Medical Records: A Time Series Approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 69–78, New York, NY, USA. ACM.

Caballero Barajas, K. L. and Akella, R. (2015). Dynamically modeling patient's health state from electronic medical records: A time series approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 69–78. ACM.

Chatzis, S. P. and Tsechpenakis, G. (2010). The infinite hidden markov random field model. *Neural Networks, IEEE Transactions on*, 21(6):1004–1014.

Cheng, B. and Titterington, D. M. (1994). Neural Networks: A Review from a Statistical Perspective. *Statistical Science*, 9(1):2–30.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Choi, E., Bahadori, M. T., and Sun, J. (2015). Doctor ai: Predicting clinical events via recurrent neural networks. *arXiv preprint arXiv:1511.05942*.

Chu, W. and Ghahramani, Z. (2009). Probabilistic models for incomplete multi-dimensional arrays. In *International Conference on Artificial Intelligence and Statistics*, pages 89–96.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Churpek, M. M., Yuen, T. C., Winslow, C., Meltzer, D. O., Kattan, M. W., and Edelson, D. P. (2016). Multicenter comparison of machine learning methods and conventional regression for predicting clinical deterioration on the wards. *Critical Care Medicine*, 44(2):368–374.

Churpek, M. M., Yuen, T. C., Winslow, C., Robicsek, A. A., Meltzer, D. O., Gibbons, R. D., and Edelson, D. P. (2014). Multicenter development and validation of a risk stratification tool for ward patients. *American journal of respiratory and critical care medicine*, 190(6):649–655.

Clermont, G., Angus, D. C., DiRusso, S. M., Griffin, M., and Linde-Zwirble, W. T. (2001). Predicting hospital mortality for patients in the intensive care unit: a comparison of artificial neural networks with logistic regression models. *Critical care medicine*, 29(2):291–296.

Clifton, D. A., Niehaus, K. E., Charlton, P., and Colopy, G. W. (2015). Health Informatics via Machine Learning for the Clinical Management of Patients. *Yearbook of medical Informatics*, 20(1):38–43.

Colopy, G. W., Clifton, D. A., and Roberts, S. J. (2015). Bayesian gaussian processes for identifying the deteriorating patient.

Cooper, G. F., Aliferis, C. F., Ambrosino, R., Aronis, J., Buchanan, B. G., Caruana, R., Fine, M. J., Glymour, C., Gordon, G., Hanusa, B. H., et al. (1997). An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial intelligence in medicine*, 9(2):107–138.

Da Costa, C. (2016). Private communication. 11 January.

Dagum, P. and Galper, A. (1993). Forecasting sleep apnea with dynamic network models. In *Proceedings of the Ninth international conference on Uncertainty in artificial intelligence*, pages 64–71. Morgan Kaufmann Publishers Inc.

Dagum, P., Galper, A., and Horvitz, E. J. (1991). *Temporal Probabilistic Reasoning: Dynamic Network Models for Forecasting*. Knowledge Systems Laboratory, Medical Computer Science, Stanford University.

De Mulder, W., Bethard, S., and Moens, M.-F. (2015). A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language*, 30(1):61–98.

Delen, D., Walker, G., and Kadam, A. (2005). Predicting breast cancer survivability: a comparison of three data mining methods. *Artificial intelligence in medicine*, 34(2):113–127.

Dietterich, T. G. (2002). Machine Learning for Sequential Data: A Review. In Caelli, T., Amin, A., Duin, R. P. W., de Ridder, D., and Kamel, M., editors, *Structural, Syntactic, and Statistical Pattern Recognition*, number 2396 in Lecture Notes in Computer Science, pages 15–30. Springer Berlin Heidelberg.

Efron, B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331.

Faul, S., Gregorcic, G., Boylan, G., Marnane, W., Lightbody, G., and Connolly, S. (2007). Gaussian process modeling of eeg for the detection of neonatal seizures. *Biomedical Engineering, IEEE Transactions on*, 54(12):2151–2162.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning*, 29(2-3):131–163.

Gangadharan, V. (2013). *Automated Multi-Parameter Monitoring of Neo-Nates*. PhD thesis, UCL (University College London).

Gers, F. A. and Schmidhuber, J. (2000). Recurrent nets that time and count. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 189–194. IEEE.

Ghahramani, Z. (2001). An introduction to hidden markov models and bayesian networks. *Int. J. Patt. Recogn. Artif. Intell.*, 15(01):9–42.

Ghassemi, M., Pimentel, M. A., Naumann, T., Brennan, T., Clifton, D. A., Szolovits, P., and Feng, M. (2015). A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *AAAI*, pages 446–453.

Goldberger, A. L. (1996). Non-linear dynamics for clinicians: chaos theory, fractals, and complexity at the bedside. *The Lancet*, 347(9011):1312–1314.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1764–1772.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):855–868.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Greene, B., de Chazal, P., Boylan, G., Connolly, S., and Reilly, R. (2007). Electrocardiogram Based Neonatal Seizure Detection. *IEEE Trans. Biomed. Eng.*, 54(4):673–682.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2015). Lstm: A search space odyssey. *arXiv preprint arXiv:1503.04069*.

Gregorcic, G. and Lightbody, G. (2005). Gaussian process approaches to nonlinear modelling for control. *IEE Control Engineering Series*, 70:177.

Güiza, F., Depreitere, B., Piper, I., Van den Berghe, G., and Meyfroidt, G. (2013). Novel methods to predict increased intracranial pressure during intensive care and long-term neurologic outcome after traumatic brain injury: Development and validation in a multicenter dataset*. *Critical care medicine*, 41(2):554–564.

Güiza, F., Ramon, J., Meyfroidt, G., Blockeel, H., Bruynooghe, M., and Van Den Berghe, G. (2006). Predicting blood temperature using gaussian processes. *Journal of Critical Care*, 21(4):354–355.

Guiza Grandas, F., Ramon, J., and Blockeel, H. (2006). Gaussian processes for prediction in intensive care. In *Gaussian Processes in Practice Workshop*, pages 1–4.

Gultepe, E., Green, J. P., Nguyen, H., Adams, J., Albertson, T., and Tagkopoulos, I. (2014). From vital signs to clinical outcomes for patients with sepsis: A machine learning basis for a clinical decision support system. *J. Am. Med. Inform. Assoc.*, 21(2):315–325.

Gultepe, E., Nguyen, H., Albertson, T., and Tagkopoulos, I. (2012). A Bayesian network for early diagnosis of sepsis patients: A basis for a clinical decision support system. In *2012 IEEE 2nd International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, pages 1–5.

Halpern, N. A. and Pastores, S. M. (2010). Critical care medicine in the united states 2000–2005: An analysis of bed numbers, occupancy rates, payer mix, and costs*. *Critical care medicine*, 38(1):65–71.

Hill, T., O'Connor, M., and Remus, W. (1996). Neural network models for time series forecasts. *Management science*, 42(7):1082–1092.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hooker, G. (2012). Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

Hravnak, M., DeVita, M. A., Clontz, A., Edwards, L., Valenta, C., and Pinsky, M. R. (2011). Cardiorespiratory instability before and after implementing an integrated monitoring system. *Crit Care Med*, 39(1):65–72.

Hughes, N. P. and Tarassenko, L. (2006). Probabilistic models for automated ECG interval analysis in Phase 1 Studies. Technical report, BSP 08-01.

Ivanov, P. C., Amaral, L. A. N., Goldberger, A. L., Havlin, S., Rosenblum, M. G., Struzik, Z. R., and Stanley, H. E. (1999). Multifractality in human heartbeat dynamics. *Nature*, 399(6735):461–465.

Jennett, B. and Bond, M. (1975). Assessment of outcome after severe brain damage: A practical scale. *The Lancet*, 305(7905):480–484.

Johnson, M. J. and Willsky, A. S. (2013). Bayesian nonparametric hidden semi-markov models. *The Journal of Machine Learning Research*, 14(1):673–701.

Kalchbrenner, N., Danihelka, I., and Graves, A. (2015). Grid long short-term memory. *arXiv preprint arXiv:1507.01526*.

Kim, S., Kim, W., and Park, R. W. (2011). A comparison of intensive care unit mortality prediction models through the use of data mining techniques. *Healthcare informatics research*, 17(4):232–243.

Knaus, W. A., Wagner, D. P., Draper, E. A., Zimmerman, J. E., Bergner, M., Bastos, P. G., Sirio, C. A., Murphy, D. J., Lotring, T., and Damiano, A. (1991). The apache III prognostic system. Risk prediction of hospital mortality for critically ill hospitalized adults. *Chest*, 100(6):1619–1636.

Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109.

Koutnik, J., Greff, K., Gomez, F., and Schmidhuber, J. (2014). A clockwork rnn. *arXiv preprint arXiv:1402.3511*.

Lachiche, N. and Flach, P. (2003). Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In *ICML*, pages 416–423.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Lange, J. M., Hubbard, R. A., Inoue, L. Y., and Minin, V. N. (2015). A joint model for multistate disease processes and random informative observation times, with applications to electronic medical records data. *Biometrics*, 71(1):90–101.

Lapuerta, P., Azen, S. P., and LaBree, L. (1995). Use of neural networks in predicting the risk of coronary artery disease. *Computers and Biomedical Research*, 28(1):38–52.

Le, Q. V., Jaitly, N., and Hinton, G. E. (2015). A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*.

Le Compte, A. J., Lee, D. S., Chase, J. G., Lin, J., Lynn, A., and Shaw, G. M. (2010). Blood glucose prediction using stochastic modeling in neonatal intensive care. *IEEE Trans. Biomed. Eng.*, 57(3):509–518.

Lee, B., Lee, T., Na, B., and Yoon, S. (2015). Dna-level splice junction prediction using deep recurrent neural networks. *arXiv preprint arXiv:1512.05135*.

Liniger, T. J. (2009). *Multivariate hawkes processes*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 18403, 2009.

Lipton, Z. C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Lipton, Z. C., Elkan, C., and Naryanaswamy, B. (2014). Optimal Thresholding of Classifiers to Maximize F1 Measure. In Calders, T., Esposito, F., Hüllermeier, E., and Meo, R., editors, *Machine Learning and Knowledge Discovery in Databases*, number 8725 in Lecture Notes in Computer Science, pages 225–239. Springer Berlin Heidelberg.

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzell, R. (2015). Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*.

Lugovaya, T. S. (2005). Biometric human identification based on ECG. PhysioNey.

Mani, S., Ozdas, A., Aliferis, C., Varol, H. A., Chen, Q., Carnevale, R., Chen, Y., Romano-Keeler, J., Nian, H., and Weitkamp, J.-h. (2014). Medical decision support using machine learning for early detection of late-onset neonatal sepsis. *J. Am. Med. Inform. Assoc.*, 21(2):326–336.

Mani, S., Shankle, W. R., Dick, M. B., and Pazzani, M. J. (1999). Two-stage machine learning model for guideline development. *Artificial intelligence in medicine*, 16(1):51–71.

McGlynn, E. A., Asch, S. M., Adams, J., Keesey, J., Hicks, J., DeCristofaro, A., and Kerr, E. A. (2003). The quality of health care delivered to adults in the united states. *New England journal of medicine*, 348(26):2635–2645.

Meehl, P. E. (1986). Causes and effects of my disturbing little book. *Journal of personality assessment*, 50(3):370–375.

Meyer, D., Leisch, F., and Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, 55(1):169–186.

Meyfroidt, G., Güiza, F., Ramon, J., and Bruynooghe, M. (2009). Machine learning techniques to examine large patient databases. *Best Pract. Res. Clin. Anaesthesiol.*, 23(1):127–143.

Moon, H., Ahn, H., Kodell, R. L., Baek, S., Lin, C.-J., and Chen, J. J. (2007). Ensemble methods for classification of patients for personalized medicine with high-dimensional data. *Artificial intelligence in medicine*, 41(3):197–207.

Morgan, M. A. (2015). Receiver operating characteristic (ROC) curve. http://radiopaedia.org/images/11592546. (Accessed on 08/02/2016).

Morik, K., Imboff, M., Brockhausen, P., Joachims, T., and Gather, U. (2000). Knowledge discovery and knowledge validation in intensive care. *Artificial Intelligence in Medicine*, 19(3):225–249.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Nodelman, U., Shelton, C. R., and Koller, D. (2002). Continuous time bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 378–387. Morgan Kaufmann Publishers Inc.

Ohmann, C., Moustakis, V., Yang, Q., Lang, K., Group, A. A. P. S., et al. (1996). Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain. *Artificial intelligence in medicine*, 8(1):23–36.

Ongenae, F., Van Looy, S., Verstraeten, D., Verplancke, T., Benoit, D., De Turck, F., Dhaene, T., Schrauwen, B., and Decruyenaere, J. (2013). Time series classification for the prediction of dialysis in critically ill patients using echo statenetworks. *Engineering Applications of Artificial Intelligence*, 26(3):984–996.

OpenCV (2014). Introduction to Support Vector Machines — OpenCV 2.4.12.0 documentation. http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. (Accessed on 29/11/2015).

Oresko, J. J., Jin, Z., Cheng, J., Huang, S., Sun, Y., Duschl, H., and Cheng, A. C. (2010). A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing. *Information Technology in Biomedicine, IEEE Transactions on*, 14(3):734–740.

Orr, G. B. and Müller, K.-R. (2003). *Neural Networks: Tricks of the Trade*. Springer.

Pan, J. and Tompkins, W. J. (1985). A real-time qrs detection algorithm. *Biomedical Engineering, IEEE Transactions on*, 3:230–236.

Pham, V., Bluche, T., Kermorvant, C., and Louradour, J. (2014). Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 285–290. IEEE.

Picard, R. R. and Berk, K. N. (1990). Data splitting. *The American Statistician*, 44(2):140–147.

Pimentel, M. A., Santos, M. D., Springer, D. B., and Clifford, G. D. (2015). Heart beat detection in multimodal physiological data using a hidden semi-markov model and signal quality indices. *Physiological measurement*, 36(8):1717.

Pirracchio, R., Petersen, M. L., Carone, M., Rigon, M. R., Chevret, S., and van der Laan, M. J. (2015). Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): A population-based study. *The Lancet Respiratory Medicine*, 3(1):42–52.

Pregibon, D. (1981). Logistic Regression Diagnostics. *The Annals of Statistics*, 9(4):705–724.

Quinonero-Candela, J., Rasmussen, C. E., and Williams, C. K. (2007). Approximation methods for gaussian process regression. *Large-scale kernel machines*, pages 203–223.

Rabiner, L. and Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Mag.*, 3(1):4–16.

Ramon, J., Fierens, D., Güiza, F., Meyfroidt, G., Blockeel, H., Bruynooghe, M., and Van Den Berghe, G. (2007). Mining data from intensive care patients. *Advanced Engineering Informatics*, 21(3):243–256.

Rangayyan, R. M. (2004). *Biomedical image analysis*. CRC press.

Rasmussen, C. E. and Williams, C. K. (2006). Gaussian processes for machine learning. *The MIT Press, Cambridge, MA, USA*, 38:715–719.

Ryan, D. P., Daley, B. J., Wong, K., and Zhao, X. (2013). Prediction of icu in-hospital mortality using a deep boltzmann machine and dropout neural net. In *Biomedical Sciences and Engineering Conference (BSEC), 2013*, pages 1–4. IEEE.

Santos-Garcıa, G., Varela, G., Novoa, N., and Jiménez, M. F. (2004). Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble. *Artificial Intelligence in Medicine*, 30(1):61–69.

Sargent, D. J. (2001). Comparison of artificial neural networks with other statistical approaches. *Cancer*, 91(S8):1636–1642.

Saria, S., Rajani, A. K., Gould, J., Koller, D., and Penn, A. A. (2010). Integration of Early Physiological Responses Predicts Later Illness Severity in Preterm Infants. *Sci. Transl. Med.*, 2(48):48ra65–48ra65.

Sboner, A. and Aliferis, C. F. (2005). Modeling clinical judgment and implicit guideline compliance in the diagnosis of melanomas using machine learning. In *AMIA Annual Symposium Proceedings*, volume 2005, pages 664–668. American Medical Informatics Association.

Scalzo, F., Asgari, S., Kim, S., Bergsneider, M., and Hu, X. (2012). Bayesian tracking of intracranial pressure signal morphology. *Artificial intelligence in medicine*, 54(2):115–123.

Scalzo, F. and Hu, X. (2013). Semi-supervised detection of intracranial pressure alarms using waveform dynamics. *Physiological measurement*, 34(4):465.

Schaefer, A. J., Bailey, M. D., Shechter, S. M., and Roberts, M. S. (2005). Modeling medical treatment using markov decision processes. In *Operations research and health care*, pages 593–612. Springer.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5(2):197–227.

Shavdia, D. (2007). *Spetic Shock: Providing Early Warnings Through Multivariate Logistic Regression Models*. PhD thesis, MIT.

Shen, T.-W., Tompkins, W., and Hu, Y. (2002). One-lead ecg for identity verification. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint*, volume 1, pages 62–63. IEEE.

Siegelmann, H. T. and Sontag, E. D. (1995). On the computational power of neural nets. *Journal of computer and system sciences*, 50(1):132–150.

Singh, B., Kapur, N., and Kaur, P. (2012). Speech recognition with hidden markov model: a review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(3).

Smielewski, P. (2011). Cambridge university: Neurosurgery unit | about ICM+. http://www.neurosurg.cam.ac.uk/pages/ICM/about.php. (Accessed on 11/02/2016).

Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J Mach Learn Res*, 15(1):1929–1958.

Stanculescu, I., Williams, C. K., and Freer, Y. (2014). Autoregressive hidden markov models for the early detection of neonatal sepsis. *Biomedical and Health Informatics, IEEE Journal of*, 18(5):1560–1570.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1139–1147.

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

Sutton, C. and McCallum, A. (2010). An Introduction to Conditional Random Fields. *arXiv:1011.4088 [stat]*.

Svátek, V., Ríha, A., Peleska, J., and Rauch, J. (2003). Analysis of guideline compliance–a data mining approach. *Studies in health technology and informatics*, 101:157–161.

Tarassenko, L., Clifton, D. A., Pinsky, M. R., Hravnak, M. T., Woods, J. R., and Watkinson, P. J. (2011). Centile-based early warning scores derived from statistical distributions of vital signs. *Resuscitation*, 82(8):1013–1018.

Temko, A., Thomas, E., Marnane, W., Lightbody, G., and Boylan, G. (2011). EEG-based neonatal seizure detection with Support Vector Machines. *Clinical Neurophysiology*, 122(3):464–473.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse gaussian processes. In *AISTATS*, volume 12, pages 567–574.

Tong, Y., Frize, M., and Walker, R. (2002). Extending ventilation duration estimations approach from adult to neonatal intensive care patients using artificial neural networks. *IEEE Trans. Inf. Technol. Biomed.*, 6(2):188–191.

Townsend, G., Graimann, B., and Pfurtscheller, G. (2004). Continuous EEG classification during motor imagery-simulation of an asynchronous BCI. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 12(2):258–265.

Tsien, C. L., Kohane, I. S., and McIntosh, N. (2000). Multiple signal integration by decision tree induction to detect artifacts in the neonatal intensive care unit. *Artif. Intell. Med.*, 19(3):189–202.

Tu, J. V. and Guerriere, M. R. J. (1993). Use of a Neural Network as a Predictive Instrument for Length of Stay in the Intensive Care Unit Following Cardiac Surgery. *Computers and Biomedical Research*, 26(3):220–229.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.

Unnikrishnan, R., Pantofaru, C., and Hebert, M. (2005). A Measure for Objective Evaluation of Image Segmentation Algorithms. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pages 34–34.

Vairavan, S., Eshelman, L., Haider, S., Flower, A., and Seiver, A. (2012). Prediction of mortality in an intensive care unit using logistic regression and a hidden markov model. In *Computing in Cardiology (CinC), 2012*, pages 393–396. IEEE.

Vikman, S., Mäkikallio, T. H., Yli-Mäyry, S., Pikkujämsä, S., Koivisto, A.-M., Reinikainen, P., Airaksinen, K. J., and Huikuri, H. V. (1999). Altered complexity and correlation properties of RR interval dynamics before the spontaneous onset of paroxysmal atrial fibrillation. *Circulation*, 100(20):2079–2084.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269.

Wang, Z., Kuruoğlu, E. E., Yang, X., Xu, Y., and Huang, T. S. (2011). Time varying dynamic bayesian network for nonstationary events modeling and online inference. *Signal Processing, IEEE Transactions on*, 59(4):1553–1568.

Williams, C., Quinn, J., and McIntosh, N. (2006). Factorial Switching Kalman Filters for Condition Monitoring in Neonatal Intensive Care. *Neural Inf. Process.*

Xu, R., Wunsch II, D., and Frank, R. (2007). Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 4(4):681–692.

Yao, K., Cohn, T., Vylomova, K., Duh, K., and Dyer, C. (2015). Depth-gated recurrent neural networks. *arXiv preprint arXiv:1508.03790*.

Yi, W., Park, K., and Jeong, D. (2003). Personal identification from ecg measured without body surface electrodes using probabilistic neural networks. In *World Congress on Medical Physics and Biomedical Engineering*.

Yoo, I., Alafaireet, P., Marinov, M., Pena-Hernandez, K., Gopidi, R., Chang, J.-F., and Hua, L. (2011). Data Mining in Healthcare and Biomedicine: A Survey of the Literature. *J Med Syst*, 36(4):2431–2448.

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zhai, H., Brady, P., Li, Q., Lingren, T., Ni, Y., Wheeler, D. S., and Solti, I. (2014). Developing and evaluating a machine learning based algorithm to predict the need of pediatric intensive care unit transfer for newly hospitalized children. *Resuscitation*, 85(8):1065–1071.

Zhang, Y., Silvers, C., and Randolph, A. (2007). Real-Time Evaluation of Patient Monitoring Algorithms for Critical Care at the Bedside. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007*, pages 2783–2786.

Zhang, Y. and Szolovits, P. (2008). Patient-specific learning in real time for adaptive monitoring in critical care. *Journal of biomedical informatics*, 41(3):452–460.

Zhang, Z. (2016). When doctors meet with AlphaGo: Potential application of machine learning to clinical medicine. *Ann Transl Med*, 4(6).

Zheng, F. and Webb, G. I. (2010). Tree augmented naive bayes. In *Encyclopedia of Machine Learning*, pages 990–991. Springer.

Zhou, Z.-H., Jiang, Y., Yang, Y.-B., and Chen, S.-F. (2002). Lung cancer cell identification based on artificial neural network ensembles. *Artificial Intelligence in Medicine*, 24(1):25–36.

Zhu, L. (2013). Nonlinear hawkes processes. *arXiv preprint arXiv:1304.7531*.