Modelling skill retention in Space Fortress using machine learning

Bachelor thesis Artificial Intelligence

7.5 ECTS

April 2020

David Lieffijn

5587301

supervised by

Dr. Armon Toubman

Royal NLR

Dr. Tejaswini Deoskar

Utrecht University

Abstract

Research on the optimal training frequency for highly skilled professionals is not well established. Finding an optimal training frequency could presumably lower costs, maintain a higher performance and create a more pleasant work environment. Royal Netherlands Aerospace Centre (NLR) started doing research on skill retention/decay in highly skilled professionals such as fighter pilots. By collecting participant data from their version of the online game Space Fortress (SF) a retention model will be created. Ideally, the final model can be extrapolated to predict an optimal training schedule for pilots and other professionals.

In this thesis suitable techniques to create an accurate forecasting model for SF are explored, by studying machine learning techniques applied in Time Series Forecasting (TSF) and Knowledge Tracing (KT). After reviewing the literature, the most promising techniques will be discussed. A recommendation regarding many aspects of the challenge will be given, with the main focus on interpolation and prediction using a Long Short-Term Memory (LSTM) in combination with feature engineering.

Contents

1 Introduction **4**

2 Background **6**

2.1 Research project OiT 6
2.2 Skill retention 6
2.3 Space Fortress 6
2.4 Design 8
2.5 Output data 9
2.6 Challenges to solve 10

3 Literature study **12**

3.1 Time Series Forecasting 12
3.1.1 A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition 14
3.1.2 A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction 15
3.1.3 A Memory-Network Based Solution for Multivariate Time-Series Forecasting . . 16
3.1.4 Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches 18
3.1.5 Recurrent Neural Networks for Multivariate Time Series with Missing Values . . 19
3.1.6 Learning Adaptive Forecasting Models from Irregularly Sampled Multivariate Clinical Data 20
3.1.7 Learning to Diagnose with LSTM Recurrent Neural Networks 21
3.1.8 Predicting the Risk of Heart Failure With EHR Sequential Data Modeling 21
3.1.9 Interpolation-Prediction Networks for Irregularly Sampled Time Series 22
3.2 Knowledge Tracing 23
3.2.1 Knowledge Tracing and Prediction of Future Trainee Performance 24
3.2.2 Knowledge Tracing in Sequential Learning of Inflected Vocabulary 24
3.2.3 Modeling Skill Combination Patterns for Deeper Knowledge Tracing 25
3.2.4 Going Deeper with Deep Knowledge Tracing 26
3.2.5 Incorporating Rich Features into Deep Knowledge Tracing 26
3.2.6 Second Language Acquisition Modeling 27
3.2.7 Second Language Acquisition Modeling: An Ensemble Approach 28

4 Discussion **29**

4.1 Challenges . . . 29

4.1.1 Temporal data/forecasting . . . 29

4.1.2 High-dimensionality/feature engineering . . . 30

4.1.3 Sparse and irregularly sampled data . . . 30

5 Conclusion **31**

A Techniques **36**

A.1 Long Short-Term Memory . . . 36

A.2 Adam optimizer . . . 37

A.3 Autoencoder . . . 37

1 Introduction

Highly skilled professionals, such as fighter pilots, need to keep their skill level constant and on a high level. This is why fighter pilots spend many hours in recurrent training, with the average US fighter pilot flying approximately 200 hours each year (Haynes, 2008). Different skills, varying from situational awareness, memorization or selective attention are trained to keep the pilot's abilities proficient (Carretta et al., 1993). Although different methods of training could reduce training hours (Smith, 1976), no extensive research has been conducted to find the optimal frequency of recurrent training. Beyond that, the curricula do not differentiate between individual pilots. Why should two pilots perform the same hours of training flights if one pilot maintains their skills without effort and the other pilot has significant performance decay without training?

This is why Royal Netherlands Aerospace Centre (NLR), together with Netherlands Organisation for Applied Scientific Research (TNO), started a research project on the topic of skill acquisition and retention in complex tasks. The project consists of smaller related studies, with the NLR specifically focussing on retention: how skilled a person stays after a period of performance decay as a result of no training. A predictive model of skill retention will be built using different measurements from the pilots' training. In the end, the model should be able to differentiate between unique abilities, like motor skill, memory or procedures. Ideally this model can predict future performance decay – the measure of skill retention. More optimally timed recurrent trainings could improve the performance and work environments of the pilots, while cutting costs. In the future, this knowledge could be applied to different domains such as medical training.

Currently, there is not enough performance data available from pilots to train a predictive model. Furthermore, data collected in the real world is often distorted or noisy. To represent the different tasks a pilot has to master - in a smaller, controlled environment - NLR and TNO created their own version of the game Space Fortress and made it available to the public. Space Fortress is "a game [originally] developed in the Cognitive Psychophysiology Laboratory (University of Illinois) as an experimental task for the study of complex skill and its acquisition. The object of the Space Fortress game is to shoot missiles at and destroy a space fortress. Missiles are fired from a spaceship whose movement is controlled by the subject. In addition to destroying the fortress, the subject must protect his ship against damage" (Mané & Donchin, 1989). Anyone with a computer and internet connection can play the game online (at https://spacefortress.nlr.nl) and contribute to this study. After a training phase the participant is asked to wait for a determined number of weeks (up to a full year), and asked to play again (van der Pal & Toubman, 2020). This way, the effect of different retention intervals on performance can be studied.

Since almost every possible metric from the game is measured the output data has a high-dimensional

feature space. This is why NLR opted for a machine learning approach to process the obtained data. The algorithm has to forecast future performance of a participant and needs to handle high-dimensional and temporal data with variable intervals. Now the main question arises: *what is the most suitable machine learning technique to model skill retention with data from Space Fortress?*

In this thesis I will firstly shed light on the background of the research project, the SF game and the technical challenges that have to be solved (section 2). Then I will compare different approaches taken by other researchers, broadly grouped by field of research. The papers are divided into two main domains: Time Series Forecasting (TSF) (section 3.1) and Knowledge Tracing (KT) (section 3.2). In the end, relevant techniques found in literature are reviewed (section 4). Based on the similarities and differences between the SF project and the discussed papers, a recommendation is given on which direction to take (section 5).

2 Background

2.1 Research project OiT

NLR and TNO started the research project Education & individual Training (OiT) with the goal to build up knowledge for the Dutch Ministry of Defence about personalizing training schedules. To be more precise, it is aimed at the retention/continuation training of fighter pilots (and other personnel) to maintain their combat readiness. These pilots need hours of expensive and intensive retention training throughout the year to maintain their proficiency. With enough data, a retention model could forecast future skill decay and therefore be able to predict if and when a pilot needs training. More optimally timed recurrent trainings could have several advantages, for instance:

- No unneeded trainings have to be carried out, presumably lowering costs;
- A higher average performance can be maintained by detecting the optimal moment a pilot needs training;
- The work environment may become more pleasant as there will be less overly complex or frustrating trainings.

2.2 Skill retention

The OiT project is focussing on skill retention or, alternatively, skill decay. Retention means how well a learned skill is maintained as time passes. Skill retention is already researched for more than a century (see e.g. Reed, 1918). A differentiation can be made between short-term retention and long-term retention. Short-term retention functions were found to be highly variable, although practice repetitions minimize forgetting. In long-term retention, motor skills are believed to have a high retention, meaning the skill is well maintained over months or years (Adams, 1987). As complex tasks in the OiT project require different skills, we could expect to see different retention curves for different skills.

2.3 Space Fortress

Real world data from fighter pilots is scarce, as there are not many pilots available for research and data collection is difficult. To create an initial retention model, another manner of generating data of complex skills was sought and found in the video game Space Fortress (SF). The game was initially developed as an experimental task for the study of complex skill and its acquisition (Mané & Donchin, 1989). Skills involved in mastering the game are multidimensional: perceptual, cognitive, motor skills together with specific knowledge of the rules and strategy are required of the player.

While playing the game, the player controls a spaceship in a rectangular space (Figure 1). The objective is to weaken the fortress in the middle of the screen by shooting missiles at it, and eventually destroying it. However, the fortress shoots back at the player. Dodging shots while maneuvering through space requires motor skills (CNTRL) and the destruction of the fortress requires timing since shots have to be timed in order to be successful (INTRVL). To make the game challenging, the player has to remember certain letters at the beginning of the game. In the game mines will appear with a specific letter showing on screen. The player can tell if the mine is friendly or foe by remembering if the letter is in the learned letters (IFF). Lastly, power-ups will appear, forcing the player to divide its attention and make the decision between short-term or long-term rewards. In the end, the aim is to collect as many points possible.

Figure 1: An in-game snapshot of Space Fortress (SF).

The skills required for SF are similar to the tasks that pilots have to master. The Israeli Air Force flight school even showed that training in SF can improve certain skills of fighter pilots, such as efficient

control and management of attention under high task load (Gopher et al., 1994). When a retention model is created from the SF data, the model can be applied to other projects in the OiT research. The model could be tested and updated with simulator data and eventually be applied on real life fighter jet training.

2.4 Design

The SF project is an experimental research, where a group of participants playing the game online provides the data. Furthermore, the research is exploratory, meaning it seeks to generate a posteriori hypotheses by examining the data and conclude from the gathered data. A custom environment was built to facilitate the study, called the Space Fortress Adaptive Instructional System (SF-AIS). The SF-AIS is an interactive website where participants learn to play SF (van der Pal & Toubman, 2020). After signing up, the participant is asked to take the initial training.

Initially, the practice sessions are configured to be simpler than the full game and the difficulty builds up to the point where the participant is able to play the game at a sufficient level. After this level is reached, the participant enters the 'retention model' phase, where the participant is asked to not play the game for an interval between 1 and 52 weeks. This interval will be randomly decided in the beginning of the experiment. When more data becomes available the retention interval can be calculated to provide a more optimally timed training. After the interval is over, participants take the 'retention test' and play the game again to determine their current skill level. A refresher training is given to update the skill level before the participant enters the waiting period again. The whole process is visualized in Figure 2.

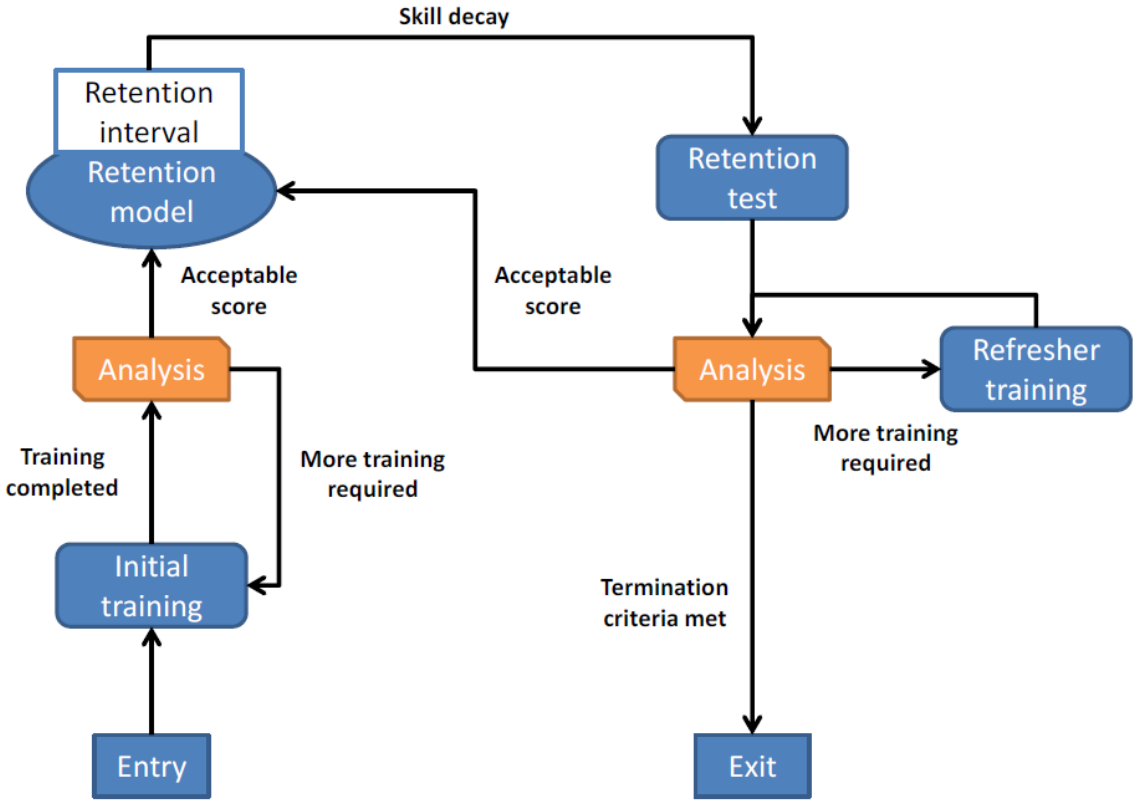

Figure 2: Workflow of the participants in the SF-AIS (image from van der Pal and Toubman, 2020).

2.5 Output data

Many variables are tracked every game, including the points, control, velocity, interval, speed and the number of shots fired. These values are displayed while playing (Figure 1). Besides that, secondary variables are registered, varying from timing of button presses to hits. In total more than 50 different metrics are stored per playing session. Each row in the data set represents one session of one player. A player usually has more than one played session, which can be either a training session or the full game (Figure 3). All these metrics result in high-dimensional data, meaning the output has many features. The metrics are a combination of discrete and continuous values.

Figure 3: A simplified visualization of what the data looks like. Columns represent the features, rows show different sessions grouped by participant (player). The continuous values of the features are represented by varying colors.

2.6 Challenges to solve

As of now there is not enough data available from the SF-AIS to make accurate predictions on trends in the output data. However, the data has certain properties that can already be identified. The eventual solution for modelling the skill retention has to be able to deal with these properties. Therefore, we can rephrase the properties as challenges that have to be solved by the implementation. Guided by these challenges we can then look for relevant literature.

The first challenge to solve is the **high-dimensionality** of the data, meaning many features (multivariate) are collected per sample and put into the model. Inputting too many features can lead to overfitting the model. Because of this, **feature engineering** could help selecting or combining features. Furthermore, we know the data is **temporal**: the data is sequentially ordered in time, with each sample having a timestamp. This comes with another problem, since the samples are not taken

at regular intervals. Participants can decide themselves when they want to play and besides that, the retention interval is inherently variable. This results in data with variable intervals with gaps from hours to weeks or even years: the data is **sparse and irregularly sampled**. As most learning algorithms require fixed sample intervals, we will review literature that creates models with missing values in the data. The main goal of the retention model is to **forecast** or predict an optimal retention interval, which could span many time steps in the future (multi-step ahead prediction). The ultimate objective is to predict the optimal retention interval for SF, which is a form of regression.

3 Literature study

Taking the challenges in section 2.6 into consideration, we review literature to explore how the authors solved related challenges. The SF data shares strong similarities with Time Series Forecasting (TSF) data (section 3.1): the data is temporal in nature and the goal is to forecast future based on trends in past values. Besides that, it often deals with high-dimensional data. However, most TSF papers deal with regularly sampled data. For data with missing values, time gaps or irregularly sampled data points, medical research could be valuable for our model. The medical research shows strong similarities with TSF as predicting clinical events in the future is often the goal. Unlike time series, medical data is usually sampled at varying intervals with many missing values or gaps. Therefore, we also investigate literature on the prediction of clinical data.

Moving away from TSF, overlap is also found between our data and Knowledge Tracing (KT) (section 3.2) data. KT attempts to predict the skill or knowledge of a student in the future, as is the objective of the OiT project. However, the time factor is usually less important as the goal is mainly to predict the next exercise outcome. A form of knowledge tracing using deep neural networks is called Deep Knowledge Tracing (DKT) and has shown some significant results in the last decade. A specific variation on knowledge tracing is Second Language Acquisition Modelling (SLAM), which uses data from online language learning courses to predict student performance.

Sixteen papers, coming from different fields of study, will be discussed below. The papers have been selected based on three criteria: relevancy, recency and importance. Relevancy was initially judged by matching the papers to the challenges posed in section 2.6. Moreover, recently published papers are preferred over older papers. As machine learning techniques improve rapidly, relatively recently published papers are generally achieving a higher performance compared to older algorithms (see e.g. section 3.1.2). Lastly, papers with more citations are favored over similar papers with less citations. All papers are summarized, with their approach and results listed. Similarities and differences between the SF data and the data from the respective paper will be discussed, with relevant techniques mentioned and further explained.

3.1 Time Series Forecasting

TSF is one of the many applications of machine learning. The aim is not only to describe or collect data, but to predict future values such as events or prices based on trends in the data set. Past samples or observations of the same variable are collected and analysed to create a model describing the relationship between values and the passing of time. This model can be used to extrapolate the series into the future, thus predicting upcoming data. Especially when little knowledge is available on what exactly generates the underlying data and how variables relate to each other, TSF is useful.

The prediction algorithms have been widely applied in many areas, for example weather forecasting (Maqsood et al., 2004), medical diagnosis (Jin et al., 2018), financial forecasting (Cao & Tay, 2003) and more specifically stock price prediction (Pai & Lin, 2005).

Two of the most widely used algorithms to model time series are Autoregressive Moving Average (ARMA) (Whittle, 1951) and Autoregressive Integrated Moving Average (ARIMA) with the Box-Jenkins method (Box et al., 2015). Furthermore, linear Support Vector Regression (SVR) (Cao & Tay, 2003) is also used frequently. Although ARMA, ARIMA and linear SVR proved to be successful in creating forecasting models, the models usually assume a certain distribution, function form or linear relationship and may not be able to capture complex underlying relationships with nonlinear data.

More recently, different forms of an Artificial Neural Network (ANN) have been employed for TSF in fields such as finance (Kim, 2006) and hydrology (Jain & Kumar, 2007). RNNs have proven to be especially useful when attempting to model non-linear dependencies on large amounts of data (Qin et al., 2017). There is, however, a drawback on the use of RNNs when attempting to model long-term dependencies: vanishing gradients. The vanishing gradient problem is a difficulty in training a backpropagated network's weights. In some cases, the gradient will become too small, preventing the weight from changing again. The opposite, exploding gradients, cause similar problems. This makes a traditional RNN less suitable for learning long-term time series (Bengio et al., 1994). A specific memory cell added to a RNN can prevent this problem, as demonstrated in a Long Short-Term Memory (LSTM) network (Gers et al., 2000). More technical details can be found in Appendix A.1.

The first four papers discussed use TSF in varying domains and focus more on high-dimensional data and less on irregularly sampled data. The last five papers do however address the irregularly sampling better, usually in clinical context.

Contents

3.1.1 A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition 14

3.1.2 A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction 15

3.1.3 A Memory-Network Based Solution for Multivariate Time-Series Forecasting . . 16

3.1.4 Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches 18

3.1.5 Recurrent Neural Networks for Multivariate Time Series with Missing Values . . 19

3.1.6 Learning Adaptive Forecasting Models from Irregularly Sampled Multivariate Clinical Data 20
3.1.7 Learning to Diagnose with LSTM Recurrent Neural Networks 21
3.1.8 Predicting the Risk of Heart Failure With EHR Sequential Data Modeling 21
3.1.9 Interpolation-Prediction Networks for Irregularly Sampled Time Series 22

3.1.1 A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition

Taieb et al., 2011

Summary The aim of this paper is to show how the choice of forecasting strategy could influence the performance of the multi-step ahead forecasts. A multi-step ahead forecast consists of predicting - as the name implies – multiple time steps into the future. Predicting multiple steps at once is a challenge, since the uncertainty increases when the forecast horizon becomes larger.

Multiple known approaches are tested on the NN5 dataset (111 daily times series drawn from a homogeneous population of empirical cash money withdrawals at ATMs) as a benchmark. Three Single-Output strategies are tested: Recursive, Direct and DirRec. Besides that, a Multi-Input Multi-Output (MIMO) strategy is selected. Finally, the DIRect and miMO (DIRMO) strategy is formed by combining Direct and MIMO.

Before testing, every configuration is subjected of several preprocessing steps. The first one is the removal of gaps, as some data contains anomalies: null values and missing observations. The gaps are replaced with the median of the surrounding data. Next, the data is deseasonalized and the dimension selection is embedded. After this, a variable selection procedure requires the setting of two elements: the relevance criterion (estimates quality of selected variable) and the search procedure (describes policy to explore the input space). Delta Test (Pi & Peterson, 1994) is adopted as the relevance criterion. Lastly, the winning model is selected. The performance of the forecasting methods over one times series was assessed by the Symmetric Mean Absolute Percentage of Error (SMAPE) measure. Based on the results, the most consistent finding is that the Multiple-Output (MIMO and DIRMO) approaches are invariably better than Single-Output approaches. Also, deseasonalization had a very considerable positive effect on the performance.

Relevance Multi-step ahead forecasts are the aim for both the SF model and this paper. A large forecasting horizon, which could be a year ahead for the SF retention model, forms a difficult problem. The comparison of Taieb et al., 2011 shows that there are good approaches to model many steps ahead. Multiple-Output approaches are preferred over Single-Output approaches. Unfortunately, the

SF model has to output only one variable, the retention interval. Besides that, the NN5 data set consists of regularly sampled data.

Although there are differences between the NN5 data and the SF data, the preprocessing steps dealing with missing values, deseasonalization and dimension selection are useful. Also, variable selection is used. These steps form a interesting takeaway for our research.

3.1.2 A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction

Qin et al., 2017

Summary Nonlinear Autoregressive Exogenous (NARX) models forecast by making use of exogenous variables: variables whose value affects the model, but the variables themselves are not affected by the model. For example, when forecasting plant growth, the weather is an exogenous variable. Most NARX models fail to appropriately capture long-term temporal dependencies and select the relevant driving series (input features) to make predictions. A Dual-Stage Attention-Based Neural Network (DA-RNN) is proposed to address these issues. The first stage selects elementary stimulus features at each time step; the second stage uses categorical information to decode the stimulus (Figure 4). This is inspired by theories of human attention that behavioural results are best modelled by a two-stage attention mechanism (Hübner et al., 2010). The first stage selects the elementary stimulus features while the second stage uses categorical information to decode the stimulus.

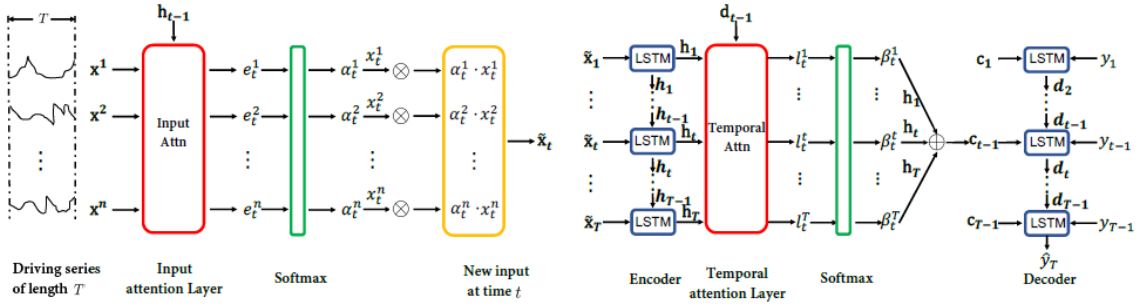

Figure 4: Graphical illustration of the DA-RNN. The left part forms the input attention mechanism. The part on the right is the temporal attention mechanism, consisting of an encoder and decoder (image from Qin et al., 2017).

First, the input selection mechanism selectively focusses on certain driving series rather than treating all the input driving series equally. The features then transfer to the temporal attention mechanism, where the encoder is formed by a RNN that encodes the input sequences into a feature representation

in machine translation. The input sequence consists of multiple driving (exogenous) series with a non-linear activation function, that could be an LSTM (Hochreiter & Schmidhuber, 1997) or a Gated Recurrent Unit (GRU) (Cho, van Merrienboer, Bahdanau, et al., 2014). Qin et al., 2017 choose the LSTM for its ability to model long-term dependencies.

Since the performance of the encoder-decoder network can deteriorate rapidly as the length of input sequence increases, a temporal attention mechanism is used to decode the data. The decoder adaptively selects relevant hidden states in the encoder across all time steps, using another LSTM network. The decoded information is the predicted output.

Training is carried out with minibatch Stochastic Gradient Descent (SGD) together with the Adam optimizer. The learning rate is reduced over time and the parameters are learned by standard back-propagation with Mean Squared Error (MSE) as the objective function. The DA-RNN is implemented in Tensorflow.

For learning two datasets are used: the SML 2010 dataset for indoor temperature forecasting (16 relevant driving series, 40 days of monitoring data, with samples every minute) and data from the NASDAQ 100 stock (81 corporations, 105 days of data with samples every minute). Various evaluation metrics are used, such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE).

The DA-RNN outperformed the benchmark models such as ARIMA, NARX RNN and other neural networks, achieving lower MAE, MAPE and RMSE.

Relevance The DA-RNN outperforms all the algorithms it is compared with and therefore could be very useful for the SF model. Especially the two-stage attention mechanism based on human behaviour is remarkable, since the SF model attempts to model human behaviour. Additionally, many input features are processed and discussed. An LSTM is used for its ability to capture long term dependencies without the problem of vanishing gradients and samples are collected at a regular interval. However, this differs from our data collection as the SF data is not generated at a fixed time step.

3.1.3 A Memory-Network Based Solution for Multivariate Time-Series Forecasting

Chang et al., 2018

Summary A deep learning based model named Memory Time-Series Network (MTNet) is proposed to effectively capture extremely long-term patterns and improve explainability. MTNet, as opposed to the DA-RNN mentioned in section 3.1.2, considers periods of time instead of particular timestamps in the past. Furthermore, it is extendable to multivariate settings, whereas DA-RNN is better suited for

univariate applications.

In MTNet model consists of a large memory component and three different embedding feature maps generated by three different encoders (Figure 5). The encoder consists of three different layers. A convolutional layer extracts short-term patterns between variables in the time dimension. Next the attention layer adaptively selects relative time across all time steps. Finally, the data is fed into a recurrent layer with the GRU and Rectified Linear Unit (ReLU) as hidden activation function.

The non-linearity of the convolutional and recurrent layer in the encoders causes the scale of the neural network output to be insensitive. To solve this problem, the final prediction is a combination of the non-linear representation from the encoders together with a linear autoregressive result.

In the training process, the MAE is adopted and all neural models are trained using the Adam optimizer.

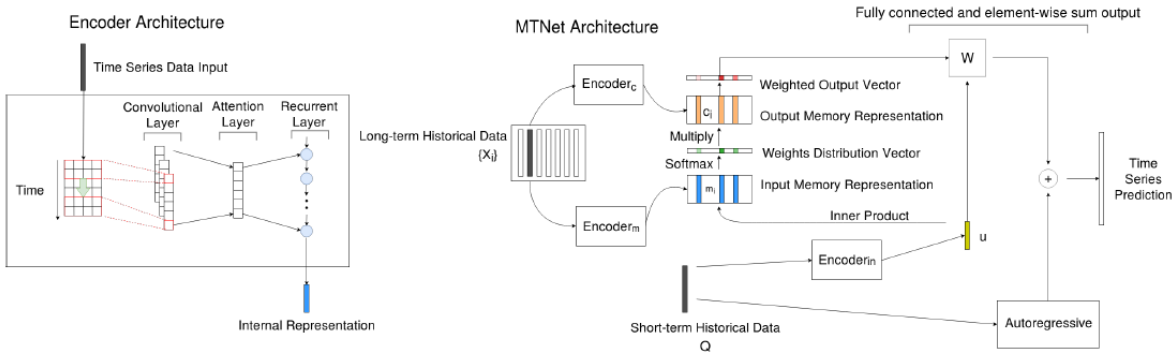

Figure 5: A graphical representation of the encoder architecture (left) and the MTNet architecture (right) (image from Chang et al., 2018).

Both univariate and multivariate experiments are conducted, with the data sets split into training (60%), validation (20%) and test (20%) in chronological order. RMSE is used as the metric, in addition to MAE for univariate tasks and both Root Relative Squared Error (RRSE) and Emperical Correlation Coefficient (CORR) for multivariate tasks.

The test results are compared with different other methods, including an Autoregressive model, RNN-GRU and the DA-RNN. Every technique is tested on multiple data sets, comparing the results per data set. MTNet outperforms these state-of-the-art methods in both univariate and multivariate time series prediction.

Relevance As the SF model has to deal with long-term patterns and interpretability of the variables would beneficial for the research, the MTNet could offer insight in the steps to take. It is able to deal with multivariate inputs and the paper explains well how the experiments are conducted.

3.1.4 Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches

Bouktif et al., 2018

Summary Electric load forecasting enables utility providers to model and forecast power loads in different time spans. Bouktif et al., 2018 specifically focussed on short-term (days to two weeks) and medium-term (weeks to months) forecasting, using the France Metropolitan's electricity energy consumption data as a case study. An LSTM is used since these networks are powerful for modelling sequential data and have the ability for end-to-end modelling, learning complex non-linear patterns and automatic feature extraction abilities.

The preprocessing of the data consists of checking of null values and outliers, scaling the data and splitting the data into train and test subsets. The measurements consist of the electric power consumption in France with thirty-minute sampling. Feature scaling is done by normalizing the data in the [0, 1] range. After this, the data set is split in 70/30 training/validation data.

To prove the effectiveness of the proposed methodology, other machine learning techniques (Ridge Regression, k-Nearest Neighbors (k-NN), Random Forests (RF), Gradient Boosting, Neural Network and Extra Trees Regressor (ETR)) were run on the same data with the complete set of features, providing a benchmark test. The MSE was used as the loss function and had to be minimized. From the performance metrics the ensemble approach ETR performed the best; the Neural Network performed the worst and took a longer time to train. Therefore ETR is used as benchmark to compare the proposed LSTM.

Bouktif et al., 2018 used wrapper techniques and embedded techniques of feature selection to validate the importance of the model inputs. This increased performance and helps against overfitting. To be more precise, recursive feature elimination and extra trees regressor showed the most important features to be the time lags. After this, hyperparameter tuning was used to improve the performance of the model. As the time lag features proved to be the most influential, a Genetic Algorithm (GA) was used to determine the optimal lag.

Generally, when a data set is larger (more samples), more hidden layers and neurons could be used without overfitting the model. In this case, the number of neurons in the input layer of the LSTM model matches the number of time lags in the input vector, hidden layers are fully connected and the output layer has a single neuron for prediction. MSE was used as the loss function between input and corresponding neurons in the output layer.

Five-fold cross validation was carried out to train five Extra Trees and five LSTM-RNN models. Multiple methods were applied as error measures, consisting of RMSE, Coefficient of Variation (CV) and MAE. Results show the GA combined with the LSTM has lower forecasting error than the other

machine learning algorithms.

Relevance This research shows that an LSTM model is performing well in a forecasting window that is similar to that of the SF model (weeks to months), with both models having many input features. To solve the high-dimensionality of the data, feature elimination is applied. Feature elimination is a solution for reducing the dimensionality of the SF data. However, the energy consumption data is sampled at an evenly spaced interval with no missing data and the paper focusses specifically on the time lag determined by the GA.

3.1.5 Recurrent Neural Networks for Multivariate Time Series with Missing Values

Che et al., 2018

Summary Missing values in multivariate time series are often correlated with the target labels: informative missingness. The missing values and patterns provide rich information about target labels in supervised learning tasks. In this paper, a novel deep learning based model GRU-D is developed to exploit two representations of informative missingness: masking (which inputs are observed or missing) and time interval (encapsulates the input observation patterns). The GRU, closely related to the LSTM, has strong prediction performance, with the ability to capture long-term temporal dependencies and variable-length observations.

In the GRU-D model, a decay mechanism is implemented. Each input variable has meaning and will return to some default value if the last observation is a long time ago. Furthermore, variables will fade away as time progresses. The decay mechanism will account for this, with each variable having its own decay rate. The patterns in the missing values could also be useful and informative and will be considered in training as well.

Different baseline methods for prediction the missing values are selected, with both non-RNN and RNN approaches. Two datasets are used to train: PhysioNet (hourly samples) and MIMIC-III (two-hourly samples). Both datasets are multivariate clinical time series with many input variables consisting of patients levels. All the RNN models are trained using the Adam optimization method and implemented with Keras and Theano. All the input variables are normalized to be of 0 mean and 1 standard deviation. The results are reported from five-fold cross validation in terms of area under the Receiver Operating Characteristics (ROC) curve (Area Under ROC Curve (AUC) score). The predictive tasks are classification problems.

The GRU-D model with trainable decays has similar running time and space complexity to original RNN models and performs better than both RNNs and non-RNNs.

Relevance Because of the similarities between the medical data and the SF data, the proposed GRU-D is an interesting approach. Although the SF data is still sampled less frequently and with larger time gaps, the approach proposed in this paper could offer insight in how to tackle the problem. Especially the decay mechanism, with a decay rate for individual variables, could be a solution for modelling the different features from SF.

3.1.6 Learning Adaptive Forecasting Models from Irregularly Sampled Multivariate Clinical Data

Liu and Hauskrecht, 2016

Summary An accurate predictive model of clinical multivariate time series could aid in understanding patient condition, the dynamics of the disease and clinical decision making. As many patient data are sparse and short span the model should be flexible and adaptive. The model learns from the population trend, is able to capture individual-specific short-term multivariate variability and adapts the prediction accordingly. The objective is to develop a model that can predict future values for a patient given a history of past observations. However, the time series of past observations for a patient may be short and patient-to-patient variability may be large. The latter means that it could be difficult to make individual predictions based on a population.

A two-stage adaptive forecasting model is proposed: AdaptLDS+reMTGP. The first stage learns a population model from clinical multivariate time series from many different patients using a Linear Dynamic System (LDS) (Kalman, 1960). In the second stage, the differences between patient and population are determined and the deviations are modelled using a Multi-Task Gaussian Process (MTGP) (Bonilla et al., 2008).

The LDS model is trained on clinical multivariate time series data from electronic health records of 500 post-surgical cardiac patients in the PCP database. The data consists of six individual time series. The test set is a random sample of 100 patients from the initial 500 selected patients.

As a benchmark, the two-stage model is compared with different common baselines. The performance is evaluated using MAPE. AdaptLDS is able to make predictions for one patient, splitting the prediction in different features. AdaptLDS+reMTGP performs better than all the other methods that are tested.

Relevance This research shows very strong similarities with the SF research. As there is scarce data for every participant, accurate predictions can be made by modelling the population. By measuring deviations from the population, individual predictions can be made. Even single skills can be modelled individually. Unfortunately, the model is not compared with deep learning methods.

3.1.7 Learning to Diagnose with LSTM Recurrent Neural Networks

Lipton et al., 2017

Summary Clinical medical data usually consists of multivariate time series of observations. It is, however, difficult to use this data effectively since the data is varying in length, is sampled irregularly and could be incomplete. An LSTM is used to model the sequences and capture long range dependencies with the aim of classification.

The anonymized data from a hospital in Los Angeles contains 10401 episodes, each containing 13 variables. These episodes vary in length from 12 hours to several months and consist of irregularly samples multivariate time series with missing values and variables. All time series are resampled to an hourly rate, filling gaps with forward- and back-filling. Missing variables are filled in by clinically normal values. Finally, all variables are rescaled to [0,1].

Models are trained on 80% of the data and tested on 10%, with the remaining 10% of the data forming the validation set. Each LSTM is trained for 100 epochs using SGD.

Area Under the ROC curve (AUC) is used as evaluation, with multiple types used. The LSTMs produce promising results and can successfully classify diagnosis. Lipton et al., 2017 show that an LSTM network outperforms other benchmark models.

Relevance Multivariate time series with irregularly sampled observations are shared between this research and the SF research. The solution, using multiple LSTMs, could be useful although their main aim is to classify medical conditions. Filling in missing values with normal values is harder in the SF model, since these values are not known. The paper is, however, very descriptive in how the techniques have been implemented.

3.1.8 Predicting the Risk of Heart Failure With EHR Sequential Data Modeling

Jin et al., 2018

Summary Time-based Electronic Health Records (EHR) containing patient data are analysed in an attempt to predict when a patient will be diagnosed. This is difficult since the data is sparse and non-standardized. Early diagnoses and treatments predicted by this model could help patients likely to have heart failure live longer and more actively. Each patient is treated as a dynamic system, measured by a set of time series such as lab tests, records and medical indicators. Two methods to process the diagnostic event sequence into the form of model input are used. The first method is one-hot encoding, the second is the word vector method. A word vector model provides a method for directly calculating the similarity between two words, providing an output vector much smaller than the length of the

dictionary of the language. After this, an LSTM is used to create a predictive model. Real-world EHRs are used to perform the experiment. The data set contains records of 5000 patients that have been diagnosed with heart failure and 15000 patients that have not been diagnosed with heart failure. Records include recording times, diagnostic events, and diagnosis time. Five-fold cross validation was used and performance was measured by comparing the results with Logistic Regression (LR), RF and AdaBoost. ROC, Precision-Recall (PR), AUC and F1 score are the metrics used to evaluate the proposed method. The accuracy of the LSTM disease prediction is higher than the other algorithms in all metrics. Besides that, word vector embedding processing performs better than one-hot encoding.

Relevance The most interesting aspect of this paper was the sparse and non-standardized data that was used for prediction. Jin et al., 2018 show that it is possible to predict events in time accurately using data that is not sampled frequently and could contain missing values. Since the SF data contains similar data in this regard, employing an LSTM network could be a good direction to take.

3.1.9 Interpolation-Prediction Networks for Irregularly Sampled Time Series

Shukla and Marlin, 2019

Summary Shukla and Marlin, 2019 propose a new model architecture for supervised learning with multivariate sparse and irregularly sampled data: Interpolation-Prediction Networks (Figure 6). The first part, the interpolation network, consists of several semi-parametric interpolation layers organized into an interpolation network. All the information contained in each input time series contributes to the interpolation of all other time series in the model. After this, the prediction network makes a prediction by using a deep learning model. Any standard supervised neural network architecture can be used, with a GRU used in this paper.

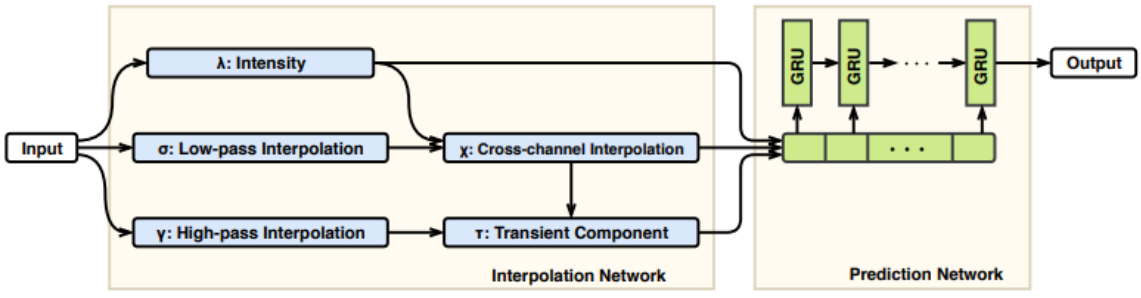

Figure 6: The architecture of the Interpolation-Prediction Network (image from Shukla and Marlin, 2019).

The proposed model is compared to a number of baseline approaches, including off-the-shelf classification and regression models and more recent approaches based on customized neural network models. For non-neural network baselines, LR, Support-Vector Machine (SVM), RF and AdaBoost are evaluated for the classification task. For the prediction tasks (length of stay in hospital) Linear Regression (LR), SVR, AdaBoost Regression and RF Regression are applied.

For neural network models, six different variants of a GRU are used as benchmark. The variations are created by differing the manner missing observations are handled and the introduction of decay. The model framework is tested on two real-world datasets: MIMIC-III (also used in section 3.1.5) and UWaveGesture. The results are reported of a five-fold cross validation experiment in terms of the average under the ROC curve (AUC score), Average Area Under the Precision-Recall Curve (AUPRC) and average cross-entropy loss for the classification task. For regression, Average Median Absolute Error (MedAE) and average fraction of EV are used as metrics.

For classification the proposed model outperforms all baseline models. In the regression task the proposed model performs similar to a couple of GRU models when measuring MedAE. However, the EV score is higher than all baseline models, which is desirable.

Relevance Multivariate sparse and irregularly sampled data form a major challenge to accurately model with. Since the clinical data in this research resembles the SF data, the Interpolation-Prediction Network is very relevant for the SF model. The proposed model by Shukla and Marlin, 2019 is able to deal with sparse and irregular data by interpolating over the entirety of the population first, followed by a prediction with a deep learning network. However, no specific attention is paid to data with a large feature space. If the Interpolation-Prediction Network would be used to model the SF retention, feature engineering has to be carried out first.

3.2 Knowledge Tracing

Similar to skill retention is the field of Knowledge Tracing (KT). KT is "an effort to model students' changing knowledge state during skill acquisition" (Corbett & Anderson, 1994). The goal of the research is to create a simple model of the student's learning process after solving simple exercises. This model allows the tutor to monitor the student's knowledge and in the end predict performance from the current knowledge state. Although KT shares similarities with the SF game, the data in KT is usually very simple and only statistical analysis is used to create a model. However, the underlying challenges when trying to model a person's skill or knowledge show strong similarities with the challenges of creating the SF retention model. We will look at two papers discussing KT to get a basic understanding of the possibilities the models offer. The first three papers attempt to model more complex KT problems. A RNN is employed to allow for models with more features

and longer prediction windows, reporting substantial improvements in prediction performance. This variant on KT is called Deep Knowledge Tracing (DKT). Lastly, we will discuss a specific application of knowledge tracing: Second Language Acquisition Modelling (SLAM). This is specifically focussed on online second language learning. The online language-learning platform Duolingo held a contest to find the best technique for predicting learners' performance. The background and results are explained in the last two papers.

Contents

3.2.1 Knowledge Tracing and Prediction of Future Trainee Performance 24
3.2.2 Knowledge Tracing in Sequential Learning of Inflected Vocabulary 24
3.2.3 Modeling Skill Combination Patterns for Deeper Knowledge Tracing 25
3.2.4 Going Deeper with Deep Knowledge Tracing 26
3.2.5 Incorporating Rich Features into Deep Knowledge Tracing 26
3.2.6 Second Language Acquisition Modeling 27
3.2.7 Second Language Acquisition Modeling: An Ensemble Approach 28

3.2.1 Knowledge Tracing and Prediction of Future Trainee Performance

Jastrzembski et al., 2006

Summary In previous research on knowledge tracing, a representation of memory decay in periods of non-practice is not included. In this paper, a new knowledge tracing equation is proposed, capable of predicting future trainee performance and the prescription of frequency and timing of optimal learning. The General Performance Equation (Anderson & Schunn, 2000) is the basis for the predictive and prescriptive mathematical model. Since learning and forgetting do not linearly improve or degrade over extended periods of time, a new equation is proposed capturing recency, frequency and spacing effects too.

Relevance The extended mathematical model can determine when a warfighter has become proficient in a skill, but could also streamline training to optimize learning. The model is able to predict future performance months in the future. Although this is very relevant, it is a simple model which does not take into account many input variables.

3.2.2 Knowledge Tracing in Sequential Learning of Inflected Vocabulary

Renduchintala et al., 2017

Summary A feature-rich knowledge tracing method for capturing the acquisition and retention of knowledge is proposed. This is applied to students learning and acquisition of a foreign language. The students translate a short inflected phrase from a new language to English. Students go through a series of interactive flash cards during a training session, each showing a different kind of exercise. The three types are: example (EX), multiple-choice (MC) and typing (TP). Many features are tracked (around 4600), forming the knowledge state of the student. The approach is called Parametric Knowledge Tracing (PKT) because student's knowledge is taken as a vector of prediction parameters (feature weights). The approach is very similar to DKT. Learning is carried out by SGD. Different update schemes are tested: redistribution (RG), negative gradient (NG) and feature vector (FG). For evaluation, the log-probability under the model of each actual response (cross-entropy) and fraction of correct responses (accuracy) were measured. The best model consists of a combination of RG and NG. The model is compared with an LSTM and performs slightly better.

Relevance As with the SF data, many features are tracked. The model performing better than an LSTM is interesting. However, the proposed model does not attempt to model multi-step ahead data nor irregularly sampled data.

3.2.3 Modeling Skill Combination Patterns for Deeper Knowledge Tracing

Huang et al., 2016

Summary This paper focusses specifically on student knowledge in complex learning activities where multiple skills are required at the same time. A known limitation of knowledge tracing is the assumption of skill independence in problems that involve multiple skills. Huang et al., 2016 assume that a set of skills is more than the sum of individual skills and skills are related. To model this, a Bayesian Network (BN) is constructed to model skill combinations. The first layer consists of basic individual skills, with the intermediate layers of skill combinations for deeper understanding. The last layer represents the mastery of the individual skills. Using two datasets from SQL and Java programming learning, different models are compared measuring the knowledge inference quality: mastery accuracy and mastery effort. Performance prediction accuracy is measured by using RMSE and AUC. The results demonstrate that incorporating skill combinations can significantly increase mastery assertion accuracy compared to traditional knowledge tracing models.

Relevance The main focus here lies on accurately modelling mastery of skills rather than forecasting future performance. Huang et al., 2016 do not focus on individual independent skills, yet show that skills are related to each other. This result could be useful for analysing the SF data.

3.2.4 Going Deeper with Deep Knowledge Tracing

Xiong et al., 2016

Summary Xiong et al., 2016 examine DKT and compare it with two other KT models: Performance Factor Analysis (PFA) and Bayesian Knowledge Tracing (BKT). PFA is a variation on a educational data mining model called Learning Factor Analysis (LFA), where learning is modelled using a mathematical equation with different variables and parameters. PFA tries to solve limitations of LFA by making it more adaptive (Pavlik et al., 2009). Basically, PFA is a form of a standard LR model with the student performance as the dependent variable. BKT models student knowledge in a Hidden Markov Model (HMM), assuming that student knowledge is represented as a set of binary variables (one per skill). The outcome of the model is also binary: the answer to each exercise is either right or wrong (Yudelson et al., 2013).

As typical RNNs suffer from vanishing and exploding gradients (see section 3.1), an LSTM model is introduced to deal with this problem. One-hot encoding is used to convert student performance into fixed length vectors. The model has 200 fully-connected hidden nodes. Mini-batch SGD was used to minimize the loss function, with a batch size of 100 (randomly selected students from all data). Dropout (Srivastava et al., 2014) was applied to the hidden layer to avoid over-fitting. Three different datasets are used to compare the algorithms. All datasets contain over 500,000 rows of student's responses and are gathered from ASSISTments (2009-2010 and 2014-2015) skill builder and the KDD Cup (2010). Students are learning different skills, with the data set containing 100+ skills. DKT did not achieve overwhelmingly better compared to PFA on the ASSISTments data set, but did perform much better on the KDD data set. Xiong et al., 2016 believe this is due to the PFA model being undermined by inaccurate item difficulty estimation.

Relevance This paper contains very specific information about the techniques applied. Although the result is not convincing on every data set, Xiong et al., 2016 show that a deep learning technique yields better results than traditional knowledge tracing models.

3.2.5 Incorporating Rich Features into Deep Knowledge Tracing

Zhang et al., 2017

Summary DKT shows promising results, as seen in section 3.2.4 where DKT outperforms BKT and PFA on most cases. However, DKT only considers the knowledge components and the correctness of the inputs. Other features that could be collected on a computer-based learning platform are

neglected. Zhang et al., 2017 seek to improve DKT by incorporating more features at the problem-level. More features are incorporated into the model, including exercise tag, correctness, first response time, attempt count and first action. Features are represented as sparse vector by a one-hot encoding. Features can be combined into cross features to improve model accuracy, but this leads to a rapid increase of the dimensionality of the input vector. An Autoencoder (Hinton & Salakhutdinov, 2006) is able to reduce dimensionality without sacrificing performance. After training the Autoencoder to reduce the features to half the input size, an LSTM with 200 hidden nodes is used to train. The loss function used is binary cross entropy and a dropout probability of 0.4 is applied. Two data sets, ASSISTments (2009-2010) and OLI Statics F2011, are used to train the network. Both are datasets from online courses containing many students' exercises. Five-fold student level cross validation is used and the result is evaluated by AUC and R2. The model incorporating the most features shows the best results, with the Autoencoder improving the model even further. The ASSISTments (2009-2010) data set is also used in the paper in section 3.2.4, with the current paper (Zhang et al., 2017) showing better results. This means that the Autoencoder with LSTM outperforms traditional models as well.

Relevance Although KT and DKT models do not account for time gaps in the model, they could offer us insight in how to set up knowledge prediction models. Zhang et al., 2017 use an Autoencoder to reduce dimensionality while maintaining performance. An Autoencoder can be used to solve the high-dimensionality of the SF data.

3.2.6 Second Language Acquisition Modeling

Settles et al., 2018

Summary Second Language Acquisition Modelling is the task of predicting errors that second language learners are likely to make at arbitrary points in the future. A competition is created to predict these errors with data from Duolingo, the online language-learning platform. The data set contains many features, with for example the user, the number of days since the learner has started learning and the current exercise format. A total of 15 teams participated in the competition, with SanaLabs performing the best, making use of a RNN combined with a Gradient Boosted Decision Tree (GBDT). RNNs work well for sequence data, while GBDTs are often the best-performing non-neural model for shared tasks using tabular data. The winning paper is discussed below, in section 3.2.7.

Relevance The task is to predict errors that learners will make in the future, based on current and past data samples. This resembles the SF skill decay model, even though the SF model is more complex and should be able to look ahead an arbitrary amount of time steps into the future instead

of predicting the next exercise.

3.2.7 Second Language Acquisition Modeling: An Ensemble Approach

Osika et al., 2018

Summary As participant in the SLAM challenge (Settles et al., 2018), the aim here is to predict students' knowledge gaps. To predict word-level mistakes an ensemble model combining a GBDT) and a RNN is built. These techniques are chosen since RNNs achieve good results for sequential prediction tasks (Piech et al., 2015) and GBDTs achieve state of the art results on various benchmarks for tabular data (Li, 2012). The RNN is implemented as generalization of the LSTM architecture. The predictions generated by the RNN and the GBDT are combined through a weighted average. By varying the proportions and measuring the AUC score, the optimal ratio is decided. All datasets were pre-partitioned into training, development and test subsets (80/10/10). The performance for this binary classification task is measured by area under the ROC curve (AUC) and F1-score. The ensemble approach shows better AUC and F1 scores than the RNN and GBDT separated. This approach outperforms all the other algorithms used in the SLAM challenge.

Relevance Predicting students' knowledge gaps is similar to predicting a participants skill in the future in SF. As with the knowledge tracing models, the time factor is not particularly important in predicting the next exercise. However, the ensemble approach shows promising results on skill prediction.

4 Discussion

In this section we discuss how the insights obtained from the literature may be used to develop a predictive model for performance on the SF game. Guided by the challenges posed in section 2.6, we can summarize the solutions the papers have for the different challenges (section 4.1).

Although many of the papers reviewed in this thesis solve problems in different fields, there are a number of techniques and measures that are more frequently used than others. For example, the approaches that are taken for training, validation and performance measure are listed in section 4.1.1. These techniques are a logical choice for application in the retention model.

4.1 Challenges

4.1.1 Temporal data/forecasting

All papers discussed deal with temporal data and attempt to forecast based on past observations. There are many different techniques employed, ranging from mathematical models to deep learning networks. As knowledge tracing is mainly focussed on predicting the next exercise instead of a specific moment in time in the future, the forecasting aspect is less relevant. Most TSF papers mentioned specifically focus on (multi-step ahead) prediction, as accuracy tends to decrease when the prediction window increases.

Therefore, all papers discussed in section 3.1 offer relevant techniques to forecast data. The most commonly used technique is a RNN, often an LSTM (see appendix A.1) or a GRU. Both are capable of dealing with long term dependencies that would cause vanishing or exploding gradients in a traditional RNN. Since the temporal aspect of the SF data is similar to the TSF data, the techniques proposed in TSF are most likely to work for the retention model.

Between papers there are similarities found in the implemented techniques to train, validate and measure the performance of the RNN. For training, most test sets are divided in a train/test/validation set, usually in the range of 80/10/10 percent. When training, the Adam optimization algorithm (see appendix A.2) is often mentioned as the preferred solution. Mean Squared Error (MSE) is used multiple times as loss function, while five-fold cross validation is the most used validation method. For performance measure, a number of techniques is specifically used for classification, others are applied to regression tasks. However, ROC (AUC) is the most commonly used. The proposed solution in 4.1.3 is tested using MedAE and EV. These measures are suitable for regression.

Recommendation An Long Short-Term Memory (LSTM) or a Gated Recurrent Unit (GRU) shows the best performance in modelling long-term temporal dependencies. For training, use an Adam optimizer.

4.1.2 High-dimensionality/feature engineering

When using many input features to train a model there will be drawbacks such as overfitting. This phenomenon is sometimes referred to as the curse of dimensionality. An unfeasible amount of samples is required to account for all the combinations of the many features. Besides that, it is also computationally inefficient and there is an increased risk of overfitting a neural network. To scale down the number of features some form of feature engineering can be applied. Solutions mentioned are:

- Encoder-decoder network with temporal attention mechanism (section 3.1.2);
- Recursive Feature Elimination (section 3.1.4);
- Extra Trees Regressor (ETR) (section 3.1.4);
- Autoencoder (section 3.2.5).

The Autoencoder is able to learn data codings in an unsupervised manner. The dimensionality of the input space can be reduced while maintaining performance. An Autoencoder showed promising results with many input features in section 3.2.4. Technical details can be found in appendix A.3.

Recommendation An Autoencoder is recommended, since this is a proven technique for reducing the input dimensionality while maintaining performance.

4.1.3 Sparse and irregularly sampled data

The presence of gaps in the data, or irregular intervals between samples, poses a challenge. Most TSF algorithms deal with regularly spaced data. Medical data however is difficult to sample regularly. The papers in sections 3.1.5, 3.1.6, 3.1.7, 3.1.8 and 3.1.9 offer interesting solutions for the problem. Especially the AdaptLDS+reMTGP (Liu & Hauskrecht, 2016) in 3.1.9 shares many similarities with the SF research. In the SF data, there is not enough data available per participant to make individual predictions. The AdaptLDS algorithm that trains on the whole population could be a possible solution. Then the reMTGP can be used to make personal predictions. Other papers show that a RNN can effectively be used to model long-range dependencies with irregularly spaced data. As mentioned in section 4.1.1, LSTMs or GRUs show promising results.

Recommendation The most relevant paper uses interpolation layers on the population, then a RNN to make a personal prediction for a participant. This is a good direction to take.

5 Conclusion

As far as we know, there are no papers written on skill retention forecasting with sparse, irregularly sampled data. However, we can combine solutions from other papers into a new model. High-dimensionality can be solved using feature engineering (section 4.1.2). Based on the papers reviewed, the use of an Autoencoder to reduce dimensionality would be a good direction to take. However, the most important challenge proved to be the irregularly sampled data. Medical forecasting uses similar data and offers some very interesting approaches (section 4.1.3). The (a) interpolation on the population combined with (b) personalized prediction using regression in AdaptLDS+reMTGP is a promising solution. After interpolation, an Long Short-Term Memory (LSTM) or a Gated Recurrent Unit (GRU) is the best solution for predicting the optimal retention interval for SF.

References

Adams, J. A. (1987). Historical Review and Appraisal of Research on the Learning, Retention, and Transfer of Human Motor Skills. *MOTOR SKILLS*, 34.

Anderson, J. R., & Schunn, C. (2000). Implications of the ACT-R learning theory: No magic bullets. *Advances in instructional psychology, Educational design and cognitive science*, 1–33.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157–166.

Bonilla, E. V., Chai, K. M., & Williams, C. (2008). Multi-task Gaussian process prediction, In *Advances in neural information processing systems*.

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. (2018). Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches †. *Energies*, *11*(7), 1636. https://doi.org/10.3390/en11071636

Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, *59*(4-5), 291–294.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control*. John Wiley & Sons.

Cao, L., & Tay, F. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, *14*(6), 1506–1518. https://doi.org/10.1109/TNN.2003.820556

Carretta, T. R., Rodgers, M. N., & Hansen, I. (1993). *The identification of ability requirements and selection instruments for fighter pilot training* (tech. rep.). ARMSTRONG LAB BROOKS AFB TX.

Chang, Y.-Y., Sun, F.-Y., Wu, Y.-H., & Lin, S.-D. (2018). A Memory-Network Based Solution for Multivariate Time-Series Forecasting [arXiv: 1809.02105]. *arXiv:1809.02105 [cs, stat]*. Retrieved March 18, 2020, from http://arxiv.org/abs/1809.02105

Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, *8*(1), 6085. https://doi.org/10.1038/s41598-018-24271-9

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches [arXiv: 1409.1259]. *arXiv:1409.1259 [cs, stat]*. Retrieved March 25, 2020, from http://arxiv.org/abs/1409.1259

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine

Translation [arXiv: 1406.1078]. *arXiv:1406.1078 [cs, stat]*. Retrieved March 25, 2020, from http://arxiv.org/abs/1406.1078

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*(4), 253–278.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Gopher, D., Well, M., & Bareket, T. (1994). Transfer of Skill from a Computer Game Trainer to Flight. *Human Factors*, *36*(3), 387–405. https://doi.org/10.1177/001872089403600301

Haynes, M. (2008). Pilot reaches 1,000 combat hours flown. https://www.af.mil/News/Article-Display/Article/124255/pilot-reaches-1000-combat-hours-flown/

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

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

Huang, Y., Guerra-Hollstein, J. D., & Brusilovsky, P. (2016). Modeling Skill Combination Patterns for Deeper Knowledge Tracing, 9.

Hübner, R., Steinhauser, M., & Lehle, C. (2010). A dual-stage two-phase model of selective attention. *Psychological Review*, *117*(3), 759–784. https://doi.org/10.1037/a0019471

Jain, A., & Kumar, A. M. (2007). Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, *7*(2), 585–592. https://doi.org/10.1016/j.asoc.2006.03.002

Jastrzembski, T. S., Gluck, K. A., & Gunzelmann, G. (2006). Knowledge Tracing and Prediction of Future Trainee Performance, 15.

Jin, B., Che, C., Liu, Z., Zhang, S., Yin, X., & Wei, X. (2018). Predicting the Risk of Heart Failure With EHR Sequential Data Modeling. *IEEE Access*, *6*, 9256–9261. https://doi.org/10.1109/ACCESS.2017.2789324

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

Kim, K.-j. (2006). Artificial neural networks with evolutionary instance selection for financial forecasting. *Expert Systems with Applications*, *30*(3), 519–526.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, P. (2012). Robust logitboost and adaptive base class (abc) logitboost. *arXiv preprint arXiv:1203.3491*.

Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. (2017). Learning to Diagnose with LSTM Recurrent Neural Networks [arXiv: 1511.03677]. *arXiv:1511.03677 [cs]*. Retrieved March 5, 2020, from http://arxiv.org/abs/1511.03677

Liu, Z., & Hauskrecht, M. (2016). Learning Adaptive Forecasting Models from Irregularly Sampled Multivariate Clinical Data, 7.

Mané, A., & Donchin, E. (1989). The space fortress game. *Acta Psychologica*, *71*(1-3), 17–22. https://doi.org/10.1016/0001-6918(89)90003-6

Maqsood, I., Khan, M., & Abraham, A. (2004). An ensemble of neural networks for weather forecasting. *Neural Computing and Applications*, *13*(2). https://doi.org/10.1007/s00521-004-0413-4

Osika, A., Nilsson, S., Sydorchuk, A., Sahin, F., & Huss, A. (2018). Second Language Acquisition Modeling: An Ensemble Approach, In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, Louisiana, Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-0525

Pai, P.-F., & Lin, C.-S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, *33*(6), 497–505. https://doi.org/10.1016/j.omega.2004.07.024

Pavlik, P. I., Cen, H., & Koedinger, K. R. (2009). Performance Factors Analysis – A New Alternative to Knowledge Tracing, 8.

Pi, H., & Peterson, C. (1994). Finding the embedding dimension and variable dependencies in time series. *Neural Computation*, *6*(3), 509–520.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing, In *Advances in neural information processing systems*.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction [arXiv: 1704.02971]. *arXiv:1704.02971 [cs, stat]*. Retrieved March 18, 2020, from http://arxiv.org/abs/1704.02971

Reed, H. (1918). Associative aids: I. Their relation to learning, retention, and other associations. *Psychological Review*, *25*(2), 128.

Renduchintala, A., Koehn, P., & Eisner, J. (2017). Knowledge Tracing in Sequential Learning of Inflected Vocabulary, In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, Vancouver, Canada, Association for Computational Linguistics. https://doi.org/10.18653/v1/K17-1025

Settles, B., Brust, C., Gustafson, E., Hagiwara, M., & Madnani, N. (2018). Second Language Acquisition Modeling, In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, New Orleans, Louisiana, Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-0506

Shukla, S. N., & Marlin, B. M. (2019). INTERPOLATION-PREDICTION NETWORKS FOR IRREGULARLY SAMPLED TIME SERIES, 14.

Smith, J. F. (1976). Continuation Versus Recurrent Pilot Training.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

Taieb, S. B., Bontempi, G., Atiya, A., & Sorjamaa, A. (2011). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition [arXiv: 1108.3259]. *arXiv:1108.3259 [cs, stat]*. Retrieved March 18, 2020, from http://arxiv.org/abs/1108.3259

van der Pal, J., & Toubman, A. (2020). An Adaptive Instructional System for the Retention of Complex Skills.

Whittle, P. (1951). *Hypothesis testing in time series analysis. Peter Whittle.* Uppsala, Almqvist och Wiksells boktryck.

Xiong, X., Zhao, S., Inwegen, V., & Beck, J. E. (2016). Going Deeper with Deep Knowledge Tracing, 6.

Yudelson, M. V., Koedinger, K. R., & Gordon, G. J. (2013). Individualized Bayesian Knowledge Tracing Models (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik, Eds.). In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Artificial Intelligence in Education*. Berlin, Heidelberg, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-39112-5_18

Zhang, L., Xiong, X., Zhao, S., Botelho, A., & Heffernan, N. T. (2017). Incorporating Rich Features into Deep Knowledge Tracing, In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale - L@S '17*, Cambridge, Massachusetts, USA, ACM Press. https://doi.org/10.1145/3051457.3053976

A Techniques

A.1 Long Short-Term Memory

Normal RNNs are very powerful, capable of instantiating almost arbitrary dynamics. However, the backpropagated error signal depends exponentially on the magnitude of the weights, risking either vanishing or blown up errors. Hence standard RNNs fail to learn when time certain time gaps are in place between relevant input events and target signals. To solve this problem, the Long Short-Term Memory (LSTM) (Gers et al., 2000) has a memory block as hidden layer in its network, combined with a pair of adaptive, multiplicative gating units which gate input and output to all cells in the block (Figure A.1). Most importantly, LSTMs have forget gates, which learn to reset the memory blocks once their contents are out of date. Where continuous input streams usually require occasional resets of the network, LSTM networks learn to reset at appropriate times. This results in a RNN that can not only model but also predict future values in temporal data.

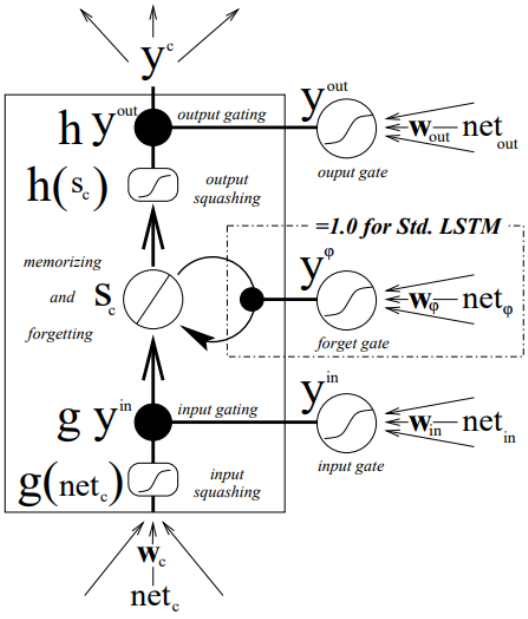

Figure A.1: The LSTM cell has a linear unit with a recurrent self-connection (image from Gers et al., 2000).

The Gated Recurrent Unit (GRU) is very similar to the LSTM. Whereas the LSTM has three gates (input, output, reset), the GRU has two gates (reset, update) (Cho, van Merrienboer, Gulcehre, et al., 2014).

A.2 Adam optimizer

Adam (short for Adaptive Moment Estimation) is a form of SGD, an iterative method for optimizing an objective function. More specifically, Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments (Kingma & Ba, 2014). Empirical results demonstrate that the algorithm compares favorably to other stochastic optimization methods.

A.3 Autoencoder

An Autoencoder is a multi-layer neural network with a small central layer that can convert high-dimensional data to low-dimensional representative encodings. These encodings can subsequently be used to reconstruct the high-dimensional input vectors. In this way the dimensionality is reduced without the loss of important information (Hinton & Salakhutdinov, 2006). If linear activations are used, the optimal solution to an Autoencoder is strongly related to Principal Component Analysis (PCA) (Bourlard & Kamp, 1988). An Autoencoder shows good results in reducing dimensionality. However, compared to a PCA, an Autoencoder loses explainabilty.

Acronyms

ANN Artificial Neural Network. 13

ARIMA Autoregressive Integrated Moving Average. 13, 16

ARMA Autoregressive Moving Average. 13

AUC Area Under ROC Curve. 19, 21–23, 25, 27–29

AUPRC Average Area Under the Precision-Recall Curve. 23

BKT Bayesian Knowledge Tracing. 26

BN Bayesian Network. 25

CORR Emperical Correlation Coefficient. 17

CV Coefficient of Variation. 18

DA-RNN Dual-Stage Attention-Based Neural Network. 15–17

DIRMO DIRect and miMO. 14

DKT Deep Knowledge Tracing. 12, 24–27

EHR Electronic Health Record. 21, 22

ETR Extra Trees Regressor. 18, 30

EV Explained Variation. 23, 29

GA Genetic Algorithm. 18, 19

GBDT Gradient Boosted Decision Tree. 27, 28

GRU Gated Recurrent Unit. 16, 17, 19, 20, 22, 23, 29–31, 36

HMM Hidden Markov Model. 26

k-NN k-Nearest Neighbors. 18

KT Knowledge Tracing. 1, 5, 12, 23, 24, 27

LDS Linear Dynamic System. 20, 30, 31

LFA Learning Factor Analysis. 26

LR Linear Regression. 23

LR Logistic Regression. 22, 23, 26

LSTM Long Short-Term Memory. 1, 13, 16, 18, 19, 21, 22, 25–31, 36

MAE Mean Absolute Error. 16–18

MAPE Mean Absolute Percentage Error. 16, 20

MedAE Average Median Absolute Error. 23, 29

MIMO Multi-Input Multi-Output. 14

MSE Mean Squared Error. 16, 18, 29

MTGP Multi-Task Gaussian Process. 20, 30, 31

MTNet Memory Time-Series Network. 16, 17

NARX Nonlinear Autoregressive Exogenous. 15, 16

NLR Royal Netherlands Aerospace Centre. 1, 4–6

OiT Education & individual Training. 6, 8, 12

PCA Principal Component Analysis. 37

PFA Performance Factor Analysis. 26

PKT Parametric Knowledge Tracing. 25

PR Precision-Recall. 22

ReLU Rectified Linear Unit. 17

RF Random Forests. 18, 22, 23

RMSE Root Mean Squared Error. 16–18, 25

RNN Recurrent Neural Network. 13, 15–19, 23, 26–30, 36

ROC Receiver Operating Characteristics. 19, 21–23, 28, 29

RRSE Root Relative Squared Error. 17

SF Space Fortress. 1, 5–8, 11, 12, 14–17, 19, 20, 22, 23, 25, 27, 29, 31

SF-AIS Space Fortress Adaptive Instructional System. 8–10

SGD Stochastic Gradient Descent. 16, 21, 25, 26, 37

SLAM Second Language Acquisition Modelling. 12, 24, 27, 28

SMAPE Symmetric Mean Absolute Percentage of Error. 14

SVM Support-Vector Machine. 23

SVR Support Vector Regression. 13, 23

TNO Netherlands Organisation for Applied Scientific Research. 4, 6

TSF Time Series Forecasting. 1, 5, 12, 13, 29, 30