Heart Rate Forecasting for Adaptive Virtual Reality Exposure Therapy for Public Speaking Anxiety

Comparing univariate and multivariate time series forecasting models with heart rate data

Maja B. Nilsen

Thesis submitted for the degree of
Master in Robotics and Intelligent Systems
60 credits

Department of Informatics
The Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Spring 2022

Heart Rate Forecasting for Adaptive Virtual Reality Exposure Therapy for Public Speaking Anxiety

Comparing univariate and multivariate time series forecasting models with heart rate data

Maja B. Nilsen

Heart Rate Forecasting for Adaptive Virtual Reality Exposure Therapy for Public Speaking Anxiety

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

An increase in the burden of disease associated with mental health has been observed in the last few years, and has been exacerbated by the recent global pandemic. This includes public speaking anxiety (PSA), as people have not been able to interact in the same way as before, in which large crowds could be gathered with no restrictions. PSA is widely treated by using exposure therapy to gradually ease the stress in these scenarios. Gathering groups of people to enable such exposure scenario however can be logistically challenging, especially in the context of the current health crisis. Virtual Reality (VR) is a newer alternative which has been applied as part of the treatment for a wide variety of mental health problems due to its ability to viscerally and safely simulate a large array of exposure scenarios. In particular, VR environments have been shown to provide a useful environment from which to perform exposure therapy for people living with PSA. However for this type of treatment to work optimally the exposure session needs to genuinely challenge the subject during the session. In other words, the patient will experience psychological/physiological arousal due to the presence of stressors. However, over time the person might get accustomed to a particular exposure scenario, reducing their level of arousal in the current situation without actually addressing the underlying issue. Therefore the capability of adapting these systems in real-time, for example by changing the crowd's behaviour, would be conductive to more efficient sessions. Acceleration in heart rate can be used as a proxy to detect activation of the sympathetic nervous system and therefore the level of arousal of the patient. In other words, if one could forecast the subject's heart rate, it would be possible to detect in advance when the person would become accustomed to the session and preemptively modify the session so as to keep the person challenged throughout. Therefore, this thesis main goal is to perform heart rate forecasting within the context of virtual exposure therapy by exploring and comparing different state of the art time series forecasting machine learning algorithms.

Heart rate data is normally represented as a type of time series data. Time series data in the machine learning field is normally used for time serie forecasting. Heart rate is influenced by several factors, like stress, physical activity and health conditions. Therefore it is sufficient to define the future heart rate as a combination of different factors, although physical activities plays the biggest role to determine the heart rate. Therefore it can be more challenging to predicting heart rate in a mental health case.

In this thesis three time serie forecasting models are tested against each other, which are the Long Short-Term Memory (LSTM) Networks, the Probabilistic Forecasting with Autoregressive Recurrent Networks (Deep AR) and the Temporal Fusion Transformer (TFT). The INTROMAT reasearch project has done an experiment to check the effectiveness of PSA exposure therapy through VR environments. The dataset from this project includes heart rate and head rotation data from the VR headset. To predict the heart rate of this dataset both univariate and multivariate experiments were tested, meaning that they predict with one or more variables of time series. In this case just the heart rate or with rotation data in addition to the heart rate. However the INTROMAT dataset is relatively small, so there needs some exploration with another dataset to get clearer results.

The conclusion of the experiments was that TFT and Deep AR showed the best results in general when it came to predicting the trend of the heart rate, when the heart rate increases and when it decreases in both cases. However none of the models showed a result that beat the Baseline when it came to predicting the actual heart rate. The Baseline model predicts the forecast by using the last known value of the target variable, so it will not catch any development in the time serie as time passes. In general the heart rate will normally not change as much in a small period of time and when a model predicts wrong for whether the heart rate increases or decreases the metrics for these models will be more affected by this than the Baseline. So that could be the reason why the Baseline resulted with better metric scores. Although in this thesis the focus is more about how the heart rate changes over time, so even if the Baseline has better metric scores it is not a sufficient method to predict heart rate.

The metrics of the univariate and multivariate experiments showed little difference of the performance of the models. This could be that the datasets still does not provide enough information to properly predict the heart rate with these deep learning models and additional biosignal modalities would be needed to effectively measure the level of arousal of patients undergoing virtual reality-based exposure therapy.

Contents

List of Figures

Preface

Foremost, I would like to thank my main supervisor for this master project, Ulysse Teller Masao Côté-Allard, for outstanding help with software, theory and programming, as well as guidance for writing this thesis. He has given me motivation to get through this project and has had a lot of patience with me during this time.

I would also like to thank my other supervisor, Jim Tørresen, which has helped me with encouragement and pushed me to work forward. He helped me with my confidence to make this possible and finally be able to finish my thesis. He set up monthly check ups in a group of other students also finishing their master's degree, which was great to get an insight to the plans of the other students for inspiration.

Thank you to the Robotics and Intelligent Systems (ROBIN) research group for letting me have the opportunity to finish this degree and has provided me with a laptop with great computing power so that I could run my machine learning algorithms whenever and wherever I wanted seemlessly.

I am also really thankful for all my kind and encouraging fellow students, who has motivated me throughout the entire degree of five years. They have helped me understand concepts and tasks even in the most complex subjects.

I want to give a special thanks to my family and friends who has been there for me this whole time by showing endless support and always believing in me. I am so grateful for every single one of you.

Chapter 1

Introduction

Mental health is an important part of peoples general well-being, and mental disorders can greatly impact an individual's quality of life. One of the most common mental issues relates to Social Anxiety Disorder (SAD) [14]. A subgroup of anxiety disorders is called Public Speaking Anxiety (PSA), which translate to a fear of speaking in public audiences. This can include speaking in a large friend group, having a presentation in front of a class, or any way an individual will have to perform anything in front of a crowd. This will often overpower the individuals capability of getting their point across, which often leads to their minds going blank, stuttering and shaking [9]. The feeling that is being experienced is often high levels of stress and uncomfortableness [9]. Some also get really anxious, and it affects their mental health in a negative way. PSA can often be mistaken with being shy or general stress. It is normal to be stressed whenever people want to impress the audience they are talking to, while stress often can be manageable in the moment. Shy people are most likely not debilitated by the discomfort in these public speaking situations, and can usually perform without as much pain as the people with PSA.

When properly treated, the symptoms associated with PSA can generally be alleviated [9]. This can be done by exposure of the activity the patients get their anxiety from. With this kind of exposure, it will allow the patients to gradually get exposed to their anxiety or fear with the goal of minimizing the symptoms and reactions to the situation [19]. However studies including this kind of treatment will require a full audience for each session, which is logistically hard to obtain.

Virtual Reality (VR) is a relatively new field in the mental health domain, and is widely used to various treat psychological issues. By using this technology instead of hiring people to experiments is more attractive as it does not depend on other people and is therefore more practical on a day-to-day basis. From previous studies, it is found that virtual reality is effective for the treatment of mental health issues [3, 9, 19, 20]. The goal is to build on this idea.

By adapting the virtual reality environment to the patients and their levels of anxiety can increase the effectiveness of the treatment [13]. Heart rate can be a good indicator of a person's stress levels. By predicting future heart rates in real-time, it would be easier to adapt the VR system to give the subjects enough challenge in their sessions for optimal treatment. Any additional data that can influence the heart rate is also useful to measure, for example physical activity and movement plays a big role of determining the heart rate [5]. Machine learning methods such as Recurrent Neural Networks (RNNs), Amazon's Probabilistic Forecasting with Autoregressive Recurrent Networks (Deep AR) [22] and the Temporal Fusion Transformer (TFT) [12] are some state of the art models that are frequently used for predicting time series, both for univariate and multivariate cases. This means that they can predict a target variable by one or more variables. So the goal of this thesis is to compare state of the art models to forecast heart rate to enable adaptive VR exposure therapy sessions for PSA.

1.1 INTROMAT

This master's thesis is part of the INTROMAT (INtroducing personalized Treatment Of Mental health problems using Adaptive Technology) research project, financed by the Research Council of Norway, 2016-2021, as one of three IKTPLUSS lighthouse projects: intromat.no. INTROMAT has a goal to improve public mental health with innovative technologies. The project owner is Haukeland University Hospital.

This thesis will focus on early intervention and treatment for social anxiety disorder in adolescents, which is a project within INTROMAT. This project uses virtual reality in the mental heath sphere by giving the clientele exposure therapy for public speaking anxiety through a VR headset, tracking the head movements and wearing a smart watch to track the heart rate. By predicting the future heart rate of the clientele, the system can potentially become adaptive and therefore change the environment in the virtual reality to adapt to the situation by changing the crowd's behaviour [18].

Chapter 2

Background

2.1 Virtual Reality and Mental Health

Virtual reality is a technology which lets a user interact with a simulated three-dimensional space with the help of a device such as goggles with one screen for each eye. This environment is often created to simulate the real world or a fictional world, depending on the purpose of the environment. This space is normally represented by three dimensions and the purpose is to interact with different objects and/or people that also are simulated in the environment [10]. This is commonly used for video games, but is increasingly applied within the mental health sphere. More researchers are now using this newer technology to help patients with their conditions through realistic exposure [3, 9, 19, 20].

VR can also be a useful tool within the education system as outlined in [10]. In that paper they discuss the idea of using VR for classes like in general studies, engineering, special needs and medical education. In cases such as these, VR can provide the opportunity to make students learn things in a way that is hard to obtain in a regular classroom. They could get to handle tools that are too expensive or dangerous to take care of for all students.

In this thesis the problem will be about the PSA case and the use of a VR environment to help adolescents with their mental issues regarding this case. It has been shown that VR treatment can help significantly with mental health problems such as PSA as proposed in [9]. The experiment had a virtual classroom with other simulated students or characters who were sitting on their desks, as in a regular classroom, going to watch the user present in front of them. In Figure 2.1 is an illustration of the environment that was described. To measure the process, they were using a smart watch to track the heart rate of the patient. The greater the heart rate, the more uncomfortable and nervous the patient was.

Lindner et al [15] examined the effectiveness of VR exposure therapy for PSA in routine care. This study found that there was a significantly decreased rate of self-rated PSA after the three hour session performed in the study.

Lin et al [13] suggested an underlying arousal feedback-based machine learning model based on heart rate data to influence the difficulty and challenge the subjects in PSA exposure therapy sessions in VR environments. In this study, the subjects had to give six 2 minute speeches to a virtual audience with different behaviours. While they were giving their speeches, the system gave feedback on their arousal levels and then modified the behaviour of the audience accordingly. The study showed that this type of treatment is acceptable and safe for most of the subjects and showed effectiveness for treating PSA.

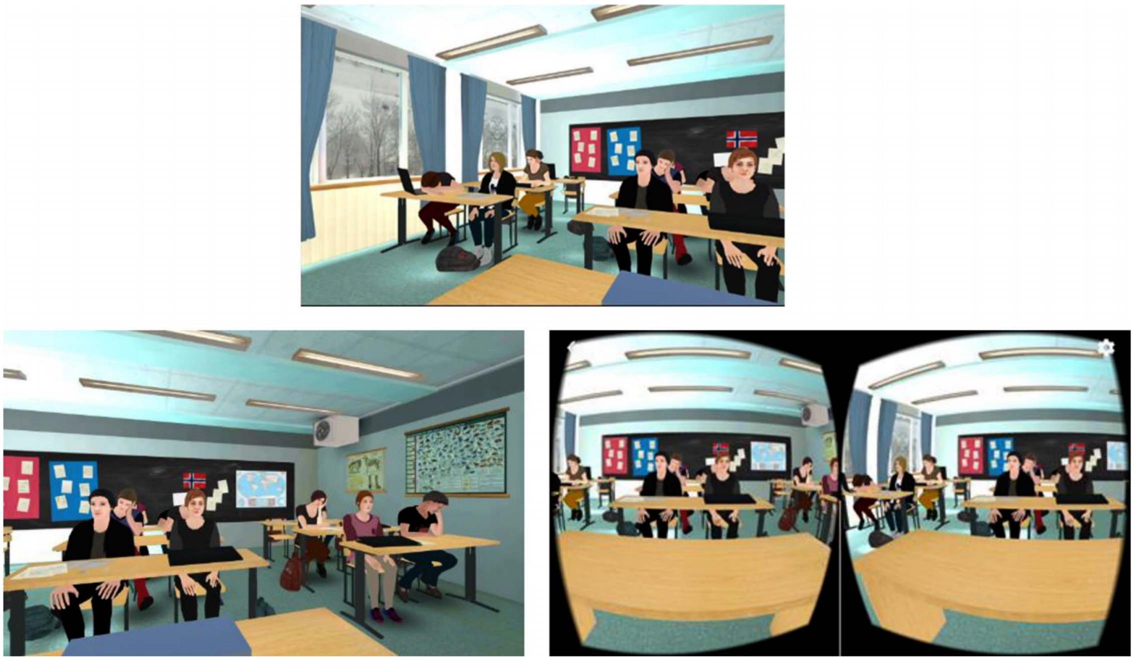


Figure 2.1: Screenshot of the virtual classroom from [9].

2.2 Supervised Learning

Machine Learning is a field within artificial intelligence where the goal is to learn and adapt from empirical data. Machine learning can be split into different categories based on the problem to be solved. Supervised Learning is a task where a model has an input set of example data we want to predict a label for, and a corresponding desired output set of labels. In this context an example data will be data that is collected which describes for example a situation or an object, and the labels represents what this example data is classified as. This model then is learning a function which will map the training set examples to their associated labels with the goal of being able to map new examples to their associated labels as accurately as possible. So the goal here is for the model to be able to generalize to unseen data. An important assumption generally taken in supervised learning is that the data is independent and identically distributed.

Often used supervised learning techniques are feed-forward neural networks (FFNNs). These types of architectures will have an input layers, some hidden layers and an output layer with different numbers of neurons, as illustrated by Figure 2.2 with a single hidden layer. In a supervised learning context, neural networks are often use to address both classification and regression problems. Neural networks with several hidden layers are called Deep Neural Networks (DNNs), and is used for Deep Learning (DL) algorithms.

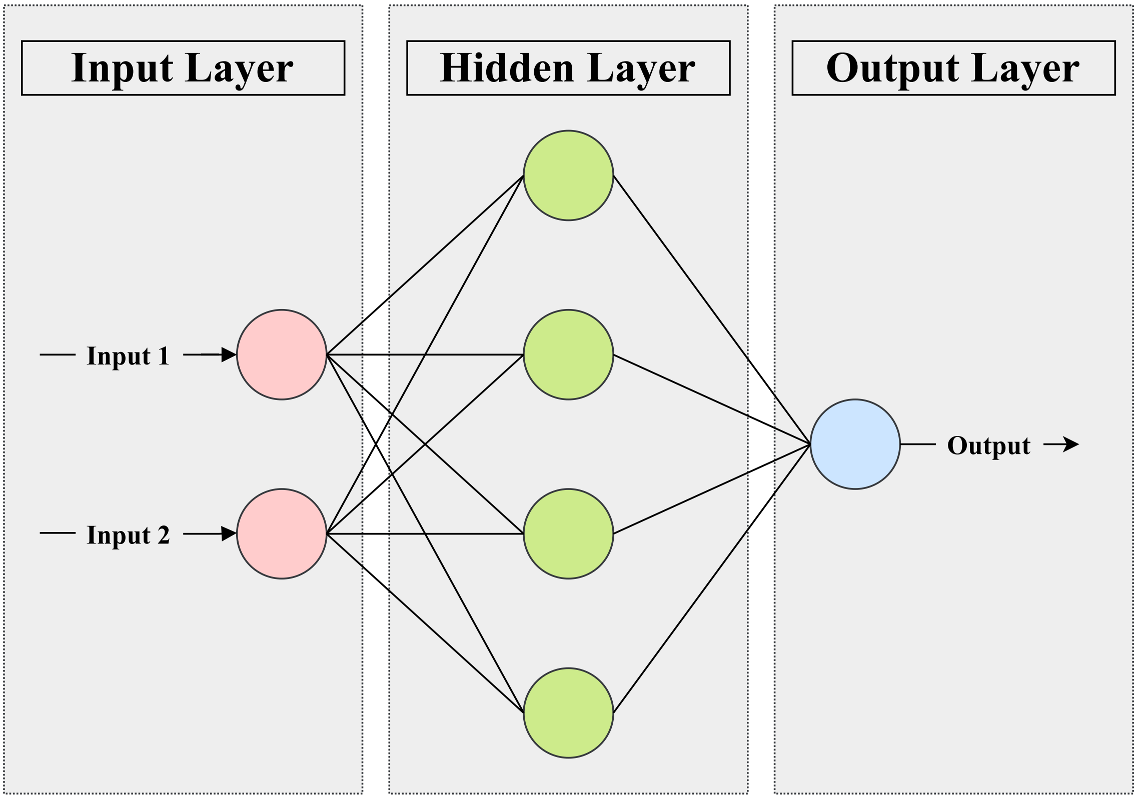


Figure 2.2: An illustration of the flow of a Neural Network. In this architecture the input layer contains two input neurons which gets fed through to the first hidden layer. In this figure the number of hidden layers is just one with four hidden neurons with corresponding weights. The outputs of the hidden neurons gets fed though to the output layer, which contains one neuron in this case. Something which is not shown in the figure is that bias nodes can also be present.

2.3 Supervised Learning for Sequence Data

Sequence data contains data where future input likely depends on previous inputs and are thus not independent and identically distributed. This can include text streams [11], audio or video clips, time series data, etc. Therefore these kinds of datasets will need a model that can handle these dependencies. FFNNs are however not sufficient for sequenced data, as they only consider non-structured data and do not capture trends in time series data. By including sequenced data in the supervised learning algorithms, the complexity will suddenly be much greater, due to the large amount of input data. Time series is as mentioned above a type of sequenced data, and in this thesis, we will be looking further into the methods available to handle these kind of data. These datasets contain recorded values with a corresponding time stamp. Time series that only record the values from one single variable is normally referred to as univariate time series. Correspondingly, time series that record the values from more than one variable is referred to as multivariate time series, and the variables in these time series often have some form of dependency to neighboring data points. As this thesis is concerned about forecasting heart rate within the context of an exposure therapy session there will be access to a dataset with heart rate and inertial measurement unit (IMU) data of head movements, which will then go under the multivariate time series category. Models often used for handling sequential data are Recurrent Neural Networks.

2.3.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a kind of DNN architecture which were specifically developed for sequential data. The advantage of using RNNs for these types of data is the capability of feeding back their outputs to their input of previous time steps, and predicting the outcome of the next step from those results. This is not possible with a standard FFNN. The main difference of RNNs and FFNNs is that the information just flows in one direction (forward) between the layers, where the RNNs the current and previous time steps of information will be fed into the RNN cells [1], see Figure 2.3. An RNN cell can be described as:

$$h_t = tanh(W * [h_{t-1}, x_t] + b) \tag{2.1}$$

Where b is the bias, W indicates the weights, x_t is the current input and h_{t-1} is the hidden output for the previous time step, as also explained from Figure 2.4.

A regular RNN will not be able to capture long-range dependencies well, which is a problem when dealing with longer input sequences. This is because the gradients will decrease at a high rate as it gets back-propagated through the deeper layers of the network, and will downplay the importance of the earlier time steps. This is also called the vanishing gradient problem. However, this problem is partially addressed by extension of the RNN architecture. One of these implementations will be addressed in the next subsection.

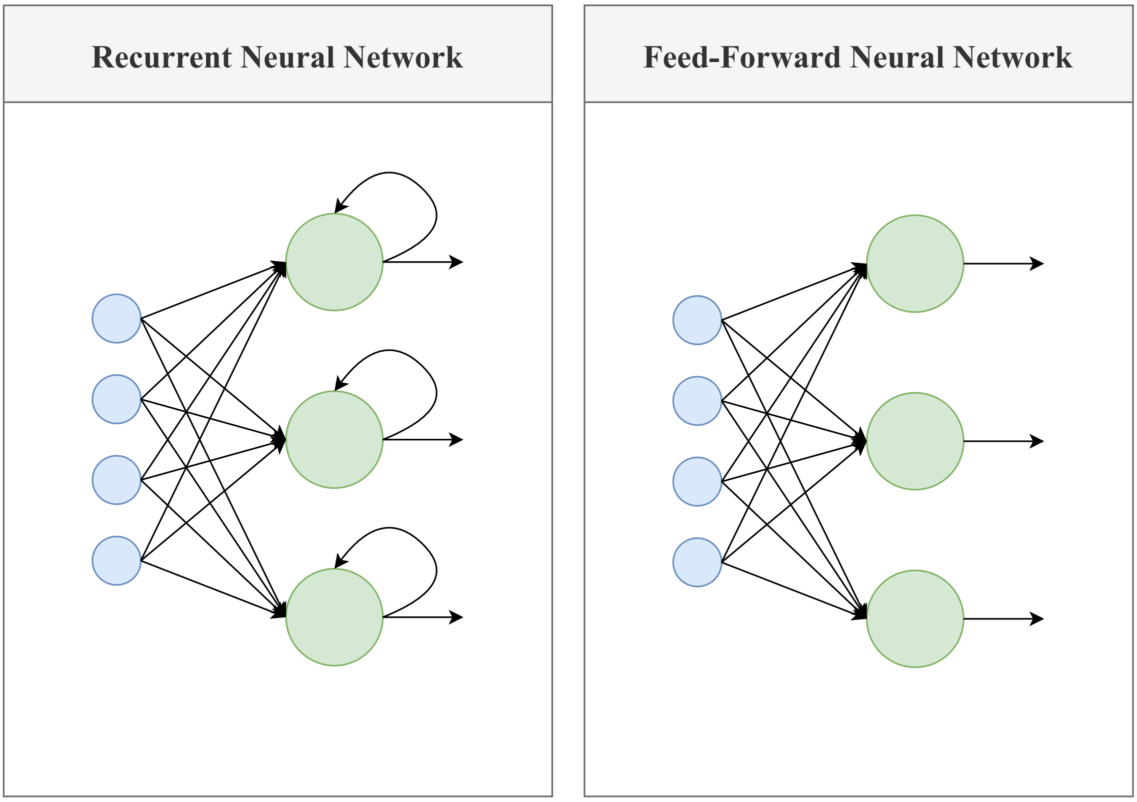


Figure 2.3: Typical Recurrent Neural Network (RNN) flow (left) compared to a regular Feed-Forward Neural Network (right). The RNN feeds back to itself in the hidden layers, while the regular neural network only feeds forward through the network.

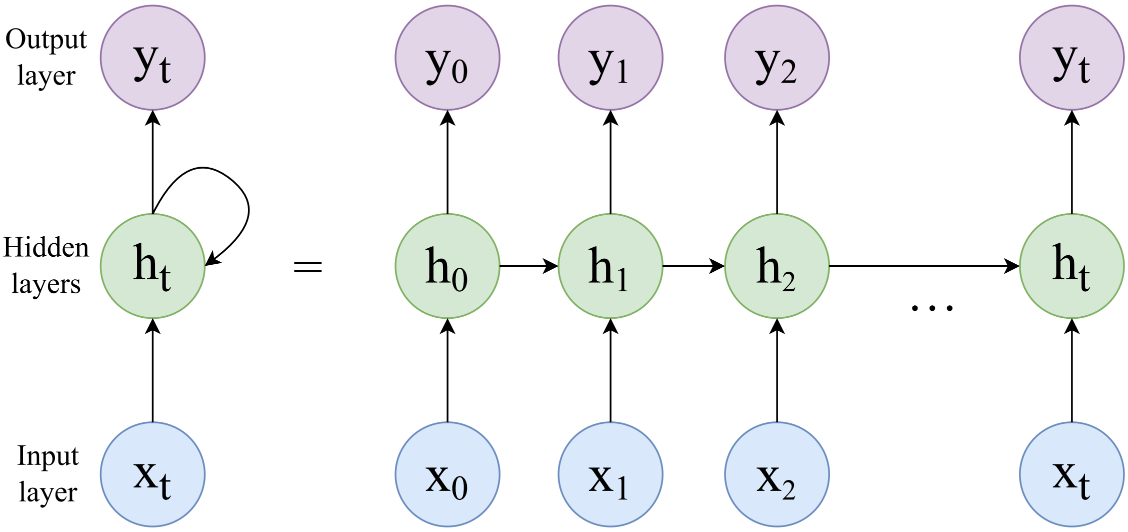


Figure 2.4: Showing the hidden layer structure in a Recurrent Neural Network (RNN). The hidden layer h_t at time step t gets info from all previous hidden layers.

2.3.2 Long Short-Term Memory

To overcome the vanishing gradient problem [7], the Long-Short Term Memory (LSTM) cell was introduced by Hochreiter et al [8]. These LSTM cells were used instead of the regular RNN cells in the vanilla RNN architecture. The main difference between the normal RNN cells and LSTM cells is the introduction of a new cell state. The LSTM cell consists of an input gate, an output gate, a forget gate and an update gate. Figure 2.5 is an illustration of the LSTM cell. This shows the extra output of the cell, $\tilde{h_t}$, which the previous cell state, h_{t-1}, is slightly adjusted by simple operations as multiplication and addition. This will enable the network to remember the inputs over a longer period of time and is therefore less sensitive to decrease as it propagates through the deeper layers of the network.

The forget gate f_t will determine what information will be forgotten, hence the name, from the previous state:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{2.2}$$

Where W_f and b_f are the current weight and bias of the forget gate.

The input gate i_t picks out the information that will be accepted into the current cell:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{2.3}$$

Where W_i and b_i are the current weight and bias of the input gate.

The update gate $\tilde{c}_t$ simply updates the cell state, and creates the following cell state output c_t:

$$\tilde{c}_t = tanh(W_c * [h_{t-1}, x_t] + b_c) \tag{2.4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{2.5}$$

Where W_c and b_c are the current weight and bias of the update gate.

The output gate o_t updates the current output, while also updating the previous $(t-1)$ hidden layer:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{2.6}$$

$$h_t = o_t * tanh(c_t) \tag{2.7}$$

Where W_o and b_o are the current weight and bias of the output gate.

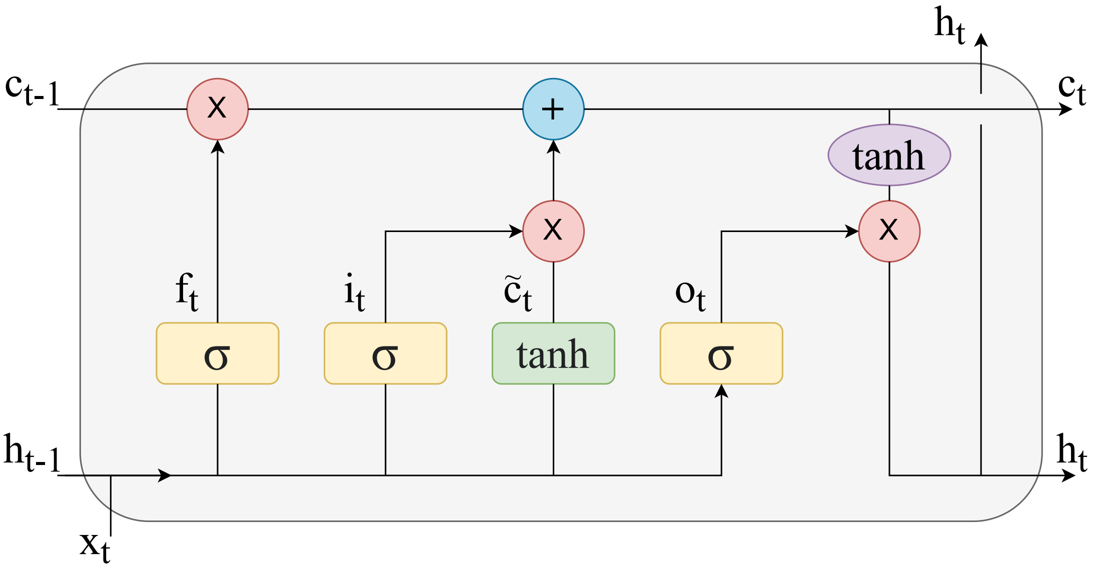


Figure 2.5: Illustration of an Long-Short Term Memory (LSTM) cell. The cell has three different inputs, the previous cell state c_{t-1}, the previous hidden state h_{t-1} and the current input value x_t. The outputs is the new cell state c_t and the new hidden state h_t. The LSTM cell contains different gates, such as the forget gate f_t, the input gate i_t, the update gate $\tilde{c}_t$ and the output gate o_t. f_t, i_t and o_t are all composed by a sigmoid function σ and a pointwise multiplication, which decides which information gets through.

2.3.3 Probabilistic Forecasting with Autoregressive Recurrent Networks (Deep AR)

Probabilistic Forecasting with Autoregressive Recurrent Networks (Deep AR) is a forecasting method for time series made by Amazon [22]. This model predicts future inputs by learning a global model from previous data of all time series in the dataset [22], which means it can handle multivariate time series. Deep AR is also able to learn seasonality and uncertain growth of data over time [4].

Deep AR uses stacked LSTM layers, as presented in Figure 2.6, and then fed into two separate linear layers. The first linear layer is to determine the mean μ of the model and the second is to determine the standard deviation σ, which is then fed into a softplus layer to make sure the values are positive [17]. These outputs are then the inputs to generate parameters of one-step-ahead Gaussian predictive distributions [12]. The inputs for the model are time series from previous time steps, the current time step for each covariate and the time series indexes, which are then fed into an embedding layer. These inputs gets concatenated before getting fed into the stacked LSTM layers.

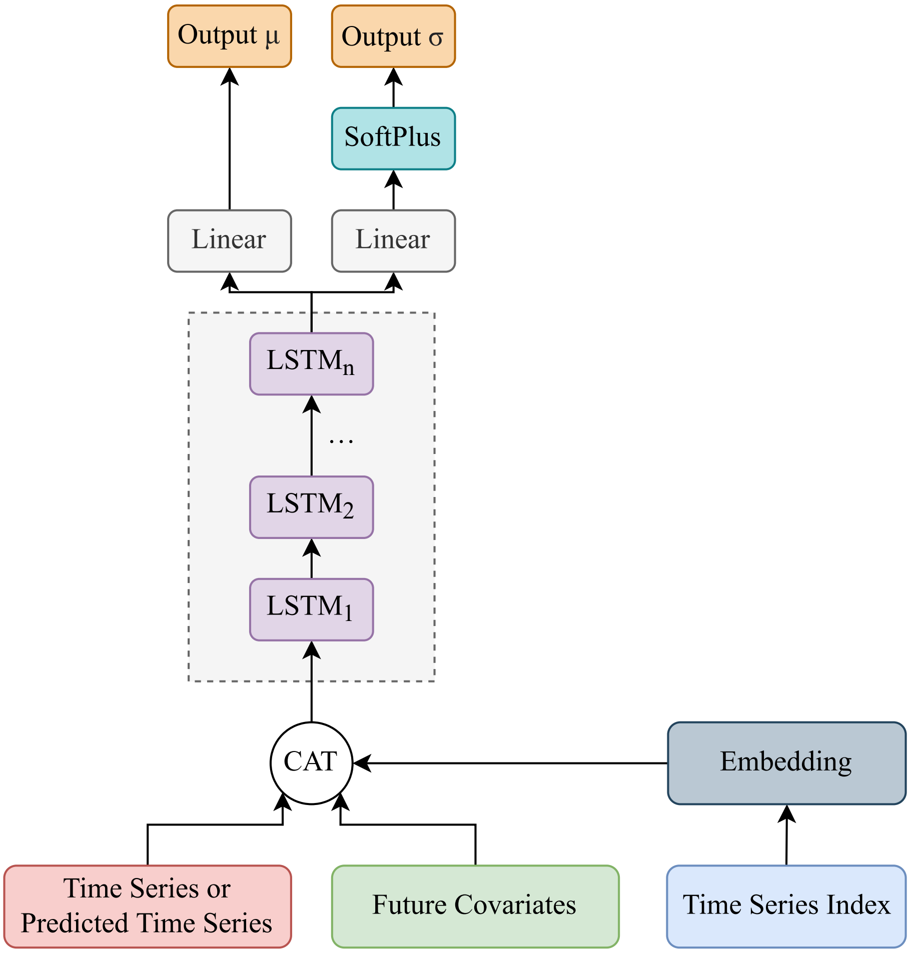


Figure 2.6: Probabilistic Forecasting with Autoregressive Recurrent Networks (Deep AR) architecture. The current or predicted time series, future covariates and the corresponding embedded time series indexes and concatenated (CAT) and then fed into a stack of Long Short-Term Memory (LSTM) layers. The output of the last LSTM layer is fed into two separate linear layers, where the output of the first linear layer determines the mean μ and the second layer is followed by a softplus layer which will determine the standard deviation σ. The softplus layer makes all the values positive.

2.3.4 Transformer Models

Transformer models are, as well as RNNs, made to process sequential data. The Transformer model was first introduced by Vaswani et al [26] to utilize more parallelization, and reducing sequential computation, to make the forecasting process more efficient. This model was originally created for language modeling, mainly to create context to text sequences and choose the importance of each part of the input with the use of attention mechanisms.

The transformer model architecture consists of an encoder-decoder structure. The encoder consists of six layers, where each layer has two sublayers. In Figure 2.7, the encoder is the left part of the illustration, and the two sublayers is first the multi-head attention layer, followed by a simple feed forward layer. Similarly, the decoder, on the right side of Figure 2.7, is also composed of six equal layers where each layer consists of three sublayers. Each sublayer of both the encoder and decoder is followed by a normalization of the outputs. This is expressed by $LayerNorm(x + Sublayer(x))$. The third layer performs a multi-head attention on the output of the encoder.

While RNNs handle the data in order, the attention mechanism will give a context to the data without processing the data in any particular order. The Transformer uses Multi-Head Attention, which consists of several Scaled Dot-Product Attention layers running in parallel.

The attention methods that are the most widely used is dot-product attention and additive attention [2]. The Transformer uses the dot-product attention method, but with a small change. In addition to the dot-product the Transformer adds a scaling factor to the attention method. Figure 2.8 shows the structure of the Scaled Dot-Product Attention to the left. The output of this structure will look like this:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.8}$$

Where Q is a matrix made up of a set of queries, and K and V are respectively the keys and values of these queries. The scaling factor $\frac{1}{\sqrt{d_k}}$ contains the dimension of the keys (K), d_k.

The Multi-Head Attention runs several attention computations in parallel, where each of these computations are called an Attention Head. This attention method will split the Q, V and K matrices in n parts, and sends each of these parts into separate Attention Heads. When all of these computations are done, they will be concatenated into a single score. Figure 2.8 illustrates this flow to the right.

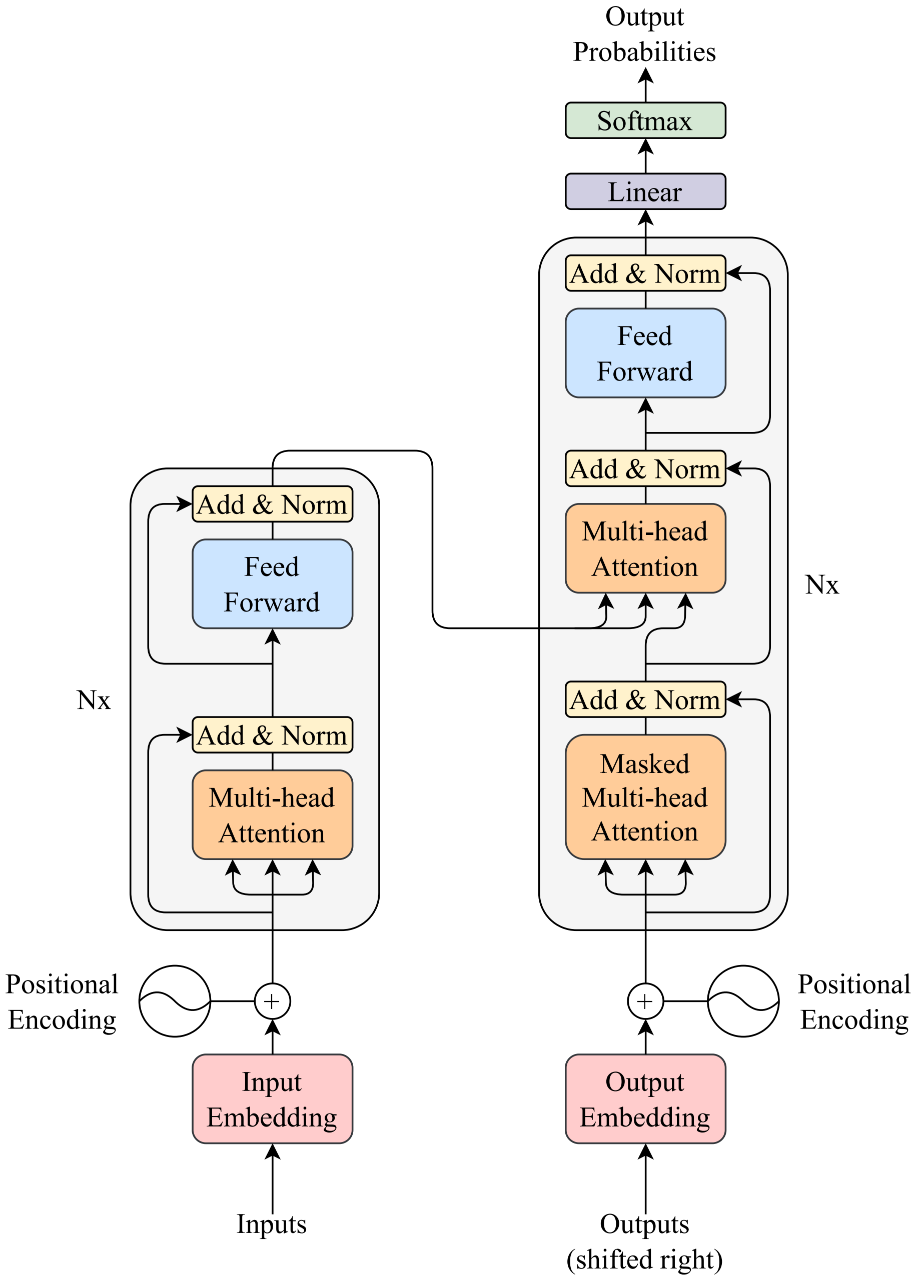


Figure 2.7: Transformer model architecture containing an encoder (left gray box) and decoder (right grey box). The encoder and decoder each consists of six equal layers where the encoder has two sublayers in each layer and the decoder has one extra sublayer in each layer. The Encoder has a Multi-head Attention layer followed by a Feed Forward layer. The decoder has an extra third sublayer where it performs Multi-head Attention on the output of the encoder. All the sublayers in both the encoder and the decoder are followed by a normalization of the output. This figure is inspired by [26].

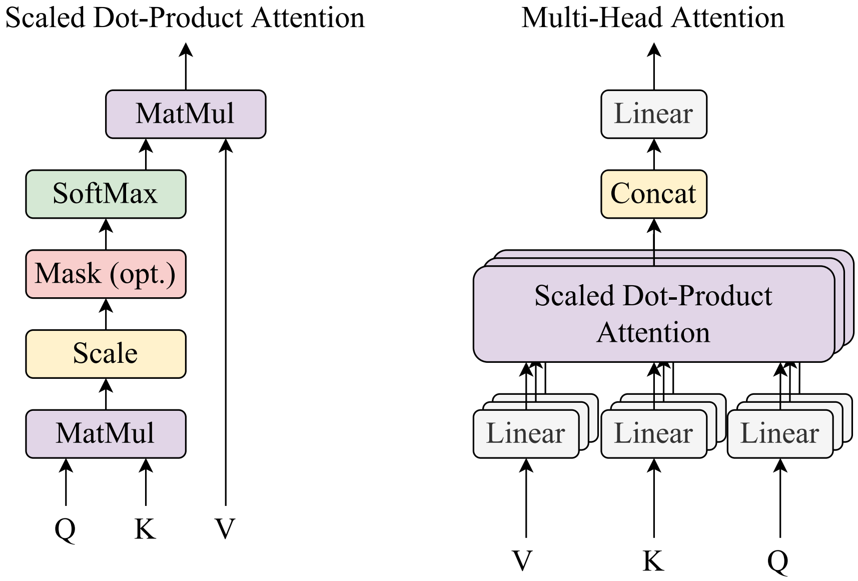


Figure 2.8: Scaled Dot-Product Attention (left) performs matrix multiplication on the queries Q and its keys K, scales the result with a value of $\frac{1}{\sqrt{d_k}}$, optionally masks the output and calculates the softmax. Lastly the values V of the queries is multiplied with the result from the softmax layer. Multi-Head Attention uses the Scaled Dot-Product method in several layers, where Q, V and K are all split in equal parts, and are later concatenated into a single score. This figure is inspired by [26].

2.3.5 Temporal Fusion Transformer

A Temporal Fusion Transformer (TFT) is a type of Deep Neural Network which is attention-based for multi-horizon forecasting, which is the prediction of variables at multiple future time steps [12].

This architecture which is shown and explained in Figure 2.9 incorporates gating mechanisms to ignore unused components of the architecture, which helps the model adapt to different datasets and scenarios. Next are variable selection networks that selects the most relevant features each time step from both static and time-varying covariates. These are then transformed into vector form and fed through Gated Residual Networks (GRN) [23]. There is a sequence-to-sequence layer to locally process known and observed input data, to learn the short-term temporal relationships of observed and known inputs, and a temporal self-attention decoder (a multi-head attention block) to learn the long-term temporal relationships within the dataset [12].

Like Deep AR, TFT is also capable of handling more than one input variable, as an advantage for the ARIMA models. This study [12] shows that TFT outperforms both ARIMA and Deep AR based on quantile loss.

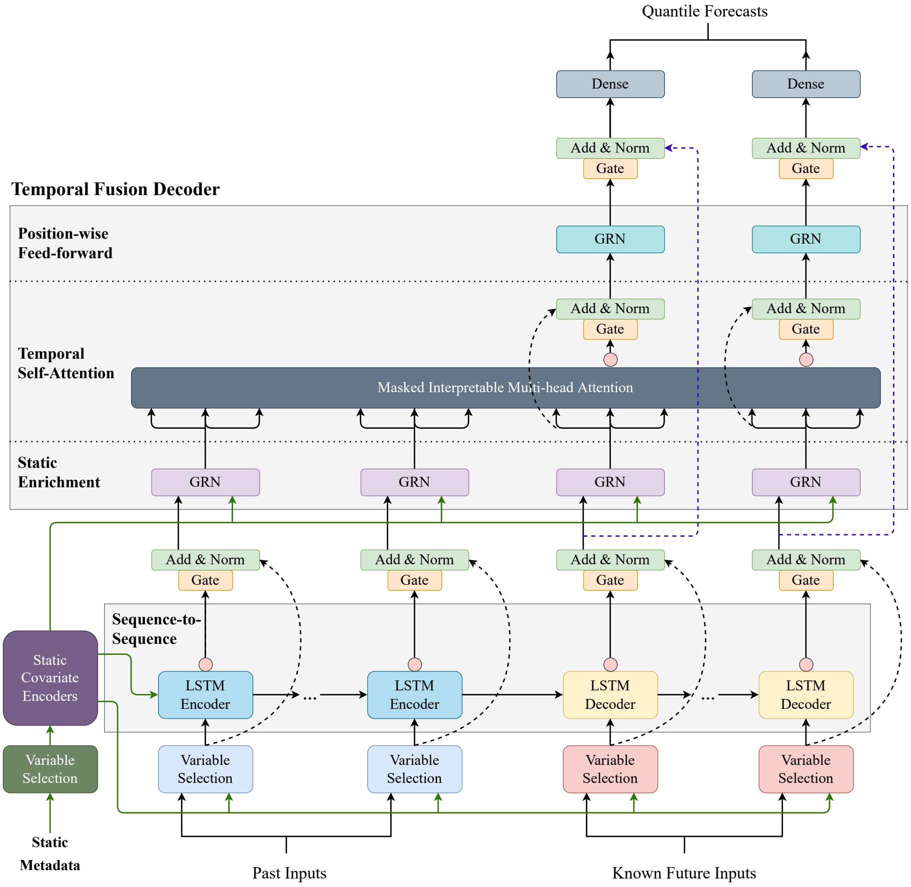


Figure 2.9: Temporal Fusion Transformer architecture. Starting with variable selection from static metatdata, followed by static covariate encoders. The output of this will together with the past and known future inputs be inputs to the variable selection method. The sequence-to-sequence layer with Long Short-Term Memory (LSTM) encoders and decoders inputs the output of the variable selection and the static covariate encoder. These outputs is later added and normalized together with the variable selection outputs and fed into the Gated Residual Networks (GRN) for static enrichment. The output of the GRNs gets fed into the temporal self-attention layer with masked interpretable multi-head attention. This is also followed by a normalization and addition of these outputs and the outputs from the GRN layer from the known future variables. This is followed by another GRN layer also followed by adding and normalization. The final outputs goes though a dense layer. This figure is inspired by [12]

2.4 Heart Rate Prediction

In general there is not very common to find work on heart rate prediction or forecasting, yet alone for heart rate forecasting for mental health projects.

INTROMAT currently has the only other study on heart rate forecasting within a mental health context. INTROMAT takes on the same problem as in this thesis with the heart rate prediction from head movement for treating PSA by VR exposure therapy [18]. This experiment and dataset will be described further in Section 3.1.2. This study seemed to show promising results testing the Zero R as a baseline, Simple Linear Regression, Random Tree and Decision Stump. The models all used 10 fold cross-validation with ten repetitions. For measuring performance, they used the Relative Squared Error, the Mean Absolute Error and the Root Mean Squared Error. All of the suggested models seemed to perform better than the baseline used in this study, and created accurate user independent models for predicting heart rate by head movement data.

Other heart rate prediction studies have mostly been within the physical activity field. One of these studies is for predicting the heart rate trend during high-intensity interval training (HIIT) by Fedorin et al [5]. In this study they tested a system which includes a pre-trained neural network with combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) layers. The goal was to predict the heart rates for the next interval during the training. For performance measure they used the Mean Absolute Error and the Root Means Squared Error, which showed great results with errors below 4 beats per minute (bpm).

A study done by Staffini et al [24] suggested heart rate forecasting for monitoring anomalies in the heart rate and address the risks of different diseases. They tested forecasting methods like an Autoregressive process, an LSTM network and a Convolutional LSTM (ConvLSTM) network. The error metrics used for this project was again the Mean Absolute Error and the Root Means Squared Error. The results of this study was that all the models performed similarly, but the Autoregressive model scored statistically significantly better than the other models. The errors were all below 7 bpm for every participant tested in the study.

Chapter 3

Datasets and Methods

This chapter will present the datasets and model implementations considered in this work.

The methods or implementations tested for these datasets are all PyTorch Forecasting models. These include the Baseline model, an Long Short-Term Memory (LSTM) network, the Temporal Fusion Transformer (TFT) and the Deep AR model.

3.1 Datasets

The goal of this thesis is to predict heart rate as a proxy to assess the level of engagement/stress the person is experiencing during an exposure therapy session. The INTROMAT project provides heart rate data including head rotation data from a virtual reality environment, as stated in Section 1.1. Other datasets containing similar types of variables are also reasonable to test time series forecasting on. Data from the mental health sphere is known to be expensive and time-consuming to produce. There is therefore a lack of time series data for heart rate for mental health, but other alternatives would then be datasets with similar types of variables. The Physical Activity Monitoring (PAMAP2) [21] contains similar variables, although the experiment for this dataset is focused on physical and not mental health.

3.1.1 Physical Activity Monitoring (PAMAP2)

The Physical Activity Monitoring (PAMAP2) dataset introduced by Reiss and Stricker [21] initially to be used in classification problems in relation to physical activity. However, as this dataset contains time series, it can also be used for time series forecasting. PAMAP2 contains time series data from three inertial measurement units (IMUs) and a heart rate monitor. The IMUs they used was the Colibri wireless IMUs, where the sampling frequency was at 100 Hz. The three IMUs were located at the wrist (right for right-handed subjects or left for left-handed), one was on the chest and the last on the ankle (left or right depending on which is dominant for the subject). The heart rate monitor was from BM innovations GmbH and had a frequency of approximately 9 Hz, which differs from the IMUs.

This dataset comprise nine subjects who performed 18 different types of physical activities such as walking, running, cycling, etc. 12 of the activities were part of the main protocol of the experiment, but some of the subjects performed 6 additional and optional activities which were related more to everyday activities such as watching the television and cleaning the house. In the dataset there are labels to determine which activities each time serie belongs to, for the classification methods. The dataset will be divided by each subject and trained individually. As the last subject has a more limited amount of data compared to the other subjects, the models will only be trained on eight of the subjects. This dataset is mainly to add additional results for heart rate prediction, but on a physical activity case.

3.1.2 INTROMAT Public Speaking Anxiety Virtual Reality

The INTROMAT Public Speaking Anxiety (PSA) Virtual Reality (VR) dataset by Kahlon et al [3] is as mentioned in Section 1.1 a dataset in which adolescent subjects were tested in a VR environment of a full classroom to see how this affected their PSA, as a form of exposure therapy. As opposed to normal exposure therapy in real life, without the VR environment, this experiment is easier to perform without the need of a full audience for each session, where all participants need to agree on every single event time.

The subjects wore a wearable wireless wristband called Empirica E4 to collect heart rate data. This heart rate data was then synchronized to a smartphone where it logged the heart rate for every second (i.e 1 Hz). They also included data from the VR headset, which measured the head rotation in the x-, y- and z-axis. The number of participants for this experiment was 21 subjects. This dataset is also divided by each subject and trained individually. All of the subjects in this dataset is being considered for this experiment. The goal of using this specific dataset is to predict the heart rate to make this VR enviroment from this experiment eventually adaptive.

3.2 Data Pre-Processing

The methods will be performed on each subject individually. Therefore the dataset will be split, so there is one time serie for each variable for each subject. Since the IMU data contains three axes, and therefore measures movement in each direction, the data will be transformed so that it measures the mean and variance of the combined axes. This is mainly to focus more on the total movement, as which direction the IMU sensor, or which body part this sensor is placed on, is not relevant to this case. The time series dataset should contain data of 1 Hz, one data point per second for both the heart rate and IMU data.

The transformation of the IMU data consists of firstly finding the norm of the movement at each time step, then subtracting the gravitational force g. This is done with:

$$IMU_{norm} = \sqrt{IMU_x^2 + IMU_y^2 + IMU_z^2} - g \tag{3.1}$$

Where IMU_{norm} is the norm of the movement vector, IMU_x, IMU_y and IMU_z are respectively the vector values of each direction on the x, y and z plane and g is the gravitational force with the value $9.81m/s^2$.

To change the data points per second, there is performed a simple moving average [25] across all the data points that adds up to 1 second. For example the PAMAP2 dataset contained heart rates of 9 Hz, so the moving average would be calculated across nine data points to get 1 Hz. The INTROMAT dataset is already at 1 Hz, so there is no need for a moving average. The simple moving average for one time step is calculated with this formula:

$$SMA(t) = \frac{1}{k} * \sum_{i=1}^{k} x(t-1) \tag{3.2}$$

Where SMA is the simple moving average, t is a specific time point, k is the total number of data points in that sequence and $x(t-i)$ is the values of the specific data points within the sequence.

The same is done to the IMU_{norm} data, creating IMU_{mean}. There was also added another variable, instead of finding the moving average, the new variable will be the variance of the IMU_{norm}, which is called $IMU_{variance}$. This is calculated as:

$$IMU_{variance}(t) = \frac{1}{k-1} * \sum_{i=1}^{k} (x(t-i) - mean(x))^2 \tag{3.3}$$

Where t is a time point in the sequence that is chosen, k is the total number of data points, $x(t-i)$ is the values of the specific data points within the sequence and $mean(x)$ is the mean value of the data points in the sequence. Naturally, as the INTROMAT dataset as mentioned already is 1 Hz, we cannot find the variance of the data within one second as we could for the PAMAP2 dataset.

As the features come from different modalities and to facilitate the learning process of their relative importance by the model during training, normalization is used where each feature is independently scaled to have 0 mean and a unit variance. The normalization method used on these datasets was the Standard Scaler [6]. This scaler transforms the data such that the mean of each feature, in this case just the heart rate, will be 0 and all the values will be between -1 and 1. This is calculated by taking each data point from one feature, subtracting the sample mean and divide by the sample standard deviation:

$$z = \frac{x-u}{s} \tag{3.4}$$

Where z is the scaled data point, also called the z-score in statistics [16], x is the value of the data point, u is the sample mean and s is the sample standard deviation. The mean and standard deviation from the training dataset is used for the training dataset, validation set and testing set.

The INTROMAT dataset contain data where the subject is in different locations in the virtual environment, meaning that not all of the time series contain data where the person is in the virtual classroom. In this experiment we want to focus on the public speaking anxiety occurring in the classroom with a full audience. Therefore the time series where the subject is in any other location than the classroom is removed.

3.3 Train, Validation and Test Data

Because the two datasets being used are of different sizes, the train, validation and test split must be different. For the finished pre-processed version of the PAMAP2 dataset, each subject has around 3000-4000 data points, or seconds. To have enough data to train on, the chosen size of the dataset is 100 seconds in the validation set, 500 seconds in the test set and the rest for training. The test data is taken from the last part of the independent exercises from the same individual so that there is no information leakage between the different datasets.

For the INTROMAT dataset, there are provided segment identification numbers to indicate new time series or sessions. Each subject in this experiment performed between six and eight sessions, so it was therefore meaningful to split the data for each subject so that the test set would contain the one last session, and the validation would have the second to last session. The rest of the sessions would be the training data.

3.4 Labeling

As the data being predicted does not get classified or categorized into any group, the labeling for forecasting will be different from classification problems. The data will be split into several sequences. These sequences will also be split into two parts where the first part of each sequence will be the input, or encoder, and the rest will be the prediction, or decoder. Depending on the size of the dataset and how long you want to predict into the future, the length of the encoder and decoder should be adjusted accordingly.

For the PAMAP2 dataset, we want to predict the heart rate 20 seconds into the future, by looking at the 80 seconds prior to this prediction. We also want to set a minimum encoder size to make the dataset more diverse with different sizes of observed data. Therefore the minimum encoder length is set to be double the prediction size, so the model always have a decent amount of data points to predict from.

The INTROMAT dataset is smaller than the PAMAP2 for each subject, so this requires a smaller total sequence size. We will want to keep the maximum encoder size to be 80, but the prediction length we want to forecast is set to be 10 seconds. The prediction length for this dataset is lower than the PAMAP2 because the total length of each recording was around one minute, which is significantly less than the PAMAP2 recordings. The minimum encoder length will then be set to 20, which is double the prediction length.

3.5 Implementation

In the implementations the goal is to see whether the models manages to guess the trend of the future heart rate. To predict the exact heart rate is therefore not the main goal of this experiment.

The methods will be ran both univariate, i.e just the heart rate as a variable, and multivariate, i.e both heart rate and IMU data. They will also be ran on both of the datasets discussed in Section 3.1.

The hyperparameters of the models will be adjusted to the PAMAP2 dataset, but also used on the INTROMAT dataset.

3.5.1 Baseline model

The Baseline model from the PyTorch Forecasting library is used here to compare the different models. The Baseline model uses the last known target value to make the prediction. That means that the last heart rate value from the encoder sequence will be predicted as the forecast for the sequence. The forecast will then be just a straight horizontal line in a plot, and is therefor a good comparison to the other methods to see if they can predict the right trend in the time sequence.

The Baseline model only takes the last known value of the target, and therefore needs no hyperparameters, and does not need to be trained on any data.

3.5.2 LSTM Network

The LSTM network to predict multiple time steps in this experiment consists of first a linear layer, followed by an LSTM layer, a dropout layer and lastly another linear layer. This flow is illustrated in Figure 3.1. The target variable for this model was the normalized heart rate, and the features used for predicting the target were the normalized heart rate and IMU data.

The first linear layer takes in the observed data, or the encoder part of the sequence, of size (batch size x encoder length x number of features) and outputs a sequence of size (batch size x encoder length x output size), where the last output size is set to be 8, to keep the number of parameters in the network on the smaller side to train faster and avoid overfitting. This size will also be the input for the LSTM layer, and the hidden size is set to be 32. The dropout layer has a dropout rate of 0.5. The last linear layer takes in an input size of the same size as the hidden size of the LSTM layer, and output size of the prediction length we want to forecast. The output of this last layer is the total output of the multi-step LSTM network. The learning rate for this method was set to be 0.01, also using the Adam optimizer to enhance performance and speed up the training. This method also used early stopping with a precision of 1e-8 and patience 10, also to speed up the training process.

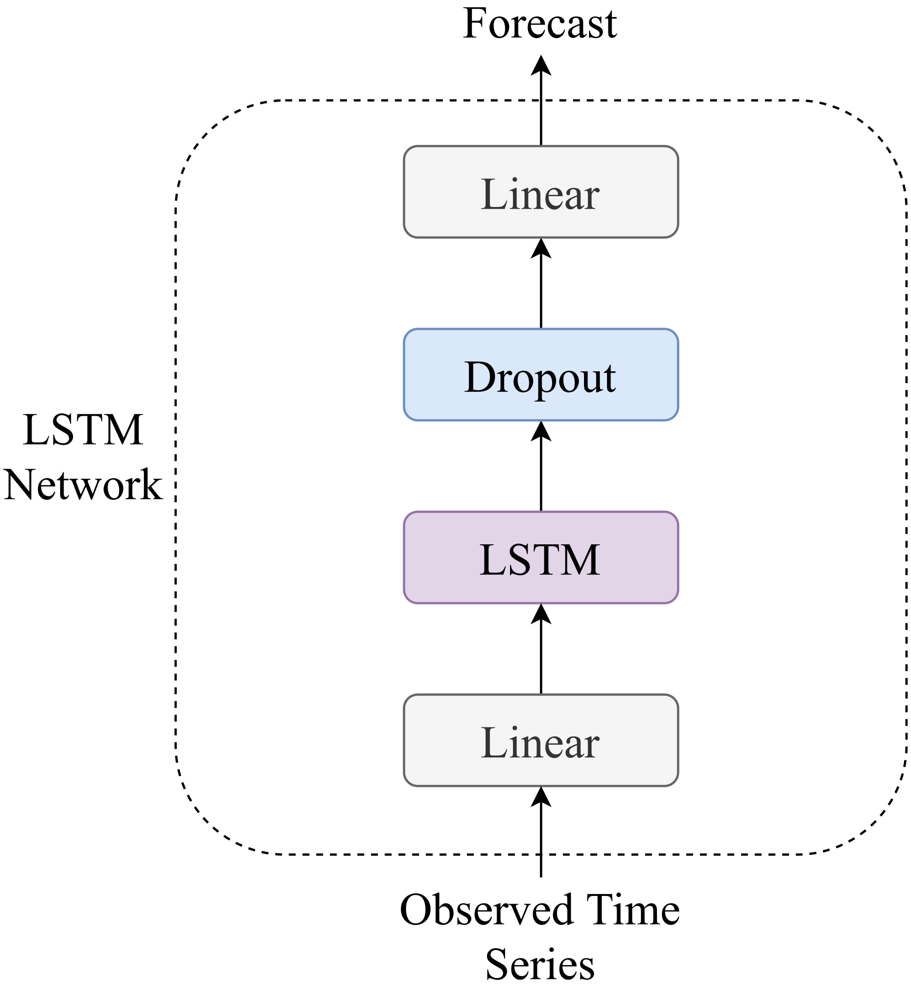


Figure 3.1: Long Short-Term Memory (LSTM) Network. Inputs observed time series data from one batch, feeds into a linear layer, then into an LSTM layer. This output gets fed into a dropout layer and directly into the last linear layer. The output of this network is the predictions for the whole batch.

3.5.3 Temporal Fusion Transformer (TFT)

The Temporal Fusion Transformer (TFT) architecture discussed in Section 2.3.6 and shown in Figure 2.9 is used to implement this method, without any modification to the architecture. The hyperparameters is chosen to be 0.02 as the learning rate. The hidden size of the network is set to be 16, and an attention head, like the ones discussed in Section 2.3.5, with a size of 1. This model has a dropout rate of 0.1. Early Stopping was also used for training and validating this model with a precision of 1e-8 and patience of

10.

The target value for this model was the normalized heart rate and the features were the normalized heart rate and IMU data.

3.5.4 Deep AR

The Deep AR model and architecture are explained in Section 2.3.3 and Figure 2.6. Deep AR handles covariates in a different way than the LSTM network and the TFT, so the choices were to either choose to lag the covariates to convert them to known future variables or to choose to make all the covariates target values of the model. The choice landed on the second option so that the model was as similar as possible to the other methods explained in this chapter.

The hyperparameters of this model is a learning rate of 0.01, a hidden size of 64 and a dropout with a rate of 0.1.

3.6 Performance Metrics

The performance metrics for comparing the results of these models is the mean absolute error (MAE), mean squared error (MSE) and symmetric mean absolute percentage error (SMAPE). These metrics were chosen to compare the performance of the models in different ways as the metrics are influenced by different factors. In this experiment the values were all rescaled to its original scale as heart rates (bpm).

3.6.1 Mean Absolute Error (MAE)

The mean absolute error (MAE) is calculated as follows:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|y(t) - \hat{y}(t)| \tag{3.5}$$

Where N is the number of data points in the sequence, t is a specific time point in the sequence, the $y(t)$ is an actual value of the real time series, and the $\hat{y}(t)$ is the models estimate of $y(t)$.

This metric measures the mean of the absolute difference between the actual and the predicted value of the model. This metric does not penalize for big outliers of the data, but is sensitive to the scale of values of the data.

3.6.2 Mean Squared Error (MSE)

The mean squared error (MSE) is calculated as follows:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \hat{y}(t))^2 \tag{3.6}$$

Where N is the number of data points in the sequence, t is a specific time point in the sequence, the $y(t)$ is an actual value of the real time series, and the $\hat{y}(t)$ is the models estimate of $y(t)$.

As opposed to the MAE, this metric will be more affected by the bigger outliers in the data because of the squared term. This is also a metric that is particularly sensitive to the scale of the data, so it is not very good to compare the performance for different datasets.

3.6.3 Symmetric Mean Absolute Percentage Error (SMAPE)

The symmetric mean absolute error (SMAPE) is calculated as follows:

$$SMAPE = \frac{100}{N} \sum_{t=1}^{N} \frac{2|y(t) - \hat{y}(t)|}{|y(t)| + |\hat{y}(t)|} \tag{3.7}$$

Where N is the number of data points in the sequence, t is a specific time point in the sequence, the $y(t)$ is an actual value of the real time series, and the $\hat{y}(t)$ is the models estimate of $y(t)$.

As the MAE this metric does not penalize for big outliers of the data, but is inaffected by the scale of the data, meaning that it is better for comparing the performance using different datasets. A weakness of this metric is that when both the actual and predicted values are close to zero the result will diverge, as the denominator of the fraction will be going towards zero.

Chapter 4

Experiments and Results

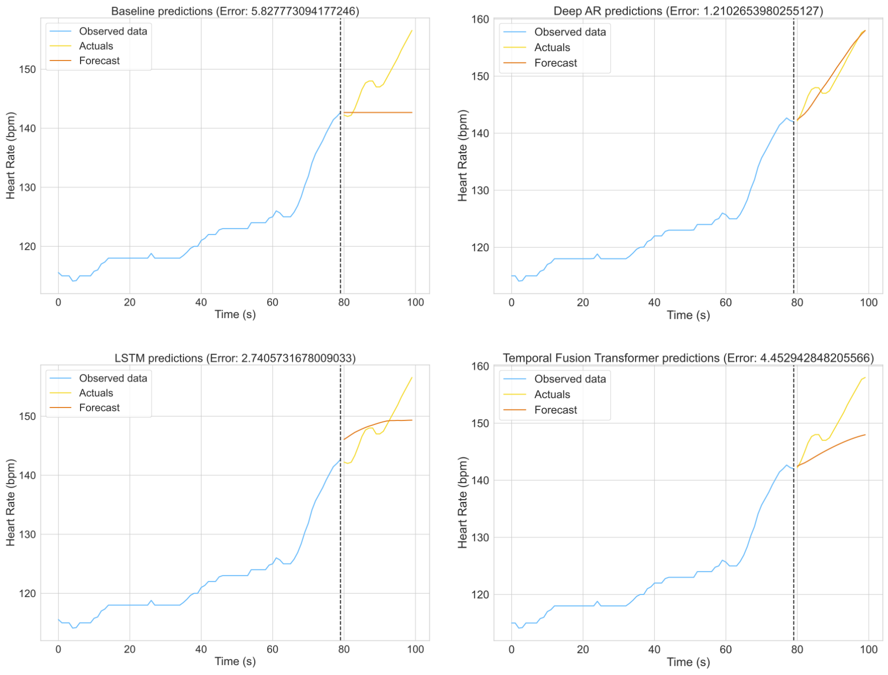
The implementations explained above in Section 3.5 is implemented for two different experiments. The experiments are divided such that there will be an analysis on both the univariate (i.e just heart rate as variable) and multivariate (i.e both heart rate and IMU data) scenarios. Both experiments will be performed on the two datasets described in Section 3.1; PAMAP2 and INTROMAT. All the plots which shows the metric scores are slightly scaled so that the big outliers will not dominate the plots.

4.1 Experiment I: Univariate

This experiment consists of running the Baseline model, LSTM network, TFT and Deep AR with just the heart rate as an explainable variable for the heart rate forecast.

4.1.1 PAMAP2

The results for the PAMAP dataset is presented as an example plot in Figure 4.1 of a sequence with the total length of 100 seconds, in which 20 of them are predicted values. From this plot there seems like the Deep AR scores the best from this specific example. However, it is not sufficient to look at just this example to conclude that this model performs the best. Additional examples of the predictions are listed in Appendix A (See Figures A.1 and A.2).


Baseline predictions (Error: 5.827773094177246)

Deep AR predictions (Error: 1.2102653980255127)

LSTM predictions (Error: 2.7405731678009033)

Temporal Fusion Transformer predictions (Error: 4.452942848205566)

Figure 4.1: PAMAP2 dataset prediction by just heart rate: Example 1. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 100 seconds, in which 20 of them are predicted values.

The MAE metric is presented in Figure 4.2, and it shows that the TFT model has the best score of all the models as it is the one that scores nearest zero. This plot also shows that the Deep AR scores a median value that is better than the Baseline, but with an upper quartile of over 9 and maximum value of over 12. The other models has an upper quartile and maximum value between 4 and 6, which shows that the Deep AR has more varied results than the other models.

]

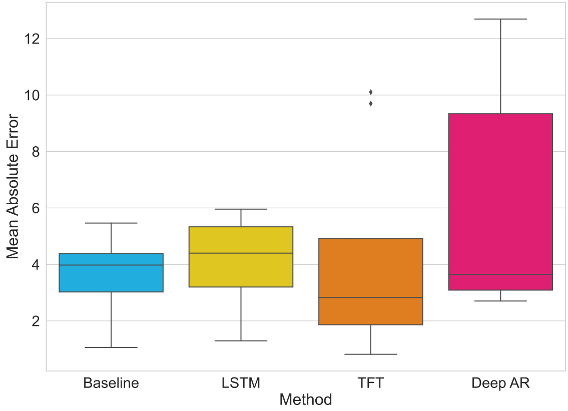


Figure 4.2: PAMAP2 dataset prediction by just heart rate: Mean Absolute Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

The MSE in Figure 4.3 shows the same results as the MAE, but in a much bigger scale. As explained in Section 3.6.2, the MSE is much more sensitive to bigger outliers, so that is what we can see here with the Deep AR model, with a maximum value of almost 250. The other models here has an upper quartile value of around 50 and maximum value of between 40 and 80. This makes a big difference from the Deep AR model.

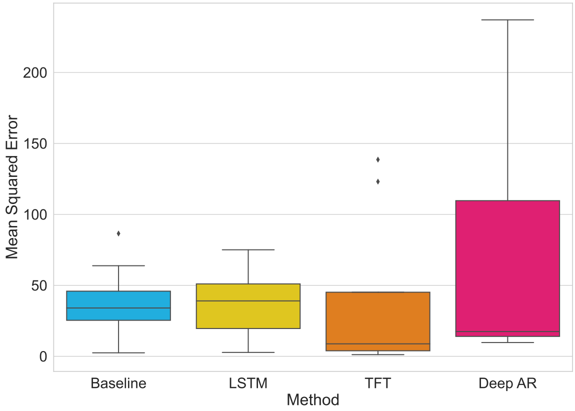


Figure 4.3: PAMAP2 dataset prediction by just heart rate: Mean Squared Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

The SMAPE plot in Figure 4.4 also draws the same conclusion as the other metrics, although this time the Baseline scores slightly better with the median value than the Deep AR. The TFT still scores the best out of all the models.

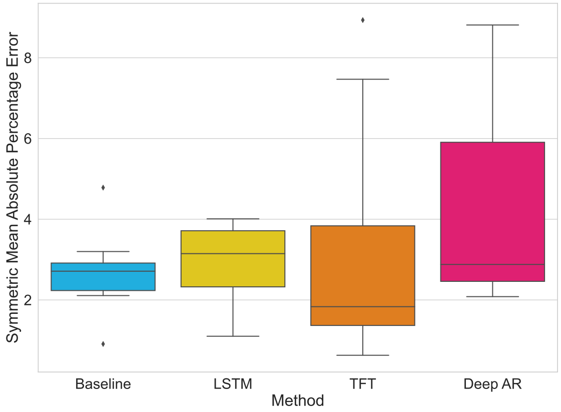


Figure 4.4: PAMAP2 dataset prediction by just heart rate: Symmetric Mean Absolute Percentage Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

4.1.2 INTROMAT

The results for the INTROMAT dataset is presented as an example plot in Figure 4.5 of a sequence with the total length of 90 seconds, in which 10 of them are predicted values. Also for the INTROMAT it seems like the Deep AR scores the best from this specific example. The LSTM has figured out the right direction of the heart rate, where as the TFT predicts the opposite direction. Additional examples of the predictions are listed in Appendix A (See Figures A.3 and A.4).

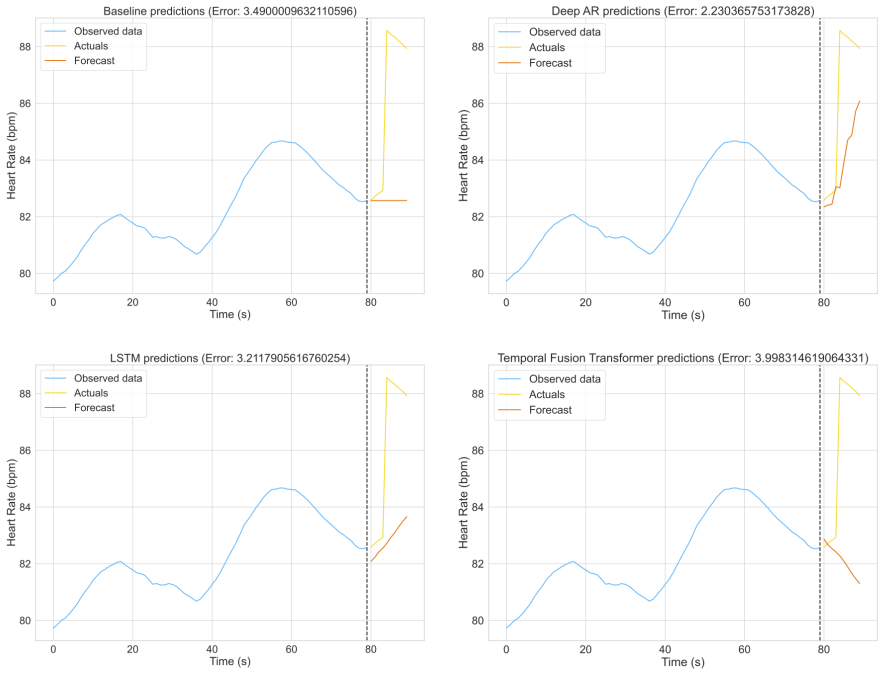


Figure 4.5: INTROMAT dataset prediction by just heart rate: Example 1. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 90 seconds, in which 10 of them are predicted values.

The MAE presented in Figure 4.6 shows that the LSTM and Deep AR model has the best median score of all the implemented models, but does not seem to score better than the Baseline. This time it seems like the TFT is the model that differs the most from the other models with bigger quartiles. TFT also has the worst median score of all the models with this metric. The plot also shows that LSTM is more stable than the other methods for this dataset of predicting the right heart rate.

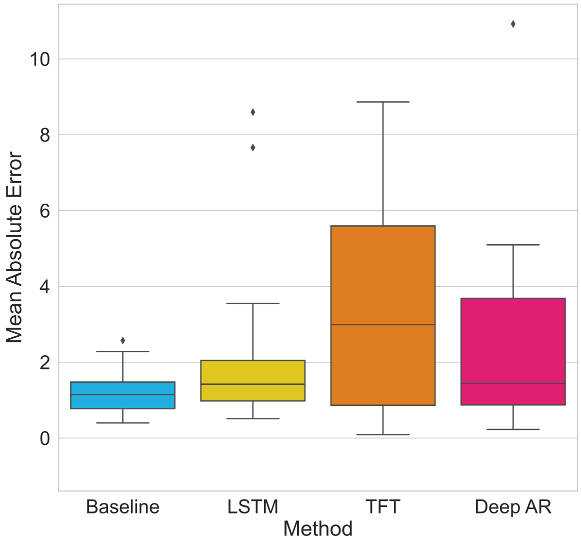


Figure 4.6: INTROMAT dataset prediction by just heart rate: Mean Absolute Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

The MSE in Figure 4.7 shows again the same results as the MAE, but in a much bigger scale, as the MSE is much more sensitive to bigger outliers. The TFT shows to have high variation in the results for the INTROMAT dataset. However this time the Deep AR scored a better median value than the LSTM, still with Baseline beating all the scores.

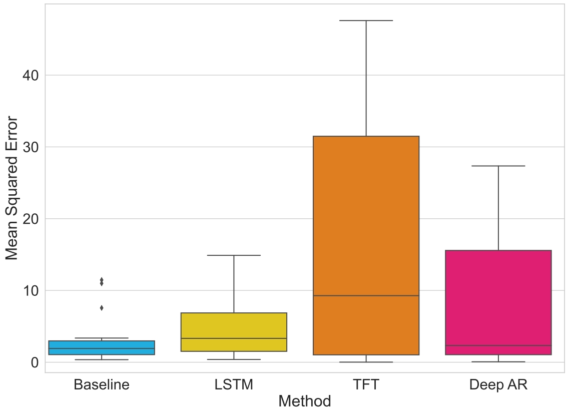


Figure 4.7: INTROMAT dataset prediction by just heart rate: Mean Squared Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

The SMAPE plot in Figure 4.8 also draws again the same conclusion as the other metrics. The Baseline yet again beats all the other models with the metric score. The median values for LSTM and Deep AR are still close, but Deep AR has more varied results with a bigger upper quartile. The TFT seems to score the worst for this metric.

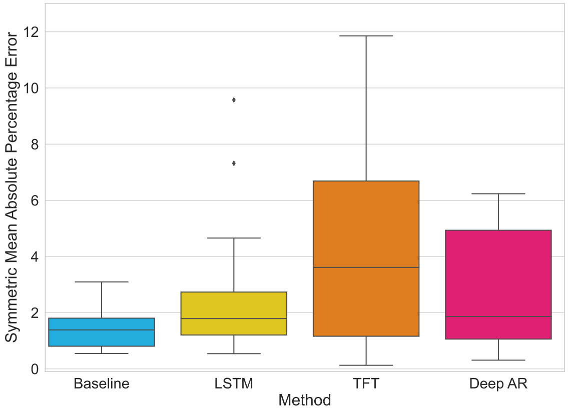


Figure 4.8: INTROMAT dataset prediction by just heart rate: Symmetric Mean Absolute Percentage Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

4.2 Experiment II: Multivariate

This experiment consists of running the Baseline model, LSTM network, TFT and Deep AR with both the heart rate and IMU data as an explainable variables for the heart rate forecast.

4.2.1 PAMAP2

Figure 4.9 shows the plot of a sequence with the total length of 100 seconds, in which 20 of them are predicted values, the same example as in Figure 4.1. This shows that just the Deep AR and TFT predicted the sudden first peak of the heart rate, but failed to predict the general upward trend. Additional examples of the predictions are listed in Appendix A (See Figures A.5 and A.6).

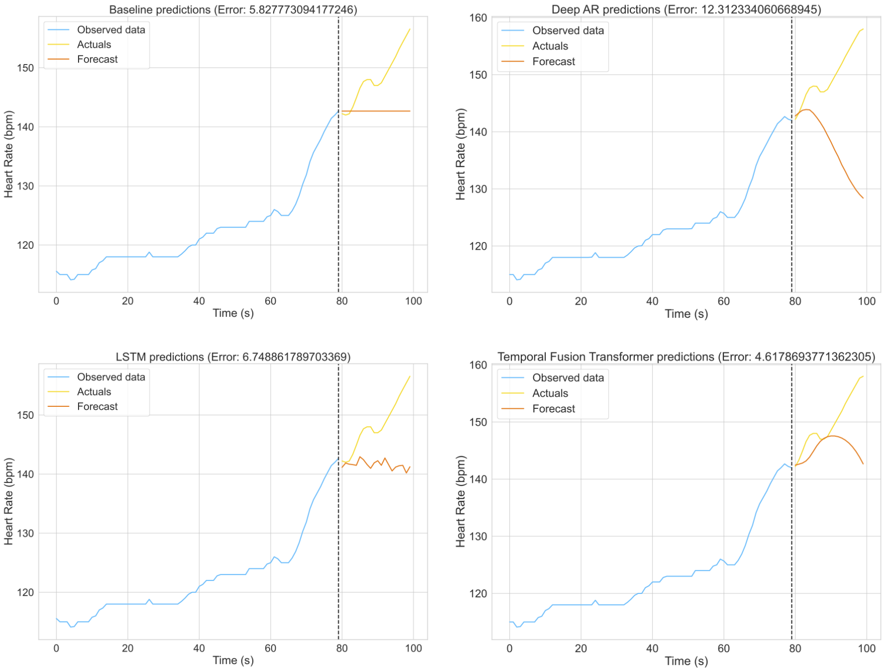


Figure 4.9: PAMAP2 dataset prediction by both heart rate and movement data: Example 1. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 100 seconds, in which 20 of them are predicted values.

The plot in Figure 4.10 shows the MAE metric, and tells us that the Deep AR here has the best median score. Although the upper quartile is the highest score of all the other models upper quartiles. The Baseline has a slightly better median score than the TFT, but the TFT also has a bigger upper quartile.

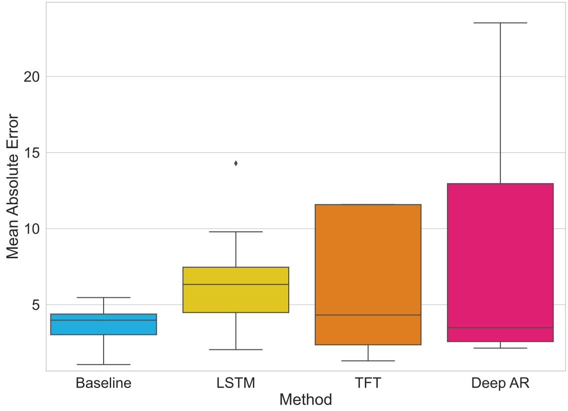


Figure 4.10: PAMAP2 dataset prediction by both heart rate and movement data: Mean Absolute Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

Figure 4.15 of the MSE of all the models shows that TFT might have bigger outliers than the Deep AR, as this metric is highly sensitive towards the bigger outliers. The upper quartile and maximum value of the TFT for this metric is almost at 300, where Deep AR is slightly behind at around 230. The LSTM and Baseline still shows little variability in their scores. The best median scores are tied between the Deep AR and TFT in this plot.

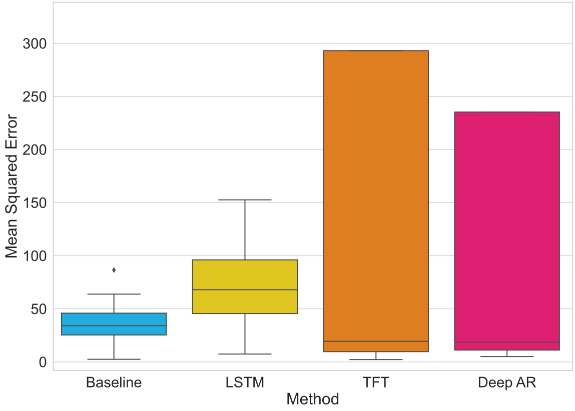


Figure 4.11: PAMAP2 dataset prediction by both heart rate and movement data: Mean Squared Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

The SMAPE metric plot in Figure 4.12 shows that the Baseline median score was the best, and the Deep AR scored slightly better than the TFT. Since the upper quartile of the Deep AR now is higher than TFT, like in the MAE plot, there seems to be higher general variability in the results for the Deep AR than in the TFT. That means that the TFT probably have bigger outliers and less general variability than the Deep AR. The LSTM had the worst median score, but the lowest in the upper quartile of the implemented models.

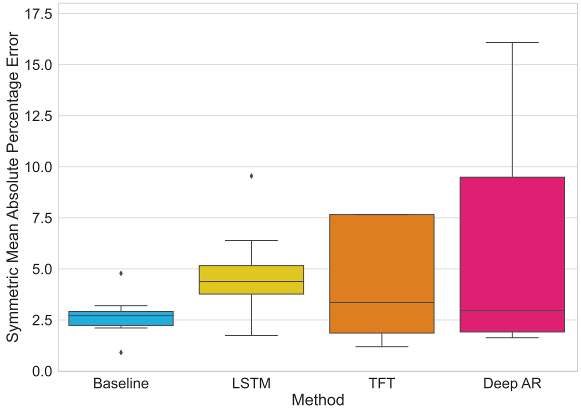


Figure 4.12: PAMAP2 dataset prediction by both heart rate and movement data: Symmetric Mean Absolute Percentage Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

4.2.2 INTROMAT

Figure 4.13 shows the plot of a sequence with the total length of 90 seconds, in which 10 of them are predicted values, the same example as in Figure 4.5. The Deep AR and LSTM in this plot seems to have predicted the right direction of the heart rate, but not the sudden big spike as the actual heart rate. As this shows the same example as in the univariate experiment the results seem similar but the Deep AR performed slightly worse in terms of just looking at this example alone. The TFT still predicts the opposite direction as it also did in the univariate version. Additional examples of the predictions are listed in Appendix A (See Figures A.7 and A.8).

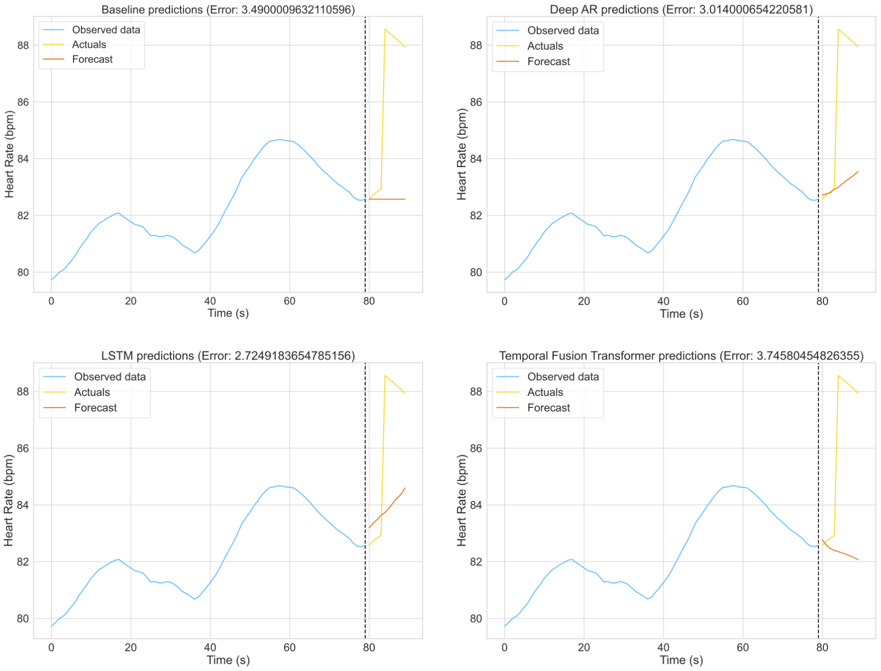


Figure 4.13: INTROMAT dataset prediction by both heart rate and movement data: Example 1. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 90 seconds, in which 10 of them are predicted values.

The MAE in Figure 4.14 shows that the Baseline model scores the best with a median score of around 1, with LSTM and Deep AR slightly behind with a median score of around 1.5. The TFT scores above 2 for the median, and also has a higher upper quartile, indicating more general varied results.

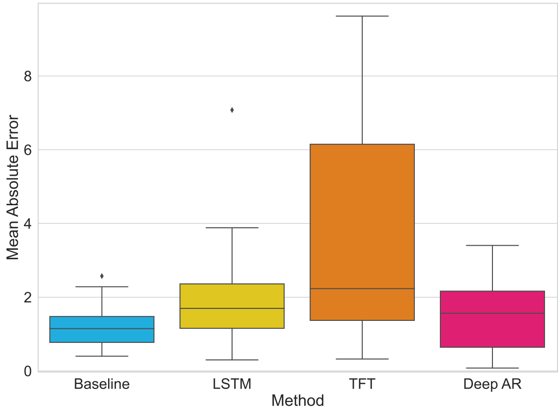


Figure 4.14: INTROMAT dataset prediction by both heart rate and movement data: Mean Absolute Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

The MSE in Figure 4.15 gives similar results as the MAE. However the median value of the Deep AR seem to be better than the LSTM. This time the LSTM har a more similar median score to the TFT. The TFT still has the most varied results, showing that it also might have some bigger outliers. The other models has a maximum value that says within a score of 30, whereas the TFT has a maximum of above 50 and an upper quartile of slightly below 40.

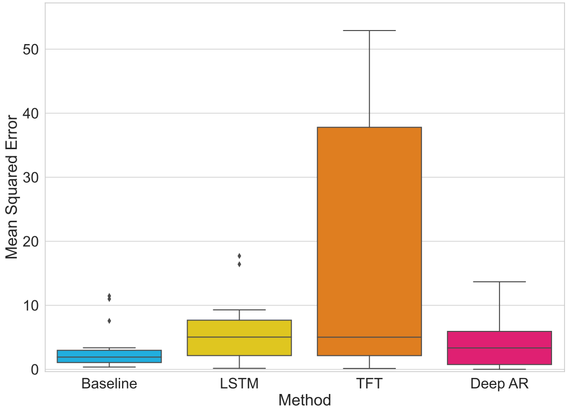


Figure 4.15: INTROMAT dataset prediction by both heart rate and movement data: Mean Squared Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

Figure 4.16 with the SMAPE results of the models shows that Deep AR scores slightly better overall than the LSTM. The Deep AR also has a low minimum score, which is really good. The TFT still shows the worst scores of all the models, with an upper quartile of above 7, where as the other models scored below 3 for this same quartile.

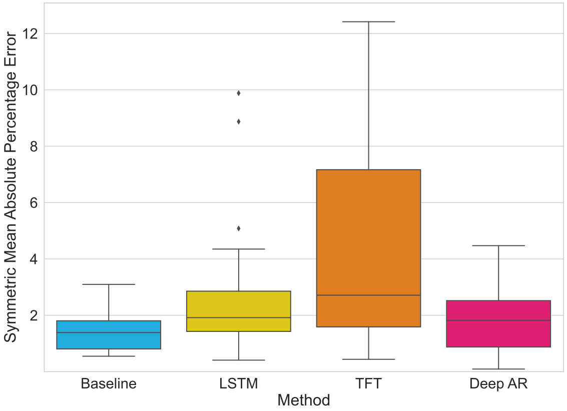


Figure 4.16: INTROMAT dataset prediction by both heart rate and movement data: Symmetric Mean Absolute Percentage Error Box plot of the Baseline model, LSTM, TFT and Deep AR.

Chapter 5

Discussion

The takeaways from all of the experiments on all the datasets is that the TFT and Deep AR seems to perform the best with the general best median scores, excluding the Baseline model. However there seems to a much higher variability of the results from the predictions. This might be because in these small time intervals (10-20 seconds) the heart rate tends to just follow one specific direction, whether it is going up or it is going down. The Baseline predicts that the heart rate just stays still since the last known heart rate value. The other models however wants to learn from more of the previous heart rates, and then often picks a direction to go for. If they pick the wrong direction, the error will be much larger than the Baseline. This can explain the larger variability for the results in the Deep AR and TFT models. The LSTM seems to have a smaller variability than the other two models, which might be because the LSTM does not predict any huge changes to the heart rate, and is performing more similarly to the Baseline because of this. ALthough in the grand scheme of things, the results from the MAE the predictions are within a 2-3 bpm difference from the actual heart rate, which shows really good results. As stated earlier, the goal is not to predict the actual heart rate, but to forecast the heart rate trend in the future. This will be more discussed in Section 5.1.

The PAMAP2 dataset has the best scores with the TFT and Deep AR model for both the univariate and the multivariate experiments. The Deep AR shows slightly better results for the median value of the metrics in the multivariate case. The TFT was the only model with a better median score than the Baseline with all metrics in the univariate case, but scored slightly worse with in the multivariate case. However this difference is not that significant.

For the INTROMAT dataset the Deep AR model seems to be generally the best scoring model for both univariate and multivariate experiments. The median scores of all the metrics of the TFT and Deep AR also seems to be slightly better with the IMU covariates than just predicting by heart rate, although the variability looks to be slightly higher for these models. The LSTM model shows the opposite with a slightly lower variability and slightly worse median value. Again not seeing substantial changes to the results.

It is challenging to compare the results for the different datasets in this current experiment because of the difference in prediction lengths. The PAMAP2 dataset predicted 20 seconds in the future and the INTROMAT only predicted 10. By having a longer prediction length, the opportunity for more error raises.

5.1 Limitations and Further Work

For several of the experiments the Baseline model scored better on the metrics of all the models. However we want the forecasting model to actually learn from past values, and not just stay the same to potentially create less error. So to test these models better, it could be more appropriate to predict the derivative of the heart rate instead, i.e predict the slope of the heart rate. This could test the models on more relevant information for this problem, but because of time limitations this was not done. It would also be great to train and test these methods on bigger datasets with even more information. This could make the models perform better by more exposure to different data. Another thing to mention is that the models could also perform better with more hyperparameter tuning.

To work further on this project, it could be an idea to generalize these models to all of the subjects so that it would adapt better to new subjects. Testing even more models on these datsets could also be something to think about. For example the Autoregressive Integrated Moving Average (ARIMA) models are good candidates to also consider working forward.

Chapter 6

Conclusion

The goal of this thesis was to facilitate for adaptation in VR exposure therapy sessions for PSA by comparing some state of the art forecasting models for heart rate prediction. As there is a lack of datasets containing heart rates in relation to mental health experiments, this comparison was tested on two different datasets, one for PSA (INTROMAT) and one for physical activity (PAMAP2).

The models tested in these experiments were an LSTM network, the TFT and Deep AR. These were tested for both univariate and multivariate cases. The univariate case focused mainly on just the heart rate, whereas the multivariate had covariates of movement data. These were all compared to the Baseline model, which just uses the last known value of the target variable in the time serie.

The comparison of the different models performed on the two datasets showed that the Baseline model scored better by the performance metrics used (the MAE, MSE and SMAPE). Although TFT and DeepAR also showed great results. The reason behind this could be that by predicting the actual heart rate, these models did not score well due to error when predicting the increase or decrease in the heart rate levels. Normally the heart rate does not change much in a small amount of time (10-20 seconds in these cases). When predicting the wrong direction of the heart rate trend the error metrics raises and creates worse results for the model. Whereas the Baseline keeps it safe by just predicting that the heart rate just stays still the entire predicted sequence. However the goal of this thesis was not to predict actual heart rate values, but to predict the trend of the heart rates in the future which the Baseline is not ideal for.

There was little change when it came to the performance of the univariate and multivariate, which might be because the data did not provide enough additional information to accurately predict the heart rate. The datasets were also relatively small, which made it harder for the models to generalize well to all the time sequences.

In conclusion these results could not provide any substantial evidence to prove that these forecasting models perform well on predicting heart rate data even with additional movement data.

Bibliography

[1] K.E. ArunKumar et al. 'Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends'. In: *Alexandria Engineering Journal* 61.10 (2022), pp. 7585–7603. ISSN: 1110-0168. DOI: https://doi.org/10.1016/j.aej.2022.01.011. URL: https://www.sciencedirect.com/science/article/pii/S1110016822000138.

[2] Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2014. DOI: 10.48550/ARXIV.1409.0473. URL: https://arxiv.org/abs/1409.0473.

[3] MM Daniels, T Palaoag and M Daniels. 'Efficacy of virtual reality in reducing fear of public speaking: A systematic review'. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 803. 1. IOP Publishing. 2020, p. 012003.

[4] Mei Dong et al. 'Deformation Prediction of Unstable Slopes Based on Real-Time Monitoring and DeepAR Model'. In: *Sensors* 21.1 (2021). ISSN: 1424-8220. DOI: 10.3390/s21010014. URL: https://www.mdpi.com/1424-8220/21/1/14.

[5] Illia Fedorin et al. 'Heart Rate Trend Forecasting during High-Intensity Interval Training Using Consumer Wearable Devices'. In: *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. MobiCom '21. New Orleans, Louisiana: Association for Computing Machinery, 2021, pp. 855–857. ISBN: 9781450383424. DOI: 10.1145/3447993.3482870. URL: https://doi-org.ezproxy.uio.no/10.1145/3447993.3482870.

[6] Thippa Reddy G. et al. 'A deep neural networks based model for uninterrupted marine environment monitoring'. In: *Computer Communications* 157 (2020), pp. 64–75. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.04.004. URL: https://www.sciencedirect.com/science/article/pii/S0140366420300542.

[7] Alex Graves. 'Long Short-Term Memory'. In: Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 37–45.

[8] Sepp Hochreiter and Jürgen Schmidhuber. 'Long Short-Term Memory'. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.

edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[9] Smiti Kahlon, Philip Lindner and T. Nordgreen. 'Virtual reality exposure therapy for adolescents with fear of public speaking: a non-randomized feasibility and pilot study'. In: *Child and Adolescent Psychiatry and Mental Health* 13 (Dec. 2019). DOI: 10.1186/s13034-019-0307-y.

[10] Dorota Kamińska et al. 'Virtual reality and its applications in education: Survey'. In: *Information* 10.10 (2019), p. 318.

[11] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep learning'. In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[12] Bryan Lim et al. 'Temporal Fusion Transformers for interpretable multi-horizon time series forecasting'. In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764.

[13] Xiangting Bernice Lin et al. 'Exposure Therapy With Personalized Real-Time Arousal Detection and Feedback to Alleviate Social Anxiety Symptoms in an Analogue Adult Sample: Pilot Proof-of-Concept Randomized Controlled Trial'. In: *JMIR Ment Health* 6.6 (June 2019), e13869. ISSN: 2368-7959. DOI: 10.2196/13869. URL: http://www.ncbi.nlm.nih.gov/pubmed/31199347.

[14] Xiangting Bernice Lin et al. 'Exposure therapy with personalized real-time arousal detection and feedback to alleviate social anxiety symptoms in an analogue adult sample: Pilot proof-of-concept randomized controlled trial'. In: *JMIR mental health* 6.6 (2019), e13869.

[15] Philip Lindner et al. 'Virtual Reality exposure therapy for public speaking anxiety in routine care: a single-subject effectiveness trial'. In: *Cognitive Behaviour Therapy* 50.1 (2021). PMID: 32870126, pp. 67–87. DOI: 10.1080/16506073.2020.1795240. eprint: https://doi.org/10.1080/16506073.2020.1795240. URL: https://doi.org/10.1080/16506073.2020.1795240.

[16] Davide Salvatore Mare, Fernando Moreira and Roberto Rossi. 'Nonstationary Z-Score measures'. In: *European Journal of Operational Research* 260.1 (2017), pp. 348–358. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2016.12.001. URL: https://www.sciencedirect.com/science/article/pii/S0377221716310207.

[17] Aleksei Mashlakov et al. 'Assessing the performance of deep learning models for multivariate probabilistic energy forecasting'. In: *Applied Energy* 285 (Mar. 2021). DOI: 10.1016/j.apenergy.2020.116405.

[18] Farzan Majeed Noori et al. 'Heart rate prediction from head movement during virtual reality treatment for social anxiety'. In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE. 2019.

[19] Theodore Oing and Julie Prescott. 'Implementations of virtual reality for anxiety-related disorders: systematic review'. In: *JMIR serious games* 6.4 (2018), e10965.

[20] David-Paul Pertaub, Mel Slater and Chris Barker. 'An experiment on public speaking anxiety in response to three different types of virtual audience'. In: *Presence* 11.1 (2002), pp. 68–78.

[21] Attila Reiss and Didier Stricker. 'Introducing a New Benchmarked Dataset for Activity Monitoring'. In: *2012 16th International Symposium on Wearable Computers*. 2012, pp. 108–109. DOI: 10.1109/ISWC.2012.13.

[22] David Salinas et al. 'DeepAR: Probabilistic forecasting with autoregressive recurrent networks'. In: *International Journal of Forecasting* 36.3 (2020), pp. 1181–1191.

[23] Abhay Srivastava and Alberto Cano. 'Analysis and forecasting of rivers pH level using deep learning'. In: *Progress in Artificial Intelligence* (Nov. 2021). DOI: 10.1007/s13748-021-00270-25.

[24] Alessio Staffini et al. 'Heart Rate Modeling and Prediction Using Autoregressive Models and Deep Learning'. In: *Sensors* 22.1 (2022). ISSN: 1424-8220. DOI: 10.3390/s22010034. URL: https://www.mdpi.com/1424-8220/22/1/34.

[25] Ivan Svetunkov and Fotios Petropoulos. 'Old dog, new tricks: a modelling view of simple moving averages'. In: *International Journal of Production Research* 56.18 (2018), pp. 6034–6047. DOI: 10.1080/00207543.2017.1380326. eprint: https://doi.org/10.1080/00207543.2017.1380326. URL: https://doi.org/10.1080/00207543.2017.1380326.

[26] Ashish Vaswani et al. 'Attention is All you Need'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Appendix A

Appendix

The figures shown in this section are additional examples for the results section.

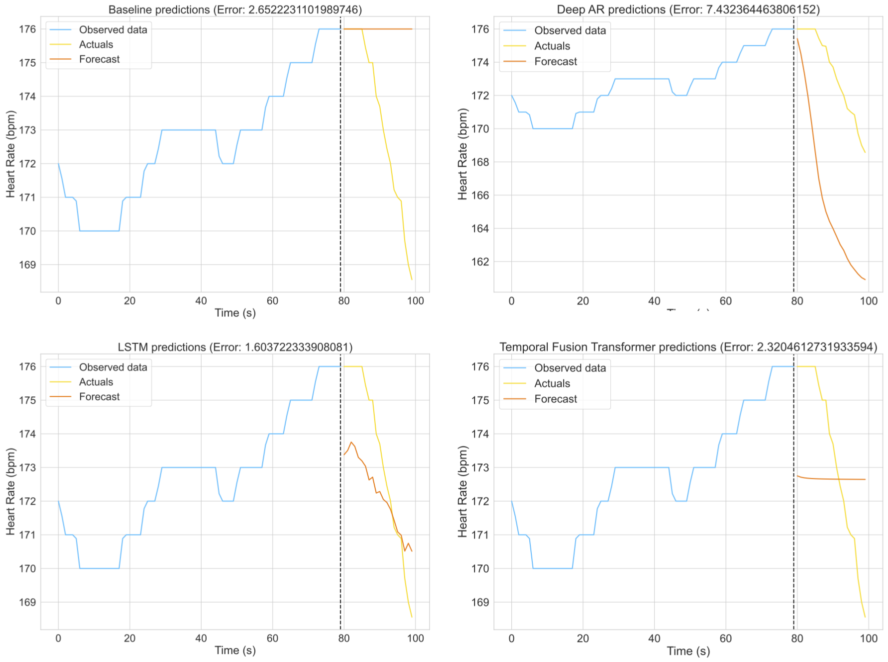


Figure A.1: PAMAP2 dataset prediction by just heart rate: Example 2. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 100 seconds, in which 20 of them are predicted values.

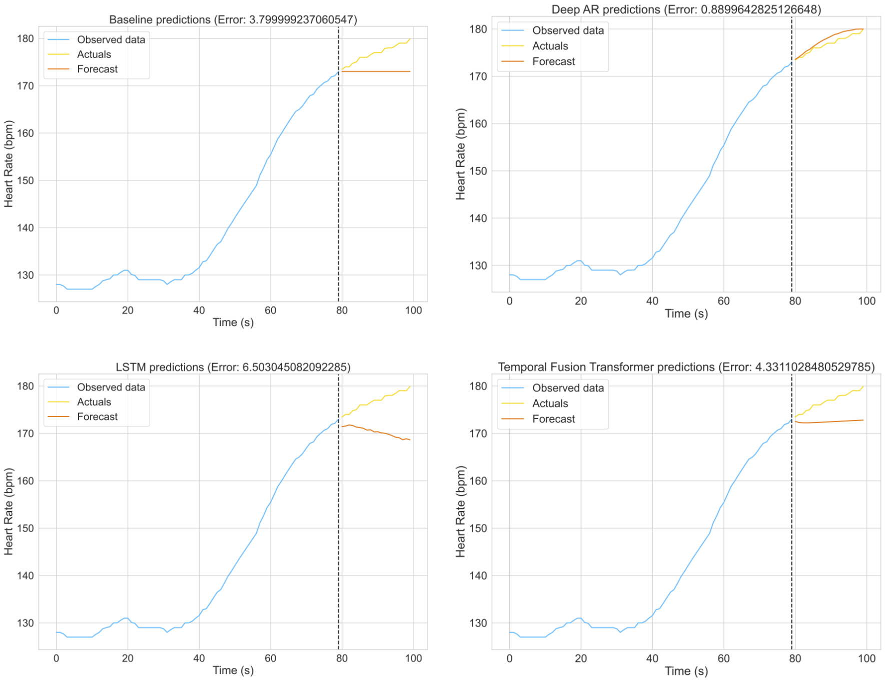


Figure A.2: PAMAP2 dataset prediction by just heart rate: Example 3. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 100 seconds, in which 20 of them are predicted values.

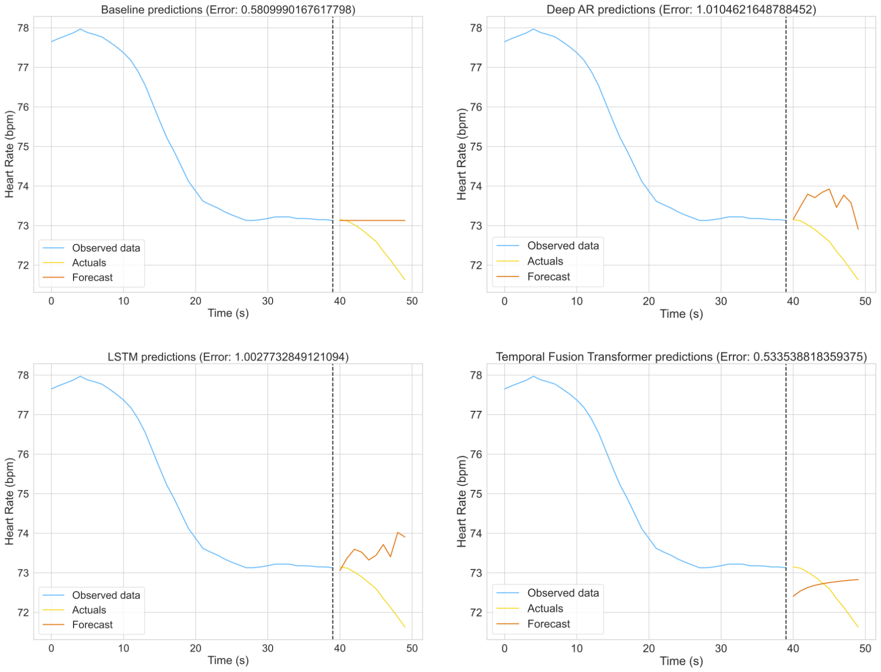


Figure A.3: INTROMAT dataset prediction by just heart rate: Example 2. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 90 seconds, in which 10 of them are predicted values.

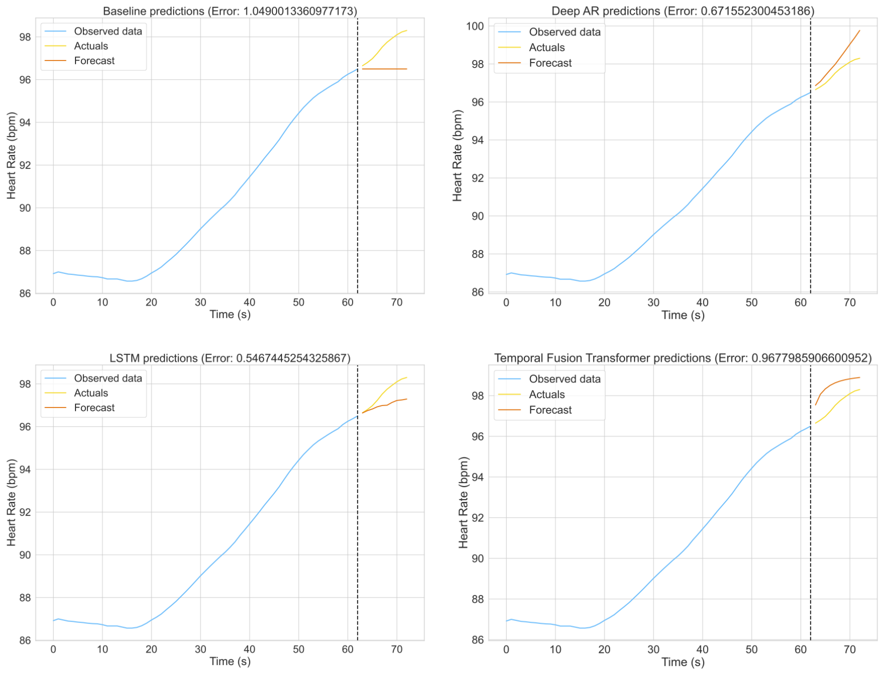


Figure A.4: INTROMAT dataset prediction by just heart rate: Example 3. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 90 seconds, in which 10 of them are predicted values.

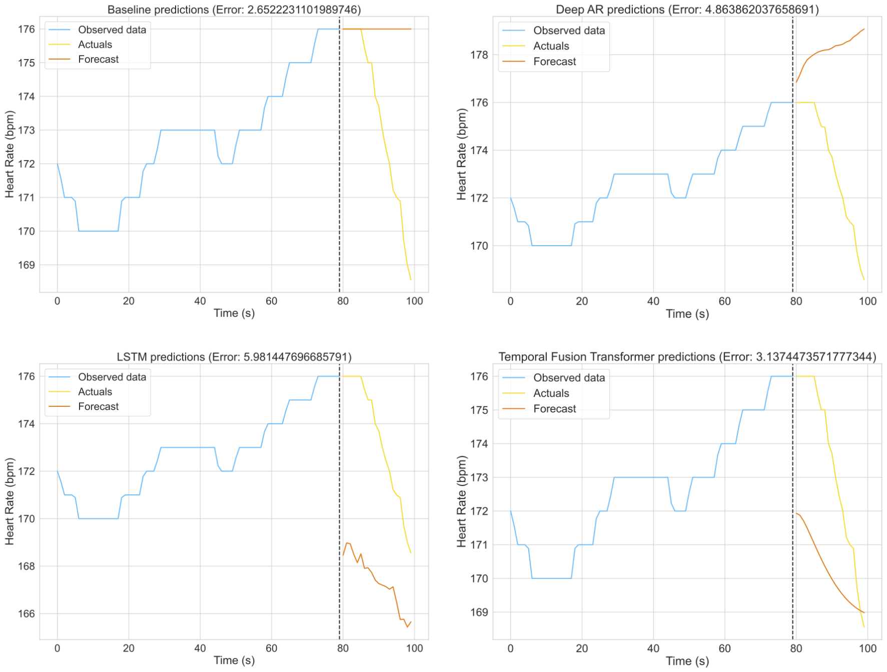


Figure A.5: PAMAP2 dataset prediction by both heart rate and movement data: Example 2. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 100 seconds, in which 20 of them are predicted values.

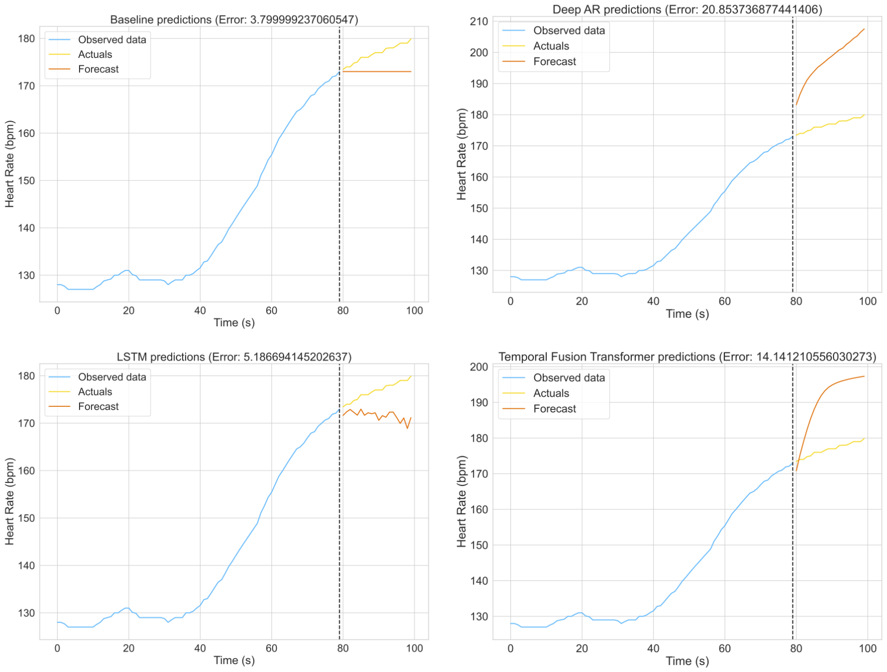


Figure A.6: PAMAP2 dataset prediction by both heart rate and movement data: Example 3. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 100 seconds, in which 20 of them are predicted values

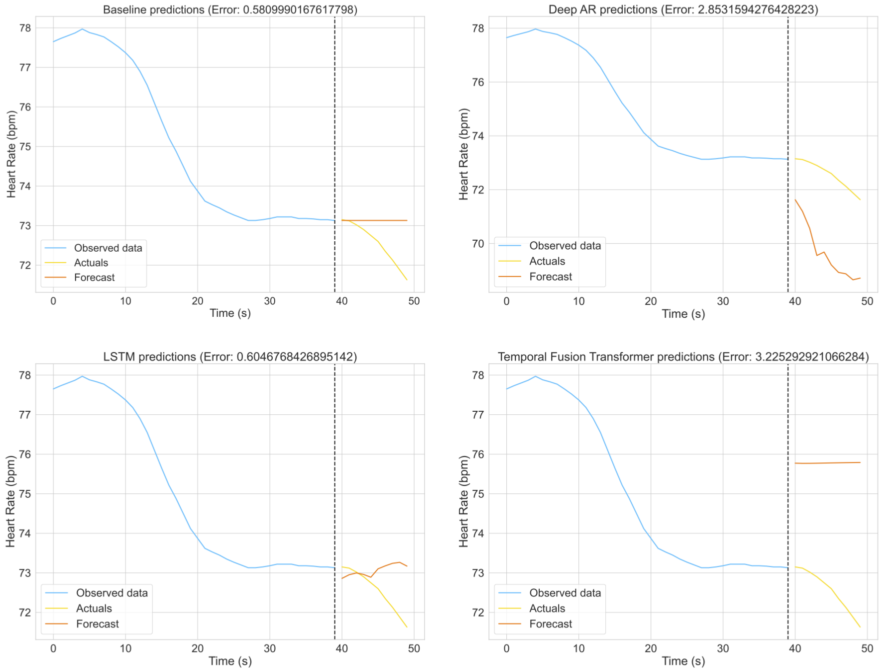


Figure A.7: INTROMAT dataset prediction by both heart rate and movement data: Example 2. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 90 seconds, in which 10 of them are predicted values

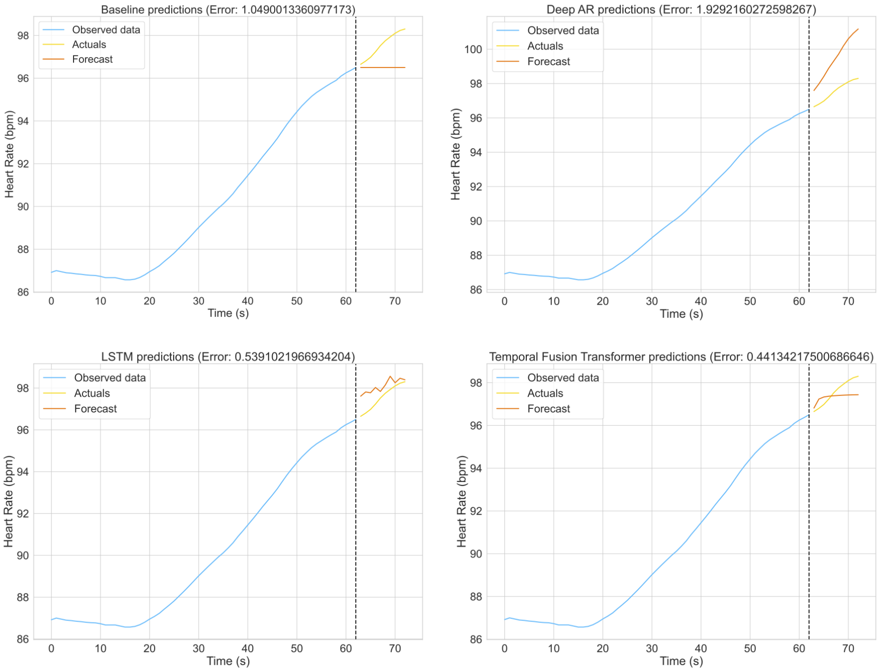


Figure A.8: INTROMAT dataset prediction by both heart rate and movement data: Example 3. Plots of observed, actual and forecast data. Top left: Baseline model, bottom left: LSTM, top right: DeepAR, bottom right: TFT. Error is measured by Mean Absolute Error. These plots shows a sequence with the total length of 100 seconds, in which 20 of them are predicted values