Utrecht University

Master Thesis

Artificial Intelligence

Adopting Time-Aware Long-Short Term Memory for Psychosis Prognosis Prediction

Author:
Maria Galanty

Supervisor:
Prof. Hugo Schnack

Daily Supervisor:
Dr. Seyed Mostafa Kia

July 22, 2022

Abstract

Schizophrenia is a complex and heterogeneous disorder because different underlying biological deficits may manifest the same symptoms across individuals. Therefore, the effective treatments can vary from one patient to another. Medicine helpful for one person, may not work or even worsen the condition of another. Individualized and accurate prediction of long-term disease course and therapy response may help navigate treatment decisions. Thus machine learning methods might be useful in treatment outcome prediction. Long-Short Term Memory (LSTM) [10] is a Recurrent Neural Network (RNN) variant capable of handling long-term event dependencies, which are common in medical data. However, it does require regular time intervals between events. In contrast to standard LSTM, Time-Aware LSTM (T-LSTM) can handle and incorporate information about irregular time intervals in data via a time decay function [3].

The goal of this study was to compare the performance of LSTM and T-LSTM models for the psychosis prognosis prediction (PPP) task, which tries to predict if a patient will be in a remission state. OPTiMiSE dataset [12], which comes from a clinical study investigating whether switching antipsychotics improves outcomes in first-episode schizophrenia patients, was used in this research. First, we checked if there was any performance difference between LSTM and T-LSTM models. Second, we investigated the effect of adjusting the T-LSTM decay function shape by learning its parameters through the backpropagation procedure. Parametric sigmoid with one and two trainable variables was used as a time decay function.

We managed to improve and obtain more stable results, while adjusting the decay function shape to the PPP application. The area under the curve (AUC) score increased from 0.65 for LSTM and T-LSTM models to 0.69 for the T-LSTM model with one trainable variable parametric sigmoid.

Acknowledgements

This work marks the end of my master's studies in Artificial Intelligence at Utrecht University. My project is a part of the research about Psychosis Prognosis Prediciton at the Pattern in Psychiatry lab at University Medical Center Utrecht. I am very grateful to have had the opportunity to be a part of this project. I would like to express my thanks to my supervisors, who agreed to guide me through this process. First, I would like to thank Mosi Kia, who acted as my daily supervisor, for his guidelines, patience and support. Second, my supervisor Hugo Schnack, I would like to extend my gratitude for the discussions and the constructive feedback.

Contents

List of abbreviations

AI Artificial Intelligence

AUC Area Under the Curve

AUROC Area Under the Receiver Operating Characteristics

CDSS Calgary Depression Scale for Schizophrenia

CGI Clinical Global Impression

FEP First Episode of Psychosis

FFNN Feed-Forward Neural Network

GAF Global Assessment of Functioning

LSTM Long-Short Term Memory

ML Machine learning

PANSS Positive and Negative Syndrome Scale

PPP Psychosis Prognosis Prediction

PSP Personal and Social Performance scale

RNN Recurrent Neural Network

ROC Receiver Operating Characteristics

SWN Subjective Well-being under Neuroleptic use

T-LSTM Time-aware Long-Short Term Memory

UKU Udvalg for Kliniske Undersogelser

1 Introduction

Schizophrenia is a chronic and complex disorder triggered by various genetic, developmental and environmental factors that disturb brain development [23]. It affects approximately 1 in 300 people worldwide and has a significant impact on patients and their surroundings. At least one-third of people with schizophrenia will obtain complete remission. While some undergo alternating periods of worsening and remission, others encounter worsening symptoms over time [31]. Psychosis is a group of clinical signs like detachment from reality, hallucinations, delusions and psychomotor anomalies, which are related but not identical to schizophrenia. This term is often used to describe the state of transition to schizophrenia, regardless of its occurrence in other disorders such as brief psychotic disorder or delusional disorders [23]. Despite a lack of recent progress in schizophrenia treatment, the understanding of the genetic and environmental causes of the disease has improved, and its relationship to neurodevelopment has become clearer [23]. Currently, the vast majority of people affected by schizophrenia are not receiving mental health care [31]. Early response to treatment for schizophrenia and related disorders is connected with a good prognosis. However, accurate prediction of disease development and choice of the appropriate treatment is still challenging for modern psychiatry. Using artificial intelligence (AI) for Psychosis prognosis prediction can make this task easier as it tries to predict if a patient will be in a remission state, which means that disease symptoms are significantly reduced. Many machine learning applications were applied to PPP tasks [16, 19, 15, 24, 27]. However, not so much research engaged information from time dimension in this process.

Long-Short Term Memory is a recurrent neural network variant capable of finding long-term dependencies in the sequence data [10]. However, it is not suitable to process longitudinal medical records, because it assumes that time intervals between events are equal. This is rarely the case in clinics and hospitals, to have equal time gaps between subsequent medical information e.g., medication intake, and blood results. In contrast to standard LSTM, Time-Aware LSTM can handle irregular time intervals in the data, which are common in medical records [3]. This architecture captures the dependencies between elements in the presence of variable time intervals through integration into the architecture, a time decay function which, based on the elapsed time between successive events, controls how much information from the previous time step can influence the current prediction. It provides the possibility to utilise the richness of medical documentation, which not only holds information about the patient's current health condition but also the timeline of the patient's disease.

1.1 Research Aims

The goal of this study is to explore the application of LSTM and T-LSTM models for PPP tasks. First, we check if there is any performance difference between LSTM and T-LSTM models and probe benchmark values for applying T-LSTM for the PPP task. Second, we test the default time decay functions used in T-LSTM. Baytas et al. [3] suggest two decay functions that can be used interchangeably depending on the amount of elapsed time. However, we think it might be more beneficial to adjust the decay function to the specific application by training models, where the function's shape is adjusted within the model training procedure using backpropagation.

1.2 Thesis Overview

We will now present an outline of this thesis. Chapter 1 gave a general introduction to the problem and described a research question. Chapter 2 provides a literature overview regarding psychosis prognosis prediction and a detailed description of LSTM and T-LSTM architectures, along with their differences. Moreover, it also includes a brief description of how the LSTM architecture evaluated from feed-forward neural networks. Chapter 3 introduces information about the dataset, methodology and experimental settings. A theoretical introduction behind using a time-decay function with LSTM is also presented in this chapter. Finally, Chapter 4 presents the results of the experiments. We conclude this thesis with discussion, strengths and limitations followed by conclusions and further work.

2 Related work

This chapter discusses work related to the research question presented in the introduction section. It covers up-to-date research on psychosis prognosis prediction and a description of the LSTM and T-LSTM architectures.

2.1 Psychosis Prognosis Prediction

Psychosis Prognosis Prediction

Schizophrenia and related diseases have heterogeneous outcomes, and their long-term courses can differ significantly between patients. A recent better understanding of causes and aberrant patterns of neurodevelopment raises hopes that disease-modifying strategies could alter the course of the disease rather than only relieve symptoms. Course-alerting intervention is especially promising around the time of the first episode of psychosis (FEP) [23]. Psychosis prognosis prediction tries to evaluate if a patient will be in a remission state, which means that disease symptoms are significantly reduced, based on the course of the disease. In contrast to diagnosis, which focuses on the current state, PPP looks into the future. During the follow-up period, many factors will influence the patient's state. Therefore PPP is a complex medical problem [11]. At a patient level, no valid prediction model for the long-term outcome of schizophrenia is available to clinicians at present [24]. The individualized long-term prediction may help improve treatment decisions.

Machine Learning Applications for Psychosis Prognosis Prediction

As a result of developed algorithms, increased availability of computational resources and collected datasets, machine learning (ML) applications have become more popular and efficient in recent years. The ability of those algorithms to handle datasets with a higher amount of variables than observations is notably useful in the medical field, where data acquisition is especially challenging due to privacy reasons [11]. Recently machine learning has emerged as a possible solution to the problem of PPP for FEP patients [16, 19, 15, 24, 27].

Koutsouleris et al. [15] tested the possibility of applying machine learning tools to predict individual treatment outcomes for FEP using only information reported by 334 patients. The goal was to predict good versus poor treatment results measured by the Global Assessment of Functioning (GAF) score (GAF $\geq$ 65 vs GAF $<$ 65, respectively) after 4 and 52 weeks of treatment. Obtained accuracy was 75% for the 4-weeks outcome and 73.8% for the 52-weeks outcome, with a decrease to 72.1% and 71.1% accordingly, when tested on geographic sites left out of the training time, concerning model generalizability to new patients. The result of this study may suggest that information obtained from the model built only on patient-reportable pre-treatment data might provide some understanding of illness trajectories, and treatment selection, which leaves an encouraging perspective for future studies.

De Nijs et al. [24] extended the use of data-driven model development based on the patients' reportable information to long-term (3 and 6 years) symptomatic and global outcomes. In this study, individualized predictions were obtained for 523 patients with psychotic disorder and variable illness duration. A support vector machine was used as a prediction model. The dataset contained baseline data and the results from the follow-up visits after 3 and 6 years. Experiments were designed to predict outcomes classified as symptomatic: in remission or not, and global outcomes, based on the GAF scale, divided into good (GAF $\geq$ 65) and poor (GAF $<$ 65). Accuracy varied between 62.2%

to 67.6% for both settings. The authors of this work emphasise that although a reasonable level of accuracy was achieved by the model, it is not suitable as a stand-alone tool to stratify treatment. The clinician's judgement is more globally constructed and includes data from the moment of the first appointment with the patient. In clinical practice, decisions are usually based on long-term information rather than on a single examination. The next natural step in psychosis prognosis prediction is exploring the models, which can predict based on longitudinal data rather than from baseline data.

Deep Learning Applications for Patients Undergoing Psychosis

Diverse neural network architectures were applied to answer research questions regarding detecting first-episode psychosis, predicting clinical improvement in psychosis and identifying patients at risk of psychosis [1, 28, 20, 29, 5, 18]. However, those models are not longitudinally informed. De Nijs et al. [24] suggested that models capable of processing long-term information might be more suitable for many medical applications. Patients suffering from various medical conditions usually have extensive data gathered about their health. Recurrent Neural Networks can handle time-series data from multiple data modalities simultaneously to forecast individual trajectories. Koppe et al. [14] explore the usage of RNNs in mobile sampling and intervention. They suggest that data coming from wearable devices, such as smartphones, fitness trackers, or smartwatches have a great potential in psychosis prediction prediction studies and can offer online feedback intervention. This work is only theoretical and does not offer any models or benchmarks. Raket et al. [26] built a dynamic electronic health record detection model (DETECT), which detects an individual risk of developing first-episode psychosis based on the electronic health records. The authors claim that early warning would allow for better preventive interventions.

2.2 Long-Short Term Memory

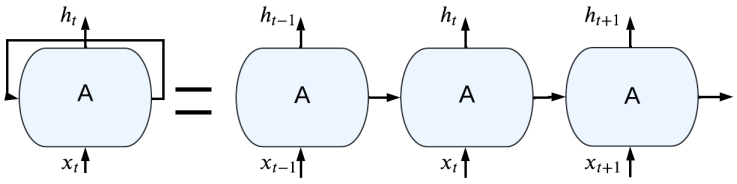


Figure 1: The repeating module in an RNN architecture. Each cell takes one input record denoted by the letter x and produce one output h, while $t-1$, t and $t+1$ in the lower index reflects the time step.

A feed-forward neural network (FFNN) allows signals to travel only one way from input to output. However, this architecture is not suitable for processing sequential data. Recurrent neural networks introduce loops (Figure 1) in the structure that enables the model to process data sequentially. This chain-like architecture can be seen as a multiple copies of the same network, where each cell is passing information to the successor. Data can enter the network sequentially, and information about earlier events can travel to the later ones. Moreover, the network's internal

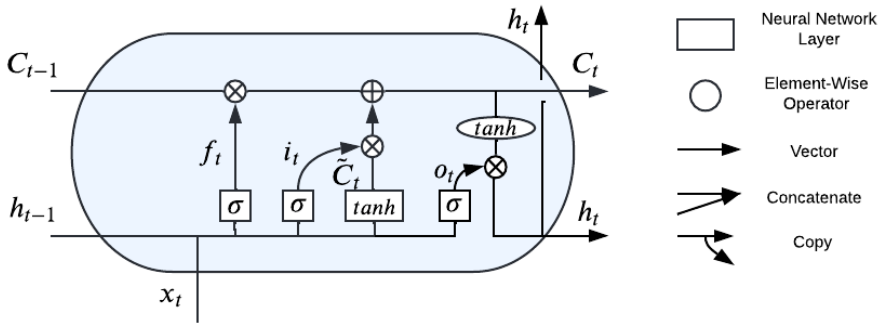
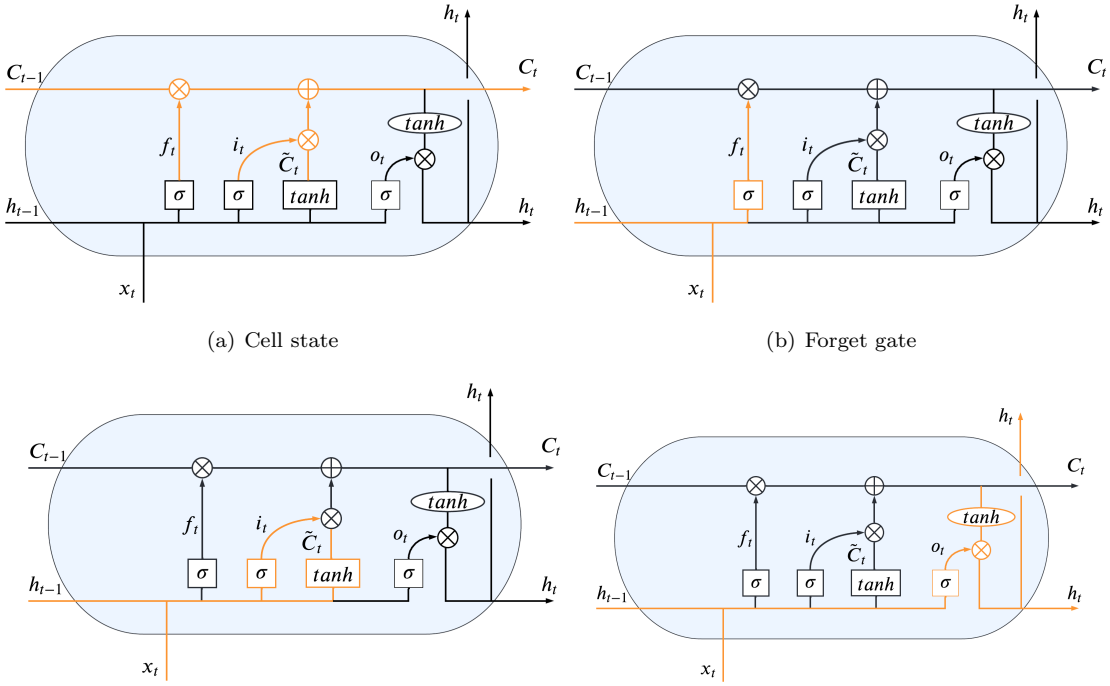


Figure 2: Illustration of the LSTM cell architecture. Single LSTM cell takes one input record denoted by the letter x and produce one output h, while t in the lower index reflects the time step.

memory can keep information from the previous hidden states. This design allows information to travel from one time step to the next and discover dependencies between different time points. RNNs are powerful architecture and were successfully used in many applications such as speech recognition [9, 8] and text classifications [17]. However, due to the vanishing gradient problem, information can travel only a limited number of times. This kind of network does not work well with long-term dependencies. In 1997 Hochreiter and Schmidhuber introduced Long-Short Term Memory, which is a network architecture that solves this issue by introducing three gates (input, forget and output) that regulate the flow of information into and out of every LSTM cell [10]. In the traditional RNN, each module has a simple structure, while in LSTM each repeated module has a more complex architecture (Figure 2), consisting of four neural network layers that interact with each other in a certain way.

The core idea behind the LSTM cell is a cell state C, the top horizontal line going through the cell. In Figure 3a it is visible that cell state is only slightly adjusted in each cell through linear interaction with the gate structures. Each of the three gates is responsible for a different task regarding the information stored in the cell state: one decides what to forget, the next one decides what to add, and the last one decides about the output of the current cell. Each gate consists of a sigmoid neural network layer and a element-wise multiplication operation. Below a step-by-stem walk through the LSTM cell is provided along with the mathematical equations.

1. First, the **forget gate** highlighted in Figure 3b decides which information from the previous cell state to forget. This decision is based on current input x_t and the hidden state from the previous cell h_{t-1}. The *sigmoid* layer produces a number between 0 and 1 for each number in the previous cell state C_{t-1}. Zero value implies that everything should be forgotten, while one keeps everything.
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
2. Next, information will be added to our cell state. Based on the previous hidden state and new data **input gate** (Figure 3c) decides which values from the previous cell state should be updated.



(a) Cell state

(b) Forget gate

(c) Input gate

(d) Output gate

Figure 3: Different components of the LSTM architecture.

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

3. Using the same information a *tangent hyperbolic (tanh)* layer creates a new state candidate $\tilde{C}_t$.

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

4. Finally, the previous cell state C_{t-1} and the new cell state candidate $\tilde{C}_t$ are combined to create a new cell state C_t.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

5. The last step is to decide what will be the cell output h_t, which can be seen as a filtered version of our cell state. An **output gate** (Figure 3d) chooses which parts of the cell state will be present in the h_t.

$$o_t = \sigma(W_o \cdot [h_{t_1}, x_t] + b_o)$$

6. Next, the cell state values are squeezed between -1 and 1 by the *tanh* and multiplied with the output gate vector. This h_t will be used both as a hidden state for the next LSTM cell and as the output for this time step.

$$h_t = o_t * tanh(C_t)$$

x_t represents the current input, h_t and h_{t-1} are current and previous hidden states, while

C_t and C_{t-1} denotes the current and previous cell memories. $\{W_f, b_f\}$, $\{W_i, b_i\}$, $\{W_c, b_c\}$ and $\{W_o, b_o\}$ are the network weights and biases of the forget gate, input gate, candidate state and output gate respectively. Those parameters are learned through backpropagation, their dimensions are determined by the input, output and the number of neurons in the unit.

The greatest limitation of the LSTM cell is an underlying assumption of uniformly distributed elapsed time between the elements of a data sequence. Therefore, the variable time intervals, which can be present in medical records, are not integrated into standard LSTM.

2.3 Time-Aware Long-Short Term Memory

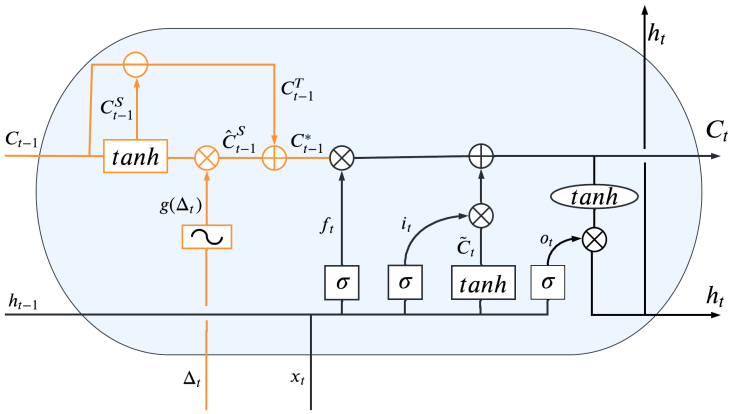


Figure 4: Illustration of the T-LSTM cell architecture. Components that are not present in the standard LSTM architecture are highlighted in orange.

LSTM architecture was designed to handle sequential data with long-term dependencies. It assumes that the collected data are uniformly distrubuted over time. That means the elapsed time between succeeding data instances is presumed to be the same throughout the dataset. However, if we think about medical records, it seems natural that the time intervals between admissions, tests and doctor appointments are irregular. Moreover, information about the timeline might indicate the patient's state. For example, frequent admissions might imply serious health concerns. In the standard LSTM architecture, this information about the varying time gap between subsequent records is not and cannot be used while processing the data, which may lead to suboptimal performance. Baytas et al. [3]introduced Time-aware LSTM, which handles those irregularities in longitudinal patient records. The idea behind T-LSTM is simple. The more time elapsed between consecutive records, the smaller should be the dependency on the previous data to predict the current outcome. Let's say that patients had their last blood test two years ago. The result of this examination should not influence the current diagnosis in a meaningful way. However, information about the patient's age or family history is still crucial. In a T-LSTM, the cell state is decomposed into long-term and short-term memory, which enables the time decay function to discount the short-term memory content according to the elapsed time, while long-term remains the same. Now, we will describe the T-LSTM architecture in detail.

The difference between T-LSTM and LSTM architectures is illustrated in Figure 4. Black elements are standard LSTM components, while orange lines are the T-LSTM additional parts. Apart from the data input t, T-LSTM also receives information on the elapsed time between current and previous records, indicated by the Δ_t. Below a step-by-stem walk through the T-LSTM cell is presented along with the mathematical equations.

1. First, the time decay function $g(\Delta_t)$ will transform the elapsed time into an appropriate weight that will discount our short-term memory in the following steps (Figure 5a).
$$g(\Delta_t)$$

2. Next, previous cell state needs to be decomposed C_{t-1} into **long-term** and **short-term memory**. Short-term memory C^S_{t-1} is obtained with the use of data-driven decomposition (Figure 5b). Parameters of this decomposition are part of the neural network weights and are learned through backpropagation.
$$C^S_{t-1} = tanh(W_d C_{t-1} + b_d)$$

3. Long-term memory C^T_{t-1} is obtained by subtracting the short-term memory values from the cell state(Figure 5c).
$$C^T_{t-1} = C_{t-1} - C^S_{t-1}$$

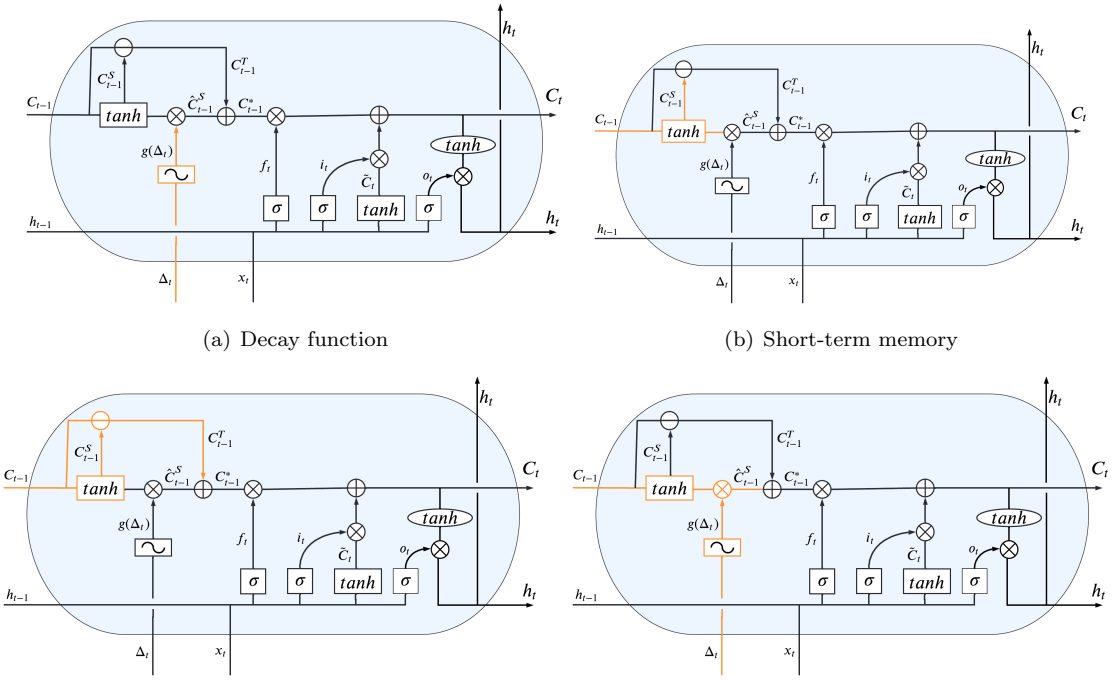
4. Afterward, $g(\Delta_t)$ is multiplied with short-term memory and discounted short-term memory C^T_{t-1} is reached (Figure 5d).
$$\hat{C}^S_{t-1} = C^S_{t-1} * g(\Delta_t)$$

5. Finally, the value of discounted short-term and long-term memories are combined to get the adjusted previous cell state C^*_{t-1}.
$$C^*_{t-1} = C^T_{t-1} + \hat{C}^S_{t-1}$$

6. The rest of the architecture remains the same as in LSTM, but the adjusted previous cell state C^*_{t-1} is used instead of the previous cell state C_{t-1}.
$$\begin{gathered} f_t = \sigma(W_f \cdot [h_{t_1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t_1}, x_t] + b_i) \\ \tilde{C}_t = tanh(W_C \cdot [h_{t_1}, x_t] + b_C) \\ C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t = \sigma(W_o \cdot [h_{t_1}, x_t] + b_o) \\ h_t = o_t * tanh(C_t) \end{gathered}$$

Parameters shared with the LSTM network denote the same meaning and were described in the previous subsection. $\{W_d, b_d\}$ are the network parameters of the subspace decomposition. Their values are learned through backpropagation.

$g(\Delta_t)$ is a heuristic decay function such that the larger the value of Δ_t, the lower the value of $g(\Delta_t)$. Therefore, the lower effect of the discount short-term memory on the adjusted cell state. Bytas et al. [3] suggest that different types of monotonically non-increasing functions can be chosen for g accordingly to the time gaps and measurements. In some domains, the elapsed time can vary from days to years, and there is a need to convert data to one type, such as days. For those reasons they recommend $g(\Delta_t) = 1/\Delta_t$ when there are small gaps of elapsed time, and $g(\Delta_t) = 1/log(e+\Delta_t)$ otherwise. Figure 6 shows both function. However, they indicate those functions as guidelines and suggest trying different kinds. In section 3.2 we will come back to this issue.



(a) Decay function

(b) Short-term memory

(c) Long-term memory

(d) Discounted short-term memory

Figure 5: Different components of the T-LSTM architecture.

3 Methodology

Chapter 3 introduces the dataset and methods used to answer the research questions.

3.1 Data

The OPTiMiSE dataset that is used in this project comes from a clinical study investigating whether switching antipsychotics (amisulpride and olanzapine) or early use of clozapine improves outcomes in first-episode schizophrenia patients [12]. It is a common practice that if the patients are not responding well to the initial treatment, they will be switched to another antipsychotic medication. Clozapine is an effective antipsychotic drug. However, due to the increased risk of side effects, it is not advised to be used as the first treatment choice, but rather when two other drugs have been insufficient. There are no established treatment algorithms for schizophrenia patients and no research evidence that the current practice is an effective way to improve the patient's well-being.

Twenty seven medical facilities in fourteen European countries and Israel were involved in this clinical study. During the first phase patients, who met the initial criteria, were prescribed amisulpride for 4 weeks. Only patients not in remission could continue to a 6-week double-blind second phase, during which participants were randomly assigned to continue amisulpride or switch to olanzapine. Participants not in remission after the first and second phases were given clozapine for an additional 12 weeks in an open-label design (phase three). Initially, 481 patients were assessed for eligibility and signed informed consent, 371 completed the first phase, 72 completed the second phase and 18 completed the third phase. Baseline data were collected before the study began. It included demographics, diagnoses, current treatment setting, and alcohol and drug use. Additionally, several psychological tests and scales were carried out: Positive and Negative Syndrome Scale (PANSS), Clinical Global Impression (CGI), Calgary Depression Scale for Schizophrenia (CDSS), Personal and Social Performance scale (PSP), Subjective Well-being under Neuroleptic use (SWN), Udvalg for Kliniske Undersogelser (UKU). The purpose of these scales is to measure symptom severity and the patient's well-being. Some were designed exclusively for patients with schizophrenia, while others can be used with various mental disorders. During the study, patients came for control visits after weeks 1, 2, 4, 6, 8, and 10-22. Those meetings involved physical examination, interviews about hospitalisation, concomitant medication, recreational drug use, alcohol use, caffeine use, and smoking. Moreover, selected psychological questionnaires and scales were repeated.

The outcome of this study shows it is not beneficial to switch antipsychotic medication. If a patient does not respond to the initial treatment, it is favourable to prescribe clozapine instead of trying another antipsychotic.

Baseline data (the first visit), along with PANSS, PSP and CGI questionnaires information from the first two phases of the clinical study (visits 2, 3, 5, 6, 8), were used to answer research questions regarding time decay functions in the T-LSTM model for psychosis prognosis prediction. Data from phase three were not employed because of very small sample size in the third phase of the study (18 samples). Information from visit 8 will serve as our target data. We will determine if the patient is in a symptomatic remission during the eighth visit, based on Andreasen et al. criteria [2], the score on chosen 8 PANSS items (P1, P2, P3, N1, N4, N6, G5 and G9) needs to be lower than 3, while the scales range from 1 to 7.

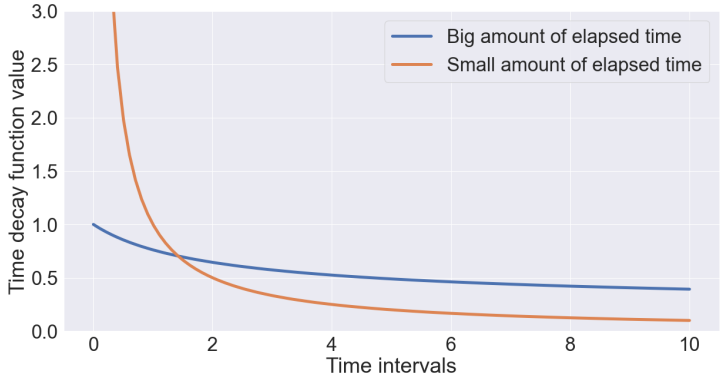


Figure 6: Decay functions proposed by the Baytas et al. [3]

3.2 The proposed method: An adaptive time decay function

T-LSTM architecture was proposed to handle variable time intervals in the data. The time decay function discounts the memory cell in the way that the longer the elapsed time, the smaller effect of the previous cell state on the current output. Baytas et al. [3] described decay function as a monotonically non-increasing discount, that the larger the value of Δ_t the smaller the value of the function $g(\Delta_t)$. The authors suggested two functions that can be used depending on the elapsed time. $g(\Delta_t) = 1/\Delta_t$ can be used with applications where time gaps are small and $g(\Delta_t) = 1/log(e + \Delta_t)$ otherwise. Both functions are convex and presented in Figure 6.

However, the expectation that this function should be monotonically non-increasing does not need to be valid for all the applications. In some cases, information from the past might be more stable than the current input. Moreover, the assumption about choosing a time decay function based on the amount of elapsed time seems not justified enough. Instead, the dynamical laws of the problems should dictate the shape of the function. Finally, discussed functions were proposed only as guidelines, and Baytas et al. [3] suggest that different functions also can be used. For those reasons, we would like to explore the option of adjusting the time decay function to the concrete application by tuning the function parameters in a data-driven manner, by learning them using backpropagation in the training procedure. We will compare performance of two architectures, where the time decay function parameters are learned using backpropagation with the results obtain for standard LSTM model and T-LSTM with $g(\Delta_t) = 1/log(e + \Delta_t)$ decay function.

Functions proposed by Baytas et al. [3] are convex, and their values decrease fast. Because the change in the behaviour of psychiatric patients for the PPP applications often does not occur quickly, we want a function, which can decrease slowly. A sigmoid function has a characteristic shape similar to the letter "S". Parametric sigmoid function $g(\Delta_t) = -\frac{1}{1+exp(b-a*\Delta_t)} + 1$ with two trainable variables a, b will be used as a decay function for this application. Parameter a controls the slope, while b determines the bending moment. To search for the best function form, we will try to learn a and b using backpropagation during the network training. This parametric sigmoid function is chosen for the following reasons. First, with these settings, it is possible to obtain function form with only a small value decrease or even no decline. Second, the chosen function is differentiable, which is required because its parameters will be adjusted using error backpropagation. Moreover,

the proposed parametric sigmoid takes values between 0 and 1 (for positive real numbers), which is necessary since they serve as a discount for short-term memory. Lastly, we do not know what function form is the best for the PPP task, and the sigmoid function is commonly used for natural science problems, including applications where specific mathematical models are unknown [6].

Non-increasing parametric sigmoid function $g(\Delta_t) = -\frac{1}{1+exp(5-a*\Delta_t)} + 1$ with one trainable variable a and b value fixed to 5, will be used in the first experiment. The parameter a controls the function slope, how quickly the function obtains a zero value. In Figure 7a, we can see how different this function looks depending on the a value. In these settings, because b value is fixed to 5, the function will always take a value of one at zero. Considering that the discount is based on the amount of the elapsed time, there should be no discount if no time has passed. It is also visible that the closer the a value is to zero, the closer function is to being constant. When the function is constantly equal to one, no discount is applied, and T-LSTM architecture turns into LSTM.

Theorem 1. *Let g be a decay function for T-LSTM model. If g is a constant function that takes the value one, then this T-LSTM model turns into an LSTM model, and the adjusted previous cell state is equal to the previous cell state.*

Proof.

$$C^S_{t-1} = tanh(W_d C_{t-1} + b_d) \quad \text{Short-term memory}$$
$$g(\Delta_t) = 1 \quad \text{Decay function is constant and equal to 1}$$
$$\hat{C}^S_{t-1} = C^S_{t-1} * g(\Delta_t) \quad \text{Discounted short-term memory}$$
$$\hat{C}^S_{t-1} = C^S_{t-1} * 1$$
$$\hat{C}^S_{t-1} = C^S_{t-1} \quad \text{Discounted short-term memory is equal to short-term memory}$$
$$C^T_{t-1} = C_{t-1} - C^S_{t-1} \quad \text{Long-term memory}$$
$$C^*_{t-1} = C^T_{t-1} + C^S_{t-1} \quad \text{Adjusted previous cell state}$$
$$C^*_{t-1} = C_{t-1} - C^S_{t-1} + C^S_{t-1}$$
$$C^*_{t-1} = C_{t-1} \quad \text{Adjusted previous cell state is equal to previous cell state}$$

□

During the second experiment, the time decay function has more freedom to adjust its shape. $g(\Delta_t) = -\frac{1}{1+exp(b-a*\Delta_t)} + 1$, where a and b are trainable variables. As previously, a controls the slope. The smaller the value of parameter a, the less steep the function will be. Variable b determines the bending moment and where the function will cross the y axis. The lower the parameter value, the lower the function will cut the axis. In Figure 7b, we can see how different this function can look depending on the a and b values. As previously mentioned, the information from the past may be more stable than the current input. If a patient says that he feels fine during today's meeting, but his condition was very disturbing two weeks ago then perhaps the information from the past is more important in assessing his future health. That is why we allow the function to be ascending. We still want the discount values to be between zero and one, which forces us to abandon the assumption that the function should take the value one in zero. Through those experiments, we would like to observe which decay function performs better in our PPP application.

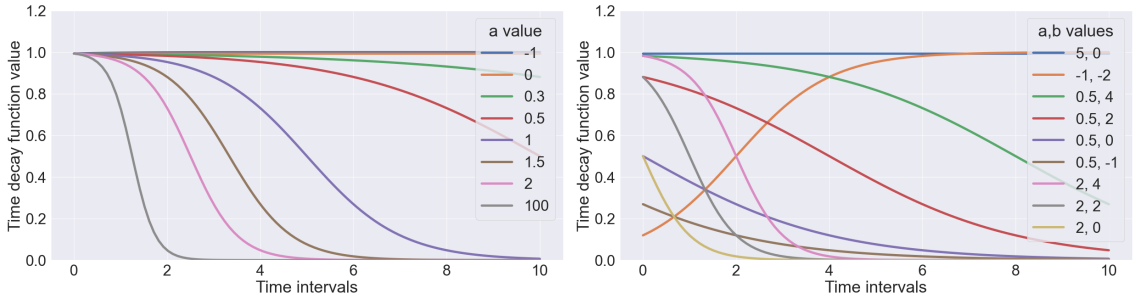


Figure 7: Various trajectories of the time decay function with (a) one parameter (b) two parameters.

3.3 Training procedure and model architecture

Data preprocessing

As previously mentioned, data from the two first phases of OPTiMiSE dataset are used in our experiments [12], including static (the features that do not change over time e.g., demographic information) and dynamic features (data that change over time e.g., weight or answers from psychological questionnaires) from 72 patients. During data preprocessing, patients' records with missing PANSS values were entirely removed, because computing remission labels was not possible for them. Consequently, the number of patients used in our experiment decreased to 66 (6 were removed). Missing values of other attributes were imputed. For continuous variables, we use median, and for binary and categorical features, the most frequent value and k-nearest neighbors imputation techniques were applied [25]. Finally, continuous data were standardised, binary features were processed by 0/1 encoding, and categorical variables were transformed with one-hot encoding.

Neural Network Architecture

We used the architecture in Figure 8 in our experiments. This is a multi-modal and multi-task architecture that can handle several static and dynamic input modalities and can predict several outcomes at the same time, thus suits perfectly the data and purpose in the PPP application. The model has four modules, including the representation learning layer, fusion layer, interaction layer and output layer. The only module that changes in our experiment is the dynamic module, where either LSTM units or T-LSTM units with different choices for the decay function are employed. We implemented both architectures [25] in Keras with Tensorflow as the backend. [4, 22].

In Figure 8 network with LSTM representation layers is presented. Each assessment scale (PANSS, PSP and CGI) and static information have separate input layers, where the number of neurons is based on the input feature size. LSTM layers, responsible for representation learning, are connected to their assessment input layer. The number of neurons in LSTM layers is fixed to 50, 10 and 5, respectively. Subsequently, the fusion layer merges the information from dynamic and static input layers. This architecture has two output modules including regression and classification. Regression modules with interaction and output layers predict PANSS, PSP, and CGI values for the next time step. This information is fed into the model along with the subsequent input data. Finally, the classification module with a softmax activation layer predicts if a patient is in symptomatic

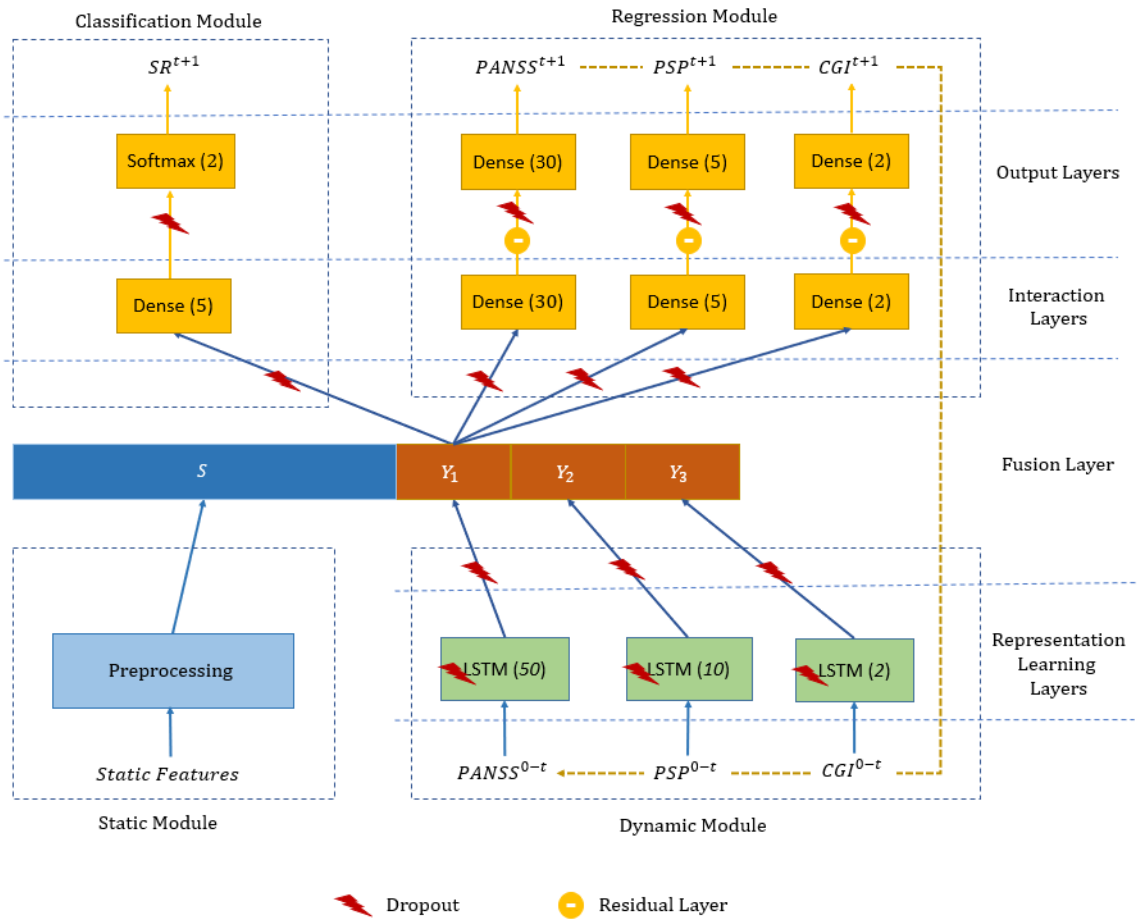


Figure 8: Model of neural network architecture used for psychosis prognosis prediction [25].

remission (SR) or not. This outcome we use to assess the performance of our models.

Red lightning in Figure 8 denotes dropout regularisation with the probability of 0.1. An Adam optimizer [13] with an exponential decay learning rate, which lowers the learning rate as the training progresses with the initial value of 0.0003, and a decay rate of 0.9 is used to minimize the total categorical cross-entropy loss across classification layers.

The described model with LSTM layers has in total 30 746 trainable parameters. In our experiments, we used four architectures, which differ only in the representation learning module. LSTM layers are replaced with T-LSTM layers with various decay functions, which changes the number of trainable parameters in the architecture. The corresponding values are presented in Table 1. There is a significant difference in parameters number between the LSTM and other models. The only difference between T-LSTM models is the number of trainable parameters used inside the decay function.

Training Setting

To obtain stable results regarding the model performances, each experiment is repeated ten times with a different, fixed random seed to make the replication possible. For each repetition, we use k-fold cross-validation with a fold number equal to 5, to obtain the most credible performance results by training and testing on different parts of the dataset. The number of epochs is fixed to 50 with a batch size equal to 1. All the weights inside the models are initialised with a glorot uniform [7]. Due to the low number of data instances, the model is first pretrained with 10000 samples of the random simulated data for two epochs, with a batch size equal to 25. Simulated data are created from a stochastic process, where a random value is generated for each variable based on the minimum and maximum feature value.

Table 1: The number of trainable parameters that were used for different models in LSTM and T-LSTM layers to process information from PANSS, PSP and CGI assesments.

Model	Trainable parameters in LSTM and T-LSTM layers		
	PANSS	PSP	CGI
LSTM	16 200	640	160
T-LSTM with $g(\Delta_t) = 1/log(e + \Delta_t)$	18 750	750	190
T-LSTM with one variable parametric sigmoid	18 751	751	191
T-LSTM with two variables parametric sigmoid	18 752	752	192

4 Results

The following chapter presents experiment results, which will be further discussed in the next section.

4.1 Experiments summary

The main results of our experiment, the performances of the different (T-)LSTM approaches, are shown in Table 2. The area under the receiver operating characteristics (AUROC) values is reported. This measure was chosen, because it captures the relationship between model sensitivity and specificity. T-LSTM with one variable parametric sigmoid with an AUC of 0.69 performed better by 0.04 than the other three models. It also had the lowest standard deviation, which indicates the most stable performance.

The Wilcoxon signed-rank test [21] with a 0.05 level of significance was performed using SciPy package [30] to investigate if obtained results differ significantly (Table 3). T-LSTM model with one variable parametric sigmoid obtained significantly different results from LSTM and T-LSTM with the $g(\Delta_t) = 1/log(e + \Delta_t)$ decay function. The p-value between the T-LSTM model with one and two trainable variables was 0.06, which is slightly above the established threshold. The difference between other models' performance is not statistically significant.

Table 2: AUROC average scores of remission predictions for FEP patients using LSTM and T-LSTM models. Presented values are mean scores after training and testing models for ten runs. The standard deviation is given after ± sign.

Model	AUROC
LSTM	0.65 ± 0.030
T-LSTM with $g(\Delta_t) = 1/log(e + \Delta_t)$	0.65 ± 0.028
T-LSTM with one variable parametric sigmoid	0.69 ± 0.026
T-LSTM with two variables parametric sigmoid	0.65 ± 0.028

Table 3: The Wilcoxon signed-rank test [21] was performed on experiment results to check if the two related paired samples come from the same distribution. The p-values of the mentioned test are presented in this table. The level of significance was set to 0.05. P-values below this threshold are bolded.

	T-LSTM $g(\Delta_t) = 1/log(e + \Delta_t)$	T-LSTM one variable sigmoid	T-LSTM two variables sigmoid
LSTM	1.00	**0.04**	0.92
T-LSTM with two variables sigmoid	0.92	0.06	-
T-LSTM with one variable sigmoid	**0.01**	-	-

4.2 T-LSTM with one trainable variable parametric sigmoid

Three different T-LSTM cells, with separately trained parameters, were processing information from three questionaries: PANSS, PSP and CGI. It resulted in three decay functions, one for each evaluation. In Table 4, a values from the $g(\Delta_t) = -\frac{1}{1+exp(5-a*\Delta_t)} + 1$ function for first fold for all runs and all assessments can be found. Figure 9 depicts shapes of the decay functions for PANSS, PSP and CGI during the first fold for every run. As described before, our model was run 10 times with 5-fold cross-validation, which sums up to 50 trained models. We have decided to present the decay function parameter values and visualisations only from the first fold of each run, but the summary results demonstrate the mean value for all 50 models.

The function shape indicates the size of the short-term memory discount for different time intervals. If the a value falls below 1, only a small discount is applied, while if the value drops below 0.60, the shape of the function becomes almost flat. As stated in the previous section, when the decay function becomes constant, no discount for short-term memory is used, and the T-LSTM model becomes the LSTM model. Based on Figure 9, it is visible that during the first fold some discount was applied during each run. The decay function inside the T-LSTM CGI cell commonly remained constant, while PANSS and PSP usually used some discount. Similar conclusions can be drawn, by looking at the mean parameter value from all runs and all folds. The standard deviation for all three assessments is substantial. It proves the instability of the results and huge variability in the learned parameter from one trained model to another.

Table 4: Values of trainable variable a from the parametric sigmoid decay function $g(\Delta_t) = -\frac{1}{1+exp(5-a*\Delta_t)} + 1$. Each assessment had its T-LSTM cell, and experiments were repeated ten times. Presented values are from the first fold of each run. At the bottom, mean values and standard deviations from the first fold and all five folds are given. To keep the table neat, we will distinguish T-LSTM layers by their input assessments(PANSS, PSP, CGI).

Run	Variable a value in the T-LSTM layer		
	PANNS	PSP	CGI
Run 1	2.46	2.67	0.79
Run 2	-0.94	1.20	-0.05
Run 3	1.43	2.02	-1.39
Run 4	1.82	-0.02	-0.28
Run 5	2.24	1.75	-0.85
Run 6	2.17	1.77	0.50
Run 7	-0.23	0.26	0.89
Run 8	-0.36	3.08	2.77
Run 9	1.91	0.63	0.80
Run 10	-0.72	2.05	2.58
Mean first fold	0.98±1.30	1.54±0.96	0.58±1.27
Mean all folds	0.70± 1.46	1.39 ± 1.18	0.19± 1.30

Based on the AUROC standard deviation, the T-LSTM model with one trainable variable inside the decay function had the most stable performance of all the models. Looking at the a parameter stability, the PSP assessment had the most stable a value, while PANSS had the most unstable.

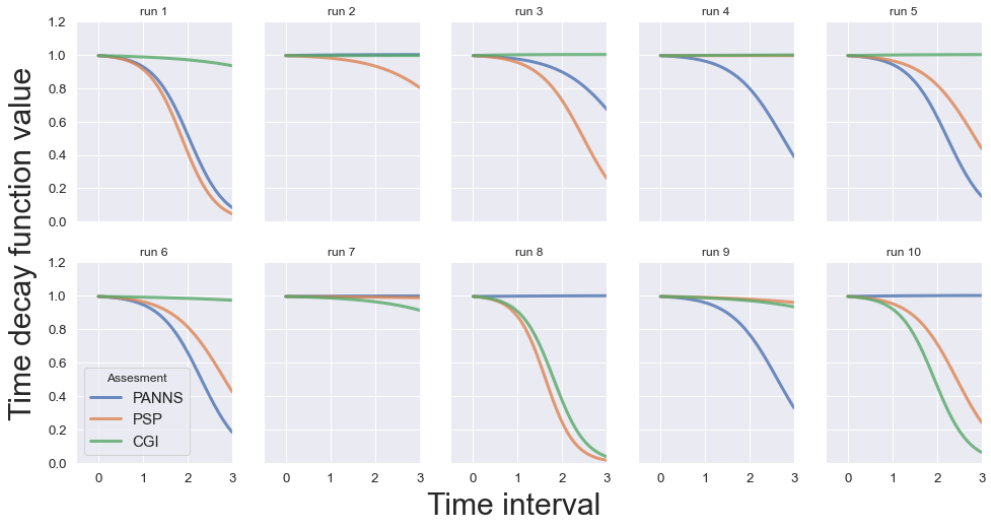


Figure 9: Decay functions with one trainable variable plotted for the first fold of every run. Parameters inside each T-LSTM cell were tuned in separately, which caused an individual decay function for PANSS, PSP and CGI assessments.

Moreover, during the first fold PSP applied a discount on 80% of runs. PANSS used some discount during 6 out of 10 runs, while CGI commonly remained constant. During the first fold, CGI a value had the biggest value range between -1.39 and 2.77.

4.3 T-LSTM with two trainable variables parametric sigmoids

The last experiment used $g(\Delta_t) = -\frac{1}{1+exp(b-a*\Delta_t)}+1$ decay function with two trainable parameters a and b. As previously, the model has separate T-LSTM cells for PANSS, PSP and CGI. Again, we will present the decay function parameter value and visualisation only for the first fold and mean values from all folds from all runs. The variables values are summed up in Table 5. Predominantly, b value was below zero, and a value was between 0.15 and 0.73 for all the assessments. Figure 10 depicts the shape of the decay functions for PANSS, PSP and CGI during every run. We can see that many of those functions are taking low values, some of them do not reach above 0.4.

Drawing conclusions from those results is challenging. Parameters values are irregular and unstable, and the functions' shapes are highly inconsistent between runs taking both ascending and descending slopes. There is no constant behaviour or repeating patterns. The goal was to make the adjusting function shape more flexible. However, it had a negative effect on the model. The AUC behaviour did not improve, and model behaviours are less stable.

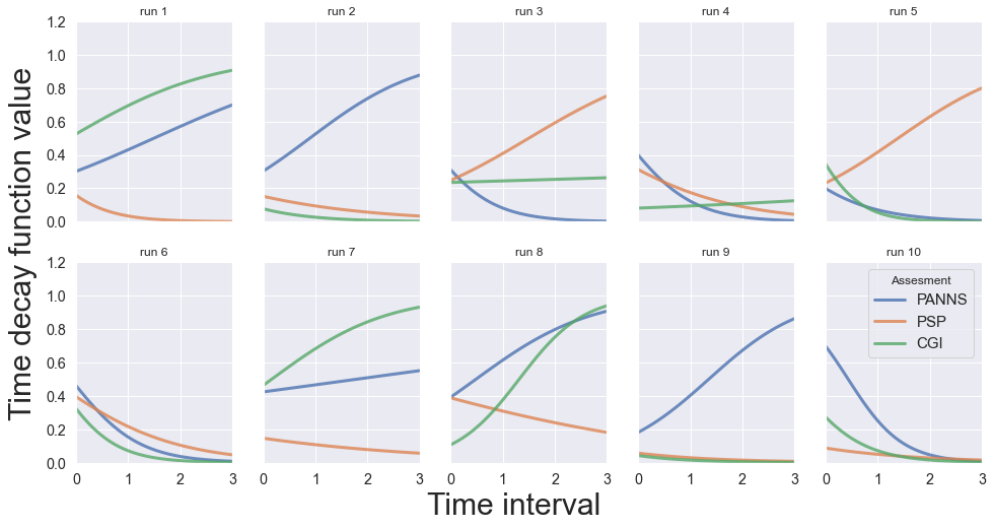


Figure 10: Decay functions with two trainable variables plotted for the first fold of every run. Parameters inside each T-LSTM cell were tuned in separately, which caused an individual decay function for PANSS, PSP and CGI assessments.

Table 5: Values of two trainable parameters a, b from T-LSTM time decay function $g(\Delta_t) = -\frac{1}{1+exp(b-a*\Delta_t)} + 1$. Each assessment had its T-LSTM cell, and experiments were repeated ten times. Presented values are from the first fold of each run. At the bottom, mean values and standard deviations for the first fold and all five folds are given. To keep the table neat we will distinguish T-LSTM layers by their input assessments(PANSS, PSP, CGI).

Run	PANSS		PSP		CGI	
Variable	a	b	a	b	a	b
Run 1	-0.56	-0.84	1.66	-1.67	-0.72	0.10
Run 2	-0.93	-0.83	0.54	-1.72	1.10	-2.48
Run 3	1.62	-0.80	-0.74	-1.11	-0.05	-1.18
Run 4	1.55	-0.41	0.76	-0.79	-0.16	-2.42
Run 5	1.18	-1.39	-0.86	-1.20	2.19	-0.64
Run 6	1.55	-0.16	0.86	-0.43	1.83	-0.74
Run 7	-0.17	-0.31	0.35	-1.77	-0.91	-0.15
Run 8	-0.90	-0.44	0.35	-0.46	-1.61	-2.12
Run 9	-1.11	-1.51	0.69	-2.81	1.07	-3.12
Run 10	1.92	0.83	0.60	-2.35	1.58	-0.98
Mean first fold	0.41±1.18	-0.59±0.63	0.42±0.70	-1.43±0.74	0.43±1.23	-1.37±1.04
Mean all folds	0.15±1.15	-0.86±0.92	0.73±0.95	-1.17±1.24	0.41±1.08	-1.15±0.95

5 Discussion

This chapter reflects on this research in three ways. First, the results presented in the previous chapter are given further interpretation. Second, we discuss the limitation of our project. Finally, based on this discussion, we will draw some conclusions and indicate opportunities for future research.

5.1 Results discussion

This study aimed to compare the performance of LSTM and T-LSTM models for psychosis prognosis prediction task for first-episode patients, and assess if adjusting the time decay function inside the T-LSTM model is beneficial for this assignment. We managed to improve the performance by adding one trainable parameter inside the parametric sigmoid decay function. The results were not only significantly better, but also more stable compared to the LSTM and T-LSTM model with a function suggested by Baytas et al. [3]. However, including two trainable parameters inside the decay function did not improve the model results, and the function shape indicates more unstable behaviour. This study used data from only 66 patients, which is promising for future research with a bigger population size. Moreover, we applied deep learning tools with longitudinal patient data which, apart from the Patterns in Psychiatry lab at Utrecht University Medical Centre architecture publication [25], was not done before for the PPP task.

Figure 9 presents the decay functions for the T-LSTM model with one trainable variable inside the parametric sigmoid. Based on those shapes and mean values, presented in Table 4, we can discuss the model behaviour. First, parameter a value has big variability for all three assessments, which might indicate that decay functions overfit the data. This may be due to a small dataset with only 66 samples, including 10% used as test sets. Moreover, we had no restrains on the decay function parameter value, which makes it even more prone to overfitting. Second, we can observe that the network found decay function parameter value as low as -1.39. However, if the a value drops below 0.60, it does not influence the shape of the decay function, in the range we use (for the time intervals of 0,1,2). The decay function remains constant. Again, putting a restrain on the parameter value, might help to find more stable results. Lastly, we recognize that the decay function is usually constant for CGI assessments, for PSP some discount is often applied, and for PANSS it varies from one run to another. This observation may suggest, if and how fast the information from different questionnaires is devaluating. Data from Clinical Global Impression remain equally important, while information from the Personal and Social Performance is becoming less important with time passing. To conclude more about how fast the information diminishes, we need more stable parameter results.

Drawing a conclusion about the decay function with two trainable variables results is more challenging, because of the huge variability in function shape and parameters values. Again this variability might be an effect of the small data sample and decay function overfitting the data. If we look at Figure 10, we can see that some decay functions take only low values. In this case, the decay function applies a huge discount on the short-term memory, almost like it is not carrying any meaningful information. When we look back at the T-LSTM architecture, described in Section 2.3, we see that short-term memory is created by data-driven decomposition applied to the previous cell state. That means that the network weights used for this decomposition, are learned using backpropagation. The overall performance of this model is quite stable, and this huge variability in decay function shapes does not influence the results. One possible explanation is that the cell

state is divided into almost non-existing short-term memory and meaningful long-term memory, which is not treated with the decay function or any other discount. All the important information would be carried by the long-term memory, which explains why this huge variability in the decay functions and taking values close to zero are not affecting our network performance. Moreover, this explanation also answers the question, of why with such instability in the decay function shapes, we can still obtain results truly similar to the LSTM and T-LSTM with a decay function suggested by Baytas et al. [3]. However, this is just an assumption, which can guide further work. Network weights should be further investigated, and restrains on their values should be applied, to see if that can bring some stabilization to the decay function parameters.

5.2 Limitation

After the discussion of the results, we see that the greatest limitation of our study is not holding information about short-term memory weights. This knowledge would allow us to tell more about the obtained results, and to either maintain or reject statements made in the previous section.

The second restrain refers to the amount of data used in this study. We had samples of only 66 patients, which required pre-training with 10000 artificial records. This is a possible reason for decay function overfitting and huge variability in the decay function parameters value.

Other limitations concern the OPTiMiSE dataset [12], which comes from a clinical trial. Firstly, only patients that agreed and did not drop out of the study are included in the dataset, which influences patients' and data heterogeneity. Second, using data from only one clinical study, does not provide external validation. Moreover, the group had strictly controlled medication. Only amisulpride and olanzapine were prescribed, while other treatments are also possible. Those data limitations might influence our study's generalisability. We built a model for psychosis prognosis prediction on data that come from a strictly controlled sample.

5.3 Conclusions and Future work

During this study, LSTM and T-LSTM neural network models were applied to the psychosis prognosis prediction task for first-episode psychosis patients using data from two first phases of OPTiMiSE clinical study [12]. The results show that adjusting the T-LSTM time decay function to a specific application by backpropagation, can improve model performance compared to the LSTM and T-LSTM model with the decay function proposed by Baytas et al. [3].

Our findings suggest that more interesting research can be done to explore potential T-LSTM applications for PPP task. First, we should look closer into cell state decomposition into short-term and long-term memory. In order to see complete model behaviour, we need to compare those findings with the decay function shapes. Second, due to high variability in the decay function parameters values, applying weights constraints and regularization (which consider weights sizes in the model loss function) might contribute to more stable results. Another direction for future research, would be adjusting the T-LSTM model architecture to have no restraints on a decay function form. Moreover, studies focused on improving prediction results are necessary, if we want to deploy models for psychosis prognosis prediction. Lastly, after solving instability issues, comparing T-LSTM results with other machine learning models would be an interesting overview. Developing PPP research is an important step in supporting the decision-making process regarding treatment and medication choice for first-episode psychosis patients, which contributes to their quality of life improvement.

References

[1] David Ahmedt-Aristizabal et al. "Identification of children at risk of schizophrenia via deep learning and EEG responses". In: *IEEE journal of biomedical and health informatics* 25.1 (2020), pp. 69–76.

[2] Nancy C Andreasen et al. "Remission in schizophrenia: proposed criteria and rationale for consensus". In: *American Journal of Psychiatry* 162.3 (2005), pp. 441–449.

[3] Inci M Baytas et al. "Patient subtyping via time-aware LSTM networks". In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 65–74.

[4] François Chollet et al. *Keras*. `https://keras.io`. 2015.

[5] Israel Elujide et al. "Application of deep and machine learning techniques for multi-label classification performance on psychotic disorder diseases". In: *Informatics in Medicine Unlocked* 23 (2021), p. 100545.

[6] Mark N Gibbs and David JC MacKay. "Variational Gaussian process classifiers". In: *IEEE Transactions on Neural Networks* 11.6 (2000), pp. 1458–1464.

[7] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[8] Alex Graves and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks". In: *International conference on machine learning*. PMLR. 2014, pp. 1764–1772.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, pp. 6645–6649.

[10] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[11] Ronald J Janssen, Janaina Mourão-Miranda, and Hugo G Schnack. "Making individual prognoses in psychiatry using neuroimaging and machine learning". In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging* 3.9 (2018), pp. 798–808.

[12] René S Kahn et al. "Amisulpride and olanzapine followed by open-label treatment with clozapine in first-episode schizophrenia and schizophreniform disorder (OPTiMiSE): a three-phase switching study". In: *The Lancet Psychiatry* 5.10 (2018), pp. 797–807.

[13] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[14] Georgia Koppe et al. "Recurrent neural networks in mobile sampling and intervention". In: *Schizophrenia bulletin* 45.2 (2019), pp. 272–276.

[15] Nikolaos Koutsouleris et al. "Multisite prediction of 4-week and 52-week treatment outcomes in patients with first-episode psychosis: a machine learning approach". In: *The Lancet Psychiatry* 3.10 (2016), pp. 935–946.

[16] Nikolaos Koutsouleris et al. "Prediction models of functional outcomes for individuals in the clinical high-risk state for psychosis or with recent-onset depression: a multimodal, multisite machine learning analysis". In: *JAMA psychiatry* 75.11 (2018), pp. 1156–1172.

[17] Siwei Lai et al. "Recurrent convolutional neural networks for text classification". In: *Twenty-ninth AAAI conference on artificial intelligence*. 2015.

[18] Pablo Lanillos et al. "A review on neural network models of schizophrenia and autism spectrum disorder". In: *Neural Networks* 122 (2020), pp. 338–363.

[19] Samuel P Leighton et al. "Development and validation of multivariable prediction models of remission, recovery, and quality of life outcomes in people with first episode psychosis: a machine learning approach". In: *The Lancet Digital Health* 1.6 (2019), e261–e270.

[20] Zhuangzhuang Li et al. "Deep learning based automatic diagnosis of first-episode psychosis, bipolar disorder and healthy controls". In: *Computerized Medical Imaging and Graphics* 89 (2021), p. 101882.

[21] Henry B Mann and Donald R Whitney. "On a test of whether one of two random variables is stochastically larger than the other". In: *The annals of mathematical statistics* (1947), pp. 50–60.

[22] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[23] Mark J Millan et al. "Altering the course of schizophrenia: progress and perspectives". In: *Nature Reviews Drug Discovery* 15.7 (2016), pp. 485–515.

[24] Jessica de Nijs et al. "Individualized prediction of three-and six-year outcomes of psychosis in a longitudinal multicenter study: a machine learning approach". In: *npj Schizophrenia* 7.1 (2021), pp. 1–11.

[25] Opstal et al. *Long short term memory for psychosis prognosis prediction*. in preparation.

[26] Lars Lau Raket et al. "Dynamic ElecTronic hEalth reCord deTection (DETECT) of individuals at risk of a first episode of psychosis: a case-control development and validation study". In: *The Lancet Digital Health* 2.5 (2020), e229–e239.

[27] Raymond Salvador et al. "Evaluation of machine learning algorithms and structural features for optimal MRI-based diagnostic prediction in psychosis". In: *PLoS One* 12.4 (2017), e0175683.

[28] Jason Smucny, Ian Davidson, and Cameron S Carter. "Comparing machine and deep learning-based algorithms for prediction of clinical improvement in psychosis with functional magnetic resonance imaging". In: *Human brain mapping* 42.4 (2021), pp. 1197–1205.

[29] Sandra Vieira et al. "Using machine learning and structural neuroimaging to detect first episode psychosis: reconsidering the evidence". In: *Schizophrenia bulletin* 46.1 (2020), pp. 17–26.

[30] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: `10.1038/s41592-019-0686-2`.

[31] World Health Organisation. *Schizophrenia*. `https://www.who.int/`, Last accessed on 10 January 2022. 2022.

A Appendix

A.1

Table 6: AUROC, balance accuracy (BAC), sensitivity and specificity average scores of remission predictions for FEP patients using LSTM and T-LSTM models. Presented values are mean scores after training and testing models for ten runs. The standard deviation is given after ± sign.

Model	AUROC	BAC	sensitivity	specificity
LSTM	0.65 ± 0.030	0.63 ± 0.030	0.78 ± 0.035	0.48 ± 0.058
T-LSTM $g(\Delta_t) = 1/log(e + \Delta_t)$	0.65 ± 0.028	0.62 ± 0.026	0.73 ± 0.070	0.50 ± 0.046
T-LSTM one variable parametric sigmoid	0.69 ± 0.026	0.63 ± 0.033	0.70 ± 0.055	0.57 ± 0.071
T-LSTM two variables parametric sigmoid	0.65 ± 0.028	0.62 ± 0.043	0.66 ± 0.091	0.58 ± 0.080

A.2

Table 7: LSTM architecture. Input layers are not connected to any other layers.

Layer	Output shape	Parameters	Connected to
LSTM_input_PANSS	(None, None, 30)	0	
LSTM_input_PSP	(None, None, 5)	0	
LSTM_input_CGI	(None, None, 2)	0	
LSTM_PANSS (LSTM)	(None, None, 50)	16 200	LSTM_input_PANSS
LSTM_PSP (LSTM)	(None, None, 10)	640	LSTM_input_PSP
LSTM_CGI (LSTM)	(None, None, 5)	160	LSTM_input_CGI
Static_input_demographics	(None, None, 86)	0	
Static_input_diagnosis	(None, None, 22)	0	
Static_input_lifestyle	(None, None, 14)	0	
Static_input_somatic	(None, None, 16)	0	
Static_input_treatments	(None, None, 1)	0	
Static_input_cdss	(None, None, 9)	0	
Static_input_swn	(None, None, 20)	0	
Static_input_mini	(None, None, 70)	0	
LSTM_concat (Concatenate)	(None, None, 65)	0	LSTM_PANNS, LSTM_PSP LSTM_CGI
Static_concat (Concatenate)	(None, None, 238)	0	All the static input layers
Concat_all (Concatenate)	(None, None, 303)	0	LSTM_concat, Static_concat
Fusion_dropout (Dropout)	(None, None, 303)	0	Concat_all
time_distributed (Time distributed)	(None, None, 30)	9120	Fusion_dropout
time_distributed_1 (Time distributed)	(None, None, 5)	1520	Fusion_dropout
time_distributed_2 (Time distributed)	(None, None, 2)	608	Fusion_dropout
subtract (Subtract)	(None, None, 30)	0	LSTM_input_PANNS, time_distributed
subtract_1 (Subtract)	(None, None, 5)	0	LSTM_input_PSP, time_distributed_1
subtract_2 (Subtract)	(None, None, 2)	0	LSTM_input_CGI, time_distributed_2
time_distributed_3 (Time distributed)	(None, None, 5)	1520	Fusion_dropout
dropout (Dropout)	(None, None, 30)	0	subtract
dropout_1 (Dropout)	(None, None, 5)	0	subtract_1
dropout_2 (Dropout)	(None, None, 2)	0	subtract_2
dropout_3 (Dropout)	(None, None, 5)	0	time_distributed_3
PANNS (Time distributed)	(None, None, 30)	930	dropout
PSP (Time distributed)	(None, None, 5)	30	dropout_1
CGI (Time distributed)	(None, None, 2)	6	dropout_2
PANNS_remission (Time distributed)	(None, None, 2)	12	dropout_3