Transform Diabetes

Harnessing Transformer-Based Machine Learning and Layered Ensemble with Enhanced Training for Improved Glucose Prediction.

RUNE ALEXANDER LAURSEN
PESHANG ALO

SUPERVISORS
Morten Goodwin, Per-Arne Andersen

University of Agder, 2023
Faculty of Engineering and Science
Department of Engineering and Sciences

Obligatorisk gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Vi erklærer herved at vår besvarelse er vårt eget arbeid, og at vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	**Ja**
2.	**Vi erklærer videre at denne besvarelsen:** • Ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • Ikke refererer til andres arbeid uten at det er oppgitt. • Ikke refererer til eget tidligere arbeid uten at det er oppgitt. • Har alle referansene oppgitt i litteraturlisten. • Ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	**Ja**
3.	Vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	**Ja**
4.	Vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	**Ja**
5.	Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	**Ja**
6.	Vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	**Ja**
7.	Vi har i flertall blitt enige om at innsatsen innad i gruppen er merkbart forskjellig og ønsker dermed å vurderes individuelt. Ordinært vurderes alle deltakere i prosjektet samlet.	**Nei**

Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven Forfatterne har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).
Oppgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	**Ja**
Er oppgaven båndlagt (konfidensiell)?	**Nei**
Er oppgaven unntatt offentlighet?	**Nei**

Acknowledgements

This master's thesis was conducted at the University of Agder (UiA) in Grimstad, Norway, by Rune Alexander Laursen and Peshang Alo, completing the master's program at the Department of Information and Communication Technology (ICT). We want to give a big thank you to Morten Goodwin for his superb direction and oversight during the period time. Your advice has been crucial in helping us complete the thesis to the desired excellence. We also like to thank Per-Arne Andersen for his advice and suggestions. And lastly, a big thanks to our families and friends for staying positive.

We would also like to acknowledge that a version of this thesis has been submitted to the SGAI International Conference on Artificial Intelligence 2023 for consideration.

We want to express our appreciation to the creators of the artificial intelligence tools that made it easier to write this thesis. OpenAI's ChatGPT, Quillbot, and Grammarly have been beneficial tools that have enhanced the content's preciseness and clarity in drafting and revising. We have only used these tools for inspiration and language cleaning; all text in this thesis is our own, and no text is directly copied from, e.g., ChatGPT.

Grimstad,
June 5, 2023
Rune Alexander Laursen and Peshang Alo

Abstract

Type 1 diabetes is a common chronic disease characterized by the body's inability to regulate the blood glucose level, leading to severe health consequences if not handled manually. Accurate blood glucose level predictions can enable better disease management and inform subsequent treatment decisions. However, predicting future blood glucose levels is a complex problem due to the inherent complexity and variability of the human body.
This thesis investigates using a Transformer model to outperform a state-of-the-art Convolutional Recurrent Neural Network model by forecasting blood glucose levels on the same dataset. The problem is structured, and the data is preprocessed as a multivariate multi-step time series. A unique Layered Ensemble technique that Enhances the Training of the final model is introduced. This technique manages missing data and counters potential issues from other techniques by employing both a Long Short-Term Memory model and a Transformer model together. The experimental results show that this novel ensemble technique reduces the root mean squared error by approximately 14.28% when predicting the blood glucose level 30 minutes in the future compared to the state-of-the-art model. This improvement highlights the potential of this approach to assist diabetes patients with effective disease management.

Contents

List of Figures

List of Tables

Glossary

CGM is an abbreviation of Continous Glucose Monitor, which is a wearable sensor that continuously measures the wearer's glucose level and transmits the data to a handheld device, typically a smartphone.. 2, 12, 13, 23, 27, 28, 31

K-Fold Cross-Validation is a method that splits a dataset into a “k" number of groups, which is used to train and test a dataset to check both if the dataset is well distributed and that it does not have different biases through the groups.. 39

Acronyms

AI Artificial Intelligence. 2, 3, 5–7, 14, 18, 23, 25

ARIMA Autoregressive Integrated Moving Average. 8, 18, 19

BGL Blood Glucose Level. vii, 2–5, 12–14, 16, 18–32, 35–37, 40, 42–49

BGLPC Blood Glucose Level Prediction Challenge. 5, 13, 29, 37–39, 47

CNN Convolutional Neural Network. 8, 18, 21, 24, 25

CRNN Convolutional Recurrent Neural Network. 4, 23, 24, 38, 40, 46

DNN Deep Neural Network. 18

DRNN Dilated Recurrent Neural Network. 23, 24

FL Fuzzy Logic. 13

GAN Generative Adversarial Network. 22

GRU Gated Recurrent Units. 10, 18

GSR Galvanic Skin Response. vii, 14, 27, 29, 43–45, 49

LEET Layered Ensemble with Enhanced Training. 5, 26, 30, 32, 43, 45, 48, 49

LSTF Long Sequence Time-series Forecasting. 19, 20

LSTM Long Short-Term Memory. 2, 3, 7, 8, 10, 13, 18–21, 23–25, 32, 35, 45, 48, 49

MAE Mean Absolute Error. 20, 23, 26, 36–42, 46, 47

MARD Mean Absolute Relative Difference. 24

ML Machine Learning. 3, 6, 23, 25, 29

MLP Multilayer Perceptron. 33, 51

MPC Model Predictive Control. 13

MSE Mean Squared Error. 7, 26, 34, 36, 45, 46

MTL Multitask Learning. 23

NLP Natural Language Processing. 7, 8, 10, 12

PID Proportional-Integral-Derivative. 13

RMSE Root Mean Squared Error. 4, 20, 23, 24, 26, 36–48

RNN Recurrent Neural Network. 2, 3, 7, 8, 10, 11, 13, 18, 19, 24, 25

SEG Surveillance Error Grid. 23

Seq2Seq Sequence to Sequence. 8, 25

STNN Spatiotemporal Transformer Neural Network. 19, 20

SVR Support Vector Regression. 19, 23, 24

Chapter 1

Introduction

This chapter begins by thoroughly explaining the underlying issue, highlighting its fascinating characteristics and the unique methods to resolve it. The motivation behind the creation of this thesis is then discussed. The goals are next stated, which prepares the audience for the presentation of the hypotheses. The distinctive contributions of this thesis are then mentioned, and the chapter concludes with an overview of the thesis.

1.1 Introduction

Diabetes is a common chronic disease in which the body cannot produce insulin and thus cannot control Blood Glucose Level (BGL). The only way to maintain the level is by measuring and adjusting them by injecting insulin. If the person fails to keep the BGL consistent, the person has an increased risk of complications such as blindness and nerve damage. Existing technologies like CGM and Insulin Pumps are widely used by people with diabetes to help them manage their disease. But even with these tools available, many people still calculate their insulin doses mentally because the tools are still insufficient, which often leads to poor calculations and unstable BGL because of all the factors that affect it. The combination of the available technologies provides a simple solution to the problem because they fail to take a holistic view of all the factors and determine how much insulin needs to be injected.

Researchers have improved their results on public and private datasets in the last ten years by adopting Artificial Intelligence (AI) techniques instead of statistical models. However, despite advanced methods like Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM), the current state-of-the-art still does not provide reliable outcomes that can be applied confidently. This limitation stems primarily from the inherent complexity and variability of the human body. Specifically, the computation of insulin dosage is a multifaceted problem due to the dynamic nature of human physiology, with many factors influencing the BGL. These factors include diet, exercise, insulin sensitivity, stress, illness, hormonal changes, and other medications. Additionally, when calculating these factors, most have a delayed effect on the BGL, meaning they must be calculated with a time dimension. This can be classified as a multivariate time series problem [1]. This type of

[1]A multivariate time series problem is a type of data analysis challenge where multiple variables, each recorded over time, interact with each other in complex, often nonlinear ways, and the goal is to predict or understand these interactions.

problem is considered complex due to the interactions between all the variables, and further, it complicates the data preprocessing.

The most common models used in AI for time series analysis are LSTM models. Still, the emergence of the Transformer model has shown promising results in many time series problems. Transformers offer a potential breakthrough in accurately predicting and managing time series data due to their ability to model complex interactions and dependencies between factors affecting a system over time. This thesis will use a Transformer-based approach to forecast BGL for people with type 1 diabetes, leveraging the ability to capture the highly dynamic dependencies and model complex interactions between the factors. One of the critical deliveries of this thesis is a novel ensemble method that uses two distinct models to outperform the state-of-the-art model by 14.28%.

1.2 Motivation

One of the authors of this thesis is a type-1 diabetes patient, which is a crucial motivating factor for this thesis. Managing diabetes is a challenging and lifelong task that requires constant monitoring and careful management of BGLs. Poorly controlled diabetes can lead to severe complications such as heart disease, kidney damage, nerve damage, and blindness. By improving the accuracy of insulin dosing and helping patients maintain stable BGLs, this approach could potentially improve the quality of life for millions of people with diabetes worldwide.

Another motivation for this project is that Transformer architecture solves the issue of vanishing gradients from RNN and LSTM. Vanishing gradient is a problem in RNNs and LSTMs where the gradient signal becomes too small, making it hard to learn long-term dependencies. Transformers solve this by using self-attention mechanisms instead of recurrent connections to capture long-term dependencies more effectively and avoid vanishing gradients. This is a significant issue when predicting BGLs on longer timescales because when data points are too far away, they will not affect the prediction. In diabetes management, some data points like exercise will affect blood glucose for up to 28 hours [50, 4].

1.3 Field of research

This thesis focuses on AI-driven solutions for type 1 diabetes treatment and is positioned at the interface of computational biology, AI, and health informatics. One of the critical tasks in efficient diabetes care is predicting the BGL from multivariate time series data using advanced Machine Learning (ML) techniques. Despite significant advancements made with models like RNN and LSTM, the complexity of human physiology and the diversity of variables affecting glucose metabolism make it difficult to forecast the BGL accurately. The unique ensemble approach introduced in this thesis advances prediction accuracy by ensembling a LSTM model and a Transformer model.

1.4 Thesis Definition

The primary goal of this thesis is to use a deep learning framework that utilizes the Transformer-based technique [60] to predict future BGLs. To achieve this, the project is broken down into the following goals.

1.4.1 Thesis goals

Goal 1: Prepare the OhioT1DM dataset for training as a multivariate multi-step time series.

Goal 2: Validate the results from the state-of-the-art by exectuting their code.

Goal 3: Utilize a Transformer model and use the preprocessed dataset to predict BGLs 30 minutes into the future with higher accuracy than the state-of-the-art model.

1.4.2 Hypotheses

Hypothesis 1: A Transformer model can outperform the state-of-the-art Convolutional Recurrent Neural Network (CRNN) model proposed by J Freiburghaus et al. [19] by predicting BGLs 30 minutes into the future with lower Root Mean Squared Error (RMSE) on the same dataset.

Hypothesis 2: Intelligent handling of missing BGL values, rather than zero-filling, could boost model accuracy.

Hypothesis 3: Opting for a 24-hour input sequence instead of a two-hour sequence could boost the Transformer model's accuracy.

1.5 Contributions

This thesis proposes an advancement in algorithmic decision-making aids for diabetes management. The primary contribution lies in the conception of a novel ensemble technique to impute missing values in a dataset, which is subsequently used by a Transformer-based model designed for multi-step time series and used to forecast the BGL 30 minutes ahead. To the best of the authors' knowledge, this approach marks a novel contribution to BGL prediction accuracy. The intricacy of the problem is emphasized by the necessity to forecast a multivariate time series impacted by diverse factors such as insulin, physical activity, and meal consumption, making this research challenging.

1.6 Pre-Project

The group conducted a preliminary project to better understand the factors necessary for utilizing AI to manage type-1 diabetes. The paper is only reviewed internally and is not published. The project examines previous work in the field of diabetes management algorithms, current datasets, and alternative data collection methods. One of the conclusions drawn in the project is that present methodologies utilize too few data points, and even if present solutions are forecasting with high accuracy, they do not predict far enough into the future to be used in real-world applications. The project also reviews the models used and their evaluation metrics. Lastly, a crucial part of the pre-project; acquiring access to the OhioT1DM dataset.

1.7 Thesis Outline

The contents of this thesis are divided into the following chapters:

- **Background** provides a comprehensive overview of the main concepts in the thesis. It explores AI and its applications, time series forecasting, Transformer architecture, diabetes, the Blood Glucose Level Prediction Challenge (BGLPC), the preliminary project, and critical factors such as missing values, data leakage, and ensemble models.
- **State-of-the-Art** provides an overview of the current state of research in BGL prediction, highlighting recent advancements and identifying gaps for further investigation.
- **Method** describes the details of the methodology employed to obtain the results, including preprocessing steps, subset division, and the introduction of the Layered Ensemble with Enhanced Training (LEET) technique. In addition, it describes the model architectures and performance metrics used in the thesis.
- **Experiments and Results** explains the experimental methodology and thesis findings. It consists of preliminary experiments, experiments with Transformer models (encoder-only and encoder-decoder), and experiments on various missing values approaches, including the LEET technique. The results demonstrate variations in model performance and the impact of different techniques on the accuracy of predictions.
- **Conclusion** summarizes the research findings and provides suggestions for future research directions and improvements.

Chapter 2

Background

This chapter explains critical concepts of AI and Diabetes to give readers some background to what comes in the following chapters.

2.1 Artificial Intelligence

A neural network is a computational system that emulates the structure and function of biological neurons in the brain [35]. The system comprises numerous interlinked processing units, called artificial neurons or perceptrons, that collaborate to analyze and acquire knowledge from complex data patterns. The basic unit of a neural network is a perceptron, which receives numerous inputs and performs a weighted summation on them, subsequently applying an activation function to generate an output. Further explanation of activation functions will be presented later in this chapter. The perceptron's weights and biases are modified during training to enhance the network's efficacy on a specific task. This is achieved through different methods, including backpropagation and gradient descent. Neural networks can perform tasks like classification and regression by mapping input data to output predictions. Utilizing a ML, algorithm proves advantageous in situations that entail the processing of large volumes of data. This is due to its capacity to obtain and deduce characteristics from the data, thereby removing the need for manual feature engineering. Neural networks can be classified into feedforward, recurrent, and convolutional networks. Each category has unique architectural and distinctive components, and selecting the right category depends on the nature of the dataset and the task under consideration. Feedforward networks are commonly employed for uncomplicated classification assignments, whereas recurrent networks are more appropriate for tasks that entail sequential data, such as language processing. Neural networks have performed exceptionally on various tasks, such as image and natural language processing, outperforming state-of-the-art statistical models. Ongoing research and development efforts continue in the field as novel architectures and techniques are being suggested to enhance the performance and capabilities of these systems.
Optimization and loss functions are crucial in ML and AI. Optimization functions, often called optimizers, are mathematical functions that adjust model parameters to minimize the result of a loss function [14]. They are vital to neural network learning as they guide the model toward the most accurate predictions. One commonly used optimization function is Gradient Descent, which iteratively adjusts parameters in a direction that minimizes the loss.

On the other hand, loss functions, also known as cost functions, measure the discrepancy between the model's predictions and the actual values. The choice of loss function depends on the specific problem at hand. For example, in regression problems, Mean Squared Error (MSE) is commonly used, whereas in classification problems, Cross-Entropy loss is often preferred. Mathematically, the model's parameters can be defined as θ, the prediction of the model as $\hat{y}$, and the true value as y. The loss function L would calculate the difference between $\hat{y}$ and y, then the optimizer would find the θ value that minimizes L as eq. (2.1) show:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, L(y, \hat{y}(\theta)) \tag{2.1}$$

Activation functions [61] are also a fundamental element of a neural network and play an essential role in deciding the output of a neuron. They introduce non-linearity into the model, allowing it to learn complex patterns. They transform a neuron's input into an output, shaping its activation in response to its inputs. The most commonly used activation functions include the Sigmoid, Tanh, and ReLU (Rectified Linear Unit). The Sigmoid function uses the range between 0 and 1, which makes it practical for models that predict probabilities. The Tanh function is like the Sigmoid function but maps its input into a range between -1 and 1. The ReLU function, on the other hand, maps the negative inputs to zero while it leaves the positive unchanged. This function is quite popular due to its efficiency but can cause exploding gradients and suffer from a problem referred to as the "dying ReLU[1]." The ELU activation function is a smooth, zero-centered function that avoids the vanishing gradient and dying ReLU problems but is more computationally expensive than ReLU.

In AI, many different neural network architectures exist to solve various tasks, as mentioned previously. RNN and LSTM are two much-used architectures that are used on sequential data like time series, speech recognition, or Natural Language Processing (NLP) [64, 65]. RNNs can capture the temporal dependencies of sequential data, allowing them to learn from the past and make predictions. It is highly effective at complex modeling patterns in data, making them a powerful tool for machine learning and AI applications. One of the shortcomings of RNNs is the problem of vanishing gradients, which is a phenomenon that occurs in traditional RNNs when the gradients of the parameters during training become very small, effectively preventing the network from learning. This happens because the gradients are calculated based on the backpropagation of error through many time steps, causing the gradients to become very small and the network to struggle to learn effectively. This can lead to poor performance and slow convergence, especially for tasks that require the network to capture long-term dependencies in sequential data. LSTM is a variation of RNN that addresses the issue of vanishing gradients in traditional RNNs, allowing the network to preserve long-term dependencies in sequential data better. It uses memory cells and gates to control information flow and stability in the network, allowing it to effectively capture long-term dependencies and make more accurate predictions.

When predicting time series, a variable's historical data points are used to predict future values. This problem is common in finance, weather forecasting, and medical analysis [6, 23,

[1]The Dying ReLU problem occurs when ReLU neurons in neural networks become perpetually inactive, reducing the model's capacity

22]. In these cases, a model is trained on past data and then used to predict future values. To do this effectively, the model needs to capture the underlying patterns and dependencies in the data, such as seasonality, trend, and autocorrelation. Further, a multi-step multivariate time series predicts multiple future steps using more than one related variable in the given historical data. This is a more complex problem than single-step univariate time series prediction because it requires the model to capture inter-dependencies between variables and predict multiple future steps. Various models can be used to tackle this problem, such as multi-output regression models, multi-step versions of Autoregressive Integrated Moving Average (ARIMA), and multivariate LSTMs. In the case of LSTMs, the network can be designed to handle multiple input variables and make predictions for various steps in the future, effectively capturing the inter-dependencies between the variables. Additionally, models such as Convolutional Neural Networks (CNNs) and Attention Mechanisms can also be used to extract relevant features from the time series data.
The attention mechanisms have addressed the information and performance loss, which is a powerful solution to these losses in Sequence to Sequence (Seq2Seq) models [58], particularly when handling long input sequences. Seq2Seq is a RNN-based encoder-decoder model for processing and generating variable-length sequences in tasks like translation and summarization. The attention model allows the decoder to focus selectively on relevant portions of the input data during output prediction by assigning weights to each encoder's hidden state [3]. This dynamic approach enhances the model's ability to capture long-range dependencies, improving performance across various applications. However, the attention mechanism in Seq2Seq models is usually limited to the encoder's hidden states, which means that the model can only consider information from the input when generating the output and does not consider any relationship between different parts of the output sequence itself.
Ashish Vaswani et al. [60] 2017 introduced the Transformer, a type of deep neural network architecture that has revolutionized the field of NLP. This has since been applied to various domains, including time series prediction. The architecture uses Attention Mechanisms to dynamically weigh the importance of different elements in a sequence, making it well-suited for sequential data such as time series. In the context of time series prediction, a Transformer model can capture the dependencies between time steps and predict future values. The input to the Transformer would be a set of historical time steps, and the target would be the future values. The self-attention layers in the Transformer would allow the network to weigh the importance of different time steps in the prediction, effectively capturing the dependencies between the time steps. It is important to note that the Transformer architecture may not always be the best choice for time series prediction on univariate data without seasonal changes, as other models such as ARIMA, SARIMA, and LSTMs may be more appropriate depending on the specific requirements of the task. However, the Transformer has shown promising results in several time series prediction tasks and is an approach worth considering.

2.2 Time Series Forecasting

Time series forecasting is a critical aspect of data analysis and predictive modeling, focusing on using historical data and statistical techniques to predict future events or observations that are influenced by time. In comparison, traditional prediction solutions ignore the temporal aspect of the data. Time series forecasting aims to uncover the underlying patterns and relationships within the data to generate precise predictions about future observations. This approach is significant in various areas, such as finance, economics, meteorology, and sales forecasting, as mentioned in the last section.
There are various time series forecasting methods, including univariate and multivariate forecasting. Univariate time series forecasting considers past observations of a single variable to make predictions. It is the simplest form of time series forecasting, as it only involves a single variable, and it can be mathematically represented as follows:

$$Y_{t+1} = f(Y_t, Y_{t-1}, Y_{t-2}, ..., Y_{t-p+1}) \tag{2.2}$$

Here, Y_t represents the variable's value at time t, f represents the forecasting model, and p represents the number of previous time steps used as input to the model [51].

Multivariate time series forecasting involves predicting future values of multiple time series simultaneously. In multivariate forecasting, the goal is to use the relationship between the different variables to improve the accuracy of the forecasts. This type of forecasting can be represented as:

$$Y_{t+1}, Y_{t+2}, ..., Y_{t+H} = f(X_t, X_{t-1}, X_{t-2}, ..., X_{t-p+1}) \tag{2.3}$$

X_t represents the additional variables for time step t, and Y_t is the predicted value for the target variable at time t. The function f considers both the past observations of the target variable and the additional variables for each time step. The H represents the number of time steps to forecast [51].
Multi-step forecasting refers to predicting future values of a time series for multiple time steps ahead. It involves predicting values beyond the next time step and is a common task in many real-world applications, such as sales forecasting, demand forecasting, and financial forecasting.

$$Y_{t+1}, Y_{t+2}, ..., Y_{t+H} = f(Y_t, Y_{t-1}, Y_{t-2}, ..., Y_{t-p+1}) \tag{2.4}$$

Where Y_t represents the target vector at time t, f represents the forecasting model, p represents the number of previous time steps used as input, and H represents the number of time steps to forecast [51].

2.3 Transformer Architecture

The Transformer model, as mentioned in section 2.1, has had a profound impact on deep learning, specifically within NLP. Before the Transformer, recurrent models like RNNs, LSTMs, and Gated Recurrent Unitss (GRUs) dominated sequence-based tasks. However, their sequential nature posed challenges to parallel computation and managing long-term dependencies.

Unlike these models, the Transformer's innovative attention mechanisms and encoder-decoder architecture handle long-range dependencies efficiently and enable parallel processing. Though it has some limitations, such as high computational requirements and dependence on large datasets, the benefits largely outweigh these drawbacks.

In this section, we'll explore the architecture of the Transformer model, focusing on the encoder-decoder framework and the attention mechanisms that are its core components. By understanding these, we can comprehend the strengths of the Transformer model and its impact on the field of NLP and other sequence-based tasks.

2.3.1 Encoder-Decoder Architecture

The encoder-decoder framework is a widely used approach in time series forecasting that uses deep learning. It separates the model into two parts: the encoder and the decoder, as shown in fig. 2.1. The encoder processes the historical time series data into a condensed representation, which the decoder then uses to make future predictions.

When applied to multi-step time series forecasting, the encoder-decoder model architecture combined with an attention mechanism incorporates temporal information into the model. The encoder contains several identical sub-layers, each consisting of a self-attention mechanism and a feedforward neural network. Meanwhile, the decoder has multiple identical sub-layers, each with a multi-head attention mechanism, a self-attention mechanism, and a feedforward neural network. The multi-head attention mechanism allows the decoder to focus on the representation generated by the encoder and make future predictions. In contrast, the self-attention mechanism helps the decoder consider the relationships between future time steps.

The encoder unit in the Transformer architecture is designed to convert the input sequence into a condensed and meaningful representation. It comprises identical sub-layers, including a self-attention mechanism explained in section 2.3.2 and a feedforward neural network. The feedforward neural network in the encoder further processes the representation from the self-attention mechanism, using linear transformations with non-linear activation functions to refine the representation and decrease its dimensionality. The output of the encoder unit is a compact and informative representation of the input sequence, which the decoder unit uses to generate forecasts. Using multiple sub-layers in the encoder enables the model to capture hierarchical representations of the input, making it capable of modeling complex dependencies in the data.

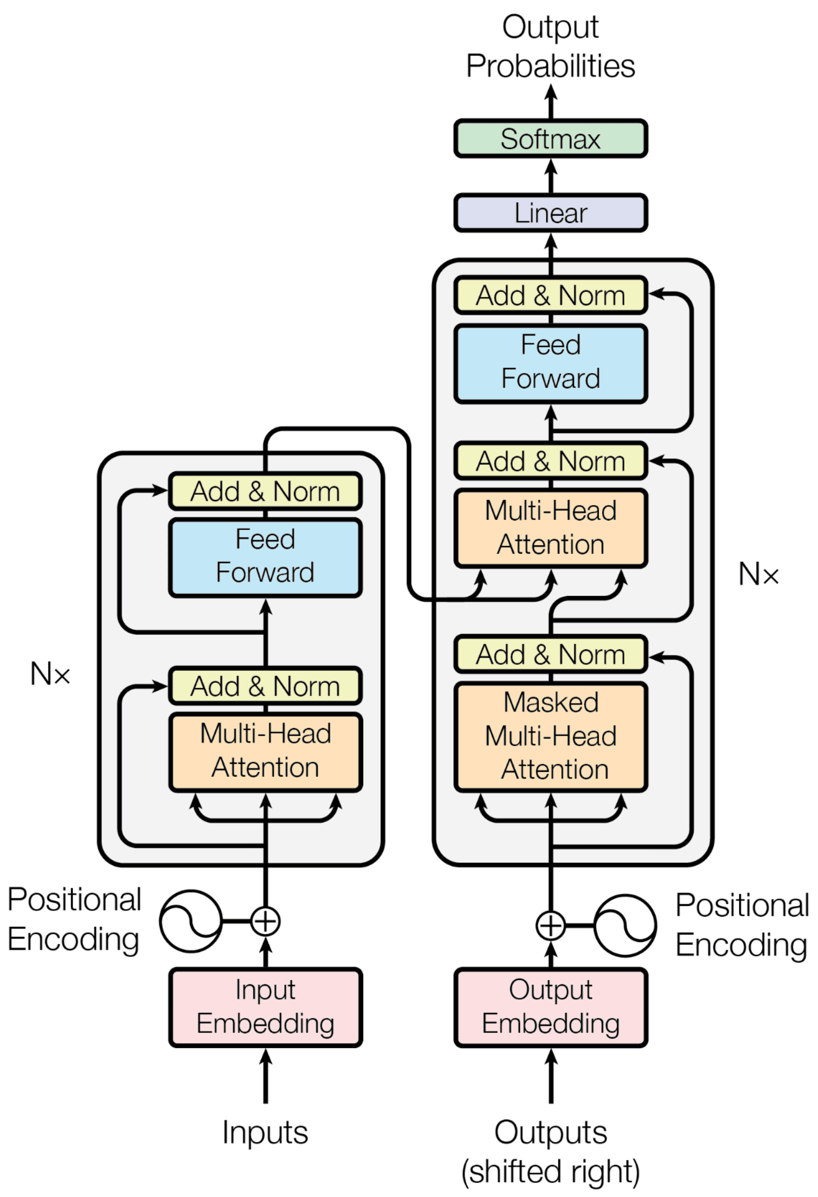

Figure 2.1: Transformer Architecture (adopted from [60])

2.3.2 Attention Mechanisms

Self-attention is a mechanism that allows the network to weigh the importance of different elements in the input sequence to predict a specific component. The idea behind self-attention is to represent each sequence element as a vector and then compute a weighted sum of the vectors based on their relationships to the current part. This allows the network to dynamically adjust the importance of each sequence component instead of relying solely on their order in the sequence [60]. This innovative approach distinguishes transformers from traditional RNN.

Multi-head attention extends the self-attention mechanism by using multiple parallel attention mechanisms with separate linear projections to compute different representation subspaces. The final representation of each element is then obtained by concatenating the outputs of all attention heads and applying another linear prediction. This allows the network to attend to multiple aspects of the input sequence in parallel, leading to better performance

than a single self-attention mechanism [60].
Further, the Multi-head attention offers an advantage to the neural network as it captures various and possibly complementary facets of the input sequence instead of a solitary, overarching representation. Moreover, it enhances the efficiency of capturing intricate patterns and interconnections in sequential data, which has significantly contributed to the triumph of the Transformer model in various NLP assignments. In general, self-attention and multi-head attention mechanisms are potent tools that facilitate neural networks in capturing intricate patterns and long-range dependencies in sequential data. These mechanisms have significantly contributed to the advancement of natural language processing and the current state-of-the-art.

2.4 Diabetes

This section explains the inner workings of diabetes, which should be understood to have the insight necessary to work towards a solution to the problem.
A healthy human body constantly adjusts the BGL by mutual secretion of insulin and glucagon hormones to maintain an optimal level between 70 - 130 (mg/dL). When carbohydrates are consumed, the digestive system breaks them into glucose, which then flows into the bloodstream. As the glucose level in the blood rise, the body begins producing insulin, which allows cells to turn glucose into energy or store them as glycogen[2] in the body. When this happens, the body trades the glucose for a different kind of energy, lowering the BGL. On the other hand, doing a hard workout will cause the BGL to drop because the muscles consume the energy stored. The body then produces glucagon, which turns glycogen back into glucose and increases BGL. This is an ongoing tug-of-war that the body is employing to maintain itself [7, 48].
Type 1 diabetes is an autoimmune condition in which the body's immune system destroys the cells in the pancreas that secrete the hormones insulin and glucagon. Without these hormones, the body cannot regulate the BGL. As a result, after consuming carbs, the BGL will increase until insulin is delivered through an injection. Alternately, if a person with diabetes exercises and the body consumes all the glucose in the muscles, the blood glucose level will plunge. The glucose level will continue to drop until carbs are ingested since the body does not have the glucagon hormone to convert glycogen into glucose. If neither situation is addressed immediately, it might result in major health issues like nerve damage and even fatality [15].
At the time of writing, there is no cure for Type-1 diabetes, implying the patient's glucose level must be constantly monitored and regulated with medical equipment. Regular BGL measurements are vital in diabetes care. Most patients in developed countries now have access to CGM[3]. The patient must determine and administer insulin dosages based on the meal's carbohydrate content by comparing the BGL before and after meals. Most patients in developed parts of the world have access to insulin pumps[4] which makes the process of

[2]Glycogen is how the body stores short-term energy, and is stored in the liver and muscles
[3]A CGM is an invasive sensor that detects blood glucose levels by penetrating the skin. The sensor typically lasts for 14 days
[4]An insulin pump is a medical device attached to the body that continuously provides insulin subcutaneously; an insulin reservoir typically lasts four days.

adjusting the BGL easier. The process of adjusting the BGL is continuous as circumstances change. Physical activity, dietary changes, and other unforeseen factors, such as illness, stress, and sleep patterns, can impact the BGL [28].

When calculating an insulin dose, some factors will increase or decrease the effectiveness of the insulin. This effectiveness is called the "insulin sensitivity factor" or "correction factor." And is an important basic indicator of how effectively the body reacts to insulin and is determined by eq. (2.5). The "Insulin Daily Total" is the total number of units from bolus and basal injections [5].

$$\text{insulin sensitivity factor} = \frac{100}{\text{Insulin Daily total}} \tag{2.5}$$

A few algorithms are implemented directly into insulin pumps to control the BGL; some are still being researched. The three main numerical algorithms that are used inside insulin pumps are Model Predictive Control (MPC), Proportional-Integral-Derivative (PID), and Fuzzy Logic (FL). MPC is an algorithm that anticipates future BGL values and changes insulin supply accordingly. Based on real-time glucose measurements, PID adjusts baseline insulin delivery to a target BGL (e.g., 126 mg/dL). The FL algorithm uses rules to mimic clinical reasoning to handle uncertainty and imprecision, such as individual variability in insulin sensitivity or dietary habits, to translate CGM sensor readings into insulin delivery actions [1, 44]. These control algorithms are tuned so that the distinctive characteristics of each algorithm dictate the aggressiveness of each system's response to BGL variations [25].

The algorithms which are still being researched are RNN and LSTM.

Because of the delayed effect of insulin and nutrients, the algorithm must use a predictive rather than a reactive approach to insulin control. The limitation of these algorithms is the lack of data. Specifically, they do not get any data on activity, and meals are only registered as a single number counting carbohydrates without the temporal information.

2.5 The Blood Glucose Level Prediction Challenge

The BGLPC [30] is a data science competition focusing on developing accurate BGL prediction algorithms. The challenge's objective is to foster research and development in the field of diabetes management, as well as collaboration between researchers and healthcare practitioners.

The challenge provides participants with a dataset containing sensor readings from sensors collecting different readings from individuals with type 1 diabetes and demographic and clinical characteristics. Among the sensor readings are the BGL, insulin injections, carbohydrate intake, and activity. The objective of the competition is to create a predictive model that can accurately predict these individuals' future BGL based on their previous glucose readings and other clinical data using machine learning and statistical techniques.

[5]Bolus injections are made before meals, whilst basal injections are made in modest doses throughout the day

2.6 Preliminary Project

The pre-project, "Diabetes management in-depth", was completed to build knowledge about the challenges of managing diabetes using algorithms, the existing approaches, and potential refinements to these approaches. A key takeaway from the pre-project is which factors affect the BGL and thus are essential to make predictions forward in time. Further, the existing methods found in the literature on AI led to the challenge described in section 2.5 and the OhioT1DM dataset used, an essential building block in this thesis. The pre-project concludes that even if predictions are accurate, their short-range forecasts do not provide any particular real-life use case. One reason there are only short-range predictions with high accuracy may be that they use too few data components when creating models and thus cannot achieve high accuracy on long-range forecasts. Lastly, the pre-project unveiled that some features can significantly affect the BGL and might improve model predictions on longer timeframes if included. The most important components are BGL, insulin, meals, and activity. Additionally, the temporal aspect of all these components should be included.

2.7 GSR

One of the values contained in the OhioT1DM dataset used in this thesis is the Galvanic Skin Response (GSR) value, which is a measurement of the skin's conductance- or electrodermal-activity. Multiple studies [56, 57, 52, 53] have shown that this value is highly correlated with the BGL, with their results giving them a correlation coefficient[6] (r) between 0.7 to 0.9. At the same time, RE Bolinger et al. [5] finds that BGL and GSR (r) for two of the patients in the OhioT1DM dataset's is -0.02 and -0.10 which indicates a poor correlation.

2.8 Missing Values

Missing values in time series data refer to situations where data points are absent or not recorded for certain periods. This can occur due to various reasons, such as equipment malfunction, data entry errors, or data corruption [40]. This is a challenge when forecasting future values in a time series dataset. The missing values can lead to biased or inaccurate forecasts, as the model may not have enough information to capture underlying patterns or trends. Additionally, missing values will cause issues in model training and evaluation, all models require a value for each observation.

There are a few common strategies for handling missing values in time series data. The first is *omission*, which means to remove the data point or the entire feature column. Another method is *imputation*, which involves replacing missing values with a substitute value, such as a constant or a value derived from other variables or a model [17]. One of these imputation approaches is *interpolation*, where missing values are estimated using the values of neighboring data points either linearly or as polynomials [54] as shown in fig. 2.2. Linear interpolation assumes a straight line between two known data points.

[6]The correlation coefficient is a value that ranges from -1 to 1, i.e., $-1 \leq r \leq 1$ which measures the correlation between two features x, and y

The forward or backward filling can also be employed, where missing values are replaced with either the preceding or following known value as shown in fig. 2.2, assuming that the time series data is relatively stable [54].

Lastly, the method of *extrapolation* leverages the predictive power of models like linear regression, MultiLayer Perceptron, or neural networks, to estimate missing values based on the patterns and correlations identified in the existing data [49].

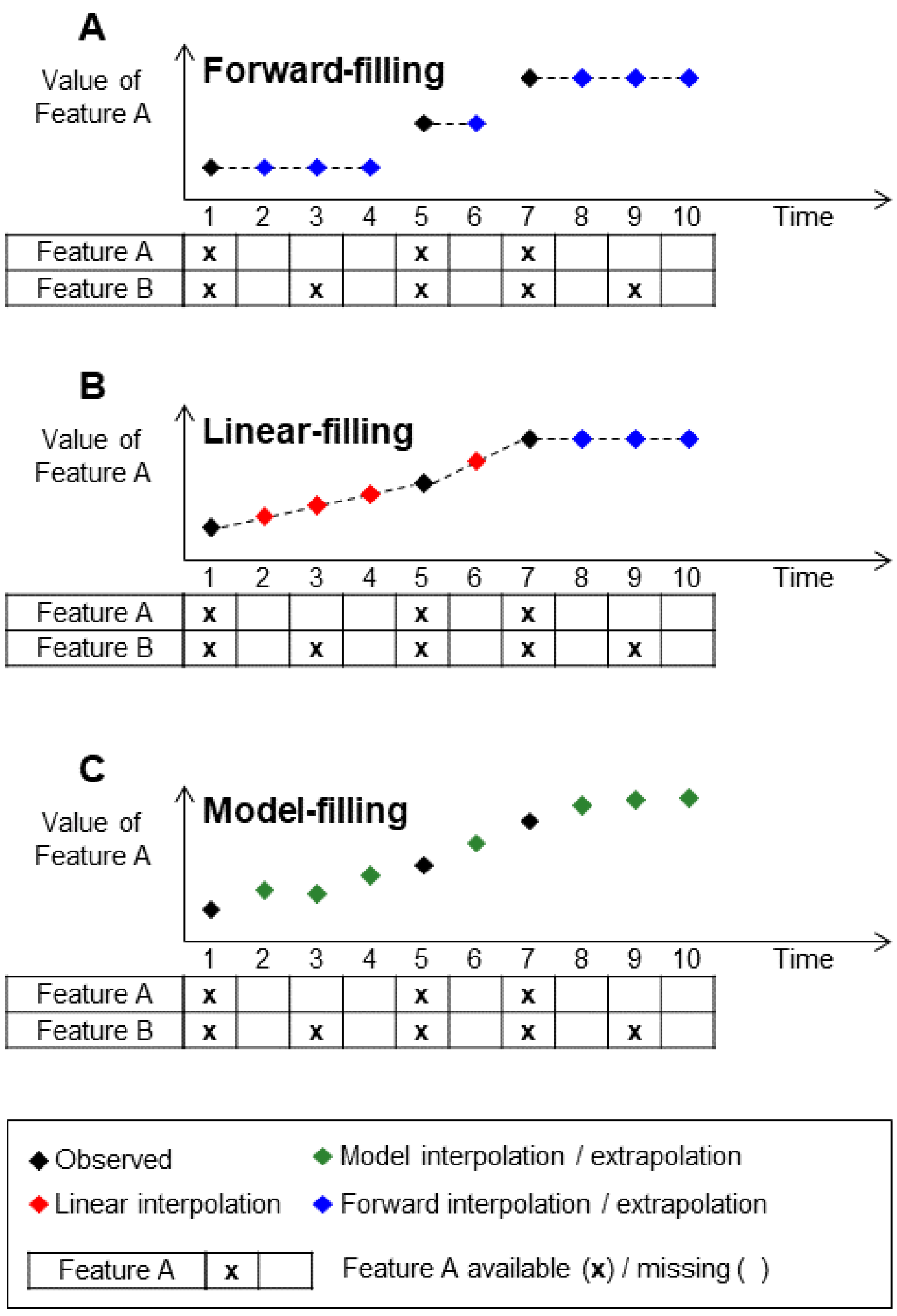

Figure 2.2: Different methods to handle missing values (adopted from [18])

2.9 Data Leakage

In machine learning, data leakage occurs when information crosses over between the training and test datasets, resulting in excessively high performance on the test set. The data leakage occurs when there is an overlap between the two sets, causing the model to memorize the training data and correctly output labels for test examples. This misleading evaluation results in significantly lower performance on truly unseen data after deployment [27].
Another form of leakage can occur when information from the future is carried backward in a time series. This can happen when using an imputation method like interpolation, and the range of imputed values is spread between two batches. For example, consider a simple time series dataset representing the BGL measured every 5 minutes as shown in table 2.1.

Table 2.1: Blood glucose levels every 5 minutes

Time	BG Level
00:00	110
00:05	115
00:10	-
00:15	130
00:20	135

Suppose the data is divided into two batches for training and validation:

- Batch 1: 00:00, 00:05, and 00:10
- Batch 2: 00:15 and 00:20

In this case, the BG level for 00:10 is missing. If linear interpolation is used to estimate the missing value for 00:10, it would be calculated as the midpoint between 00:05 and 00:15 (115 + 130) / 2 = 122.5. However, using this imputed value to train the model introduces data leakage because the information from 00:15 (a future time point) estimates the value for 00:10, which is part of the training data. The model may become biased and overfit, as it has indirectly accessed information from the validation set (Batch 2) during training.
The most straightforward technique for minimizing data leakage risk is holding back an evaluation subset during the model training. Holding back the subset, which is independent of the training set, can be used to evaluate the model's performance without incorporating any information from this dataset into the model's development. This helps ensure the model is evaluated on unseen data, providing a more reliable estimate of its performance in real-world scenarios.

2.10 Ensamble Model

Ensemble modeling [36, 21] involves executing multiple related analytical models and merging the outcomes into a unified score or spread. This approach is used in predictive analytics and data mining applications to enhance their evaluation metrics. There are three different methods used; *Bagging*, *Boosting*, and *Stacking*. The Bagging method starts with splitting the dataset into subsets, then homogeneously training one model per subset parallelly, and finally combining the resulting models. The Boosting method is similar to Bagging but trains the models sequentially instead of parallelly. Lastly, the Stacking method consists of two layers of models, the *Base Learner* layer and the *Meta Learner* layer. The base models are stacked parallelly like in the Bagging method, but unlike Bagging, the models are trained heterogeneously before their output is combined into the meta-learner.

Chapter 3

State-Of-The-Art

AI has become more prevalent in recent years and is theoretically very good at solving many problems. In particular, data and algorithms have helped researchers grasp complex mechanisms and develop more complex and interpretable algorithms, encouraging the clinical application of the AI technology [55].
In recent years, the medical field has witnessed a surge in the application of machine learning techniques, with numerous studies conducted on predicting the BGL. Researchers have long employed statistical models to solve the complex time series problem of diabetes. One such model is the ARIMA model [16], a forecasting technique that excels in dealing with linear and regular data. However, the blood glucose data obtained from patients often exhibit irregularities and non-linearity, making it difficult for such models to perform well.

3.1 Sequence Learning and Analysis

Deep learning is the most recent paradigm shift in computing. Using this tool in data-driven tasks has shown remarkable usefulness in identifying complex patterns from all types of datasets. Moreover, deep learning performs exceptionally in identifying patterns within sequential data, wherein the temporal dynamics of the data hold significant importance in information processing. Empirical evidence suggests that, in the realm of sequence data, RNNs surpass both Deep Neural Networks (DNNs) [47] and CNNs [34] as the most effective deep learning architectures. RNNs, equipped with their distinctive feedback loops, are adept at capturing long-range dependencies and temporal patterns, proving that they can provide a superior framework for tasks requiring analyzing time series data.
However, RNNs are not without their drawbacks. As the input sequence length increases, the gradient vanishing problem emerges, leading to the loss of earlier information. To address the vanishing gradient problem prevalent in RNNs, advanced model architectures such as LSTM [24] and GRU [12] are much used. These variants incorporate specialized gating mechanisms that facilitate the preservation of long-range dependencies, mitigating the vanishing gradient issue to a certain extent. However, the need to encode input data into a fixed-length hidden state vector creates an information loss problem. When dealing with long and complex sequences, the limited space cannot preserve all the essential information, affecting the model's prediction ability.
Elena Castilla et al. [8] examines many applications of time-series forecasting in several

sectors. The authors demonstrate in their paper that the Spatiotemporal Transformer Neural Network (STNN) model can recreate the phase space of a dynamic system. The proposed STNN model outperforms the ARIMA and Support Vector Regression (SVR). Using many high-dimensional short-term time-series datasets, they further examine the performance of the STNN framework. They randomly select four target variables from each dataset to be predicted. Each approach recorded the mean and standard deviation of its forecasts. As demonstrated in table 3.1, the proposed STNN model outperforms ARIMA, SVR, RNN, and other models across nearly all datasets regarding inference performance.
The results from this paper show that their transformer model outperforms a LSTM model, a RNN model, and statistical analysis models nine out of 10 times. These findings underscore the Transformer model's potential in forecasting time series data like BGL.

Table 3.1: Results on different datasets (adopted from [8])

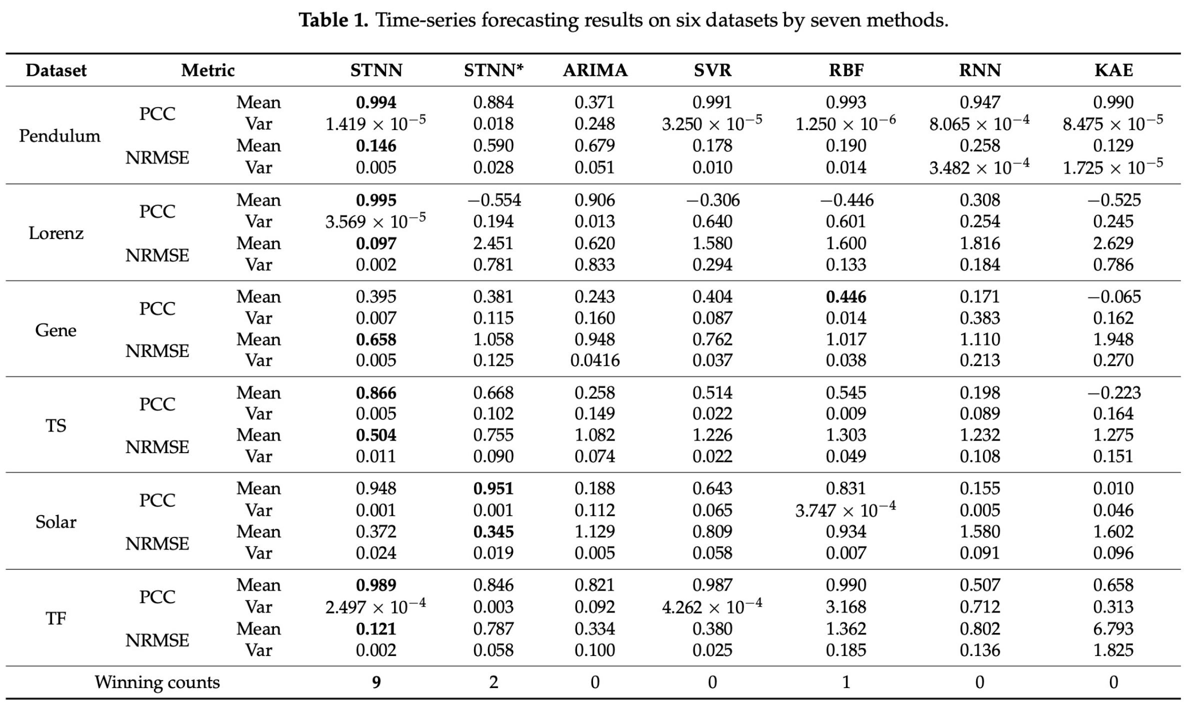

Table 1. Time-series forecasting results on six datasets by seven methods.

Dataset	Metric		STNN	STNN*	ARIMA	SVR	RBF	RNN	KAE
Pendulum	PCC	Mean	**0.994**	0.884	0.371	0.991	0.993	0.947	0.990
		Var	1.419×10^{-5}	0.018	0.248	3.250×10^{-5}	1.250×10^{-6}	8.065×10^{-4}	8.475×10^{-5}
	NRMSE	Mean	**0.146**	0.590	0.679	0.178	0.190	0.258	0.129
		Var	0.005	0.028	0.051	0.010	0.014	3.482×10^{-4}	1.725×10^{-5}
Lorenz	PCC	Mean	**0.995**	−0.554	0.906	−0.306	−0.446	0.308	−0.525
		Var	3.569×10^{-5}	0.194	0.013	0.640	0.601	0.254	0.245
	NRMSE	Mean	**0.097**	2.451	0.620	1.580	1.600	1.816	2.629
		Var	0.002	0.781	0.833	0.294	0.133	0.184	0.786
Gene	PCC	Mean	0.395	0.381	0.243	0.404	**0.446**	0.171	−0.065
		Var	0.007	0.115	0.160	0.087	0.014	0.383	0.162
	NRMSE	Mean	**0.658**	1.058	0.948	0.762	1.017	1.110	1.948
		Var	0.005	0.125	0.0416	0.037	0.038	0.213	0.270
TS	PCC	Mean	**0.866**	0.668	0.258	0.514	0.545	0.198	−0.223
		Var	0.005	0.102	0.149	0.022	0.009	0.089	0.164
	NRMSE	Mean	**0.504**	0.755	1.082	1.226	1.303	1.232	1.275
		Var	0.011	0.090	0.074	0.022	0.049	0.108	0.151
Solar	PCC	Mean	0.948	**0.951**	0.188	0.643	0.831	0.155	0.010
		Var	0.001	0.001	0.112	0.065	3.747×10^{-4}	0.005	0.046
	NRMSE	Mean	0.372	**0.345**	1.129	0.809	0.934	1.580	1.602
		Var	0.024	0.019	0.005	0.058	0.007	0.091	0.096
TF	PCC	Mean	**0.989**	0.846	0.821	0.987	0.990	0.507	0.658
		Var	2.497×10^{-4}	0.003	0.092	4.262×10^{-4}	3.168	0.712	0.313
	NRMSE	Mean	**0.121**	0.787	0.334	0.380	1.362	0.802	6.793
		Var	0.002	0.058	0.100	0.025	0.185	0.136	1.825
Winning counts			**9**	2	0	0	1	0	0

Haoyi Zhou et al. [67] investigate using a Transformer model for Long Sequence Time-series Forecasting (LSTF) and propose solutions for some of the limitations faced. The first limitation is that the quadratic dot-product calculation in the self-attention modules causes the complexity and memory usage per layer to become $\mathcal{O}(L^2)$. The second limitation is the inability to scale because of a memory bottleneck of $\mathcal{O}(J \cdot L^2)$ for larger time horizons when stacking encoder and decoder layers. Lastly, the dynamic nature of a transformer's decoding makes inference very slow when predicting long outputs.
The authors create an efficient transformer-based model, which they call *Informer*. It addresses the limitations in the following ways: Using ProbSparse self-attention, which reduces memory usage to time complexity to $\mathcal{O}(L \log L)$. Using self-attention distilling[1], to reduce the Value vectors for each attention layer in a pyramid-like scheme that halves the following layer inputs and optimizes the model for long input sequences. Their generative decoder

[1]Self-distillation in neural networks refines information from the deepest classifiers up to the shallower ones by connecting the shallow classifiers at different levels to different attention modules. [66]

predicts sequences of long output sequences simultaneously instead of one by one. This simultaneous nature further optimizes the prediction speed.
They evaluate the proposed model with four extensive datasets and demonstrate that their *Informer Model* outperforms the other types of models as shown in table 3.2 by at least 10% and offers a fresh approach to the LSTF problem.

Table 3.2: Informer Model Results (adopted from [67].)

Methods		Informer		Informer†		LogTrans		Reformer		LSTMa		LSTnet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh$_1$	24	**0.577**	**0.549**	0.620	0.577	0.686	0.604	0.991	0.754	0.650	0.624	1.293	0.901
	48	**0.685**	**0.625**	0.692	0.671	0.766	0.757	1.313	0.906	0.702	0.675	1.456	0.960
	168	**0.931**	**0.752**	0.947	0.797	1.002	0.846	1.824	1.138	1.212	0.867	1.997	1.214
	336	1.128	0.873	**1.094**	**0.813**	1.362	0.952	2.117	1.280	1.424	0.994	2.655	1.369
	720	**1.215**	**0.896**	1.241	0.917	1.397	1.291	2.415	1.520	1.960	1.322	2.143	1.380
ETTh$_2$	24	**0.720**	**0.665**	0.753	0.727	0.828	0.750	1.531	1.613	1.143	0.813	2.742	1.457
	48	**1.457**	**1.001**	1.461	1.077	1.806	1.034	1.871	1.735	1.671	1.221	3.567	1.687
	168	3.489	**1.515**	3.485	1.612	4.070	1.681	4.660	1.846	4.117	1.674	**3.242**	2.513
	336	2.723	1.340	2.626	**1.285**	3.875	1.763	4.028	1.688	3.434	1.549	**2.544**	2.591
	720	**3.467**	**1.473**	3.548	1.495	3.913	1.552	5.381	2.015	3.963	1.788	4.625	3.709
ETTm$_1$	24	0.323	**0.369**	**0.306**	0.371	0.419	0.412	0.724	0.607	0.621	0.629	1.968	1.170
	48	0.494	0.503	**0.465**	**0.470**	0.507	0.583	1.098	0.777	1.392	0.939	1.999	1.215
	96	**0.678**	0.614	0.681	**0.612**	0.768	0.792	1.433	0.945	1.339	0.913	2.762	1.542
	288	**1.056**	**0.786**	1.162	0.879	1.462	1.320	1.820	1.094	1.740	1.124	1.257	2.076
	672	**1.192**	**0.926**	1.231	1.103	1.669	1.461	2.187	1.232	2.736	1.555	1.917	2.941
Weather	24	**0.335**	**0.381**	0.349	0.397	0.435	0.477	0.655	0.583	0.546	0.570	0.615	0.545
	48	0.395	0.459	**0.386**	**0.433**	0.426	0.495	0.729	0.666	0.829	0.677	0.660	0.589
	168	**0.608**	**0.567**	0.613	0.582	0.727	0.671	1.318	0.855	1.038	0.835	0.748	0.647
	336	**0.702**	**0.620**	0.707	0.634	0.754	0.670	1.930	1.167	1.657	1.059	0.782	0.683
	720	**0.831**	**0.731**	0.834	0.741	0.885	0.773	2.726	1.575	1.536	1.109	0.851	0.757
ECL	48	0.344	**0.393**	**0.334**	0.399	0.355	0.418	1.404	0.999	0.486	0.572	0.369	0.445
	168	0.368	0.424	**0.353**	**0.420**	0.368	0.432	1.515	1.069	0.574	0.602	0.394	0.476
	336	0.381	**0.431**	0.381	0.439	**0.373**	0.439	1.601	1.104	0.886	0.795	0.419	0.477
	720	0.406	0.443	**0.391**	**0.438**	0.409	0.454	2.009	1.170	1.676	1.095	0.556	0.565
	960	**0.460**	**0.548**	0.492	0.550	0.477	0.589	2.141	1.387	1.591	1.128	0.605	0.599
Count		33		14		1		0		0		2	

To summarize the findings from this section, it is clear that the Transformer-based Informer and STNN models stand out. With the Informer model offering approximately 10% better performance than the LSTM model and the STNN model surpassing traditional models 90% of the time, it is evident that Transformer models hold considerable promise for enhancing the precision of time series data predictions.

3.2 Advances in Glucose Forecasting

Many research papers [37, 33, 46, 20, 13, 2, 31, 45, 68, 9, 39] are available on blood glucose prediction, including many different approaches to machine learning models for improving the accuracy and thus, reliability of the predictions that the model produces.
J Freiburghaus et al. [19] propose a CRNN fig. 3.1 model for predicting the future BGL of people with type 1 diabetes using four different features. The model is evaluated using the metrics RMSE and Mean Absolute Error (MAE) on different time horizons. Their methodology splits the dataset in two: *2018* and *2020*. The *2018* set is used to create a pre-trained model, and the *2020* set is split into the individual patients and used to fine-

tune the pre-trained model for each of them. Their model prediction results are listed as the best in the competition, as shown in table 3.3 below as *Paper ID 5*. The authors also note that adding more features influencing the BGL, such as stress or illness, remains to be researched. Reviewing their code unveils that missing values are solved with forward-filling as mentioned in section 2.8. It also reveals that the basal insulin feature values are assigned to one timestamp. In the real world, this value denotes the hourly rate, and when this dose is unchanged for many hours, a discrepancy emerges between the preprocessed dataset and the real world. Further, this discrepancy may negatively impact the model's predictive accuracy because of the missed data. Both of these discrepancies are areas that can be improved upon.

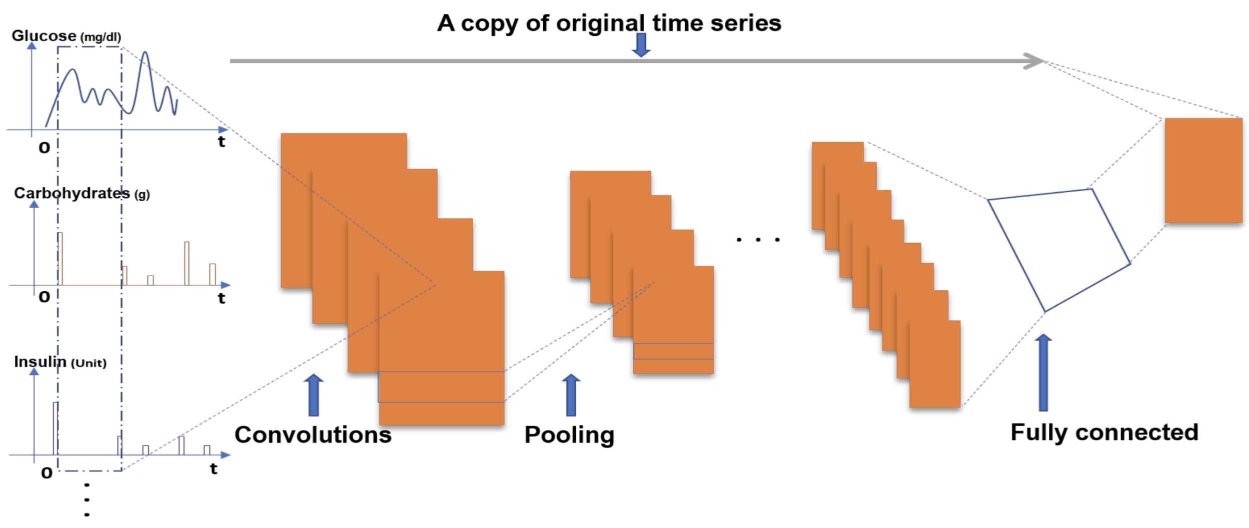

Figure 3.1: Convolutional Recurrent Neural Network (adopted from [19])

Table 3.3: Blood Glucose Level Prediction Challenge Results (adopted from [59])

	30 min		60 min				
Paper ID	RMSE	MAE	RMSE	MAE	**Overall**	Online	Personalized
5	17.45	11.22	33.67	23.25	85.59	No	Yes
13	18.22	12.83	31.66	23.60	86.31	No	Yes
6	19.21	13.08	31.77	23.09	87.15	No	Yes
16	18.34	13.37	32.21	24.20	88.12	No	Yes
15	19.05	13.50	32.03	23.83	88.41	No	Yes
1	18.23	14.37	31.10	25.75	89.45	No	No
14	19.37	13.76	32.59	24.64	90.36	Yes	Yes
8	19.01	13.73	33.37	24.98	91.09	No	Yes
11	18.99	13.73	33.39	25.04	91.15	No	Yes
4	19.79	13.62	33.73	24.54	91.68	No	Yes
7	19.60	14.25	34.12	25.99	93.96	No	Yes
9	20.03	14.52	34.89	26.41	95.85	Yes	Yes
2	21.80	15.00	35.00	25.00	96.80	Yes	Yes

Mehrad Jaloli and Marzia Cescon [26] describe deep learning-based algorithms for forecasting BGL in people with type 1 diabetes. They propose a model that uses a CNN and LSTM units to predict the BGL for different time horizons using historical glucose measurements, meal information, and insulin injections. The model is evaluated on two datasets, *Replace-BG* and *DIAdvisor*, and achieves excellent performance compared to other approaches on the same datasets. Specifically, the model yielded RMSE values of 9.28 ± 1.31 (mg/dL) and 9.81 ± 0.91 (mg/dL) for the 30-minute horizon, 16.51 ± 2.19309 (mg/dL) and 18.32 ± 2.76 (mg/dL) for the 60-minute horizon, and 23.45 ± 3.18 (mg/dL) and 25.12 ± 4.65 (mg/dL) for the 90-minute horizon on the *Replace-BG* and *DIAdvisor* datasets, respectively.

SM Lee et al. [37] propose a deep learning framework to forecast BGL in patients with type-2 diabetes using a Transformer model. The framework utilizes the encoder part of the Transformer to perform regression and classification under a unified framework. The model is trained and evaluated on a dataset containing a week of data with the BGL, collected from patients with type-2 diabetes. The authors face a class imbalance problem[2] since the events of hyperglycemia[3] and hypoglycemia[4] occur so seldom. They use a Generative Adversarial Network (GAN) model to generate augmented glucose time series data to solve this.

Another problem they face is less data from type-2 diabetes patients. They overcome this problem by using transfer learning[5], using the type-1 diabetes dataset OhioT1D to create a base model, then finetuning this model with their Type-2 data. As their results show in table 3.4, their transformer model provides accurate prediction results for both their Regression and Classification experiments, thus detecting hyperglycemia and hypoglycemia in the patient data.

Cross Validation	**Model**	**Experiment**	**Metrics**	**12 Step**	**24 Step**	**36 Step**
5-Fold	Transformer	Regression	MAPE	13.76	15.36	14.94
			STD	0.32	0.7	0.85
		Classification	F1 Score	0.68	0.66	0.71
			STD	0.02	0.036	0.01

Table 3.4: Blood Glucose Level Prediction (adopted from [37])

Their approach proves a transformer model can predict BGL. Still, since their study only relies on BGL when training the model and disregards other critical factors that influence blood glucose, a comprehensive and accurate predictive model that includes these features remains to be developed. The human body's glucose regulatory system is complex and individual variability in response to medication, diet, and activity, as mentioned in section 2.6. Since the study is centered around type-2 diabetes, their approach can be further developed to construct a transformer model for type-1 diabetes, which can predict exact numerical values.

Tomas Koutny and Michael Mayo. [33] propose a new method for forecasting BGL that are both easily understandable and have low complexity. This approach uses a rule-driven model instead of a complex deep learning model. The researchers identify multiple patterns of glucose fluctuations by extracting relevant features from the OhioT1DM dataset and subsequently conduct experiments to predict patterns. As part of the evaluation process,

[2]The class imbalance problem refers to the unequal distribution of labels in a dataset, leading to biased learning and poor performance of machine learning algorithms.

[3]Hyperglycemia is a condition where blood glucose levels are abnormally high

[4]Hypoglycemia refers to abnormally low blood glucose levels

[5]Transfer learning is a strategy in AI where a pre-trained model is fine-tuned to perform a related task, improving efficiency.

they use the Surveillance Error Grid (SEG)[6] SEG to verify that they are inside the acceptable relative error. Their model's relative error is less than 30% in 96.3% of instances when forecasting BGL on a 30-minute timeframe. The limitation of this study is that the relative error was considered zero in cases where the BGL exceeded the predetermined upper or lower limit. Given that AI models outperform simple rule-driven models, it means that there is room for further research in using complex ML models.

Nemat et al. [46] employ deep learning models and ensembling to strengthen the management of type 1 diabetes by predicting BGL at 30 and 60-minute intervals using the OhioT1DM dataset. In the study, three distinct models are employed, namely linear regression, Vanilla LSTM (VLSTM), and Bidirectional LSTM (BiLSTM) [20]. The three ensembling techniques evaluate the models; Stacking, Multivariate, and Subsequence. Their model is evaluated with a RMSE of 19.63 (mg/dL) and a MAE of 13.88 (mg/dL), showing that their ensemble model outperforms the pre-existing models they compare to by RMSE 15.64% and MAE 17.91%. Further, their stacking ensembling approach shows the highest level of performance among their models. Their method uses two LSTM models. This show there is room for more research on using advanced models such as the Transformer model in the ensembling.

Daniels et al. [13] employ a CRNN Multitask Learning (MTL) approach to develop individualized models for type-1 diabetes patients using the OhioT1DM dataset. Their MTL approach is a type of transfer learning that produces accurate models for each individual in the dataset, despite using all the data simultaneously with patients in batches. This is achieved by backpropagating the error in each batch that tunes the model weights to each patient in specific layers, which is used to train shared layers in the model that are generalized for all the patients. This method is used to tackle the problem of inter-individual variability in the BGL predictions. The dataset is divided into two groups based on the patient's blood glucose variability to increase the generalized part of the model and its accuracy. The performance of their model is compared to a SVR model. The model is evaluated on multiple forecast ranges between 30 and 120 minutes, where the 30-minute range was the most accurate. Their MTL approach outperformed the other model with a RMSE of (18.8$\pm$2.3) (mg/dL) and a MAE of (13.2$\pm$1.6) (mg/dL). These results indicate that the patient's data is hard to train as a generalized model and opens the possibility of researching models that can handle inter-individual variability. Mario Munoz-Organero. [45] develops a physiological prediction model to assist individuals with type 1 diabetes. The author uses the D1NAMO dataset comprised of actual patient data; in addition, they use the AIDA simulator[7] to generate additional data. Utilizing CGM data, carbohydrate digestion, and insulin absorption processes as features, an LSTM model is employed to forecast BGL for a time horizon ranging from 30 to 60 minutes forward in time. The prediction of BGL resulted in RMSEs of 6.42 (mg/dL) and 11.35 (mg/dL) for the actual patient data at 30 and 60 minutes, respectively. In addition, the simulator-generated data yielded an RMSE of 3.45 (mg/dL) at the 30-minute mark.

Taiyu Zhu et al. [68] employ a Dilated Recurrent Neural Network (DRNN) [9] model architecture to forecast BGL 30 minutes forward in time. Their motivation to develop the

[6]The Surveillance Error Grid labels the clinical risk an inaccurate blood glucose monitor holds [32].

[7]AIDA is an open-source software application that enables the dynamic simulation of plasma insulin and blood glucose patterns, designed for pedagogical demonstrations, instruction, and self-learning purposes [38].

DRNN is the challenges of vanishing gradients and inefficient parallelization that typical RNNs encounter when processing extended sequences. In a DRNN with skip connections, some layers are connected directly to layers that are not adjacent to them in time, allowing the network to have access to information from past or future timesteps, thus affecting the model's network's ability to capture long-term dependencies and reducing the vanishing gradient problem that can arise in such models. They incorporate the hidden state preceding the skip length rather than relying on the technique of reflecting the initial hidden state to the RNN. The skip length is an important parameter in the DRNN; it refers to the number of timesteps the network jumps over between two consecutive layers, which in their experiment was set to (1,2,4). The study uses the OhioT1DM dataset to conduct a comparative evaluation analysis of the model's performance against the following models; autoregressive, SVR, and CNN. The results in RMSEs for the autoregressive model, SVR, and traditional neural networks, are 20.1, 21.7, and 22.9 (mg/dL), respectively. In contrast, the DRNN model achieved a RMSE of 18.9 (mg/dL), indicating superior performance.

Kezhi Li et al. [39] propose a CRNN model as a deep learning approach for predicting BGL 30 and 60 minutes into the future in type-1 diabetes patients. The model uses a convolutional layer incorporating a 1D Gaussian kernel filter and a RNN layer with LSTM cells to acquire knowledge of long-term dependencies. The study utilizes data from 10 patients generated by the UVA/Padova simulator[8] and a clinical dataset of 10 patients. The experiment utilizes both actual patient data and synthesized patient data. The outcomes of the experiments are satisfactory for the 30-minute prediction horizon, as evidenced by the virtual patient data recording RMSE of 9.38 (mg/dL) and Mean Absolute Relative Difference (MARD) of 5.50 (mg/dL), as well as the actual patient data recording RMSE of 21.07 (mg/dL) and MARD of 11.61 (mg/dL). Expanding on the last three papers mentioned, it's plausible that leveraging cutting-edge machine learning models like the Transformer, which is shown promising results in sequential prediction tasks, could further refine the accuracy of predictions on the same dataset.

To summarize the state-of-the-art in blood glucose prediction, there are many different models and datasets, but there is a lack of transformers utilized in the research and with the dataset used in this thesis. In addition, as in the previous section about sequence learning and analysis, there is room for more research using transformer models to predict BGL from time series data. Table 3.5 summarizes the state-of-the-art results on BGL prediction found in this section.

[8]The UVA/Padova Simulator [42] has received approval from the FDA and can be utilized as a replacement for human subjects in pre-clinical trials aimed at assessing the efficacy of blood glucose control algorithms.

Table 3.5: Results from the models

Authors	**Method**	**Dataset**	**RMSE 30 min (mg/dL)**	**MAE (mg/dL)**
J Freiburghaus et al.	CRNN	OhioT1DM	17.45	11.22
Mehrad Jaloli and Marzia Cescon	CNN and LSTM	Replace-BG DIAdvisor	9.28 ± 1.31 9.81 ± 0.91	5.63 ± 0.96 6.60 ± 0.76
Daniels et al.	CRNN	OhioT1DM	18.8 ± 2.3	13.2
Mario Munoz-Organero	LSTM	D1NAMO	6.42	-
Mario Munoz-Organero	LSTM	AIDA sim.	3.45	-
Taiyu Zhu et al.	DRNN	OhioT1DM	18.9	-
Li et al.	CRNN	UVA/Padova sim.	21.07	-
Nemat et al.	Ensemble LSTM	OhioT1DM	19.63	13.88

3.3 Summary

In summary, the literature presented in this chapter highlights the noteworthy body of research accomplished in recent years that focuses on time series forecasting, particularly within the context of predicting BGL in individuals with diabetes. The growing interest in diabetes management algorithms underscores the potential of this approach to make meaningful contributions while **simultaneously** indicating the complexity and ongoing challenges associated with such predictions.
The various methods and AI model architectures employed to tackle this complex problem range from traditional time series analysis to advanced ML techniques. They leverage deep learning models, such as RNNs and CNNs, to predict the BGL with high accuracy and efficiency. They have also begun exploring ML approaches initially designed for natural-language processing tasks, such as Seq2Seq models and Transformers. These techniques, which excel compared to RNNs and CNNs at capturing context and dependencies in textual data, show great potential for modeling the intricate temporal patterns intrinsic to the BGL, which could further enhance prediction accuracy and advance the field.
In the current research on type-1 diabetes prediction models, there are still gaps in the research on transformer models used to overcome the problems of traditional RNN models like vanishing gradient and memory bottlenecks in long time-series datasets. As mentioned in section 3.1, the Transformer-based model results are approximately 10% better than a LSTM model when predicting a time series, which indicates there is potential to use a Transformer model to improve the prediction of BGL.
Moreover, using additional critical features with their temporal information and further improving on the data preprocessing technique could eliminate irrelevant or noisy data and thus enhancing the accuracy and general applicability of the model's predictions and ultimately improving patient care in type-1 diabetes management.

Chapter 4

Method

This chapter outlines the methodologies utilized in this thesis to forecast BGLs for type 1 diabetes patients. The aim is to use a Transformer model to outperform the state-of-the-art model used in BGL forecasting. The chapter is divided into several sections, each describing process aspects. The chapter will explain the components shown in fig. 4.1 in section 4.1 through section 4.3.
The section labeled “Dataset and preprocessing" (section 4.1) pertains to the OhioT1DM dataset utilized in the experiments, as well as the preprocessing procedures implemented to prepare the data for the model. The section clarifies the feature selection process, which is based on the importance of the features when interpreting the BGL.
In section 4.2, the LEET technique is presented. This technique is specifically developed to effectively manage missing values in the dataset and mitigate the potential leakage issue which may arise when the interpolation imputation method is applied to the evaluation data.
The section labeled “Model Architecture" (section 4.3) presents a summary of the different models employed in the research. It is subdivided into three subsections, namely 4.3.1 “Encoder Only Transformer Model," 4.3.2 “Informer Model," and 4.3.3 “LSTM Model," which provide a comprehensive description of the individual models employed in the research.
“Performance Metrics" (section 4.4) provides a summary of the evaluation metrics utilized for assessing the models' performance, namely MSE, RMSE, and MAE.
This chapter thoroughly comprehends the methodologies and techniques employed in this research, thus creating the groundwork for the subsequent chapters demonstrating the outcomes and findings.

4.1 Dataset and preprocessing

The dataset [43] used in the experiments is the one mentioned in section 2.5 and is referred to as the OhioT1DM dataset. The dataset is acquired under a Data Use Agreement between Ohio University and the University of Agder. Due to its extensive use in research, this dataset is excellent for confirming findings through comparative analysis. And because it includes several individuals and has an appropriate resolution on the timesteps (5 minutes), this diabetes dataset is substantially higher quality than others that can be found online. Half of the data was initially released for the competition held in 2018, and the other half was released for the next round held in 2020. The dataset contains eight weeks of data for

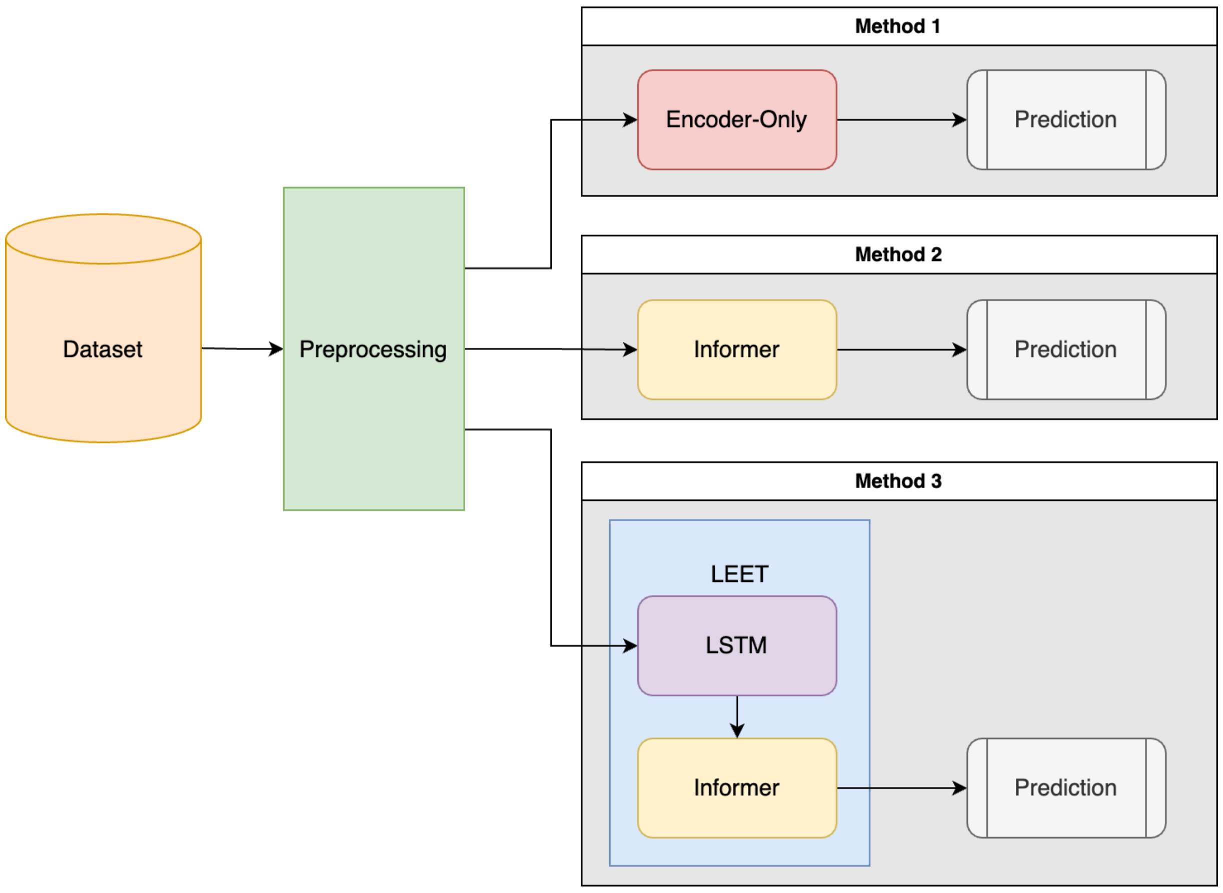

Figure 4.1: Prediction Methods

12 patients with type 1 diabetes, and each patient's data is split into training and testing files with 20 data fields. All patients in the dataset used a CGM sensor, an insulin pump, a fitness band that recorded physiological data, and a mobile app that the patients used to register the rest of the data fields.

The most significant features are BGL, insulin, meals, and activity, as concluded in the pre-project mentioned in section 2.6. Additionally, the features, heart rate, GSR, steps, and sleep were included to give the model more data and, thus, the best circumstance for correctly decoding the patients' physiologies. Further, to get the best model from the dataset, it must be interpreted correctly before preprocessing to get the correct timeline of events. As noted in section 3.2, the researchers have made an error by using the basal insulin feature's value at a single timestep. The correct way to interpret this value is *units per hour* from its timestamp to the next value's timestamp, thus making the feature a continuous injection throughout the day instead of a simultaneous one.

The following list contains the data points extracted and used in training. Figure 4.2 shows some of the features from patient 559 in the dataset. Out of the 20 data fields the dataset contains, 14 of them are chosen based on research done in the preliminary project explained in section 2.6. It should be noted that two of the features in the figure, namely *air_temp*, and *skin_temp*, were not used.

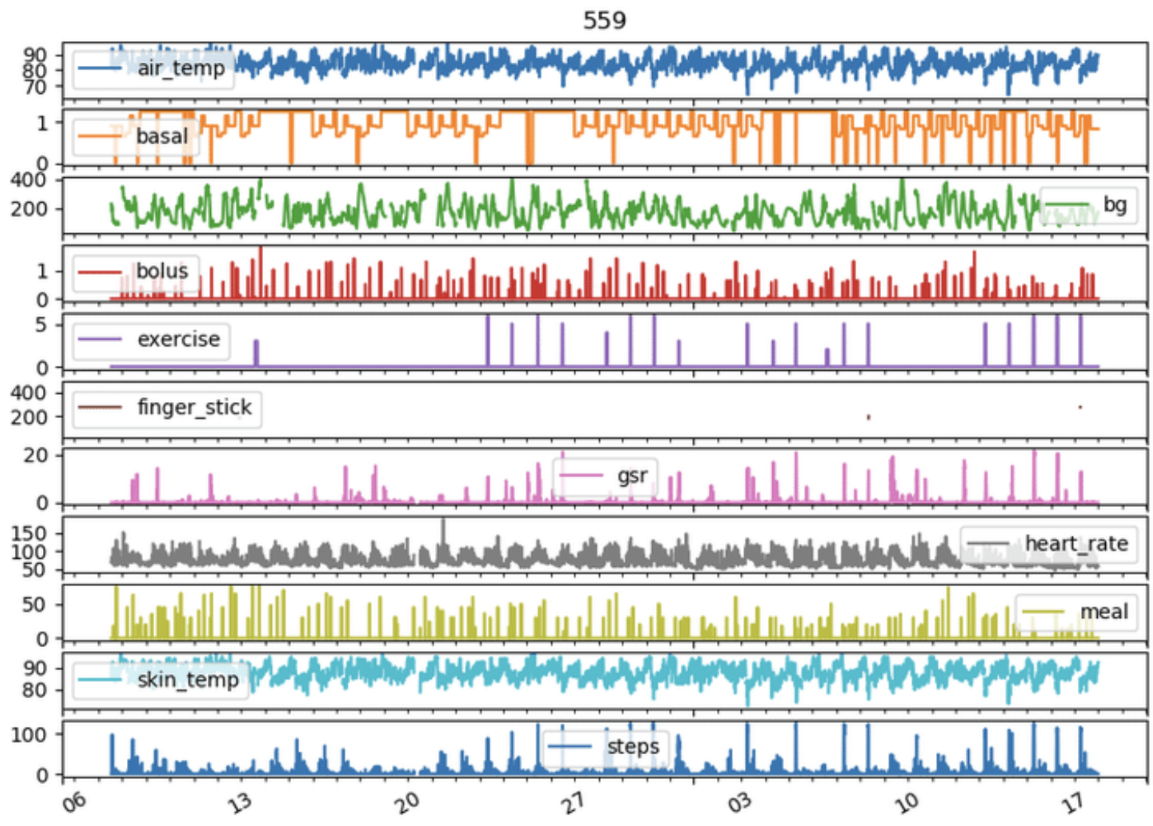

Figure 4.2: Patient 559 training data (Adopted from [63])

- Blood Glucose Level (bg), which is recorded by a CGM sensor in 5-minute intervals.
- Fingerstick, which is a manual BGL measurement often used either when waiting for a newly applied CGM sensor to complete calibration or to double-check the CGM value's correctness. This value is extracted to fill the BGL value gaps that occur when a CGM sensor is calibrating.
- Insulin consists of two main types of injection doses: continuously delivered (basal) and On-demand (bolus) insulin. The following list shows the sub-types with their slight variations that must be accounted for differently.
    1. Normal Bolus and Dual Normal are injected simultaneously at the timestamp.
    2. Square is simultaneous at the timestamp only if the start value equals the stop value, else it is administered throughout the time interval.
    3. The Square Bolus is administered throughout a time interval.
    4. Basal is the hourly rate of insulin injected between two timestamps and is often varied throughout the day to adapt to the activity level throughout the day.
    5. Temp Basal overrides the Basal with a temporary hourly rate between two timestamps.
- Meal is recorded as a value of how many grams of carbohydrates are consumed at that time.
- Heart rate is registered in 5-minute intervals.

- GSR value is registered once per minute.
- Steps are summarized in 5-minute intervals and registered at the timestamp.
- Exercise is registered with a timestamp, duration (in minutes), and an intensity value between 1 and 10, assessing the patient's activity.
- Sleep is registered with two timestamps and a value accounting for some patients' sleep quality.

The data is extracted from XML files and preprocessed, then saved as CSV files. It is preprocessed so that all the features have a common time dimension into which they can be inserted to create a multivariate time series. The preprocessing also adheres to the rules from the BGLPC mentioned in section 2.5. The common time dimension for all the data is created starting from the first BGL value and increases by five minutes for each new value until the final value. Since there is a discrepancy in the features timestamp in the dataset (ex., Meal: 10:11, Setps: 10:13), the timestamps are set to the previous 5-minute timestamp. As mentioned in section 2.8, missing values, as shown in fig. 4.3, in a dataset that is used for ML must be addressed. Since the BGL values are used as part of the model's input and output, introducing a new issue where the time step with a missing value also contains essential data points like meals or insulin must be avoided. The omission method mentioned in section 2.8 will not be used. As mentioned in section 2.4, the optimal range for BGL is between 70 - 130 (mg/dL). Moreover, the patients in the dataset range between 39 - 477 (mg/dL). This means that zero-filling the missing BGL values will introduce an error between 39 - 477 (mg/dL) for each missing value. To reduce this error, the data points from *finger stick* are used to fill some of the gaps in BGL. This approach reduces the number of missing values, but as fig. 4.4 shows, zero values are still present.

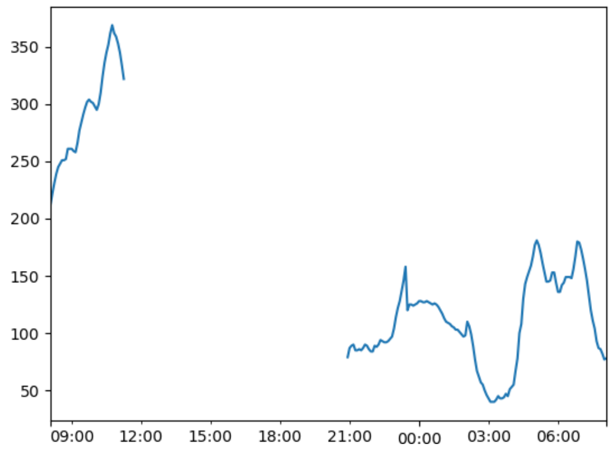

Figure 4.3: Missing Values

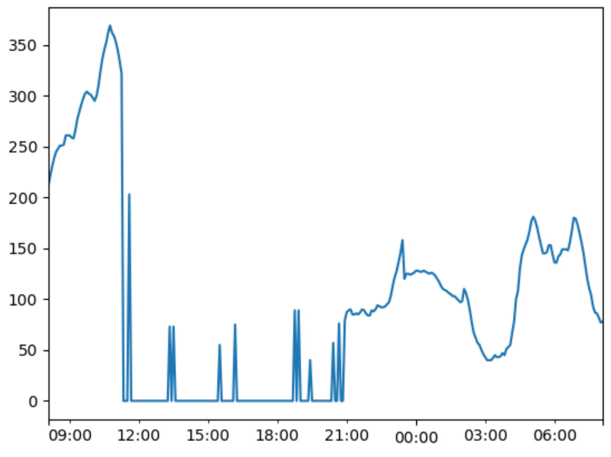

Figure 4.4: Fingerstick and Zeros

From this point, the preprocessing takes the two paths described below, where the data is processed differently for the various models and experiments.

1. The first path is to split the dataset into three subsets, training, validation, and testing. Zachary C. Lipton, David C. Kale, and Randall Wetzel [41] propose a method of adding a new feature column to indicate a missing value in the timestep. This method is used in combination with one of the three tactics below to handle the missing values:

   Tactic 1: The missing values are kept at 0 in all the subsets (zero-filled).

   Tactic 2: The missing values are imputed with linear interpolation only in the training and validation subsets.

   Tactic 3: The dataset is split into a training, validation, and testing subset. The missing values are imputed with linear interpolation in all the subsets.

2. The second path is to split into two subsets, *subset a* and *subset b*, with their divisions being 20% and 80%, respectively.

   Subset a: Only the BGL feature column is kept, and all missing values are removed. This subset is then used to train and test the first model in the LEET technique explained in section 4.2.

   Subset b: Uses the first model produced in the LEET method to impute all the missing values. Subset b is further divided into three subsets for training, validation, and testing.

The linear interpolation method used on the training data is the same as the leading paper [19] from the challenge described previously. Using interpolation is against the challenge's rules because it can introduce data leakage, as mentioned in section 2.9, but it is mitigated by **not** interpolating the testing subset. As fig. 4.5 shows, the graph is now continuous and never toches zero.

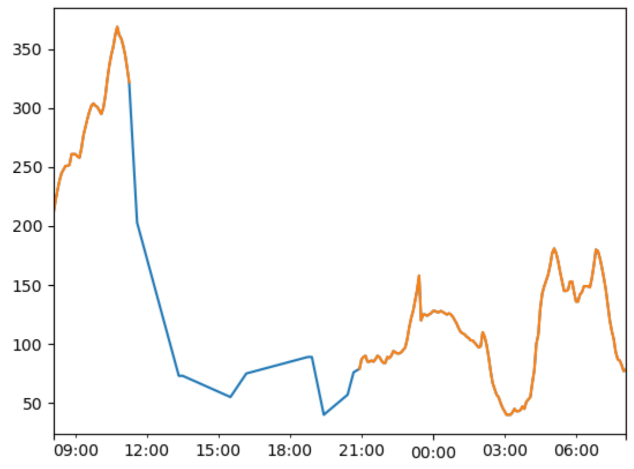

Figure 4.5: Interpolated

The following list is the key takeaways from the preprocessing

1. Read each patient's XML files, first the training file, then the testing file separately.
2. Before adding any values, a time series starting from the first BGL value's timestamp and ending at the last BGL value's timestamp is created, with a delta time of 5 minutes. All feature values' time steps are moved back to the prior whole 5-minute timestep. Given the necessity of this adjustment of the values, it should be noted that this procedure may introduce an error in the relation of one value to another of up to 5 minutes when moved.
3. **BGL** values are added to the data frame with the adjusted timestamp, as explained in step 2.
4. **Finger stick** values are added only to the timestamps where the CGM values are missing.
5. **Meal** values are added to their corresponding timestamp in the data frame.
6. **Basis heart** rate values are added to their corresponding timestamp in the data frame.
7. **Basis GSR** values are added to the data frame to their corresponding timestamps.
8. **Basis steps** values are added to their timestep in the data frame.
9. **Bolus** data's four subcategories are handled as follows: Normal Bolus and Normal Dual Bolus values are added to their timestamp directly. Square Bolus and Square Dual Bolus values are divided by the number of 5-minute intervals between their start and stop values and then added to the data frame.
10. **Basal** and **Temp basal** values are divided between their start value and the next basal value start value, then added to the data frame.
11. **Exercise** is added to the data frame after first compressing the duration to the prior 5-minute step (e.g., if the duration is 52 minutes, the new duration is 50 minutes). Secondly, the duration is divided by five to determine the number of time steps to which the intensity value should be added.
12. **Basis sleep** values are set as "1" in all the timestamps between the start and stop timestamps.

The dataset remains unshuffled primarily for two reasons. Firstly, the highest-performing outcomes in the challenge are achieved using non-shuffled data, which clearly advocates for maintaining the original sequence of the data. Secondly, because of the chronological order of time series data is essential for capturing temporal dependencies and patterns within the dataset.
Since time series data exhibits a temporal sequence, future observations often hinge on historical data. Disrupting this chronological ordering through shuffling could potentially compromise the model's ability to generate accurate predictions. This is particularly pertinent in the context of Transformer models, whose performance is predicated on recognizing and learning from sequential dependencies in the data.

4.2 Layered Ensemble with Enhanced Training

As mentioned in chapter 3, Nemat et al. use an ensemble method that delivers superior results when used. This thesis explores this method as a solution to impute the missing BGL values. The proposed method combines the Bagging and Boosting ensemble methods to form a custom technique named LEET. As fig. 4.6 shows, and explained in section 4.1, the dataset is split into subsets a and b, where subset a contains 20% of the data and is used to train and evaluate the LSTM Model. This model is then used to impute the missing values in subset b, which then is used to train, validate, and test the Informer Model.

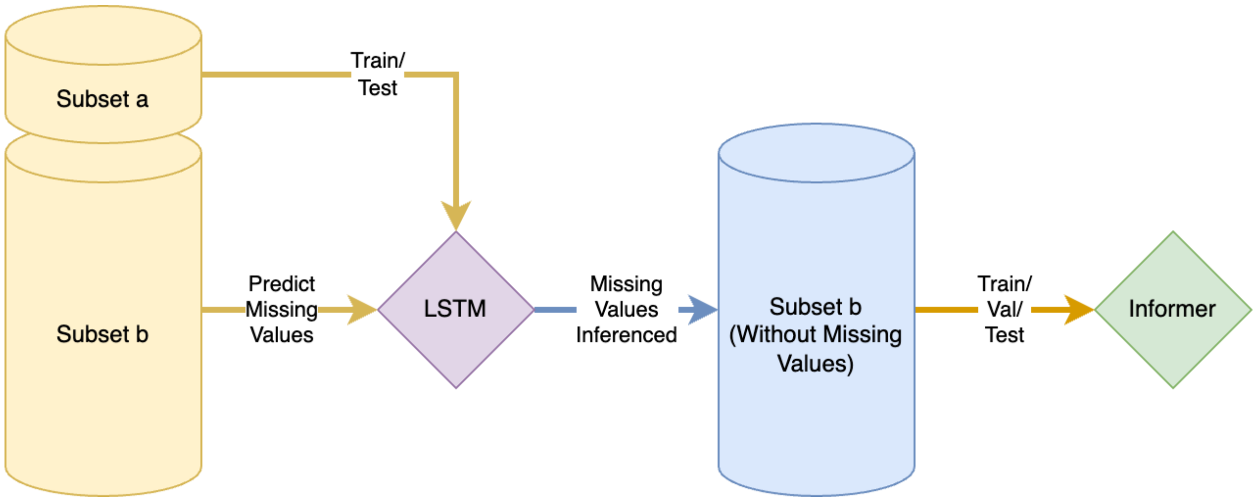

Figure 4.6: Layered Ensemble with Enhanced Training

The LEET method is used as an imputation method to solve the problem of missing values, which causes the evaluation metrics to be subpar because the loss is calculated between a predicted value of ex. 250 (mg/dL) and a missing value that is set to be 0 (mg/dL), which will always be worse because the BGL will never go below a value of 35 (mg/dL) in a human. Another problem that LEET solves is the leakage problem that the interpolation imputation method can cause when used on the testing data, as explained in section 2.9, where future values cause leakage.

4.3 Model Architecture

This section outlines the models used in this thesis, used either as disjunct models tested in a head-to-head comparison or conjunct utilizing the LEET technique. The first two (*Encoder-Only Model* and *Informer Model*) are Transformer models with some differences, but both use multivariate-multistep to forecast a sequence. The last model (*LSTM Model*) is a less complicated single-variate LSTM model that predicts the next value. These models are detailed below in section 4.3.1 through section 4.3.3.

4.3.1 Encoder-Only Model

This model consists of a specified number of Transformer encoder blocks, each containing a multi-head self-attention layer and a fully connected network. The model uses transformer encoder blocks to encode the input time series data and then uses a fully connected network

Multilayer Perceptron (MLP) to generate predictions. In the model, each block combines several layers to process incoming data. This includes layer normalization, multi-head self-attention, and a feed-forward network with 1D convolutional layers. The overall model architecture is built by sequentially connecting the transformer encoder blocks, followed by a global average pooling layer, dense layers, and a fully connected output layer.
The list below provides an overview of the main hyperparameters used in this model. Table B.1 in appendix B lists the exact values for these parameters.

- **Activation Function (activation):** The activation function used in the model's neurons.
- **Batch Size (batch_size):** The number of training examples utilized in one iteration.
- **Early Stopping (early_stopping):** The percentage of neurons that are randomly turned off during training to prevent overfitting.
- **Number of Epochs (epochs):** The number of complete passes through the entire training dataset.
- **Feed-Forward Network Dimension (ff_dim):** The dimension of the feed-forward network model.
- **Prediction Horizon (horizon):** This refers to the dimensionality of the output space of each attention head in a Transformer model.
- **Learning Rate (learning_rate):** The size of the steps taken to reach a local minimum of the loss function.
- **Look-back Period (look_back):** The length of the sequence to use as input to predict the next time step.
- **MLP Dropout Rate (MLP dropout):** The dropout rate specifically for the MLPs in the model.
- **Number of MLP Units (MLP units):** The number of neurons in the MLP.
- **Number of Attention Heads (number of heads):** This refers to the number of parallel attention layers (or "heads") in the Transformer model.
- **Number of Transformer Blocks (num_transformer_blocks):** The number of Transformer blocks in the model.

4.3.2 Informer Model

The Informer model implemented by Haoyi et al. [67] mentioned in section 3.1, is a further development of the Transformer model that improves efficiency for long input sequences in time-series. The Informer model's high-level architecture consists of two primary components: an encoder and a decoder. The encoder and decoder work seamlessly to process extensive sequential inputs and generate associated outputs. As shown in fig. 4.7, the encoder sequentially processes the raw input data, compressing it into a more understandable format. This bundled output is then transmitted to the decoder's multi-head attention head, and lastly, a fully connected layer generates the final sequence output. The data is passed from the encoder to the decoder. The encoder first processes the input data by refining it into a more compact form before passing it to the decoder. The decoder generates the output using the processed data, a start token, and a placeholder for the target sequence. Including a start token allows the decoder to produce the output in one forward procedure, significantly improving the more time-consuming dynamic decoding process. The model's performance is then evaluated by the loss function that employs MSE between the predicted and target sequences, clearly measuring the model's effectiveness.

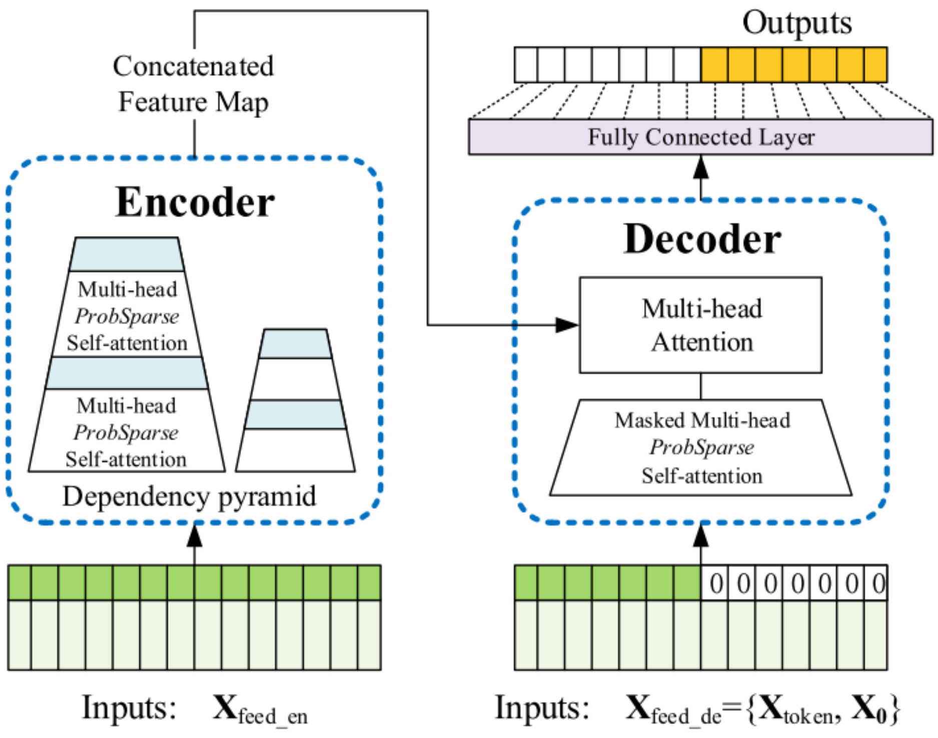
Concatenated Feature Map
Outputs
Fully Connected Layer
Encoder
Multi-head *ProbSparse* Self-attention
Multi-head *ProbSparse* Self-attention
Dependency pyramid
Decoder
Multi-head Attention
Masked Multi-head *ProbSparse* Self-attention
0 0 0 0 0 0 0
Inputs: $\mathbf{X}_{\text{feed\_en}}$
Inputs: $\mathbf{X}_{\text{feed\_de}}=\{\mathbf{X}_{\text{token}}, \mathbf{X}_{\mathbf{0}}\}$

Figure 4.7: High Level Informer model Architecture (adopted from [67])

The encoder first takes action, transforming long sequential inputs into a matrix representation. Using a mechanism known as ProbSparse self-attention[1], the encoder identifies and focuses on the most relevant features within the data. To maintain effective control over memory usage, the encoder uses an operation that progressively condenses the time dimension of the input data. This operation occurs across different layers, with each successive layer halving the time dimension of its predecessor. Thus, the output from each layer becomes the input for the next, creating the pyramid-like structure portrayed in fig. 4.7. The culmination of this process is a robust self-attention feature map characterized by reduced redundancy. The outputs of all layers are then concatenated, forming the final hidden representation of the encoder. A single stack inside the encoder is shown in fig. 4.8. It is the primary stack taking the entire input sequence of data. The red layers called *Attention Block 1, 2, and 3* are dot-product matrices that take the output from the prior layer. As the

[1]ProbSparse self-attention [10] is a modification of the self-attention mechanism in Transformer models that makes attention probabilistically sparse, allowing each token to attend to a fixed number of other tokens, thus reducing computational complexity from quadratic to linear for longer sequences.

figure shows, each subsequent block is halved in size by applying **self-attention distilling** explained in section 3.1 to the layer, which enables efficient handling of very long input sequences. Lastly, all the stacks' feature maps are concatenated and returned as the encoder's output.

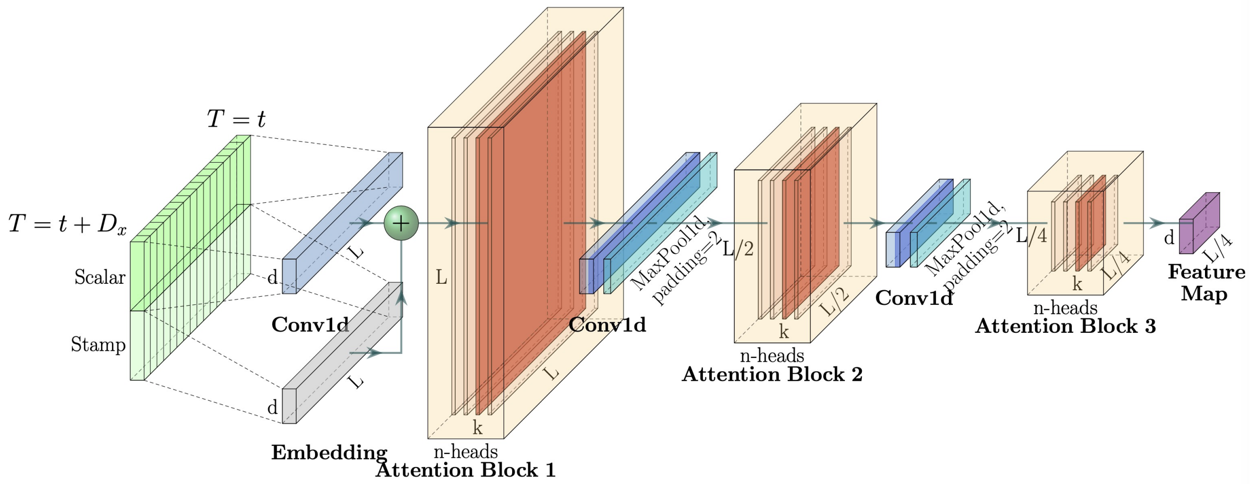

Figure 4.8: A single stack in the encoder (adopted from [67])

The decoder, also seen in fig. 4.7, uses the bundled output from the encoder. Using a generative inference approach, the decoder can create long sequential outputs more efficiently than traditional dynamic decoding. It receives a start token and a placeholder for the target sequence as inputs. The start token is derived from an earlier slice of the input data, while the placeholder is initialized to zero. Once the inputs are ready, the decoder employs masked multi-head attention, as explained in section 2.3.2. This strategy safeguards each position from attending to future positions, avoiding auto-regression. Finally, the output is predicted by a fully connected layer.

4.3.3 LSTM Model

The objective of this model is to take a sequence of BGL values and predict the next value. The LSTM model architecture, as shown in the general architecture in fig. 4.9, starts with an input layer receiving a sequence of BGL values. Following this, there are two LSTM layers, each with 50 neurons. The first LSTM layer is designed to return sequences, meaning it passes its output to the next LSTM layer. The second LSTM layer does not return sequences; instead, it directly passes its output to the following layer. The last layer in this architecture is a Dense layer with one neuron, a fully connected layer that outputs the prediction.

The LSTM model predicts BGL values one step (5 minutes) into the future using the previous six steps (30 minutes). To train and test the LSTM model, test data from both the 2018 and 2020 datasets are used. All missing BGL values from the subset used are removed to ensure the model's integrity.

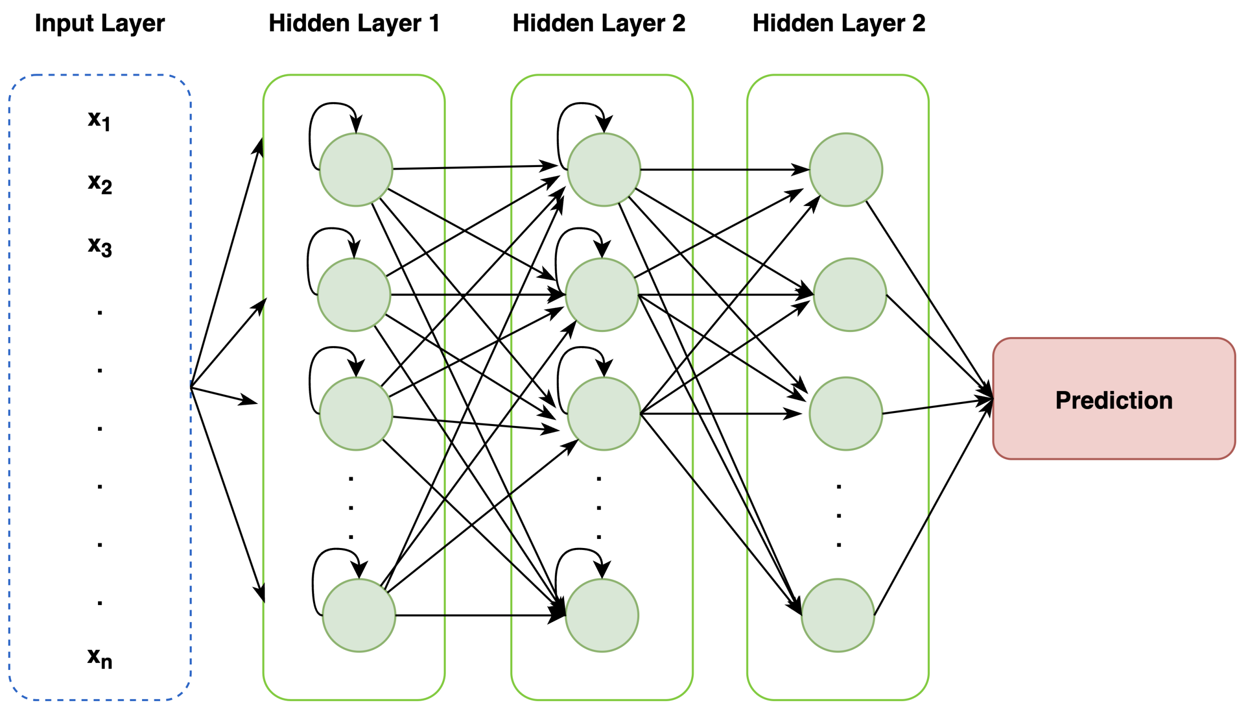

Figure 4.9: Long Short-Term Memory Model Architecture

4.4 Performance metrics

This section presents an overview of the diverse range of error metrics proposed for evaluating the performance of model predictions in regression models [11].
MSE eq. (4.1) is a widely used metric for evaluating regression models, as it measures the average squared difference between predicted and actual values. For each prediction, it computes the average of the squared errors, with more significant errors having a greater impact due to the squaring operation.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.1}$$

RMSE eq. (4.2) is the square root of the mean squared error. It provides an interpretable measurement of the prediction error, described in the same units as the target variable, i.e., BGL. It is advantageous because it is described in the same units as the original data, allowing a more intuitive comprehension of model accuracy and prediction error magnitude. As the equation below shows, n is the total number of observations, y_i is the true value, $\hat{y}_i$ is the predicted value, and $(...)^2$ indicates the square of the difference.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.2}$$

MAE eq. (4.3) is a widely used metric for evaluating regression models, as it measures the average of absolute differences between predicted and true values. The MAE is simple to understand and works well for applications where reducing the model's average error is the main objective. As the equation below shows, n is the total number of observations, y_i is the true value, $\hat{y}_i$ is the predicted value, and $|...|$ indicates the absolute value.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.3}$$

RMSE and MAE are options for evaluating the difference between predicted and actual values because they provide an intuitive way to compare the performance of various models. Additionally, the results from BGLPC in section 2.5 use these performance metrics to predict BGL 30 minutes into the future. Thus, the results obtained in the present paper are likewise exhibited utilizing identical metrics, thereby enabling their validation through comparative analysis. The RMSE and MAE are helpful tools for assessing a model's performance. They are not sensitive to extreme outliers and can be very helpful when the errors are symmetrically and evenly distributed through the dataset. They are used to evaluate a regression model's performance and are often used in conjunction with each other.

Chapter 5

Experiments and Results

This chapter presents the experiments that test all the hypotheses stated in section 1.4.2. The experimental results are quantified using the performance metrics RMSE and MAE as mentioned in section 4.4, and every experiment and its result is discussed in each section or subsection. Some experiments and their results included in this chapter do not reflect the final results. They are included merely to give further insight into what works best out of multiple configurations and, thus, which direction forward is the most promising.
The experiments in this thesis are completed using Nvidia Tesla V100 32GB GPU, Python 3.8, and PyTorch 1.12. They are logged and plotted using the MLOps platform Weights and Biases [62]. The code is written and tested using Apple's M1 chip, which has GPU acceleration enabled in PyTorch and TensorFlow via the Metal Performance Shaders Graph (MPS).

5.1 Preliminary Experiments

This section will look at two preliminary experiments that do not test any of the hypotheses but are conducted to verify and confirm insights. The first experiment validates the experimental results from the paper leading the BGLPC mentioned in section 2.5, and the second unveils statistical insights into the dataset.

5.1.1 Validation of Previous Results

This preliminary experiment examines and executes the leading contributor's code to reproduce and validate their published results. As mentioned in chapter 3, they split the dataset into two subsets. The first subset containing the 2018 data is further split into a train and validation subset, which are used to train their proposed CRNN model. The second subset, comprising the 2020 data, is further split into a train, validation, and test subset, which are used to train and evaluate one fine-tuned model for each patient. Their score is then calculated by the average of all the patients RMSE which in this experiment results in an average RMSE of 17.45 (mg/dL) and MAE of 11.22 (mg/dL).
Discussion: The preprocessing on the 2020 subset has a concerning line of code that causes data to leak from the train and validation sets into the test set, which should only contain unseen data, as shown in listing 5.1. The snippet is pulled from the public code repository [29]. The leakage problem invalidates their published result, and after fixing their issue and

re-training the model, the new results have a RMSE of 18.49 (mg/dL) and a MAE of 12.11 (mg/dL), which is close to their published results but still remains on the top of the BGLPC ranking.

Listing 5.1: Leakage code snippet from pre-processing

```
# Load 80% of the data
df_train = all_train_df[train_contrib_id]
...
# split ratio
split_idx = int(len(df_train) * 0.8)
df_val = df_train.iloc[split_idx+48:]
df_train = df_train.iloc[:split_idx]
...
df_test = all_test_df[test_contrib_id]
...
# Data leakage:
df_test = pd.concat([df_train.dropna().iloc[-12:], df_test], axis=0)
```

5.1.2 Analysis Experiment

In this preliminary experiment, the dataset is preprocessed similarly to the other contributors in the BGLPC. The data is then used to train a model sequentially with one patient's data at a time to produce the best model for each patient. And finally, the results are analyzed. This experiment is meant to ascertain if the order of training the patients matters and if the generalized model gives different results from one patient to another. The model is trained using the Transformer outlined in section 4.3.1. It is trained on one patient at a time without shuffling, as described in section 3.2. K-Fold Cross-Validation ensures that the data is reasonably distributed and that there are no clusters of biased data.

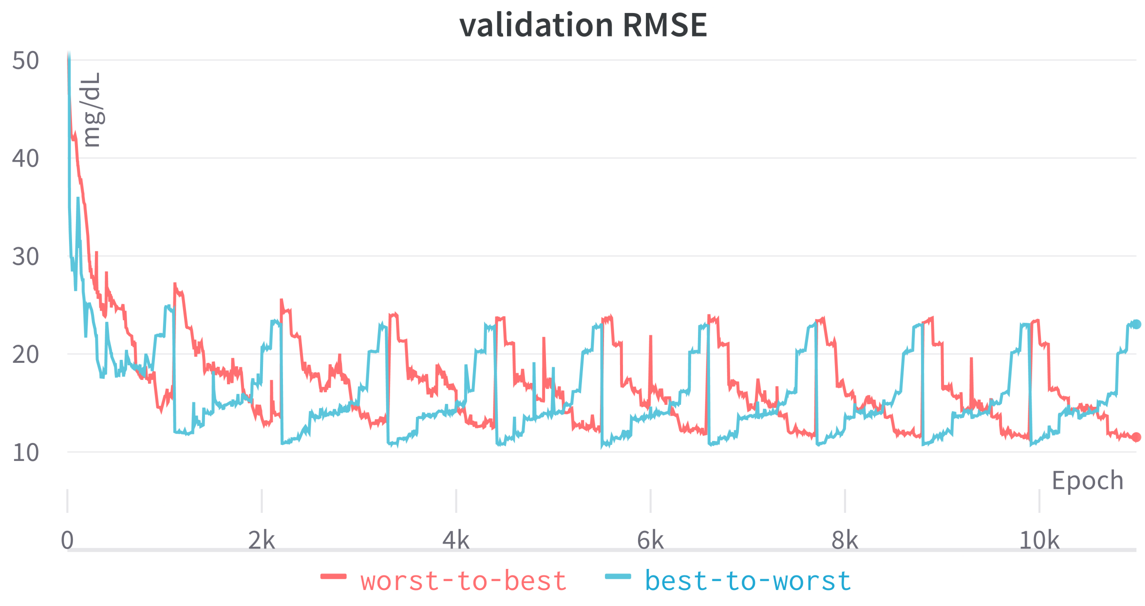

Figure 5.1: One-by-one

Discussion: The resulting model is generalized for all the patients but reveals some unexplainable differences in the patients' data as shown in fig. 5.1. The figure also shows the order of training the patients, from “best" to “worst" or in the opposite order, does not affect the model. These results conclusively tell us that some inherent discrepancies in the patients' data make it challenging to make a generalized model for them.

5.2 Transformer models

In this section, two different approaches to the Transformer model are used to test **hypothesis 1**, which states that **“A Transformer model can outperform the state-of-the-art CRNN model proposed by J Freiburghaus et al. [19] by predicting BGLs 30 minutes into the future with lower RMSE on the same dataset."** The experimental results from the models in this section are compared to the state-of-the-art model as described in section 3.2.

5.2.1 Experiment 1: Encoder-Only

This experiment tests **hypothesis 1** by using a Transformer model with only the encoder, as explained in section 4.3.1 to outperform the state-of-the-art model. The model is trained on the 2018 subset, which is then fine-tuned on all the patients in the 2020 subset at once to create **a generalized model for all patients**. This is similar to the approach mentioned in section 3.2, except that they train **one model for each patient** in the 2020 subset. During the experiments, adjusting the number of feedforward layers in the neural network slows the training process substantially and results in a lower loss. Various activation functions are tested but do not improve the model performance.

The Transformer model shows good results when evaluated on both the training and validation sets, with a RMSE values of 15 (mg/dL) and 11 (mg/dL), respectively, and a MAE values of 9.5 (mg/dL) and 7.8 (mg/dL), respectively. The model's efficacy drops when applied to the unseen test data, as seen by the lesser RMSE value of 27 (mg/dL) and MAE of 24.2 (mg/dL). A detailed graphical illustration of these findings can be found in fig. 5.2.

Discussion: Multiple factors can cause this experiment's poor performance of the encoder-only model. First, the model looks overfitted on the training and validation data, resulting in poor generalization and a low score when evaluated with unseen test data. Another explanation could be that the inherent complexity of BGL prediction is better suited for a CRNN model, like the one from section 3.2, which combines convolutional layers for feature extraction and recurrent layers for temporal relationships. The encoder-only model, in contrast, relies entirely on self-attention mechanisms, which may not be as effective at capturing the time-series nature of BGL data. In addition, using only the encoder from the Transformer model may limit its ability to learn complex patterns in the data since it does not have the full sequence-to-sequence powers of a complete encoder-decoder Transformer model. This conclusively shows that the Transformer model, which uses an encoder-only architecture, underperforms compared to the state-of-the-art model. Thus, hypothesis 1 is rejected.

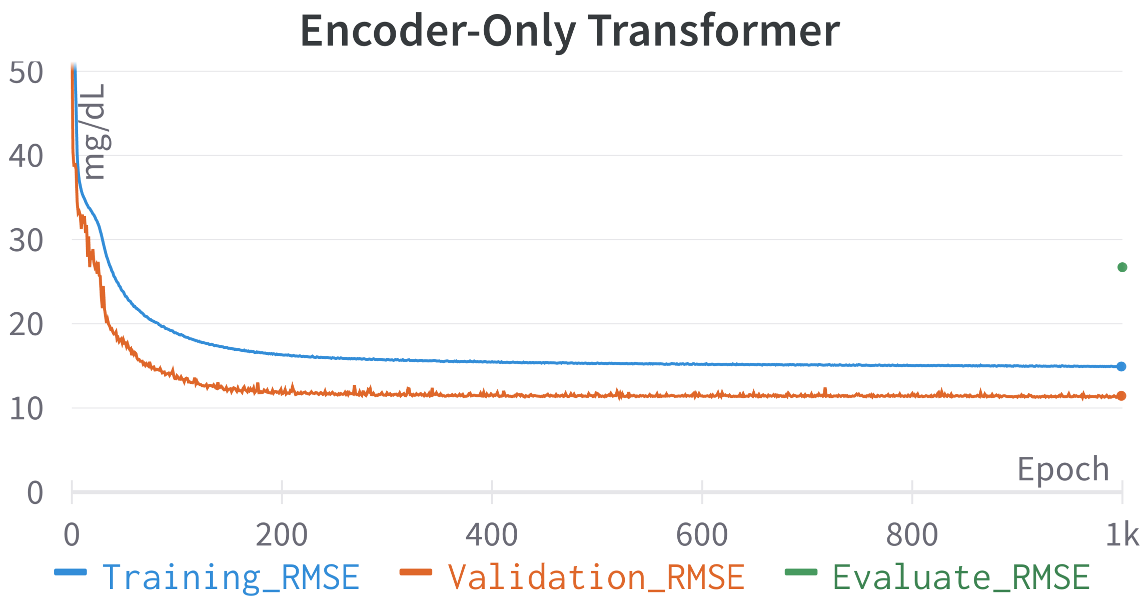

Figure 5.2: Transformer Model With Only Encoder

5.2.2 Experiment 2: Encoder-Decoder

In this experiment, **hypothesis 1** is tested by using the Informer model mentioned in section 3.1, a transformer model with an encoder-decoder model architecture as explained in section 4.3.2 to outperform the state-of-the-art model. This experiment contains multiple runs and, thus, multiple models, where the main differences are how the missing values in the dataset are handled. The first is where they are addressed by zero-filling, and in the second, they are managed by linear interpolation. Further, numerous tests are completed to optimize the model and its results. These tests included various varieties of features, hyperparameter tuning, and different data-splitting methods between the training, validation, and testing sets. Figure 5.3 shows that the model trained with interpolated data outperforms the model trained on non-interpolated data. The best result is obtained using interpolation, which gives a RMSE of 14.76 (mg/dL) and a MAE of 11 (mg/dL), while the best results for non-interpolated data produce a RMSE of 21.4 (mg/dL) and a MAE of 11.2 (mg/dL).
Discussion: The Informer is a great leap from previous implementations of transformer models with their new ingenious methods to increase efficiency, giving the ability to test different hyperparameters swiftly. A key component of this model is its decoder, which is crucial for forecasting future values. It generates future steps auto-regressively, leveraging its preceding outputs as context for upcoming predictions. The decoder's attention mechanism helps to focus on relevant parts of the input sequence, while the masked self-attention preserves the data's temporal sequence. When experimenting with the hyperparameters in this model, no matter how they are tweaked, the default hyperparameters provide the best results, indicating that they are highly optimized for this model. The Informer model also performs better than the encoder-only from the previous experiment, no matter how the dataset is preprocessed. Table A.1 in appendix A shows a concise overview of the results from different tests. The results from the experiment conclusively show that the Informer model underperforms the state-of-the-art model. Thus, hypothesis 1 is rejected.

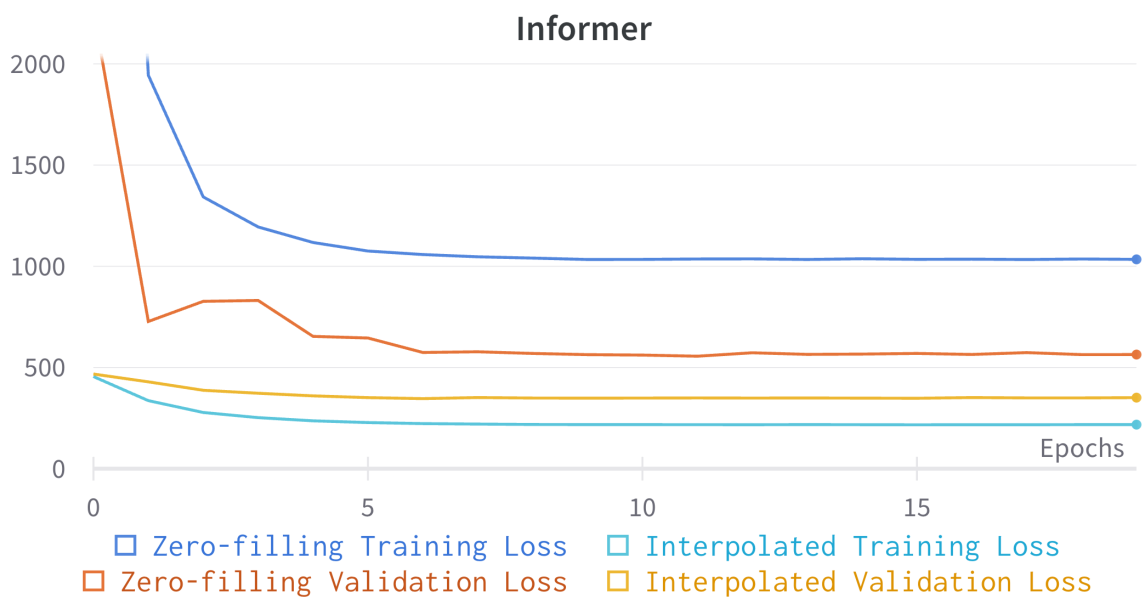

Figure 5.3: Informer Results

5.3 Missing Values

Hypothesis 2 states that: **"Intelligent handling of missing BGL values, rather than zero-filling, could boost model accuracy."** This hypothesis is tested with three experiments outlined in section 5.3.1 through section 5.3.4. The following experiments were performed at different points over the project's timeline. As a result, the outcomes are not directly comparable due to the modifications made to the underlying code at each stage. These changes, critical to the iterative development and refinement process, introduced changes in each experiment's operating conditions, thus influencing the comparability of the results.

5.3.1 Experiment 3: Extra Column

In this experiment, **hypothesis 2** is tested by training two different models. Their respective performance metrics are compared to ascertain the superior model. The first model is trained using the initial feature columns as discussed in section 4.1. The second model is trained with an extra column named "missing_bg." The extra column contains a binary integer value to indicate all the timesteps where there are missing values in the "BGL" column. The resulting models from this experiment are evaluated with the performance metrics RMSE and MAE. The model with the extra column "missing_bg" achieves a RMSE of 34 (mg/dL) and a MAE of 25.6 (mg/dL). The model without the column achieves a RMSE of 43 (mg/dL) and a MAE of 34.5 (mg/dL). The graphs in fig. 5.4 also show that the model trained with the extra column is superior to that trained without the new column.

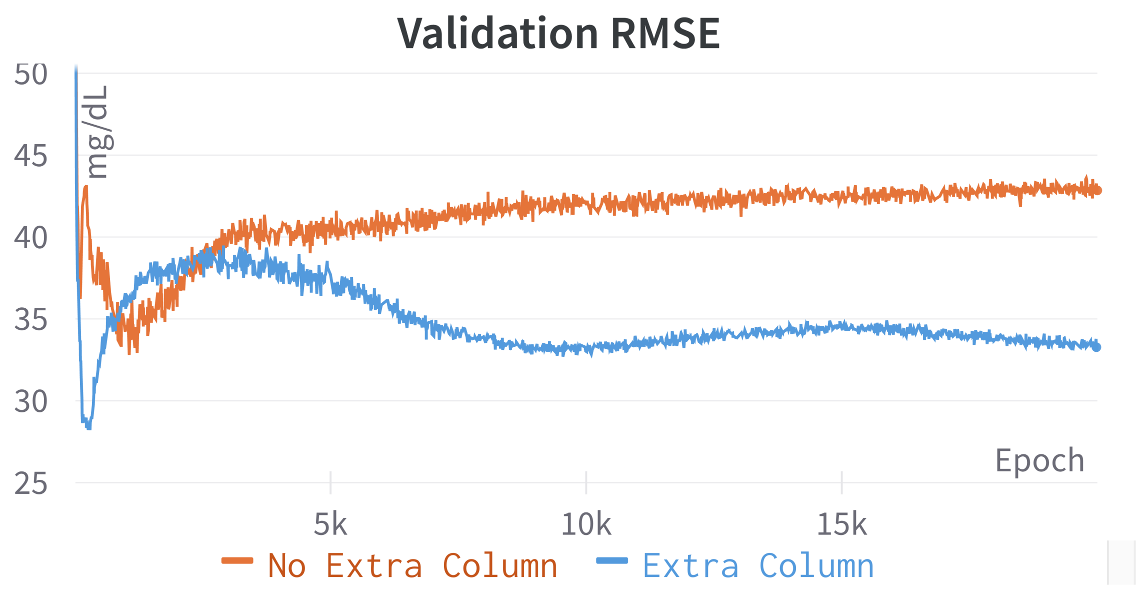

Figure 5.4: Missing Values - Extra Column

Disscussion: Including the "missing_bg" column as an indicator for missing values provides additional context about the dataset to the model, successfully turning missing values from a challenge into a usable feature. The decrease in the RMSE metric results from the model's ability to understand patterns and dependencies related to the missing values, which leads to more accurate predictions. This conclusively shows that intelligent handling of missing values instead of zero-filling can boost the Encoder-Only model's accuracy, thus confirming hypothesis 2.

5.3.2 Experiment 4: Interpolation

In this experiment, **hypothesis 2** is tested by training two different models. Their respective performance metrics are compared to ascertain the superior model. Both these models use the extra column from the prior experiment to enhance the result further. The first model is trained using the zero-filled data, and the second is trained using interpolated data. The models' resulting validation RMSE are 19 and 12, respectively, as shown in fig. 5.5.
Discussion: The results show that the model trained on interpolated data is superior to that trained on non-interpolated data. The experiment illustrated in the figure uses the Encoder-only model. In addition, the same results are also reflected when using the Informer model mentioned in section 5.2.2. This conclusively shows that intelligent handling of missing values instead of zero-filling can boost the model's accuracy, thus confirming hypothesis 2.

5.3.3 Experiment 5: Ensemble Method Part 1

This experiment is the first part of the LEET method mentioned in section 4.2, which uses two distinct models to yield the best results. This experiment does not directly test any of the hypotheses but is an experiment to evaluate the best-performing Model 1 out of two options. The first option is using GSR, and the second is using BGL to predict BGL, and the best one is used in the next experiment to construct the best-performing ensembled model.

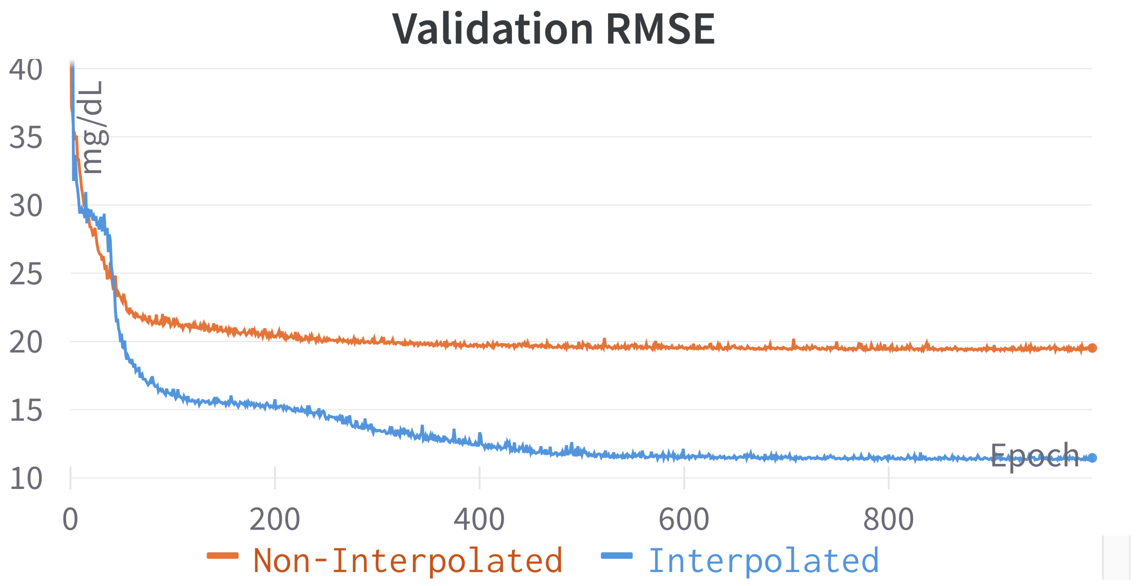

Figure 5.5: Missing Values - Interpolation

The model used in this experiment is detailed in section 4.3.3 and is utilized to handle the missing values in the dataset. And as explained in section 4.2, the dataset is split into two subsets, where subset a is used in this experiment to train the first of the two ensembled models.

The first alternative of this experiment employs GSR to predict the BGL value in the same timestep. The trained model yields the evaluation RMSE of 64 (mg/dL). Further, using a simple statistical analysis as shown in fig. 5.6, reveals that the best correlation coefficient between any of the patients' GSR and BGL in the dataset is $r = -0.12$.

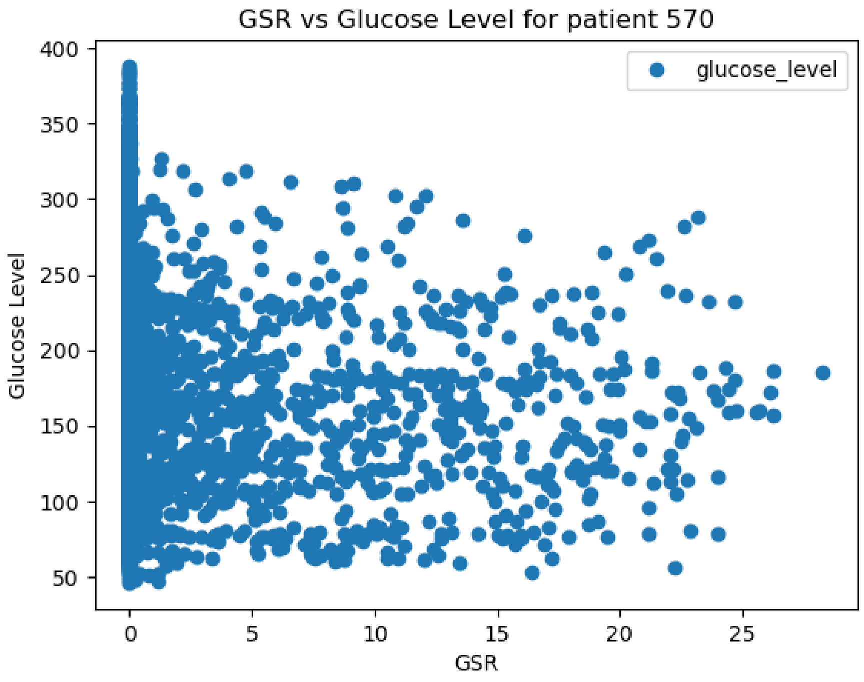

Figure 5.6: Correlation between BGL and GSR.

The second alternative of this experiment employs prior values of the BGL feature to predict the next BGL value. Using the prior six BGL values (30 minutes) gives the best resulting model, which is evaluated to have a RMSE of 4.5 (mg/dL).

Discussion: As mentioned in section 2.7, GSR is an intriguing feature because of its high correlation to BGL, but the RMSE of 64 (mg/dL) is an abysmal score. And further, the correlation of $r = -0.12$ between GSR and BGL indicates a very poor correlation of the features. This verifies the findings from the paper [5] mentioned in section 2.7 that the OhioT1DM dataset has a very poor correlation between these values even in the patients with the highest correlation. This low correlation paved the way to use BGL to predict the next BGL value, which gives a far superior evaluation RMSE of 4.5 (mg/dL).

5.3.4 Experiment 6: Ensemble Method Part 2

This experiment is the second part of the ensemble technique. The LSTM model from the prior experiment is combined with the Informer model to construct the LEET method and test **hypothesis 1** to outperform the state-of-the-art model. Additionally, this experiment tests **hypothesis 2** to outperform the prior imputation methods used for handling missing values in this thesis. The Informer model is the same model used in experiment 2, which is used to predict BGL values several steps into the future. The dataset is split into two subsets, where subset b is used to train Model 2, as explained in section 4.2. Model 1 extrapolates the missing values in subset b, which replaces all the zero-filled values. The newly preprocessed subset b is then used to train the Informer model, which yields an RMSE of 15.85 (mg/dL) when predicting 30 minutes into the future. Lastly, fig. 5.7 shows the training and validation loss using a MSE loss function. This figure also shows that this model converges after very few training epochs.

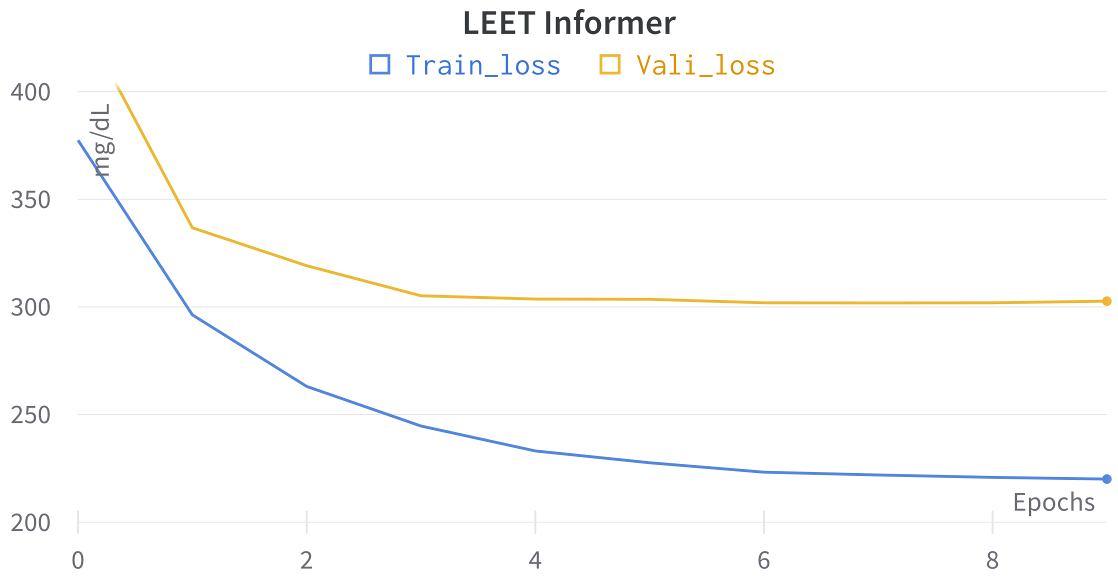

Figure 5.7: LEET Informer

Discussion: The method in this experiment outperforms both the models and methods in the prior experiments and the state-of-the-art model. In addition, this method does not suffer from the data leakage problem that the interpolation method does, which means this method can be used in a real-life scenario. Furthermore, as shown in fig. 5.7, the convergence of the graphs is a sign that the model has stopped learning because it has been trained to a solution that reduces the error as much as possible, given the model's structure and the available training data. This experiment conclusively shows that a Transformer model can outperform the state-of-the-art CRNN model from [19] by predicting the BGL 30 minutes into the future with lower RMSE on the same dataset also that intelligent handling of missing values instead of zero-filling can boost the model's accuracy, thus confirming both hypothesis 1 and hypothesis 2.

5.4 Experiment 6: Longer Input Sequence

Hypothesis 3 states that: **"Opting for a 24-hour input sequence instead of a two-hour sequence could boost the Transformer model's accuracy."** In this experiment, the hypothesis is tested by training the Informer model with the input sequence lengths of 24 and 288 timesteps, representing two hours and 24 hours, respectively, to evaluate the best-performing sequence length. When trained on the short and long sequences, the models resulting RMSE is 15.8 (mg/dL) and 16.3(mg/dL), respectively, and MAE of 9.0 (mg/dL) and 9.3 (mg/dL), respectively. Further, fig. 5.8 shows the training and validation loss using a MSE loss function, these graphs show that short and long sequences are very similar, but the short validation loss is slightly lower.

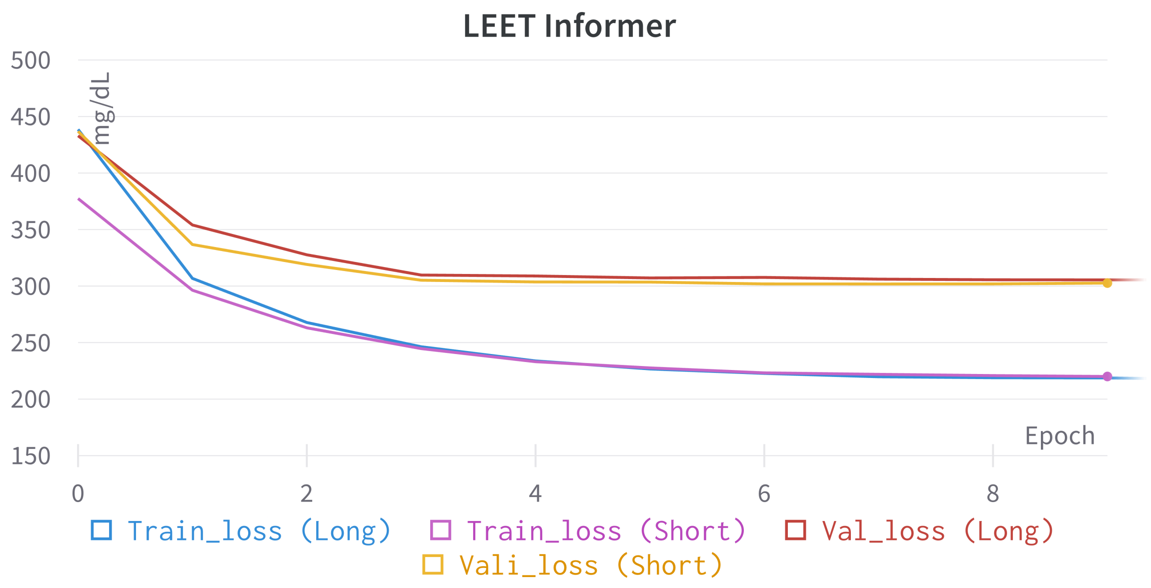

Figure 5.8: Short vs. Long Input Sequence

Discussion: As mentioned in section 2.6, activity affects the BGL up to 24 hours into the future by increasing the body's insulin sensitivity. This extended timeframe of the activity's effect on the BGL is the reason for the hypothesis tested in this experiment. But even though the model has all these extra timesteps with data, the accuracy does not improve. This may be caused by the model's ability to extract the activity information from a much smaller input sequence in the form of insulin sensitivity. And additionally, the extra data may be an unwanted introduction of noise which reduces the model's ability to find trends and correlations. This verifies the time frame used in state-of-the-art algorithms attempting to solve the same problem. This conclusively shows that opting for a 24-hour input sequence instead of a two-hour sequence **does not** boost the Transformer model's accuracy, thus rejecting hypothesis 3.

5.5 Summary

Table 5.1 lists the results achieved by the different experiments in this thesis, as well as some of the best results from the BGLPC mentioned in section 2.5. The results are listed with the evaluation metrics RMSE and MAE and "Decrease from 18.49," which is the result of the state-of-the-art model after fixing the data leakage as explained in section 5.1.1.

Table 5.1: Comparison of Results

Method	RMSE (mg/dL)	MAE (mg/dL)	% Decrease from 18.49
Informer with interpolation	**14.76**	11	20.23 %
LEET	**15.85**	9	14.28 %
SotA	17.45	11.22	-
SotA (fixed)	18.49	12.11	0 %
Informer	21.4	11.2	-15.75 %
Encoder-Only	27	24.2	-46.09 %
Missing value column	34	25.6	-84.11 %

Chapter 6

Conclusions

This thesis presents a comprehensive investigation into applying Transformer-based models for predicting blood glucose levels in type 1 diabetes patients using the much-used OhioT1DM dataset. Most significantly, we propose using a novel ensemble technique named LEET which uses a LSTM model as its first stage and then uses a Transformer model as its second stage. The task is highly complex due to the intricate nature of the human body, characterized by its multivariate nature and the temporal dependencies inherent in the time series data.

Section 1.4.1 introduces the three following goals. First, validate the result from the state-of-the-art CRNN model from section 3.2. Second, preprocess the dataset for the transformer model. Lastly, develop a transformer model that reaches higher accuracy than the state-of-the-art model when predicting BGL 30 minutes into the future. When validating the state-of-the-art model in the preliminary experiment, we found that they made a mistake when preprocessing the basal column by using it as a dose instead of an hourly rate. Additionally, we discovered an error in their code, causing data leakage and, thus, invalidating their published result. The dataset's features are preprocessed, so all features have a common time dimension. This gives us a multivariate time series that resembles the real-world as close as possible.

Several models are suggested and subsequently employed in our experiments to evaluate our hypotheses and enhance the precision of model predictions. The OhioT1DM dataset, which contains eight weeks of data for each of the 12 individuals with type 1 diabetes, is utilized for these model training and evaluations. The accuracy of the Encoder-Only model is inferior to the state-of-the-art model. Further, the Informer model's accuracy outperforms the Encoder-Only model but is still inferior to the state-of-the-art model. Lastly, the proposed LEET technique produces a model with a RMSE of 15.85, which is 14.28% lower than the state-of-the-art model. Additionally, this novel technique avoids data leakage. To our knowledge, this represents the best results in a model trained using the OhioT1DM dataset mentioned in 4.1 when predicting 30 minutes into the future.

Further, when using the linear interpolation imputation method on the dataset, the informer model produces the highest accuracy in this thesis. However, this approach leads to data leakage, rendering the model unsuitable for practical use. The LEET technique is the superior method used in this thesis to handle missing values in the OhioT1DM time-series dataset. Although, as initially hypothesized, an extended input sequence would yield higher

accuracy, our experiment proves this is not the case, underscoring the importance of selecting an optimal sequence length for time-series prediction tasks.
The quantity and quality limitations of the OhioT1D dataset may be two significant factors that have negatively impacted the results presented in this thesis. These limitations indicate the need for another dataset without these limitations.
The following list presents further validations and improvements that can be used as the next steps to continue the research presented in this thesis:

- Further Exploration into Data Shuffling should be done using the Encoder-Only model without shuffling. The reason to try this is due to the sequential nature of time-series data. By retaining this sequentiality, prediction accuracy might be improved.
- The Encoder-Only Model can be utilized with LEET to improve the result further. This could further validate the effectiveness of the LEET technique and give it broader relevance with different models.
- The LEET Technique can be further investigated using a Transformer Model instead of a LSTM model as the first stage in the technique. This modification could boost the quality of the imputed values, thus, boosting the second model's prediction accuracy and, in the end, contributing to more precise BGL predictions.
- Experimentation with other datasets is highly recommended to validate and enhance model performance. The limited size and the poor correlation between BGL and GSR in OhioT1DM are good motivations for acquiring a bigger and better-correlated dataset. If the first model in the LEET technique is improved, then this would most likely enhance the second model's efficiency and prediction results.
- Future investigations could explore methods for model explainability using techniques such as SHAP or LIME. This is crucial for regulatory compliance and ethical considerations in medical applications, as it ensures transparency, accountability, and trustworthiness in the models' predictions.
- Extended Forecasting with LEET: The model's impressive short-term prediction performance encourages exploration into forecasting further into the future, such as 1 and 2 hours ahead. This could validate LEET effectiveness on longer horizons as well.

Appendix A

Informer Experiment List

Table A.1: Informer Experiments

Experiment Description	Data Type	Train Distribution	Validation Distribution	Test Distribution	Test RMSE
Remove 4x basis columns (note: 2018, train=0.9, test=0.001)	Interpolated	90%	0.9%	0.1%	14.929
Combine Basal + Bolus into insulin (note: 2018, train=0.9, test=0.001)	Interpolated	90%	0.9%	0.1%	15.478
Remove missing_bg (91) (note: 2018, train=0.9, test=0.001)	Interpolated	90%	0.9%	0.1%	15.094
	Interpolated			0.1%	15.27
batch size=8, seq_len, label_len = 24	Interpolated	-	-	-	14.762
Transfer learn on (90) with 2020 data, then train Patient 540	Interpolated	-	-	-	28.3
Transfer learn on (90) with 2020 data:, then train Patient 544	Interpolated	-	-	-	25.23
	Interpolated			-	27.55
All patients_non_interpolated (train = 0.7, val = 0.1, test = 0.2)	Non-Interpolated	70%	10%	20%	28.8
2018_2020, non-interpolate train+test, !scale, all features (train=0.8, val=0.2, test=0.001)	Non-Interpolated	80%	20%	0.1%	9.241
2018_2020, non-interpolate train+test, !scale, all features (train=0.8, val=0.1, test=0.1)	Non-Interpolated	80%	10%	10%	22.2
2018_2020, non-interpolate train+test, !scale, all features (train=0.8, val=0.1, test=0.2)	Non-Interpolated	80%	10%	20%	36

Appendix B

Encoder-Only Hyperparameters

Table B.1: Model Hyperparameters

Hyperparameter	**Value**
activation	relu
batch_size	512
dropout_rate	0.25
early_stopping	100
epochs	11000
ff_dim	8
head_size	8
horizon	6
learning_rate	0.0001
look_back	12
MLP dropout	0.4
MLP units	512
number of heads	8
num_transformer_blocks	4

Bibliography

[1] N Allen, A Gupta - Diagnostics, and undefined 2019. "Current diabetes technology: striving for the artificial pancreas." In: *mdpi.com* (). URL: https://www.mdpi.com/428238 (visited on 03/09/2023).

[2] Muhammad Asad, Usman Qamar, and Muhammad Abbas. "Blood Glucose Level Prediction of Diabetic Type 1 Patients Using Nonlinear Autoregressive Neural Networks." en. In: *Journal of Healthcare Engineering* 2021 (Feb. 2021). Publisher: Hindawi, e6611091. ISSN: 2040-2295. DOI: 10.1155/2021/6611091. URL: https://www.hindawi.com/journals/jhe/2021/6611091/ (visited on 04/21/2023).

[3] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015). arXiv: 1409.0473 Publisher: International Conference on Learning Representations, ICLR. (Visited on 04/06/2023).

[4] Krishna V. Bhaskarabhatla and Richard Birrer. "Physical activity and diabetes mellitus." In: *Comprehensive Therapy* 31.4 (Dec. 2005). Publisher: Springer, pp. 291–298. ISSN: 00988243. DOI: 10.1385/COMP:31:4:291/METRICS. URL: https://link.springer.com/article/10.1385/COMP:31:4:291 (visited on 03/16/2023).

[5] Brian Bogue-Jimenez et al. "Selection of Noninvasive Features in Wrist-Based Wearable Sensors to Predict Blood Glucose Concentrations Using Machine Learning Algorithms." In: *Sensors* 22.9 (2022). Publisher: MDPI, p. 3534.

[6] Longbing Cao. *AI in Finance: A Review*. en. SSRN Scholarly Paper. Rochester, NY, July 2020. DOI: 10.2139/ssrn.3647625. URL: https://papers.ssrn.com/abstract=3647625 (visited on 06/01/2023).

[7] *Carbohydrates and Blood Sugar | The Nutrition Source | Harvard T.H. Chan School of Public Health*. URL: https://www.hsph.harvard.edu/nutritionsource/carbohydrates/carbohydrates-and-blood-sugar/ (visited on 11/07/2022).

[8] Elena Castilla et al. "Spatiotemporal Transformer Neural Network for Time-Series Forecasting." In: *Entropy 2022, Vol. 24, Page 1651* 24.11 (Nov. 2022). Publisher: Multidisciplinary Digital Publishing Institute, p. 1651. ISSN: 1099-4300. DOI: 10.3390/E24111651. URL: https://www.mdpi.com/1099-4300/24/11/1651/htm (visited on 02/21/2023).

[9] Shiyu Chang et al. "Dilated Recurrent Neural Networks." In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/32bb90e8976aab5298d5da10fe66f21d-Abstract.html (visited on 04/13/2023).

[10] Yuanhong Chang et al. "Efficient temporal flow Transformer accompanied with multi-head probsparse self-attention mechanism for remaining useful life prognostics." en. In: *Reliability Engineering & System Safety* 226 (Oct. 2022), p. 108701. ISSN: 0951-8320. DOI: 10.1016/j.ress.2022.108701. URL: https://www.sciencedirect.com/science/article/pii/S095183202200326X (visited on 05/30/2023).

[11] Davide Chicco, Matthijs J. Warrens, and Giuseppe Jurman. "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation." en. In: *PeerJ Computer Science* 7 (July 2021). Publisher: PeerJ Inc., e623. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.623. URL: https://peerj.com/articles/cs-623 (visited on 05/30/2023).

[12] Junyoung Chung et al. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." In: (Dec. 2014). arXiv: 1412.3555. URL: http://arxiv.org/abs/1412.3555 (visited on 04/06/2023).

[13] John Daniels, Pau Herrero, and Pantelis Georgiou. "A multitask learning approach to personalized blood glucose prediction." In: *IEEE Journal of Biomedical and Health Informatics* 26.1 (2021). Publisher: IEEE, pp. 436–445.

[14] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. en. Google-Books-ID: pFjPDwAAQBAJ. Cambridge University Press, Apr. 2020. ISBN: 978-1-108-47004-9.

[15] Linda A. DiMeglio, Carmella Evans-Molina, and Richard A. Oram. "Type 1 diabetes." In: *The Lancet* 391.10138 (June 2018). Publisher: Elsevier, pp. 2449–2462. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(18)31320-5. (Visited on 11/07/2022).

[16] Arul Earnest et al. "Using autoregressive integrated moving average (ARIMA) models to predict and monitor the number of beds occupied during a SARS outbreak in a tertiary hospital in Singapore." In: *BMC Health Services Research* 5 (May 2005). Publisher: BioMed Central Ltd. ISSN: 14726963. DOI: 10.1186/1472-6963-5-36. (Visited on 04/06/2023).

[17] Craig K. Enders, Han Du, and Brian T. Keller. "A model-based imputation procedure for multilevel regression models with random coefficients, interaction effects, and nonlinear terms." In: *Psychological methods* 25.1 (2020). Publisher: American Psychological Association, p. 88.

[18] *Figure 5. Different strategies to impute missing data. (A)...* en. URL: https://www.researchgate.net/figure/Different-strategies-to-impute-missing-data-A-Forward-filling-imputed-missing-values_fig3_335623844 (visited on 05/18/2023).

[19] J Freiburghaus et al. "A deep learning approach for blood glucose prediction of type 1 diabetes." In: *ceur-ws.org* (2020). (Visited on 03/16/2023).

[20] Alex Graves and Jürgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM networks." In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 4. IEEE, 2005, pp. 2047–2052.

[21] Junqi Guo et al. "An AI-Application-Oriented In-Class Teaching Evaluation Model by Using Statistical Modeling and Ensemble Learning." en. In: *Sensors* 21.1 (Jan. 2021). Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, p. 241. ISSN: 1424-8220. DOI: 10.3390/s21010241. URL: https://www.mdpi.com/1424-8220/21/1/241 (visited on 05/03/2023).

[22] Pavel Hamet and Johanne Tremblay. “Artificial intelligence in medicine.” en. In: *Metabolism*. Insights Into the Future of Medicine: Technologies, Concepts, and Integration 69 (Apr. 2017), S36–S40. ISSN: 0026-0495. DOI: 10.1016/j.metabol.2017.01.011. URL: https://www.sciencedirect.com/science/article/pii/S002604951730015X (visited on 06/01/2023).

[23] Sue Ellen Haupt et al. “Machine Learning for Applied Weather Prediction.” In: *2018 IEEE 14th International Conference on e-Science (e-Science)*. Oct. 2018, pp. 276–277. DOI: 10.1109/eScience.2018.00047.

[24] S Hochreiter, J Schmidhuber - Neural computation, and undefined 1997. “Long short-term memory.” In: *ieeexplore.ieee.org* 9.8 (1997), pp. 1735–1780. URL: https://ieeexplore.ieee.org/abstract/document/6795963/ (visited on 04/06/2023).

[25] FJ Doyle III et al. “Closed-loop artificial pancreas systems: engineering the algorithms.” In: *Am Diabetes Assoc* (). URL: https://diabetesjournals.org/care/article-abstract/37/5/1191/38217 (visited on 03/09/2023).

[26] Mehrad Jaloli and Marzia Cescon. “Long-term Prediction of Blood Glucose Levels in Type 1 Diabetes Using a CNN-LSTM-Based Deep Neural Network.” In: *Journal of Diabetes Science and Technology* (Apr. 2022). Publisher: SAGE Publications Inc. ISSN: 19322968. DOI: 10.1177/19322968221092785/ASSET/IMAGES/LARGE/10.1177_19322968221092785-FIG6.JPEG. URL: https://journals.sagepub.com/doi/full/10.1177/19322968221092785 (visited on 02/21/2023).

[27] Balajee JM. “Data wrangling and data leakage in machine learning for healthcare.” In: (2018).

[28] Eric L. Johnson et al. “Standards of Medical Care in Diabetes—2021 Abridged for Primary Care Providers.” In: *Clinical Diabetes : A Publication of the American Diabetes Association* 39.1 (Jan. 2021). Publisher: American Diabetes Association, p. 14. ISSN: 08918929. DOI: 10.2337/CD21-AS01. URL: /pmc/articles/PMC7839613/ (visited on 11/07/2022).

[29] Jonas. *BLGP-HES-SO*. original-date: 2020-07-09T12:02:28Z. Sept. 2020. URL: https://github.com/JonasFreibur/BLGP-HES-SO/blob/ae8e387da720da13bcec410f8a5fc3a385c1aca7/BLGP_HES-SO.ipynb (visited on 04/13/2023).

[30] *KDH 2020 - BGLP Challenge*. URL: https://sites.google.com/view/kdh-2020/bglp-challenge (visited on 02/21/2023).

[31] Dae-Yeon Kim et al. “Developing an Individual Glucose Prediction Model Using Recurrent Neural Network.” en. In: *Sensors* 20.22 (Jan. 2020). Number: 22 Publisher: Multidisciplinary Digital Publishing Institute, p. 6460. ISSN: 1424-8220. DOI: 10.3390/s20226460. URL: https://www.mdpi.com/1424-8220/20/22/6460 (visited on 04/13/2023).

[32] David C. Klonoff et al. “The Surveillance Error Grid.” In: *Journal of Diabetes Science and Technology* 8.4 (June 2014), pp. 658–672. ISSN: 1932-2968. DOI: 10.1177/1932296814539589. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4764212/ (visited on 04/20/2023).

[33] Tomas Koutny and Michael Mayo. “Predicting glucose level with an adapted branch predictor.” en. In: *Computers in Biology and Medicine* 145 (June 2022), p. 105388. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2022.105388. URL: https://www.sciencedirect.com/science/article/pii/S0010482522001809 (visited on 04/13/2023).

[34] A Krizhevsky et al. “Imagenet classification with deep convolutional neural networks.” In: *dl.acm.org* 60.6 (June 2017). Publisher: Association for Computing Machinery, pp. 84–90. DOI: 10.1145/3065386. URL: https://dl.acm.org/doi/abs/10.1145/3065386 (visited on 04/06/2023).

[35] Anders Krogh. “What are artificial neural networks?” en. In: *Nature Biotechnology* 26.2 (Feb. 2008). Number: 2 Publisher: Nature Publishing Group, pp. 195–197. ISSN: 1546-1696. DOI: 10.1038/nbt1386. URL: https://www.nature.com/articles/nbt1386 (visited on 06/01/2023).

[36] Alok Kumar and J. Mayank. “Ensemble learning for AI developers.” In: *BApress: Berkeley, CA, USA* (2020). Publisher: Springer.

[37] SM Lee et al. “Glucose Transformer: Forecasting Glucose Level and Events of Hyperglycemia and Hypoglycemia.” In: *ieeexplore.ieee.org* (). URL: https://ieeexplore.ieee.org/abstract/document/10035395/ (visited on 02/21/2023).

[38] Eldon D. Lehmann. “The freeware AIDA interactive educational diabetes simulator.” In: *Med Sci Monit* 7.3 (2001). Publisher: Citeseer, pp. 504–515.

[39] Kezhi Li et al. “Convolutional recurrent neural networks for glucose prediction.” In: *IEEE journal of biomedical and health informatics* 24.2 (2019). Publisher: IEEE, pp. 603–613.

[40] Wei-Chao Lin and Chih-Fong Tsai. “Missing value imputation: a review and analysis of the literature (2006–2017).” In: *Artificial Intelligence Review* 53 (2020). Publisher: Springer, pp. 1487–1509.

[41] Zachary C. Lipton, David C. Kale, and Randall Wetzel. “Modeling Missing Data in Clinical Time Series with RNNs.” In: (June 2016). arXiv: 1606.04130. DOI: 10.48550/arxiv.1606.04130. URL: https://arxiv.org/abs/1606.04130v5 (visited on 01/19/2023).

[42] Chiara Dalla Man et al. “The UVA/PADOVA type 1 diabetes simulator: new features.” In: *Journal of diabetes science and technology* 8.1 (2014). Publisher: SAGE Publications Sage CA: Los Angeles, CA, pp. 26–34.

[43] Cindy Marling and Razvan Bunescu. “The OhioT1DM Dataset for Blood Glucose Level Prediction: Update 2020.” In: *CEUR workshop proceedings* 2675 (2020). Publisher: NIH Public Access, p. 71. ISSN: 16130073. URL: /pmc/articles/PMC7881904/ (visited on 03/16/2023).

[44] Richard Mauseth et al. “Use of a “fuzzy logic” controller in a closed-loop artificial pancreas.” In: *liebertpub.com* 15.8 (Aug. 2013), pp. 628–633. DOI: 10.1089/dia.2013.0036. URL: https://www.liebertpub.com/doi/abs/10.1089/dia.2013.0036 (visited on 03/09/2023).

[45] Mario Munoz-Organero. “Deep physiological model for blood glucose prediction in T1DM patients.” In: *Sensors* 20.14 (2020). Publisher: MDPI, p. 3896.

[46] Hoda Nemat et al. “Blood glucose level prediction: advanced deep-ensemble learning approach.” In: *IEEE Journal of Biomedical and Health Informatics* 26.6 (2022). Publisher: IEEE, pp. 2758–2769.

[47] J Schmidhuber - Neural networks and undefined 2015. “Deep learning in neural networks: An overview.” In: *Elsevier* (). URL: https://www.sciencedirect.com/science/article/pii/S0893608014002135 (visited on 04/06/2023).

[48] Rei Noguchi et al. “The selective control of glycolysis, gluconeogenesis and glycogenesis by temporal insulin patterns.” In: *Molecular Systems Biology* 9.1 (2013). Publisher: European Molecular Biology Organization, p. 664. ISSN: 17444292. DOI: 10.1038/MSB.2013.19. URL: /pmc/articles/PMC4039368/ (visited on 11/07/2022).

[49] Jangho Park et al. “Long-term missing value imputation for time series data using deep neural networks.” en. In: *Neural Computing and Applications* 35.12 (Apr. 2023), pp. 9071–9091. ISSN: 1433-3058. DOI: 10.1007/s00521-022-08165-6. URL: https://doi.org/10.1007/s00521-022-08165-6 (visited on 05/29/2023).

[50] N. S. Peirce. “Diabetes and exercise.” In: *British Journal of Sports Medicine* 33.3 (June 1999). Publisher: British Association of Sport and Excercise Medicine, pp. 161–172. ISSN: 0306-3674. DOI: 10.1136/BJSM.33.3.161. URL: https://bjsm.bmj.com/content/33/3/161 (visited on 03/09/2023).

[51] Rob J. Hyndman; George Athanasopoulos. “Forecasting: Principals and Practice; chapter: 3.1. Some simple forecasting methods.” In: (2012). ISBN: 978-0-9875071-1-2, p. 377. URL: https://books.google.com/books/about/Forecasting_principles_and_practice.html?id=_bBhDwAAQBAJ (visited on 02/10/2023).

[52] W. H. M. Saad et al. “Analysis on continuous wearable device for blood glucose detection using GSR sensor.” In: *Int. J. Nanoelectron. Mater* 13.8 (2020).

[53] Wira Hidayat Mohd Saad et al. “Implementation of Continuous Wearable Low Power Blood Glucose Level Detection using GSR Sensor.” In: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10.2-6 (2018), pp. 81–85.

[54] Cátia M. Salgado et al. “Missing data.” In: (2019).

[55] Alice Segato et al. “Artificial intelligence for brain diseases: A systematic review.” In: *APL Bioengineering* 4.4 (Oct. 2020). Publisher: AIP Publishing LLC AIP Publishing, p. 041503. ISSN: 24732877. DOI: 10.1063/5.0011697. URL: https://aip.scitation.org/doi/abs/10.1063/5.0011697 (visited on 01/19/2023).

[56] U. Snekhalatha et al. “NON-INVASIVE BLOOD GLUCOSE ANALYSIS BASED ON GALVANIC SKIN RESPONSE FOR DIABETIC PATIENTS.” In: *https://doi.org/10.4015/S101623721850009* 30.2 (Mar. 2018). Publisher: World Scientific Publishing Company. ISSN: 10162372. DOI: 10.4015/S1016237218500096. (Visited on 02/07/2023).

[57] U. Snekhalatha et al. “Non-invasive blood glucose analysis based on galvanic skin response for diabetic patients.” In: *Biomedical Engineering: Applications, Basis and Communications* 30.02 (Apr. 2018). Publisher: National Taiwan University, p. 1850009. ISSN: 1016-2372. DOI: 10.4015/S1016237218500096. URL: https://www.worldscientific.com/doi/abs/10.4015/S1016237218500096 (visited on 05/02/2023).

[58] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to sequence learning with neural networks.” In: *Advances in Neural Information Processing Systems* 4.January (2014). arXiv: 1409.3215 Publisher: Neural information processing systems foundation, pp. 3104–3112. ISSN: 10495258. (Visited on 04/06/2023).

[59] *The BGLP Challenge: Results, Papers, and Source Code.* URL: http://smarthealth.cs.ohio.edu/bglp/bglp-results.html (visited on 03/16/2023).

[60] Ashish Vaswani et al. “Attention is All you Need.” In: *Advances in Neural Information Processing Systems* 30 (2017). (Visited on 01/19/2023).

[61] Yingying Wang et al. “The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition.” en. In: *Applied Sciences* 10.5 (Jan. 2020). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 1897. ISSN: 2076-3417. DOI: 10.3390/app10051897. URL: https://www.mdpi.com/2076-3417/10/5/1897 (visited on 06/01/2023).

[62] *Weights & Biases – Developer tools for ML*. URL: https://wandb.ai/site/,%20http://wandb.ai/site (visited on 05/11/2023).

[63] Jinyu Xie and Qian Wang. “Benchmark Machine Learning Approaches with Classical Time Series Approaches on the Blood Glucose Level Prediction Challenge.” In: *KHD@ IJCAI*. 2018, pp. 97–102.

[64] Wenpeng Yin et al. *Comparative Study of CNN and RNN for Natural Language Processing*. arXiv:1702.01923 [cs]. Feb. 2017. DOI: 10.48550/arXiv.1702.01923. URL: http://arxiv.org/abs/1702.01923 (visited on 06/01/2023).

[65] Yong Yu et al. “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures.” In: *Neural Computation* 31.7 (July 2019), pp. 1235–1270. ISSN: 0899-7667. DOI: 10.1162/neco_a_01199. URL: https://doi.org/10.1162/neco_a_01199 (visited on 06/01/2023).

[66] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. “Self-Distillation: Towards Efficient and Compact Neural Networks.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.8 (Aug. 2022), pp. 4388–4403. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3067100.

[67] Haoyi Zhou et al. *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. arXiv:2012.07436 [cs]. Mar. 2021. DOI: 10.48550/arXiv.2012.07436. URL: http://arxiv.org/abs/2012.07436 (visited on 04/17/2023).

[68] Taiyu Zhu et al. “Dilated recurrent neural networks for glucose forecasting in type 1 diabetes.” In: *Journal of Healthcare Informatics Research* 4 (2020). Publisher: Springer, pp. 308–324.