Multi-step Heart Rate Forecasting in ICU Patients using the Temporal Fusion Transformer

Abstract

Heart rate monitoring is vital in intensive care units (ICUs) for early detection of patient deterioration. Accurate forecasting can enhance proactive clinical interventions. This study applies the Temporal Fusion Transformer (TFT) to predict heart rate six hours ahead using the Medical Information Mart for Intensive Care IV (MIMIC-IV) dataset. Leveraging its ability to integrate static and dynamic patient data, the TFT offers improved forecasting accuracy compared to Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models. These findings highlight the TFT's potential to enhance patient monitoring and inform clinical decision-making in ICUs.

1 Introduction

1.1 Background

Heart rate monitoring is a vital component (pun intended), of patient care. This is especially so in intensive care units (ICUs), where it can provide real time insight into the patients' health, stress, medicinal effects, etc. It also plays an important role in guiding medical professionals to make better informed decisions[1]. Now, with the growing integration of technologically advanced patient observation systems, and electronic health records (EHRs), an abundance of health data is available. This presents an opportunity for machine learning methods, which rely on such data, to develop predictive models that can potentially further enhance patient outcomes and support proactive care [2].

However, despite the availability of this rich data, in forecasting specifically, predicting vital signs remains a difficult task. The human system, to the surprise of no one, is complex. There are numerous influential nonlinear factors such as: medications, comorbidities, external stimuli, etc., [3] making it difficult for traditional time series models to perform effectively [4][5]. These limitations show a need to utilize a more advanced machine learning model that can not only effectively comprehend the temporal patterns, short and long, but also integrate diverse data types [6].

1.2 Aim

We explore the Temporal Fusion Transformer (TFT) [7], an extension of the Transformer, for multi-step forecasting of heart rate for patients specifically in the Intensive Care Unit. The model is trained using the Medical Information Mart for Intensive Care IV (MIMIC-IV) dataset. The model's strengths lie in its ability to grasp temporal dependencies, especially long-term trends, while also integrating various patient features. We train the TFT to forecast heart rate 6-hours into the future and then evaluate how it did by benchmarking it against traditional time series models like Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks. This work

hopes to highlight the potential in integrating this into medical workflows. Improving the accuracy of these types of forecasts enables early detection of adverse events, optimize resource allocation, and empowers health experts with knowledge into their patients.

2 Background

2.1 AI in Healthcare

Its common knowledge that Artificial Intelligence is rapidly becoming more integrated into more aspects of our lives. So, it is inevitable that AI has made its way into healthcare as well [8][9]. This revolution in the field enables data-driven approaches to diagnosis, treatment, and patient care, and with the growing availability of large-scale clinical datasets, this only adds fuel to the rocket. AI systems are being leveraged in a wide range, from image-based diagnostics [10] to predictive models for patient deterioration in ICUs [11]. Among these, time series data—such as physiological metrics recorded in EHRS—are particularly valuable for understanding patient health trajectories [12]. However, the irregular sampling, nonlinear dynamics, and high dimensionality of these datasets present challenges that require sophisticated forecasting techniques.

2.2 Related Works & Shortfalls

Conventional statistical models are widely used due to their simplicity and interpretability [13]. For example, the Autoregressive integrated Moving Average (ARIMA) model has been used for patient length of stay prediction in intensive care units [14], and to monitor and manage seasonal diseases.

However, these methods often rely on assumptions of linearity and stationarity in the data, making them inadequate for capturing complex, non-linear patterns characteristic of physiological signals. To address these limitations, deep learning techniques—particularly recurrent neural networks (RNNs) and their variants—have gained significant attention. Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have been utilized for several assignments such as: predicting patient outcomes, tracking disease progression, and anticipating hospital readmissions [15][16].

Despite their widespread success in time series forecasting problems and similar sequential data tasks, these models also face several limitations. One of the primary challenges lies in their difficulty handling very long sequences effectively. While it is true that both architectures use gating mechanisms designed for this very issue by selectively choosing relevant information to pass on, when dealing with rather long sequences, performance degrades. In the same vein, when backpropagating through more than a few layers/steps, gradients can either grow exponentially and “explode” or shrink exponentially and “vanish”. This problem effectively loses critical information that could help generalize for those long-term dependencies. Computational inefficiency is another drawback. Both models rely on sequential processing, preventing parallelization during training and evaluation. This significantly increases the computational cost and stops making them more scalable for larger applications[17]. These limits show the need for alternative models to address the shortcomings of LSTMS and GRUs, particularly in clinical forecasting, where scalability, adaptability, and robustness are a must.

3 Data

3.1 MIMIC Dataset

The Medical Information Mart for Intensive Care IV (MIMIC-IV) [18][19][20] is a comprehensive, semi-publicly available database containing de-identified health data for over 70,000 ICU admissions at the Beth Israel Deaconess Medical Center in Boston, Massachusetts between 2008 and 2019. Building on its predecessor, MIMIC-III, it offers improved data quality, additional variables, and a more user-friendly data schema. The database includes a wide array of information such as demographics, hourly vital signs, laboratory results, medications, and clinical notes, making it an invaluable resource for developing predictive models in critical care.

MIMIC-IV complies with all relevant ethical standards and regulations, including the Health Insurance Portability and Accountability Act (HIPAA). The dataset, as mentioned previously, is de-identified to protect patient privacy: patient identifiers were removed or replaced with anonymized codes, and dates were shifted to prevent re-identification of individuals. Access to MIMIC-IV requires completion of training on human subjects' research and agreement to a data use agreement. This ensures ethical standards while using the dataset.

For this study, we utilized MIMIC-IV to create a time-series dataset [21] for forecasting heart rate six hours ahead. The target variable, heart rate, was complemented with dynamic physiological measurements and static patient characteristics to potentially enhance predictive power, allowing us to model both short-term and long-term variations in heart rate and its covariates.

3.2 Data Preprocessing

Extracting and preprocessing are vital steps especially for data such as the MIMIC-IV dataset. Although the data isn't totally raw and has been somewhat preprocessed for ease-of-use by the creators, there are multiple things that need to be done. Patients are filtered to contain at least 24 hours of ICU data, and physiological outliers outside of the normal ranges - for example, abnormal heart rates - were filtered out. Static categorical variables for patients, such as admission type, demographics, and illnesses are appended. Illness diagnosis codes (ICD-9 and ICD-10) [22][23] are utilized to group illnesses into their broader categories (e.g., circulatory, blood, etc.,) and then one hot-encoded. Continuous variables include other vital signs such as, respiratory rate, oxygen saturation, temperature, and past heart rate which are normalized. For TFTs, the dataset must be shaped into a time-series. All the data is aggregated into an hourly interval to align inputs. Missing values are also handles with forward and backward filling for short gaps, and linear interpolation for small missing segments in continuous variables. Patients with over 20% missing data are excluded to maintain integrity. The target variable is defined by shifting heart rate measurements six hours into the future. This enables the model to utilize historical heart rate measurements (18 hours).

3.3 Feature Selection & Analysis

To optimize model performance and address hardware limitations that constrained computational resources, feature analysis and engineering was done. Initially, 42 variables were considered, including static variables such as age, gender, length of stay, admission type, and illness categories, and time varying unknown reals such as heart rate and other vital signs.

To identify the most influential static features, we analyze importance scores using a built-in method from the TFT's library.

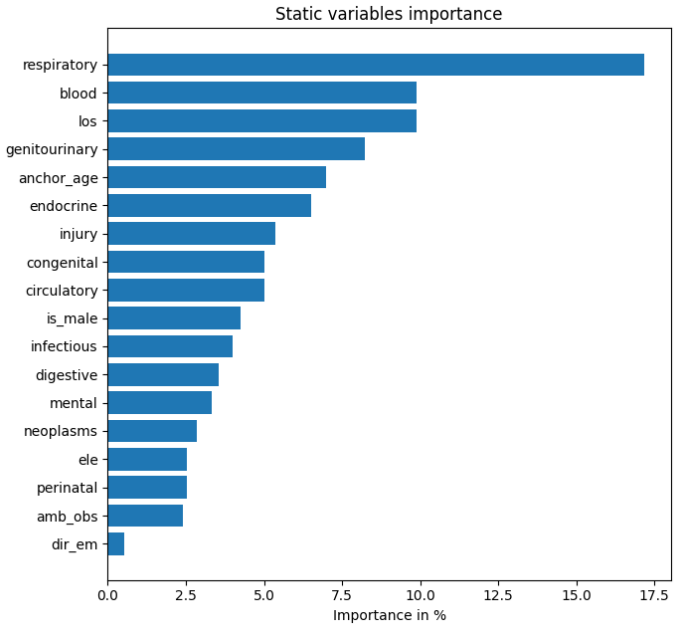

Figure 1: Variable Importance for static variables

The analysis of variable importance scores shows that for certain static features—like age, length of stay, and key illnesses, particularly those related to respiratory and blood conditions—significantly influenced heart rate forecasting. Conversely, features such as specific admission types and less common or unrelated illness categories were found to have minimal impact.

Guided by these insights and mindful of hardware constraints, we reduced the feature set by prioritizing the most impactful variables, narrowing it from 42 to 30. When comparing the trimmed model to the original full-feature set, we observed a slight performance improvement. This result suggests that focusing on the most critical features enhances the model's generalization by reducing noise introduced by less relevant inputs. This feature reduction not only resulted in a more efficient and interpretable model but also maintained, and in some cases improved, performance. Additionally, it enabled the development of more complex TFT architectures, such as those with additional heads or layers, within the hardware limitations. The detailed benchmark is provided in Section 5.

4 Methodology

4.1 Temporal Fusion Transformer (TFT) Architecture

The Temporal Fusion Transformer (TFT), as mentioned previously, is specifically designed to handle multi-horizon forecasting problems and capture long-sequence temporal patterns. It comprises of several key components that together make it effective:

- **Variable Selection Networks (VSNs):** Employed at both the static and temporal levels to select the most relevant features dynamically. This allows the model to focus on important covariates.

- **Gated Residual Networks (GRNs):** Used for feature transformation, enabling the model to handle nonlinear relationships and interactions between variables. They incorporate gating mechanisms to control information flow and mitigate issues like vanishing gradients.
- **Temporal Attention Layer:** Allows the model to weigh different time steps differently, focusing on periods that are more informative for the forecasting task. It captures long-range temporal dependencies without the need for sequential processing, as required in RNNs[24].
- **Static Covariate Encoders:** Process static features, such as patient demographics, and influence the temporal dynamics by modulating the temporal layers based on static information.

The model outputs quantiles. By outputting probabilistic forecasts across multiple quantiles, a quantile range can be established which provides an intuitive aspect in evaluating the model's performance.

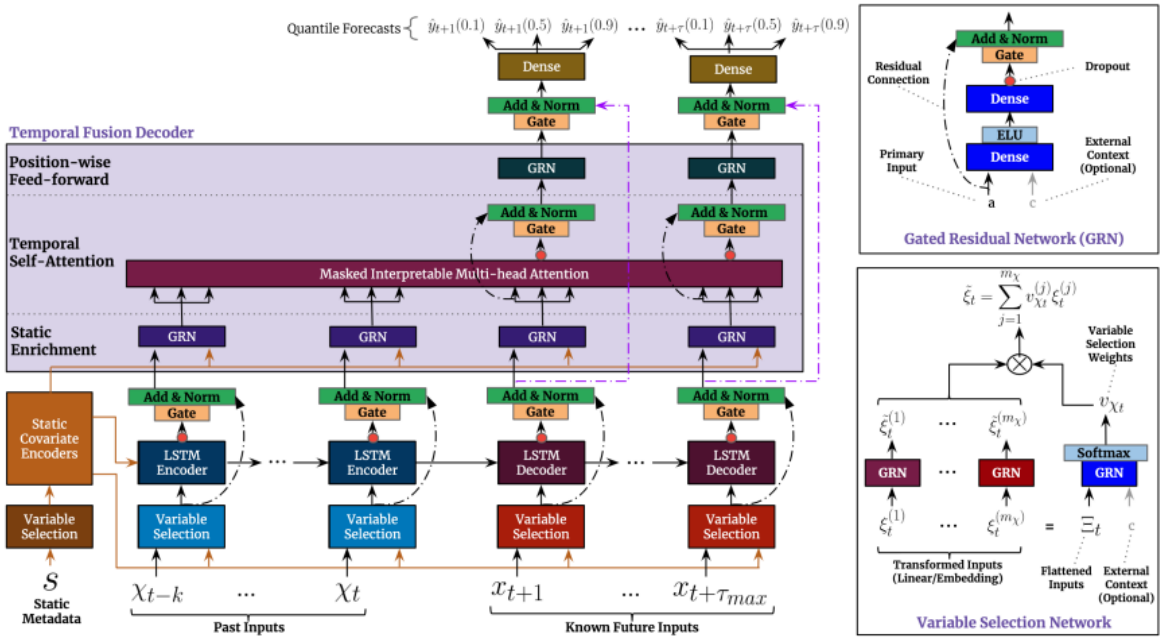

Figure 2. TFT architecture

The TFT also addresses several of the shortcomings of LSTMs and GRUs. It can handle long sequences and avoid the problem of exploding and/or vanishing gradients by using its attention-based architecture. Instead of relying on sequential processing, the multi-head attention mechanisms allow it to directly focus on important parts of the sequence, regardless of how far away it is. Additionally, the TFT leverages the transformer's capacity to process data in parallel, dramatically improving computational efficiency and scalability.

In our implementation, we configured the TFT to process 18 hours of historical data to predict heart rate for the subsequent 6 hours. The model inputs include time-varying covariates (e.g., vital signs) and static covariates (e.g., age, gender). We utilized the PyTorch Forecasting library [25] to

implement the TFT, with specific attention to handling irregular time series data common in clinical datasets.

Hyperparameters were optimized using Optuna [26], focusing on parameters such as the number of attention heads, hidden layer sizes, dropout rates, and learning rates. The final model architecture for the Temporal Fusion Transformer included 4 attention heads, a hidden size of 64, a dropout rate of 0.2, a learning rate of 0.002, a hidden continuous size of 9, and gradient clipping at 0.09.

4.2 Procedure

The performance of the Temporal Fusion Transformer was evaluated with two recurrent neural network-based models, also created using the PyTorch Forecasting library, Long Short-Term Memory (LSTM) [27] and Gated Recurrent Units (GRUs) [28]. These models are widely used for time series forecasting due to their ability to model temporal dependencies in similar problems.

The LSTM was configured with 64 hidden units and 2 layers, utilizing the gating mechanisms to capture long-term dependencies through its gating mechanisms. This setup is particularly effective for tasks that require important information to be passed over many iterations. Similarly, the GRU was configured with 64 hidden units and 2 layers, but is incorporated with a simpler architecture, making it computationally lighter while retaining the ability to model sequential patterns. Both models, along with the TFT, were trained using quantile loss to enable probabilistic forecasting.

All three models-TFT, LSTM, and GRU- utilize the Adam optimizer and use the quantile loss function, allowing for evaluations of the accuracy of predictions across multiple quantiles ranges. All training and experiments were performed on an NVIDIA GeForce 3060 Graphics Processing Unit (GPU) with 12GB VRAM to ensure computational consistency. Model checkpoints and training logs were saved and logged using PyTorch Lightning's modules as well as Tensorboard [30] to facilitate reproducibility and evaluate performance. The final preprocessed dataset consists of more than 3.5 million rows with 20 features, both time-varying and static. This dataset structure is consistent, using 18 past time steps of varying physiological variables as input features and forecasting heart rate 6 future time steps ahead. To do this, a "sliding window" approach slid across every row to generate input-output pairs that help preserve the temporal relationships in the data [29]. All models were trained for a maximum of 30 epochs, with early stopping applied monitoring validation loss to prevent overfitting if no improvements occurred within 7 consecutive epochs. A modest batch size of 32, was used due to the size of the data and hardware limitations.

By comparing these models, this study aims to evaluate the comparative effectiveness of these models against each other. By leveraging probabilistic forecasting through quantile loss, all models also address the inherent uncertainty present in clinical time series data. Ultimately, this comparison showcases the utility of such models in healthcare environments, especially in the Intensive Care Units.

5 Results & Analysis

5.1 Evaluation Metrics

Model performance was evaluated using the following metrics:

- Mean Absolute Error (MAE): Measures the average magnitude of errors in a set of predictions, without considering their direction.
- Mean Absolute Percentage Error (MAPE): Expresses accuracy as a percentage, making it easier to interpret the error magnitude relative to the actual values.
- Root Mean Squared Error (RMSE): Provides a quadratic scoring rule that measures the average magnitude of the error, giving higher weight to larger errors.
- Symmetric Mean Absolute Percentage Error (SMAPE): An alternative to MAPE that treats over- and under-forecasts equally.
- Quantile Loss: Evaluates the accuracy of probabilistic forecasts across multiple quantiles, capturing the model's ability to estimate the distribution of the target variable.

5.2 Model Performances

This section presents the performance of the Temporal Fusion Transformer (TFT) model in forecasting heart rate, including comparisons between:

- The original TFT model with all features and the reduced TFT model with selected features.
- The TFT model and the baseline models (LSTM and GRU).

MODEL	MAE	MAPE	RMSE	SMAPE	LOSS
Original TFT model	2.86	4.13	3.76	4.08	1.86
Reduced TFT model	2.77	4.02	3.66	3.94	1.77

Table 2. Performance Metrics Comparison of Original and Reduced TFT Models

The reduced TFT model slightly outperforms the original model across all evaluation metrics. This improvement indicates that removing less important static features helps the model generalize better by reducing overfitting and computational complexity.

Model	MAE	MAPE	RMSE	SMAPE	LOSS
TFT (reduced)	2.77	4.02	3.66	3.94	1.77
LSTM	3.63	5.28	4.90	5.21	2.48
GRU	3.55	5.16	4.78	5.09	2.38

Table 2. Performance Metrics Comparison with Baseline Models Metrics

5.3 Comparative Analysis and Discussion

The TFT exceeds the LSTM and GRU models across all performance metrics. The TFT achieves MAE of 2.77 and MAPE of 4.02%, this shows higher accuracy in absolute and relative terms. The RMSE of (3.66), SMAPE (3.94%), and quantile loss (1.77), reflect improved forecasting performance and uncertainty estimation. On the other hand, the poorer performance of the LSTM and GRU models are probably attributed to their architecture. They process sequences sequentially and struggle to comprehend longer temporal series and complex patterns due to their inherent limitations. Overall, the TFT's ability to model intricate temporal dynamics and feature interactions results in more accurate and reliable heart rate forecasts, making it more suitable for clinical applications than the LSTM and GRU models

5.4 Visualizations

To further illustrate the TFT model's performance, a visualization of its heart rate predictions over a 6-hour forecasting horizon for a selected patient is presented.

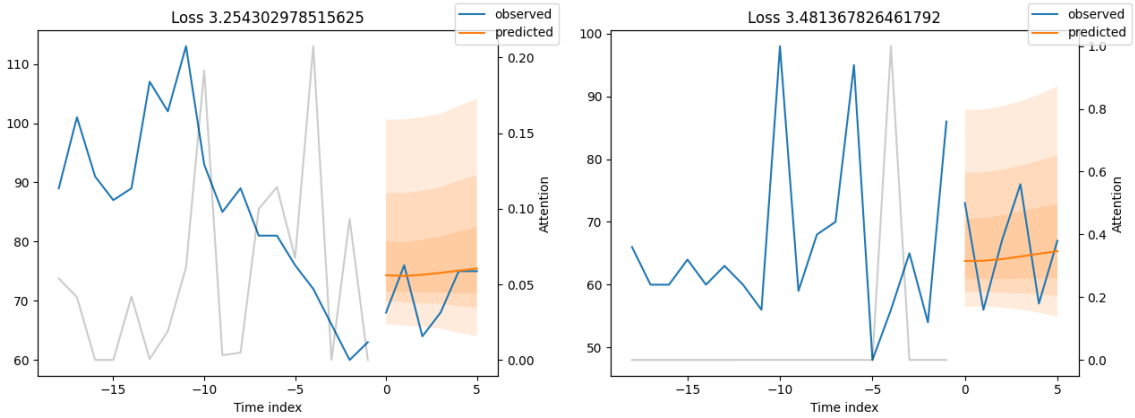

Figure 3. Example of TFT heart rate predictions over a 6-hour horizon with quantile forecast range.

Breaking down the example visualizations, the shaded regions represent the attention weights over the forecast horizon, showing the model's confidence and/or uncertainty. The predictions encapsulate the actual heart rate trajectories, essentially capturing short and long-term trends. The quantile ranges account for any sudden increases and decreases in heart rate, which shows that the model recognizes that heart rate can vary.

6 Conclusion

This work explored the application of the Temporal Fusion Transformer (TFT) for forecasting heart rate for ICU patients' multiple steps into the future using the MIMIC-IV dataset. After preprocessing the data and training several models, metrics show that the TFT exceeds the LSTM and GRU models across the board, including MAE, MAPE, RMSE, SMAPE, and quantile loss. Its much better performance is most likely ascribed to its advanced architecture, comprising of multiple components, utilizing both static and time-varying covariates, which all together efficiently capture intricate temporal dependencies both long and short.

However, despite these encouraging results, limitations such as reliance on data from a single hospital in Boston, Massachusetts and strict hardware computational complexity should be

acknowledged. In the future, further work could focus on broadening where the data comes from to enhance generalizability, integrate new relevant patient variables that weren't present in the MIMIC set, and utilize a more robust hardware system to allow for the creation of more complex models, which in turn will capture more complex relationships, enhancing model performance.

In conclusion, the use of the Temporal Fusion Transformer for heart rate forecasting represents a promising avenue in healthcare analytics, and a very real prospect of integration into clinical workflows. By leveraging its accurate prediction capabilities, and uncertainty estimates with quantiles, healthcare providers can potentially improve patient outcomes through a more guided insight into their health.

References

[1] Rolfe, S. (2019). The importance of respiratory rate monitoring. *British Journal of Nursing, 28*(8), 504-508. https://doi.org/10.12968/bjon.2019.28.8.504

[2] Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, 380(14), 1347-1358.

[3] Malik, M., & Camm, A. J. (2017). Dynamic electrocardiography. In *Electrophysiological Disorders of the Heart* (2nd ed., pp. 75-81). Elsevier.

[4] Pereira, C. R., Weber, S. A., Hook, C., & Papa, J. P. (2016). Deep learning-aided heart condition assessment based on ECG signals for mobile devices. *Neurocomputing*, 166, 117-123.

[5] https://medium.com/metaor-artificial-intelligence/the-exploding-and-vanishing-gradients-problem-in-time-series-6b87d558d22

[6] Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A survey of recent advances on deep learning techniques for electronic health record (EHR) analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5), 1589-1604

[7] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748-1764.

[8] Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine Learning in Medicine. *New England Journal of Medicine*, 380(14), 1347–1358. https://doi.org/10.1056/NEJMra1814259

[9] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24-29.

[10] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & van Ginneken, B. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

[11] Cai, X., Xu, Z., Wu, W., Li, J., & Zhang, Q. (2022). A survey of deep learning for electronic health records. *Applied Sciences, 12*(22), 11709. https://doi.org/10.3390/app122211709

[12] Goldstein, B. A., Navar, A. M., Pencina, M. J., & Ioannidis, J. P. A. (2017). Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 24(1), 198–208. https://doi.org/10.1093/jamia/ocw042

[13] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.

[14] Jones, S. S., Thomas, A., Evans, R. S., Welch, S. J., Haug, P. J., & Snow, G. L. (2008). Forecasting daily patient volumes in the emergency department. *Academic Emergency Medicine*, 15(2), 159–170. https://doi.org/10.1111/j.1553-2712.2007.00032.x

[15] Lipton, Z. C., Kale, D. C., & Wetzel, R. (2016). Modeling Missing Data in Clinical Time Series with RNNs. In *Proceedings of the Machine Learning for Healthcare Conference* (pp. 253–270).

[16] Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1), 6085.

[17] Pascanu, R., Mikolov, T., & Bengio, Y. (2012). *On the difficulty of training recurrent neural networks*. arXiv. https://arxiv.org/abs/1211.5063

[18] Johnson, A., Bulgarelli, L., Pollard, T., Gow, B., Moody, B., Horng, S., Celi, L. A., & Mark, R. (2024). MIMIC-IV (version 3.1). *PhysioNet*. https://doi.org/10.13026/kpb9-mt58.

[19] Johnson, A.E.W., Bulgarelli, L., Shen, L. et al. MIMIC-IV, a freely accessible electronic health record dataset. Sci Data 10, 1 (2023). https://doi.org/10.1038/s41597-022-01899-

[20] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

[21] Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

[22] National Center for Health Statistics (NCHS). (2009). *International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM)*. U.S. Department of Health and Human Services. https://www.cdc.gov/nchs/icd/icd9cm.htm

[23] Centers for Medicare & Medicaid Services (CMS). (2022). *ICD-10-CM/PCS: International Classification of Diseases, 10th Revision, Clinical Modification/Procedure Coding System*. U.S. Department of Health and Human Services. https://www.cms.gov/Medicare/Coding/ICD10

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, 5998–6008.

[25] Otte, C. (2020). PyTorch Forecasting: Time Series Forecasting with Deep Learning. Retrieved from https://github.com/jdb78/pytorch-forecasting

[26] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631.

[27] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[28] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv*. https://arxiv.org/abs/1406.1078

[29] Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

[30] Falcon, W., & The PyTorch Lightning Team. (2019). *PyTorch Lightning*. GitHub. https://github.com/Lightning-AI/lightning