Computers in Biology and Medicine 194 (2025) 110406

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Multivariate multi-horizon time-series forecasting for real-time patient monitoring based on cascaded fine tuning of attention-based models

Hager Saleh [a,b,c,*], Shaker El-Sappagh [d,e], Michael McCann [f], Saeed Hamood Alsamhi [a,g,h], John G. Breslin [a]

[a] Insight Research Ireland Centre for Data Analytics, School of Engineering, University of Galway, University Road, Galway, H91 TK33, Ireland
[b] Atlantic Technological University, Letterkenny, Donegal, Ireland
[c] Faculty of Computers and Artificial Intelligence, Hurghada University, Hurghada, Egypt
[d] Faculty of Computer Science and Engineering, Galala University, Suez, 435611, Egypt
[e] Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, Banha, 13518, Egypt
[f] Department of Computing, Atlantic Technological University, Letterkenny, Donegal, Ireland
[g] Department of Computer Science and Engineering, College of Informatics, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul, 02841, Republic of Korea
[h] Department of Electronics Engineering, IBB University, Ibb, Ibb, Yemen

ARTICLE INFO

Keywords:
Real-time patient monitoring
Time-series data forecasting
Intensive care unit
Sequence-to-sequence modeling
Multi-horizon forecasting
Attention-based modeling
Deep learning

ABSTRACT

The real-time forecasting of critical physiological indicators in intensive care units (ICUs) is essential for early intervention and clinical decision support. This study introduces a novel framework, StreamHealth Multi-Horizon AI, which has been designed to perform multivariate, multi-horizon time-series forecasting for vital signs, specifically for a person's blood oxygen saturation level (SpO2) and respiratory rate (RR). The framework leverages advanced attention-based models, with a particular emphasis on the Temporal Fusion Transformer (TFT) and Temporal Convolutional Network (TCN), and we benchmark its performance against classical deep learning architectures, including LSTM, GRU, Bi-LSTM, Bi-GRU, CNN, and Sequence-to-Sequence (Seq2Seq) models with and without attention mechanisms. Both univariate and multivariate forecasting tasks are explored across multiple prediction horizons (i.e., 7, 15 and 25 min), using physiological time-series data from the MIMIC-III database. The proposed system incorporates a cascaded fine-tuning strategy, wherein the TFT model is sequentially fine-tuned on individual patients' data, significantly enhancing the model's generalizability to unseen patient profiles. Empirical results demonstrate that the TFT model consistently outperforms baseline models across all forecasting settings, achieving lower RMSE and MAE values, and exhibiting superior capacity for capturing long-sequence dependencies and temporal feature dynamics.

To validate its applicability in real-time clinical environments, the framework integrates a simulated streaming infrastructure using Apache Kafka and Apache Flink, enabling continuous data ingestion, forecasting, and visualization of vital signs. This end-to-end deployment underscores the system's potential for ICU monitoring, allowing clinicians to anticipate patient deterioration proactively. In summary, we introduce a comprehensive framework that uniquely integrates TFT with cascaded fine-tuning for multivariate, multi-horizon forecasting of critical ICU indicators. Additionally, we demonstrate a simulation for a real-time deployment pipeline using Kafka and Flink, enabling robust and generalizable ICU monitoring in clinical settings. As a result, this work has contributed a robust and clinically relevant AI solution for real-time healthcare monitoring.

1. Introduction

Blood oxygen saturation (SpO2) and respiratory rate (RR) are critical indicators of a patient's health, particularly in intensive care settings [1–3]. For example, chronic obstructive pulmonary disease (COPD) patients can experience fluctuating SpO2 levels, underlining the limitations of intermittent measurements (that could miss these fluctuations) and the importance of predictive monitoring [4]. The continuous monitoring of RR is equally important, particularly for patients with COPD, respiratory infections, or asthma, as it helps to

* Corresponding author at: Insight Research Ireland Centre for Data Analytics, School of Engineering, University of Galway, University Road, Galway, H91 TK33, Ireland.
E-mail address: hager.saleh@insight-centre.org (H. Saleh).

https://doi.org/10.1016/j.compbiomed.2025.110406
Received 21 October 2024; Received in revised form 15 May 2025; Accepted 17 May 2025
Available online 10 June 2025

detect disease severity and ensure timely interventions. Predicting SpO2 and RR simultaneously provides a more complete understanding of respiratory conditions, enabling early detection of breathing distress [3].

Integrating artificial intelligence (AI) into healthcare has enabled significant advances in both monitoring and predictive capabilities [5]. AI techniques, including deep neural (DL) networks and transformer models, are being increasingly deployed to analyze complex patterns in medical data and to predict vital signs, thereby bringing transformative improvements to patient care. These studies have shown promising results in anticipating changes in vital signs over time [6–9]. Erion et al. [10] applied ML and DL models to predict hypoxemia or non-hypoxemia using time series data based on different features such as demographic data, real-time measurements of vital signs, and laboratory results. Bandopadhaya et al. [11] proposed an e-health solution that integrates deep learning (DL) models with IoT devices for the early detection of problematic SpO2 levels in COVID-19 patients. Priem et al. [12], Zhang et al. [13] and Tonmoy et al. [14] applied DL models to predict SpO2 using a photoplethysmogram (PPG). Shuzan et al. [15] applied machine learning (ML) and DL algorithms as single regression models to predict RR and SpO2.

The authors in [16–18] used the Beth Israel Deaconess Medical Center (BIDMC) [19] dataset to train and evaluate ML models to predict RR and SpO2. Kumar et al. [16] applied different DL models such as LSTMs, CNNs, LSTMs with an attention layer, hybrid CNN-LSTMs, bidirectional-LSTMs (Bi-LSTMs), and Bi-LSTMs with an attention layer for one-step ahead prediction of RR using 32 s and 64 s windowing on the Capnobase and MIMIC-II (via BIDMC) datasets. Lee et al. [17] presented a method combining gradient boosting (GB) with autocorrelation-based power spectral feature extraction to predict RR using the BIDMC dataset.

In multi-horizon forecasting, predictions are provided for multiple future horizons or time points rather than just one step ahead as in traditional time series forecasting. This can also be applied to univariate or multivariate time series [20], and UMH or MMH is used in our paper to refer to univariate or multivariate multi-horizon (time series) forecasting respectively. Most literature studies, especially in the ICU domain, depend on a single input feature and give predictions one step into the future [21]. For example, the authors in [16] predicted RR just one second into the future, which significantly limits the practical medical utility of their results, as ICU physicians require longer prediction horizons to make informed clinical decisions.

Models that can interpret multivariate data and provide multi-horizon time series forecasting are crucial in sensitive medical domains, especially in settings that need real-time monitoring and forecasting, such as in ICUs [22]. Real-time processing of time series data is crucial for providing timely and accurate decisions for effective interventions. Therefore, combining the capabilities of advanced AI models and stream processing can enable real-time ingestion, collection and analysis, and provide timely and accurate decision-making processes [23]. To address the challenges mentioned above, we pose the following research questions:

1. How can multi-task learning models be designed to accurately and simultaneously forecast multiple physiological signals (SpO2 and RR) over multiple future horizons in real-time ICU environments?
2. In what ways can transformer-based architectures, specifically the Temporal Fusion Transformer (TFT), outperform traditional deep learning models in capturing long-term dependencies and multivariate dynamics in ICU data?
3. How does the proposed StreamHealth Multi-Horizon AI framework enhance model generalizability and enable practical deployments in intensive care settings through its integration of cascaded fine-tuning and real-time streaming infrastructures?
4. How can distributed systems such as data lakes and streaming data architectures improve the real-time processing of time series data for healthcare monitoring?

We have found no studies in the literature that have investigated these research questions using transformers and time series data in ICU settings [24]. MMH time series forecasting is critical in domains requiring continuous monitoring and decision-making, particularly in sensitive healthcare environments like ICUs. These settings demand real-time predictions of multiple physiological indicators such as RR and SpO2 to enable proactive interventions and improve patient outcomes. While classical deep learning models, including GRUs, LSTMs, Bi-LSTMs, Bi-GRUs and CNNs, have shown promise in time series analysis, they can struggle with capturing long-sequence dependencies and inter-variable interactions in multivariate forecasting tasks [16].

Enhancements such as attention mechanisms in sequence-to-sequence (S2S) architectures (e.g., S2S-LSTM, S2S-GRU) partially address these limitations, but have yet to be shown to demonstrate robust performance in multi-horizon forecasting within ICU settings. Transformer-based models, particularly the Temporal Fusion Transformer (TFT) [25], have emerged as a powerful alternative due to their ability to model complex temporal relationships and dynamic feature selection. Despite their potential, the application of transformer models to MMH forecasting in ICUs remains underexplored.

Existing research often focuses on single-variable or short-horizon predictions, leaving a significant gap in addressing the unique challenges posed by ICU datasets, such as those from the MIMIC-III dataset [17]. These challenges include high-dimensional data, variability in physiological signals, and long-term predictions to support critical care decisions. This study aims to bridge this gap by comprehensively evaluating classical DL models, S2S architectures, and transformer-based models for MMH time series forecasting in ICUs. Leveraging the MIMIC-III dataset, we focus on understanding the performance limitations of these models, and highlight the superiority of transformers like TFT in handling long-sequence dependencies, generalizability, and real-time forecasting capabilities. By addressing these gaps, this research contributes to advancing predictive analytics in healthcare, and supports the development of robust, real-time monitoring systems tailored to critical care environments.

We have implemented the novel StreamHealth Multi-Horizon AI (SMHA) framework to address the essential challenges of multivariate multi-horizon (MMH) real-time healthcare monitoring. This framework is designed to overcome limitations in existing methods, such as handling complex multivariate time series data, intermittent measurements, and the need for accurate synchronization across multiple variables. By integrating AI with big data streaming technologies, SMHA leverages an attention-based encoder–decoder model alongside a robust data infrastructure, including a data lake and streaming architecture. This approach enhances predictive accuracy for RR and SpO2 over various time horizons, marking a significant advancement in real-time healthcare analytics.

The proposed framework also demonstrates the practical implementation of MMH forecasting by combining multivariate data aggregation from IoT devices and sensors with a distributed file system using Apache Flink and InfluxDB for efficient time series data handling. Grafana further supports the system by enabling real-time analysis and visualization of raw and predicted data. This comprehensive integration ensures high availability, fault tolerance, and actionable insights, ultimately contributing to proactive healthcare management and decision making. By addressing existing gaps and advancing real-time monitoring, the SMHA framework paves the way for improved predictive analytics in healthcare settings.

The current study investigates in detail the capabilities of time series transformers (e.g., TFTs) to be able to learn from multivariate time series data, and how they can be used to provide multi-horizon predictions that are compared with classical DL models and models

with bidirectional, S2S, and attention features. The study explores the robustness, generalizability, and stability of these models. In addition, the architecture is extended to provide real-time monitoring. The main objectives of our paper can therefore be summarized as follows:

- *Introduction of the SMHA Framework:* A novel framework has been developed for the real-time multivariate multi-horizon forecasting of critical ICU indicators such as SpO2 and RR, integrating data lakes, streaming data, and Temporal Fusion Transformers.
- *Exploration of Transformer Architectures:* For the first time, the use of advanced transformer models, specifically the TFT, has been investigated for MMH forecasting of physiological indicators in ICU settings, addressing some of the limitations with classical deep learning methods.
- *Comparison with Classical Deep Learning Models:* Comprehensive experiments have been conducted with classical and sequence modeling methods, including LSTM, GRU, Bi-LSTM, Bi-GRU, CNN, S2S, and S2S-Attention, benchmarking their performance in both UMH and MMH forecasting tasks.
- *Demonstration of TFT Superiority:* The TFT model significantly outperforms classical deep learning approaches in both the UMH and MMH tasks, particularly in handling long-sequence dependencies and complex temporal dynamics.
- *Robust Cascaded Fine-Tuning:* A cascaded fine-tuning approach has been implemented and validated, demonstrating the generalizability and robustness of the TFT model using unseen patient data from the MIMIC-III dataset.
- *Real-Time Forecasting System:* A simulated sensor system was developed, leveraging Python, Kafka, and Apache Flink for real-time data generation, slicing, and forecasting using S2S-Attention models for both SpO2 and RR in parallel.
- *Advancing Healthcare Monitoring:* This work has contributed to improving healthcare decision-making by enhancing the predictive accuracy of real-time ICU monitoring systems.
- *Validation on MIMIC-III Dataset:* The framework has been validated using a dataset of 20 patients, demonstrating the feasibility and efficacy of the approach on real-world clinical data.
- *Generalizable Insights for Physiological Data Modeling:* Key challenges in multivariate time series forecasting were addressed, including temporal synchronization and variability, thereby providing insights applicable to broader physiological and healthcare data analytics domains.
- *Potential for Broader Adoption:* The practical applicability of the framework has been highlighted in real-time healthcare monitoring systems, paving the way for future research and deployment in clinical settings.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 presents an overview of big data streaming platforms. Section 4 presents the methodology and our architecture. Section 5 discusses the experimental results and has an associated discussion. Section 6 presents some of the limitations and our ideas for future work. Finally, Section 7 concludes the paper.

2. Related work

This section is structured into four subsections to discuss the literature related to four aspects: predicting SpO2, predicting RR, transformer-based models, and finally, real-time systems for medical applications.

2.1. Predicting SpO2

Various studies have applied ML and DL classification models to predict hypoxemia or non-hypoxemia. Erion et al. [10], applied logistic regression (LR), XGBoost, a one-dimensional convolutional network (1DCNN), and LSTM on a patient's blood oxygenation data to predict hypoxemia or non-hypoxemia using a private dataset that collected from an academic medical center's Anesthesia Information Management System (AIMS). This dataset included demographic data (age, sex, height, weight), patient information, diagnoses, treatments, and observations. Annapragada et al. [26] presented two stages of DL models: regression and classification to predict SpO2 and to classify SpO2 levels into hypoxemia and non-hypoxemia. For the regression stage, SWIFT (SpO2 Waveform ICU Forecasting Technique) used two different LSTM architectures for forecasting SpO2 30 min into the future. The first deep LSTM architecture consisted of five hidden layers, and the second shallow LSTM architecture consisted of two hidden layers. Both had batch normalization and an output layer. The two models, used to predict the level of SpO2, were evaluated by MSE. Then, each time point was classified as hypoxemia and non-hypoxemia for the classification stage based on an SpO2 threshold of 92%.

Bandopadhaya et al. [11] proposed an e-health solution that integrates DL models with IoT devices for the early detection of problematic SpO2 levels in COVID-19 patients. A time series of SpO2 levels for patients was collected via IoT and used to train the encoder–decoder LSTM model and predict SpO2 levels. The results showed that the encoder–decoder LSTM model recorded the lowest error. More studies have applied ML and DL regression models to forecast RR and SpO2, and recently, deep learning methods have been used to leverage PPG. Shuzan et al. [15] applied different regression models: support vector regression (SVR), Gaussian process regression (GPR), ensemble trees, linear regression, and decision tree regression (DTR) to forecast RR and SpO2 separately using photoplethysmograms (PPG). They applied different feature selection (FS) methods using the feature (selection) Ranking Library (FSLib) to reduce the dimensionality of the PPG data. GPR was chosen as the best ML algorithm for both RR and SpO2, and the FS methods FitRGP and ReliefF gave the best performance for RR and SpO2, respectively.

Gurvan et al. [12] applied the DL model to predict SpO2 using PPG signals collected from the BiOSENCY BORA Band SpO2 Validation Study (BORA) dataset. Zhang et al. [13] presented a customized dataset collected from wearable sensing of PPG signals that were used to monitor SpO2. They then applied linear/nonlinear models to predict SpO2, resulting in a low RMSE of 1.8%. Tonmoy et al. [14] applied different ML regression methods: logistic regression (LR), decision tree regression (DTR), random forest regression (RFR), support vector regression (SVR), and K-neighbors regression to predict SpO2 using a private dataset of PPG signals collected using a smartphone. The results showed that LR recorded the best performance with the lowest MAE.

Chowdhury et al. [18] proposed the ROSE-Net model inspired by DenseNet and ConvMixer and then adapted it for one-dimensional data. It included three stages: a projection stage, a convolution stage, and a pooling with SpO2 estimation stage. Convolution layers with a patch size equal to the stride were used to reduce the representation of a single input. The output of each layer was concatenated with all the other layers in the convolution layer, which were densely connected. Then, the final features were pooled using the Global Average Pooling Layer to estimate the SpO2 level. The model was trained on clinical PPGs from BIDMC and was tested on the rPPG dataset. These results demonstrated the model's ability to estimate SpO2 levels. As observed, most literature studies, especially in the ICU domain, depend on a single input feature and can only predict one time step into the future.

2.2. Predicting RR

As mentioned, Kumar et al. [16] have applied different DL models (CNN, LSTM, LSTM with an attention layer, a hybrid CNN-LSTM model, Bi-LSTM, and Bi-LSTM with an attention layer) in order to be able to predict RR one step ahead of time. They used 32- and 64-second windowing on the BIDMC (extracted from MIMIC-II) and Capnobase

datasets, which include electrocardiograms (ECGs) and PPGs, plus surface electromyogram (sEMG) data that was collected from various biosensors. The results showed that Bi-LSTM with an attention layer recorded the best performance with the lowest MAE for the one-step head prediction of RR. Baker et al. [27] proposed a simple and effective respiratory quality index (RQI) scheme to assess the degree of quality of each modulation-extracted respiration signal from ECG and PPG data extracted from MIMIC-III. They then developed a BiLSTM model for estimating RR, which employs calculated RRs and their supplementary quality indices as input features. They estimated RR based on 20-, 30-, and 60-second segments, and impressive MAE results were shown even down to the shortest segment length. The results showed that their proposed RQI with a BiLSTM recorded the best results for continuous and non-invasive respiratory rate monitoring.

Soojeong et al. [17] presented a method that combined gradient boosting (GB) with an autocorrelation function-based power spectral feature extraction process to predict RR using the BIDMC dataset, which achieved higher stability and accuracy. Their method recorded the best performance when compared to LSTMs and SVR. Bian et al. [28] proposed a DL model based on a network architecture (ResNet) to predict RR values. PPG time-series data was used to train the model through a process that augmented the data with a synthetic PPG dataset to overcome the insufficient data problem often encountered in DL. Their results showed that the proposed DL model scored the best compared to classical methods. Again, most literature studies, especially in the ICU domain, depend on a single input feature and predict a single time step into the future.

2.3. Transformer-based models

Classical ML and DL models have some limitations in terms of long sequence time-series forecasting. Transformer-based models using attention mechanisms can learn long temporal dependencies [29]. TFT is a well-known and powerful transformer-based model for multi-horizon time-series forecasting [25], which has been used in different domains. Li et al. [30] presented a novel model that combined CNNs and TFTs to detect Obstructive Sleep Apnea (OSA) using single-lead ECG signals. The model used a deep residual shrinkage module, a multi-scale convolutional attention (MSCA) module, and a multilayer convolution module to extract rich time-frequency features from short ECG sequences efficiently.

In an ICU setting, Sun et al. [31] proposed the Static and Multivariate Temporal Attentive Fusion Transformer (SMTAFormer) to predict short-term ICU readmission risks by integrating static and dynamic temporal clinical data, using the MIMIC-III dataset to construct the Readmission Risk Assessment (RRA) dataset. Leveraging a transformer encoder for temporal feature representation and a multi-head attention mechanism, SMTAFormer captures intra-correlations among multivariate temporal features and inter-correlations with static data.

He et al. [32] proposed TFT-multi, an extension of the Temporal Fusion Transformer (TFT), designed for simultaneous multivariate time-series forecasting of vital sign trajectories in the ICU. Addressing the challenges of predicting multiple interconnected variables, the model enhanced the original TFT by modifying its input–output structure and loss function to handle multivariate data more efficiently. Focusing on the healthcare domain, it predicted five vital signs: mean arterial blood pressure, pulse, SpO2, temperature, and respiratory rate, using data from MIMIC-IV plus an independent institutional dataset. These studies highlighted the TFT's potential in multi-horizon forecasting of multivariate ICU features. However, there is a research gap in this literature whereby the TFT has not been deeply tested under different settings, including single-task, multitask, and cascaded fine-tuning for improved generalizability. We investigate in much more detail the capabilities of TFTs compared with classical DL models to predict multivariate outcomes under different training methodologies.

Recent advances in time-series forecasting have introduced models such as N-BEATS [33], Informer [34], FEDformer [35], and Prophet [36]. N-BEATS is a powerful univariate architecture based on backward and forward residual links, primarily designed for interpretable univariate forecasting tasks. Prophet, developed by Facebook, is widely used for business-oriented seasonal forecasting but is less suitable for multivariate, high-frequency, and real-time applications. Informer and FEDformer are efficient transformer-based architectures optimized for extremely long sequences and massive-scale datasets (e.g., weather and traffic), often requiring high computational resources and lacking the fine-grained temporal interpretability necessary for ICU monitoring. In contrast, TFT explicitly supports multivariate multi-horizon forecasting, incorporates static and time-varying covariates, and provides interpretable outputs, making it well-suited for healthcare time-series analysis. Therefore, although we recognize the contributions of these models, we focused our evaluation on architectures that align closely with the clinical forecasting goals of this study.

2.4. Real-time systems for medical applications

Various research studies have used Spark with ML to solve medical problems. For example, Nair et al. [37] introduced a real-time system for predicting heart disease using a decision tree (DT) and Spark. The system was tested using health attributes that were extracted from streaming tweets. Then, the DT was applied to predict the health status of the user. Abderrahmane et al. [38] developed a real-time system for predicting cancer diseases based on Spark and DTs. Firstly, an offline model was developed using preprocessing and by analyzing historical cancer datasets. Then, the model was integrated with Spark to give predictions in real time. Ed-daoudy et al. [39] applied six ML models with feature selection methods to develop offline models that were also used in real time. Kafka was used to received streaming health tweets and they were then ingested into Spark. The Spark Streaming extension was used to extract health attributes, and random forest (RF) was applied to give predictions in real time.

Ahmed et al. [21] applied various ML models in order to determine the best model which could be used to predict heart disease in real time. They used univariate and relief feature selection methods with DT, SVM, RF and LR applied on the heart disease dataset. Kafka was used to read data from Twitter and stream it to Spark. RF was then applied to predict heart disease in real time. Farki et al. [40] investigated techniques for real-time blood pressure forecasting, and integrated ML techniques with real-time streaming data processing systems like Apache Spark and Kafka. Web-based tools were implemented to test the scalability of this technology for remote patient tracking and individualized healthcare. Ahmed et al. [41] also investigated a real-time system for forecasting systolic blood pressure (SBP). They applied DL models such as LSTM, Bi-LSTM and GRU on historical time-series BP data, in order to determine the best model for forecasting. The Bi-LSTM model was found to deliver the best results in predicting near-future values for SBP in real time. Simulated sensors were used to generate streaming SBP values, which were then sent to a Kafka topic (unit of organization). The Spark Streaming extension was then used to read this Kafka data in a streaming form, after which sliding window sizes were applied to the data, and this was sent to the Bi-LSTM model to predict near-future SBP values.

Liang Tan et al. [42] proposed a 5G-enabled real-time monitoring for COVID-19 patients using big data platforms and DL models. Firstly, they developed CNN and LSTM models for predicting COVID-19 using ECG signals. Secondly, 5G was used to send and receive data from wearable sensors. After that, the Flink streaming data processing framework was applied to access electrocardiogram data.

Table 1
Comparing literature studies based on highlights and limitations.

Papers	Years	Highlights	Limitations	Datasets
Erion, Gabriel et al. [10]	2017	Applied DL to forecast hypoxemia using just SpO2. Used a personalized dataset (AIMS).	Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	Private dataset
Nair et al. [37]	2018	Introduced a real-time system with DT to predict heart disease using a structured dataset, Spark and Kafka. Tested predictions in real time using health-related attributes extracted from streaming tweets.	Used structured datasets, did not use time-series data. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, or weight optimization using multi-patient time-series data.	MIMIC-III
Abderrahmane et al. [38]	2018	Introduced a real-time system with DT to predict cancer disease using a structured dataset, Spark and Kafka. Tested predictions in real time using a simulated dataset.	Used structured datasets, did not use time-series data. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, or weight optimization using multi-patient time-series data.	Structured cancer disease dataset
Ahmed et al. [21]	2020	Introduced a real-time system with ML and feature selection methods to predict heart disease using a structured dataset, Spark and Kafka.	Used structured datasets, did not use time-series data. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, or weight optimization using multi-patient time-series data.	Structured heart disease dataset
Ahmed et al. [41]	2021	Investigated a real-time system for forecasting BP in real time using time-series data, based on a DL model, Spark and Kafka.	Used only one feature based on time-series data. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, or weight optimization using multi-patient time-series data.	MIMIC-III
Annapragada et al. [26]	2021	Forecasted SpO2 using a DL model and time-series data. Applied classifications and regressions. Applied DL to the prediction of hypoxemic events using SpO2. Proposed their SWIFT model to estimate results.	Forecasted SpO2 only. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	Private dataset
Baker et al. [27]	2021	Respiratory modulation signals were extracted from ECG and PPG waveforms to estimate RR. Applied several different neural network (NN) structures to predict RR. Developed an RQI scheme to assess the results.	Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	MIMIC-III
Kumar et al. [16]	2022	Applied LSTM and GRU DL models to estimate RR and breathing patterns. Applied the attention mechanism to improve the algorithm's performance.	Did not explore models for long-period time series. Used multi-second segments from the BIDMC dataset. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	BIDMC, a subset of MIMIC-II
Soojeong et al. [17]	2022	Combined GB with an autocorrelation-based power spectrum to extract features. Applying ML and DL models to predict RR.	Did not explore models for long-period time series. Used multi-second segments from the BIDMC dataset. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	Private dataset
Zhang, Qingxue et al. [13]	2022	Forecasted SpO2 using linear/nonlinear models based on personalized time-series data.	Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	Private dataset
Shuzan et al. [15]	2023	Applied ML models to estimate RR and SpO2 from PPGs. Applied a feature selection approach.	Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	BIDMC, a subset of MIMIC-II
Bandopadhaya et al. [11]	2023	Forecasted SpO2 using an encoder–decoder model based on time-series data. Applied model on own dataset.	Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	Private dataset
Tonmoy et al. [14]	2024	Forecasted SpO2 using ML models based on personalized time-series data. Applied different pre-processing steps to enhance results.	Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	Private dataset
Chowdhury et al. [18]	2024	Proposed ROSE-Net to estimate SpO2 using DL. Included three stages: a projection stage, a convolution stage, and a pooling with SpO2 estimation stage.	Did not explore models for long-period time series. Used multi-second segments from the BIDMC dataset. Did not cover multi-task transformer models, multivariate multi-horizon forecasting, integrating big data platforms with models, or weight optimization using multi-patient time-series data.	BIDMC, a subset of MIMIC-II

2.5. Gaps in the literature

Table 1 shows a comparison of the research studies related to the areas discussed. The comparison highlights some of the gaps in the literature as regards physiological signal forecasting in ICU settings. Most prior work has focused on single-task modeling, typically addressing either SpO2 or RR independently, without exploiting the potential of multitask learning for joint signal forecasting. Furthermore, most of these studies were limited to univariate and single-horizon time-series predictions, thus overlooking the complex temporal dependencies and interactions between multiple physiological variables over varying forecast horizons. Although transformer-based architectures, particularly the Temporal Fusion Transformer (TFT), have shown promising performance in other domains, their application remains underexplored in healthcare time-series forecasting, with few studies addressing their potential for multi-horizon multivariate predictions. Additionally, existing research has rarely integrated real-time big data infrastructures such as Apache Kafka and Flink with predictive models, relying instead on offline processing or structured datasets that are not representative of streaming ICU environments. Moreover, none of the surveyed approaches adopted patient-level cascaded fine-tuning strategies to improve model generalizability. They did not evaluate model robustness across heterogeneous patient data from large-scale clinical repositories such as MIMIC-III. These limitations collectively underscore the novelty and necessity of the proposed StreamHealth framework, which uniquely integrates multivariate multi-horizon forecasting with attention-based modeling, patient-specific fine-tuning, and real-time deployments to advance continuous and clinically relevant ICU monitoring.

3. Platforms

This section describes various platforms that we used to develop our real-time forecasting framework. It should be noted that integrating a machine learning model into a real-time monitoring system has a well-known and almost standard approach [37–39,41]. We followed the same streaming methodology, but with a different problem setting.

3.1. Apache Kafka

Apache Kafka is an open-source event-streaming system. Event streaming captures real-time data from sensors, databases, cloud services, and applications to form a stream of events. This stream of events can be routed to different applications for storage and processing. Kafka is designed to pipe these events from the source to the desired location [43]. Kafka has three main capabilities. The first is the most basic function: to publish and subscribe to streams of events for writing and reading data. The second function is to store data streams for as long as is needed. The third is to process streams in real time or in a batch format [43]. The Kafka cluster consists of one or more brokers

that manage the system's reading and writing functionality. The cluster uses Apache Zookeeper to maintain its state. Events are stored in topics (a topic is a category or unit of organization in Kafka), and are stored in an append-only sequence. Kafka brokers can hold more than one partition for one topic [43].

Kafka has several APIs for exporting and reading data. The Producer API allows an application to publish a stream of events on one or more Kafka topics. The Consumer API reads one or more topics and processes the events they produce. The Streams API allows stream processing in real-time or batched processing [43]. This can also be used to group topics or perform fundamental transformations, enabling the configuration of input and output streams. Kafka also has a Connect API, which is used to build and run reusable data connectors to import from or export to external applications or systems. Some big data systems have also built their own native connectors to Kafka to allow for easier transferal of events.

3.2. Apache flink

Apache Flink is an open-source framework developed by the Apache Software Foundation for distributed processing of streaming data and batch data. It is designed with data streams in mind, and offers a very low latency and a high level of fault tolerance. Flink works well for both streaming data and big data as it has good scalability in terms of performance, and it can be deployed on several systems, including YARN, Kubernetes, and Mesos [44]. It includes the following functions:

- Table API is an API that offers both traditional tables, similar to those found in SQL, and dynamic tables, which are used for representing data streams. This API provides functionality that is more or less identical to SQL queries with standard commands such as SELECT, JOIN, GROUP BY, etc.
- FlinkML provides a set of scalable ML algorithms and an intuitive API. It contains algorithms for supervised learning, unsupervised learning, data preprocessing, recommendation, and other utilities.
- DataStream API. This allows users to process data streams and work with the data in real time. This is typically not used unless it is needed for significant optimization. The stream and batch data processing layer is where both bounded and unbounded data streams can be processed. This layer contains both the DataStream and DataSet APIs. The DataStream API works with real-time data, while the DataSet API is for batch processing. Batch processing can be specified as a window, either tumbling or sliding, and allows for time-based batching.

3.3. InfluxDB

InfluxDB is a real-time focused database service built with time-series analytics in mind. InfluxDB is also very scalable, allowing for clustering and cloud-based auto-scaling of clusters. InfluxDB is heavily optimized for time-series data [45], which tends to be generated in small sizes but very quickly. This could be up to as high as millions of data points per second, which is far too fast for conventional databases. InfluxDB has been developed for high-availability data retrieval, fast storage, IoT sensor data, and providing real-time analytics [45]. InfluxDB is a time-series database that combines the concepts of a database with retention time and policies. InfluxDB's measurements function acts much like tables do in a relational database, and includes tags, fields and associated timestamp values. The tags are indexed columns, and fields are not. Each record in the InfluxDB is associated with a timestamp in a nanosecond-precision format. This timestamp is essential for writing and processing the data in subsequent layers [45].

3.4. Grafana

Grafana is an open-source software system that allows services to be monitored in a real-time and user-friendly way. It enables querying, visualization, and alerts on data or application logs. It allows developers to create live, easy-to-process dashboards and send live alerts to users. Alerts can be distributed using Grafana Alerting, which can go through several notifiers, including PagerDuty, SMS, email, and Slack [46].

Dashboards are the main feature of Grafana. Dashboards allow for a broad range of data visualizations. A sample dashboard could be one used for monitoring Kubernetes clusters. Such a dashboard would show many visualization types, including metrics over time between locations. It would also give information on any errors, what clusters they originate from, and storage availability. Such dashboards are designed to make logs as readable and user friendly as possible. Plugins allow many data types to be interpreted and various visualizations to be shown to users, such as the outcomes of data analyses [46]. Grafana allows many frameworks and technologies to be monitored in one location. For example, a dashboard could be created for developers to ensure correct data flows from various data producers (sensors) to data consumers (sinks) [46].

4. Methodology

This section gives details of the dataset description, the problem formulation, and the proposed StreamHealth Multi-Horizon AI (SMHA) framework. The proposed framework has two pipelines, as shown in Fig. 1, including (1) a pipeline of model deployment and (2) a pipeline of online forecasting.

4.1. Dataset description

The goal of this work is to determine the best model and use it to evaluate our system in real time. To develop the offline model, SpO2 and RR (multivariate) time series were extracted minute-by-minute for 20 ICU patients with chronic diseases, and the data was obtained from the Medical Information Mart for Intensive Care (MIMIC-III) [47] database. The extracted patients were aged from 53 to over 80 years, and the dataset includes 11 female and 9 male patients. Table 2 shows the database characteristics and some sample entries.

4.2. Problem formulation

Multi-horizon time-series forecasting is vital for real-time patient monitoring in ICUs, as it enables clinicians to anticipate changes in a patient's health over multiple future intervals, supporting proactive decision making. This forecasting provides insights into critical variables such as heart rate, oxygen saturation, and blood pressure, which can exhibit complex temporal dependencies and abrupt changes. Leveraging advanced architectures such as encoder–decoder models and temporal transformers, which effectively integrate static and time-varying covariates, can improve interpretability and accuracy in these scenarios. Accurate predictions empower clinicians to adjust treatments dynamically, reducing risks and improving patient outcomes. In this study, we investigate the prediction of univariate and multivariate time-series data using diverse DL models, including RNNs and transformers, which are well-known models for interpreting time-series data. A crucial step is the formulation of the dataset as a regression task. Formulating the dataset as a regression task using different windows involves defining a structured approach to divide the time-series data into input–output pairs. The process is described as follows:

(1) Define the look-back window (i.e., input sequence) where a fixed number of past observations, referred to as the *look-back window* (k), is used as input features. This window consists of time-dependent input features such as observed values, known inputs, and exogenous

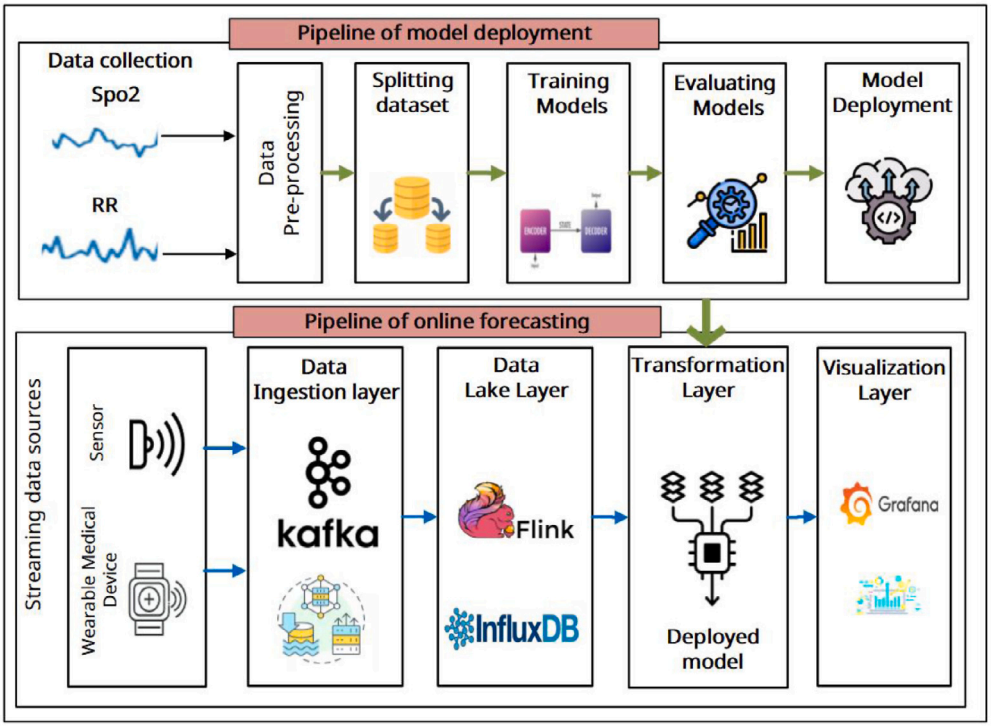

Fig. 1. StreamHealth Multi-Horizon AI (SMHA) framework architecture.

Table 2
Database characteristics and some sample entries.

Patients	Age	Gender	RR mean ± std	RR min	RR max	SpO2 mean ± std	SpO2 min	SpO2 max
Training set	69 ± 10.95	8F/7M	20.561 ± 3.11	9	38	97.85 ± 1.15	82	100
Testing set	64 ± 4.88	3F/2M	23.017 ± 2.23	10	43	97.80 ± 0.96	88	100
Patient 1	70	Female	24.45 ± 6.06	10	35	97.80 ± 2.13	88	100
Patient 2	71	Female	24.95 ± 6.80	13	43	99.40 ± 0.87	96	100
Patient 3	60	Female	19.21 ± 3.69	12	27	96.88 ± 2.44	88	100
Patient 4	60	Male	24.76 ± 5.17	14	35	98.17 ± 2.09	88	100
Patient 5	62	Male	21.69 ± 4.49	11	33	96.74 ± 2.22	87	100

variables. For example, for a time series y_t, the input at time t is defined as:

$$X_t = [y_{t-k}, y_{t-k+1}, \dots, y_{t-1}]$$

(2) Define the forecast horizon (i.e., output sequence), denoted as τ, which specifies the number of steps into the future for which predictions will be made. The output for the model at time t is the target value(s) at future time steps:

$$Y_t = [y_{t+1}, y_{t+2}, \dots, y_{t+\tau}]$$

As shown in Fig. 2, a sliding window technique creates overlapping input–output pairs throughout the dataset. This ensures that each time series step is considered for input and output creation. For each timestamp t, an input–output pair is created as:

Input: $X_t = \{[y_{t-k}, y_{t-k+1}, \dots, y_{t-1}], \text{covariates for } t-k \text{ to } t-1\}$

Output: $Y_t = \{y_{t+1}, y_{t+2}, \dots, y_{t+\tau}\}$

As shown in Fig. 2, we investigate different settings by formulating the problem as a univariate (i.e., RR or SpO2 alone) and multivariate (i.e., RR and SpO2 together) multi-horizon time-series forecast with different input and output window sizes. Different DL models are evaluated for these different problem settings.

We tested several settings for prediction horizons (i.e., 7, 15 and 25 min) based on different lag windows (i.e., 3, 7 and 15 min).

1. The clinical justification for prediction horizons (7, 15 and 25 min): This allows the system to provide:
   (a) Short-term prediction (7 min): Immediate forecasting is critical for detecting acute events such as sudden drops in SpO2 or rapid changes in RR. These short-term forecasts allow swift interventions, such as adjusting oxygen therapy or administering emergency treatments.
   (b) Mid-term prediction (15 min): A mid-range horizon is clinically relevant for anticipating trends that may not require immediate action but indicate the potential for future instability. For example, a gradual decline in SpO2 or a rising RR trend over 15 min might signal the onset of hypoxemia or respiratory distress, providing a window for preemptive measures.
   (c) Long-term prediction (25 min): Long-term predictions are essential for resource planning and proactive patient management in intensive care units (ICUs). 25 min forecasts can enable clinicians to assess the effectiveness of ongoing interventions and adjust care plans proactively.
2. The clinical justification for lag windows (3, 7 and 25 min): The selected lag windows reflect the temporal dependencies necessary for accurate multi-horizon forecasting:
   (a) Short lag (3 min): Captures immediate trends and high-frequency variations in vital signs, essential for real-time monitoring and quick adjustments.

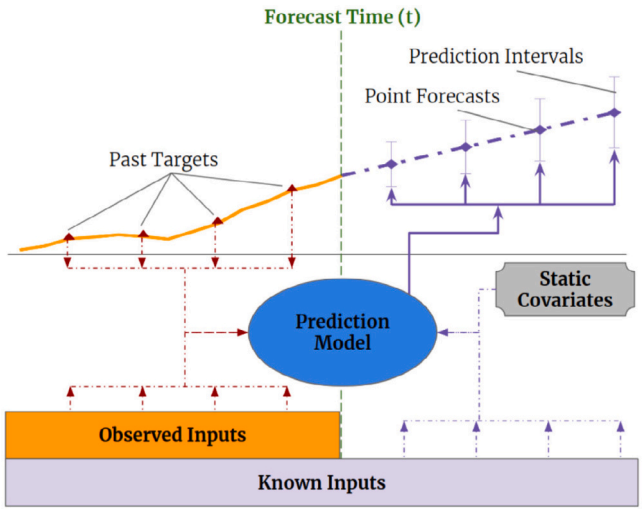

Fig. 2. Problem formulation as a time-series forecasting regression task [25].

(b) Moderate lag (7 min): Balances between capturing short-term variations and incorporating additional historical context, providing a robust input for mid-term forecasts.
(c) Long lag (25 min): Includes extensive historical data to model longer-term patterns and dependencies, enhancing the accuracy of long-term predictions.

3. The alignment with real-world clinical decision-making: These intervals are chosen to mirror critical decision-making time-frames in ICUs, where clinicians rely on continuous monitoring to make time-sensitive decisions:

   (a) Real-time alerts: Short-term predictions (7 min) align with the need for immediate alerts and intervention.
   (b) Trend analysis: Mid-term forecasts (15 min) support trend analysis, guiding decisions on the escalation of care or changes in monitoring intensity.
   (c) Proactive management: Long-term predictions (25 min) aid in planning interventions and allocating resources, such as preparing for potential intubation or transfer to higher levels of care.

4. The AI significance of these settings is as follows:

   (a) The integration of AI-driven models like the TFT ensures that these horizons are not arbitrarily selected but are optimized based on the model's ability to learn and predict meaningful temporal dependencies. The model's self-attention mechanisms enable dynamic weighting of lagged inputs, ensuring that the chosen time frames provide actionable and clinically interpretable predictions.
   (b) AI enhances not only the accuracy of these predictions, but also their utility by aligning with clinical workflows, supporting continuous monitoring, and reducing the cognitive burden on healthcare providers.

4.3. Pipeline of model deployment

Fig. 3 shows the piplines of developing models to foresacte RR and Spo2. The main goal of the first pipeline is to obtain the best model for forecasting RR and SpO2 time series, which is then integrated with the second pipeline in the data transformation layer to forecast RR and SpO2 in real time. We compare the performance of several DL algorithms with the transformer model. These models are tested for forecasting univariate multi-horizon (UMH) and multivariate multi-horizon (MMH) time series. Different experiments are executed to optimize the many models and test their learning capabilities. Firstly, the various DL models such as long short-term memory (LSTM), gated recurrent unit (GRU), bidirectional LSTM (Bi-LSTM), Bi-GRU, and CNN are used for predicting SpO2 or RR in the UMH setting. Secondly, the MMH setting is tested using sequence-to-sequence (S2S) models such as S2S-LSTM and S2S-GRU, and sequence-to-sequence with attention (S2S-A) models such as S2S-A-BiLSTM and S2S-A-BiGRU. Classical DL models achieve promising results but have limitations in handling long sequences. This limitation can be solved using transformer architectures, especially time-series transformers such as the temporal fusion transformer (TFT).

The TFT model is tested for UMH and MMH tasks, and is compared with the other classical DL models. All models are trained and tested using long time-series data for individual patients. Each patient's data is split into 80% training data and 20% testing data, and the model is trained and tested on single-patient data. The TFT achieved superior results compared to all classical DL models. SpO2 and RR MMH time-series forecasting are further investigated using a TFT transformer. To test the model's generalizability and robustness, a cascaded fine-tuning approach is used to sequentially and cumulatively fine-tune the TFT model with a set of patients, and then test it with a different dataset. To perform this experiment, we collect a dataset of 20 MIMIC-III patients and divide them into 15 patients for training and five for testing. The TFT model is fine-tuned 15 times in a cascading way. Then, the model is tested using the five testing examples. The TFT model achieves promising generalization results on the unseen data from the five patients used for testing. A simulated sensor that generates RR and SpO2 time-series is developed using a Python script, and the data is then stored in a Kafka topic.

4.3.1. Data pre-processing

Data pre-processing includes two steps: filling in missing values and normalizing data.

- Filling in missing values: we replace null values using forward fill, a data imputation technique used to handle missing values in datasets, particularly time-series data [48].
- Normalizing data: The data is scaled from the original range to a new range of 0 to 1 to improve and simplify model training. Python's MinMaxScaler has been utilized for scaling numbers to be between 0 and 1. Predicted outcomes are rescaled to the original range using an inverse_transform function [49] in order to evaluate the models.

4.3.2. Sequence-to-sequence autoencoder model

Sequence-to-Sequence (S2S) autoencoder models are comprised of an Encoder, Decoder, RepeatVector and a TimeDistributed layer. They include one Encoder layer and one Decoder layer. S2S models contain LSTMs and GRUs that are utilized like Encoders and Decoders [50]. The input to the models is a sequence of past time steps and the number of features (RR and SpO2), and the output of the model is a sequence of future time steps with several features.

The Encoder part consists of various layers. The Encoder's inputs module processes sequences of inputs structured as (n_past, n_features), where n_features is the total of all the features per time step, including SpO2 and RR data, and n_past is the number of time steps that have previously been taken into consideration. The architecture's GRU layer skillfully captures the temporal relationships present in these input sequences, enabling efficient sequence modeling. To reduce overfitting during training, a dropout layer is deployed to deliberately avoid relying too much on specific nodes. After sequence processing, the Encoder's states module aggregates the final states, including the LSTM

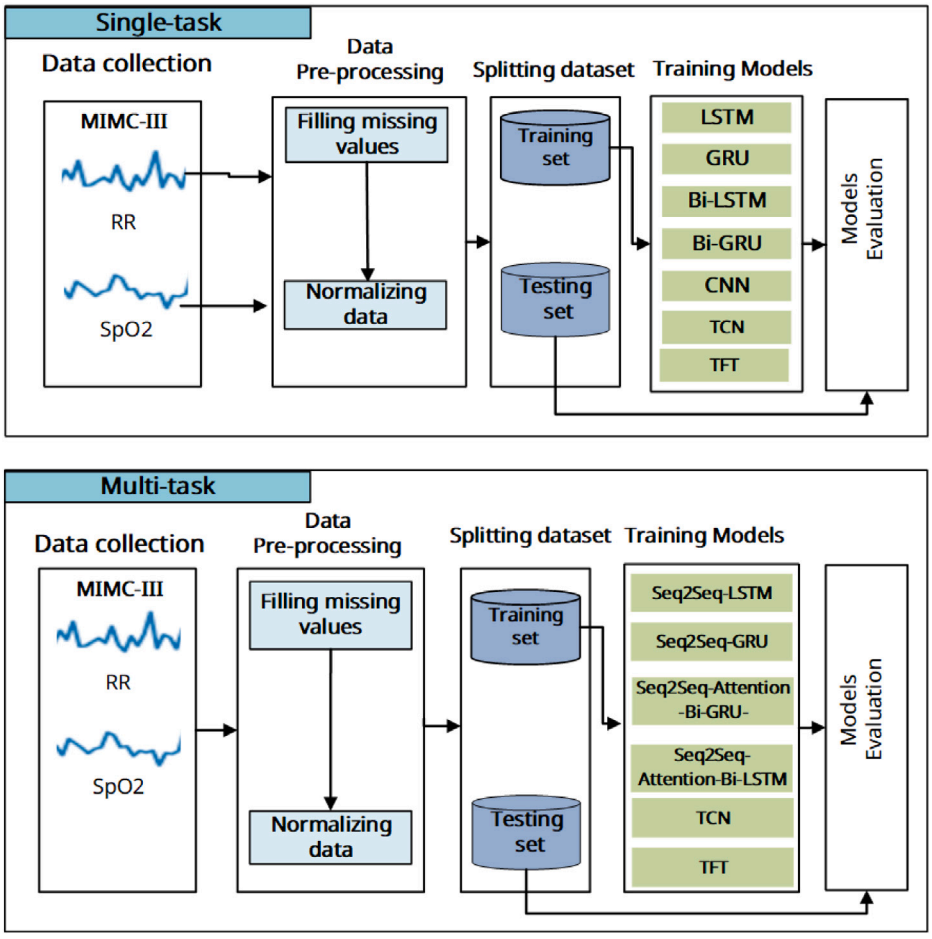

Fig. 3. Univariate multi-horizon and multivariate multi-horizon steps.

layer's hidden and cell states. These combined states function as a succinct synopsis of the input sequence, and are essential for guiding further decoding operations in Algorithm 1.

The RepeatVector layer is an essential design component, as it reproduces the Encoder's output states over several steps and aligns them with every step in the output sequence. This technique ensures that the Encoder provides the Decoder with complete context, improving prediction accuracy and coherence. Then, the Decoder module's LSTM layer analyzes these recurrent Encoder states along with earlier outputs, using this context to repeatedly predict future outputs while training. To efficiently combine these predictions, the TimeDistributed dense layer applies a dense transformation to each time step of the output sequence. The deployment of this layer makes it easier to generate a series of output vectors, each of which adds to the overall forecasting power of the model, as shown in Algorithm 1.

Algorithm 1 Encoder and Decoder process.

1: **Encoder process**
2: **Encoder inputs:** $(n\_past, n\_features)$
   - n_past: Number of time steps in the past.
   - $n\_features$: Number of aggregated features for RR and SpO2.
3: **GRU layer:** Captures temporal dependencies in the input sequences effectively.
4: **Dropout layer:** Prevent overfitting.
5: **Encoder states:** LSTM layer after processing the input sequence.
6: **Decoder process**
7: **RepeatVector layer:** Repeats the Encoder's output.
8: **GRU layer:** Processes the repeated Encoder states.
9: **TimeDistributed dense layer:** Applies a dense transformation to each time step independently.

4.3.3. Sequence-to-sequence with attention models

The core component of the S2S-A model is an attention-based Encoder–Decoder architecture that employs either Bi-GRU or Bi-LSTM, and such models are referred to as either S2S-A-BiLSTM or S2S-A-BiGRU respectively. It consists of two main parts: the Encoder and the Decoder, as shown in Algorithm 2. The Encoder part begins by taking an input sequence of shape (n_past, n_ features), where n_past represents the number of past time steps, and n_features is the number of aggregated features per time step (using RR and SpO2). Bi-GRU processes the input sequence, collecting past and future dependencies, and concatenating the forward and backward hidden states to create the final hidden state.

The GRU output is then subjected to an attention layer, which computes scores between each Encoder's and Decoder's hidden states. These scores are then normalized using the softmax function, enabling the model to concentrate on distinct segments of the input sequence. The Encoder output computes weights for the inputs to prepare for the Decoder based on this attention mechanism. This creates a context vector, representing significant portions of the input sequence and repeating them n_future times via the RepeatVector layer. This repeated attention output serves as the Decoder's input. It processes the sequence using a GRU layer to produce a series of output states. Considering the

Algorithm 2 Process for S2S-A models.

Initializations $(n\_past, n\_features)$, $e_{i,t}$ is the attention score, $\alpha_{i,t}$ is the normalized attention weight
Encoder part
Encoder inputs: $(n\_past, n\_features)$
Process the input sequence.

$$\vec{h}_t = \mathrm{GRU}_f(\vec{h}_{t-1}, x_t)$$

$$\overleftarrow{h}_t = \mathrm{GRU}_b(\overleftarrow{h}_{t+1}, x_t)$$

Concatenate forward and backward hidden states.

$$h_t = [\vec{h}_t; \overleftarrow{h}_t]$$

Apply attention mechanism to the output of the Bi-GRU.
Compute attention scores.

$$e_{i,t} = \mathrm{score}(s_{i-1}, h_t)$$

Normalize the attention weights.

$$\alpha_{i,t} = \frac{\exp(e_{i,t})}{\sum_{t=1}^{T} \exp(e_{i,t})}$$

Calculate weighted inputs based on the attention mechanism.
Apply attention weights.
Generate a context vector.

$$c_i = \sum_{t=1}^{T} \alpha_{i,t} h_t$$

Repeat the attention output.
Output the context vector.
Decoder part
Takes the previous input from the Encoder.
Applied to the repeated attention output sequence.
Processes the output sequence.

$$s_i = \mathrm{GRU}(s_{i-1}, y_{i-1}, c_i)$$

Generate values at each time slot.
Forecast values at a specific time.
Generate the whole output sequence.

$$o_i = W_o[s_i; c_i] + b_o$$

$$P(y_i | y_{<i}, X) = \mathrm{softmax}(o_i)$$

End

context vector and hidden states, the Decoder produces a single value at each time, and the output states added together make up the whole output sequence.

4.4. Temporal convolutional networks (TCNs)

TCN is a type of neural network architecture designed for sequence modeling tasks that handle entire sequences in parallel with stable gradient propagation [51]. A CNN is typically associated with images; TCNs tweak that robust architecture for sequence modeling tasks using one-dimensional convolutions that slide over the input sequence [52]. The real power of TCNs lies in how they manage sequence modeling and time-series forecasting. Its model handles long-term dependencies and respects the causality of time without suffering from memory degradation. The combination of dilated convolutions and parallel processing allows TCNs to outperform traditional methods, while its architecture goes deep enough to capture both short-term and long-term dependencies. At the heart of TCNs is the convolution operation [52]. Specifically, TCNs use dilated convolutions, which can be expressed mathematically as [53]:

$$y(t) = \sum_{i=0}^{k-1} f(i) \cdot x(t - d \cdot i)$$

where $y(t)$ is the output at time step t, $f(i)$ is the filter of size k, and $x(t - d \cdot i)$ is the input sequence. In addition, d represents the dilation factor, which aims to control the spacing between the filter elements to allow the network to expand the receptive field exponentially, corresponding to its depth. Determining the receptive field of a TCN is critical to determine the scope of the input that can be seen by the network. It is calculated from [53]:

$$R = 1 + (k-1) \sum_{i=0}^{k-1} f(i)$$

4.4.1. Multi-horizon time-series forecasting using a temporal fusion transformer

The Temporal Fusion Transformer (TFT) is a novel deep learning model designed for interpretable multi-horizon time-series forecasting. The TFT combines high forecasting accuracy with the ability to provide detailed insights into the underlying temporal dynamics, making it a powerful tool for real-world applications in retail, healthcare, and finance [25]. Fig. 4 presents a high-level architecture of the TFT, which is explicitly designed to address the challenges of multi-horizon forecasting by incorporating diverse data sources and providing interpretability. The architecture integrates three main input types: static covariates, past observed inputs, and a priori known future inputs, ensuring each input type is handled appropriately to capture its contribution to the forecast. The Variable Selection Networks (VSNs) layer dynamically identifies the most salient features for each time step, thus focusing computational effort on relevant inputs. Gated Residual Networks (GRNs) are used extensively throughout the model to enable an efficient flow of information while allowing the network to bypass unnecessary computations. Local time-dependent patterns are captured through LSTM layers, while multi-head attention layers learn long-term temporal dependencies across the dataset. Additionally, context vectors derived from static covariates are integrated at multiple points, allowing static metadata to condition temporal dynamics effectively. Prediction intervals are generated using quantile regression, providing probabilistic forecasts across all horizons. The architecture is modular, combining interpretable components with high-performing temporal layers, enabling the integration of robust forecasting and actionable insights into the model's behavior.

The TFT achieves superior performance, outperforming existing benchmarks across datasets with diverse temporal dynamics, including simple univariate and complex multivariate time series. It enables dynamic feature selection by employing VSNs to identify and prioritize relevant input features at each time step, enhancing predictive accuracy and reducing noise. With Gated Residual Networks (GRNs), the model dynamically skips unnecessary computations, adapting to datasets of varying complexity. Moreover, the TFT provides unified input handling by integrating static metadata, observed historical data, and known future inputs, ensuring comprehensive modeling for diverse datasets. Formally, the goal of multi-horizon forecasting is to predict $y_{t+\tau}$ for $\tau \in \{1, 2, \ldots, \tau_{\max}\}$ based on various input types: static covariates $s \in \mathbb{R}^{m_s}$, observed past inputs $z_{t-k:t} \in \mathbb{R}^{m_z}$, and known future inputs $x_{t-k:t+\tau_{\max}} \in \mathbb{R}^{m_x}$. The predictive model is formulated as:

$$\hat{y}_{t+\tau} = f(s, z_{t-k:t}, x_{t-k:t+\tau}, \tau)$$

where f represents the learnable architecture. The TFT generates probabilistic forecasts for different quantiles q by minimizing the quantile loss:

$$\mathrm{QL}(y, \hat{y}, q) = q(y - \hat{y})_+ + (1 - q)(\hat{y} - y)_+$$

where $(x)_+ = \max(0, x)$. The total loss for training is the sum of the quantile losses across all time steps and quantiles:

$$L(\Omega, W) = \sum_{y_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1}^{\tau_{\max}} \mathrm{QL}(y_t, \hat{y}(q, t-\tau, \tau), q)$$

where Ω represents the training dataset and W are the model parameters. Gated Residual Networks (GRNs) are a core component of TFTs,

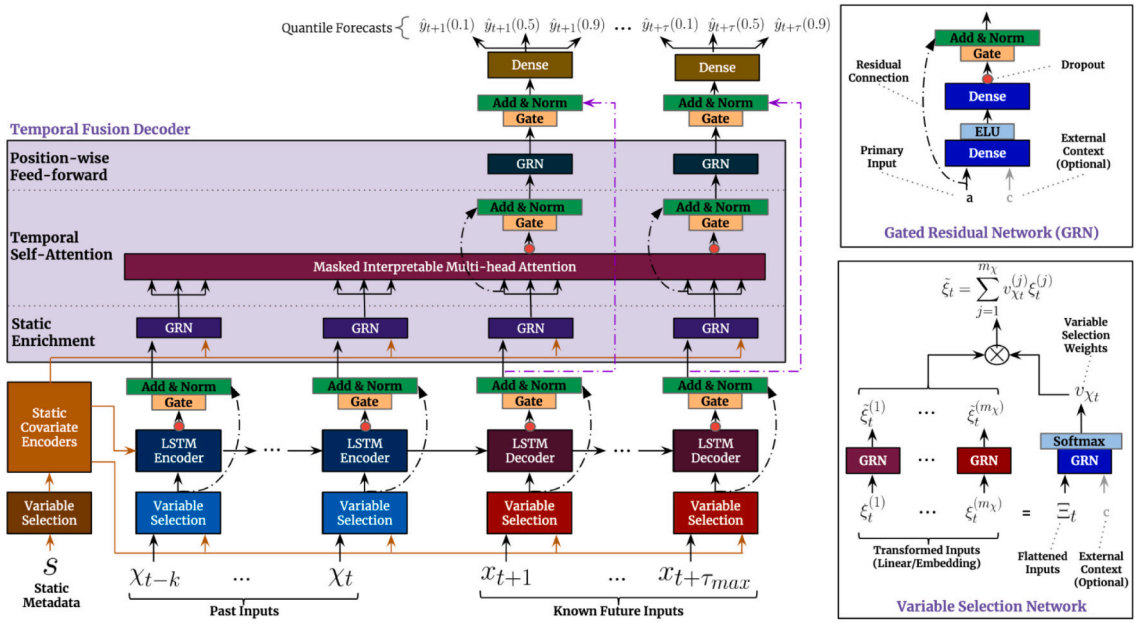

Fig. 4. Temporal fusion transformer [25].

designed to enable adaptive non-linear processing. Given an input a and optional context c, the GRN computes:

$$\mathrm{GRN}(a,c) = \mathrm{LayerNorm}(a + \mathrm{GLU}(\eta_1))$$

where:

$$\eta_1 = W_1\eta_2 + b_1, \quad \eta_2 = \mathrm{ELU}(W_2a + W_3c + b_2)$$

GLUs are used within GRNs to provide selective gating:

$$\mathrm{GLU}(\gamma) = \sigma(W_4\gamma + b_4) \odot (W_5\gamma + b_5)$$

where σ is the sigmoid function, $\odot$ denotes element-wise multiplication, and W and b are learnable parameters. To capture long-term temporal dependencies, the TFT employs interpretable multi-head attention. Standard scaled dot-product attention is given by:

$$\mathrm{Attention}(Q,K,V) = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

where Q, K, and V are the query, key, and value matrices, and d is the dimensionality. The interpretable multi-head attention modifies this as:

$$\mathrm{InterpretableAttention}(Q,K,V) = \frac{1}{m_H}\sum_{h=1}^{m_H}\mathrm{Attention}(QW_Q^h, KW_K^h, VW_V),$$

where W_Q^h, W_K^h and W_V are learnable weights, and m_H is the number of attention heads. For locality enhancement, the TFT uses an LSTM-based sequence-to-sequence layer. The output of this layer, with n specifying the position index, is denoted as:

$$\phi(t,n) \in \{\phi(t,-k), \ldots, \phi(t,\tau_{\max})\}$$

The outputs are processed via a gated skip connection:

$$\phi'(t,n) = \mathrm{LayerNorm}(\phi(t,n) + \mathrm{GLU}(\phi(t,n)))$$

To analyze persistent patterns, attention weights are aggregated across time steps and horizons. The contribution of a feature at time n is measured by:

$$\alpha(t,n,\tau) = \frac{1}{T}\sum_{t=1}^{T}\alpha(t,n,\tau)$$

where $\alpha(t,n,\tau)$ is the attention weight for position n at horizon τ. To detect regime shifts, the attention patterns are compared using the Bhattacharyya distance:

$$d(p,q) = \sqrt{1-\rho(p,q)}$$

where:

$$\rho(p,q) = \sum_j \sqrt{p_j q_j}$$

For a given time step t, the distance metric is computed as:

$$\mathrm{dist}(t) = \frac{1}{\tau_{\max}}\sum_{\tau=1}^{\tau_{\max}} d(\alpha(\tau), \alpha(t,\tau))$$

4.5. The pipeline of online forecasting

The pipeline of online forecasting consists of four main layers, each of which transfers data to the next layer: data ingestion, data lake, data transformation, and data visualization. The following is a detailed discussion of each layer, after some initial comments on streaming data.

- **Streaming data resources:** Various streaming data sources, such as wearable sensors, continuously generate data streams containing physiological variables like SpO2 or RR. These data streams are ingested into Kafka topics, where they are processed and aggregated in real time using Apache Flink. Both Kafka and Flink have previously been discussed in detail.
- **Data ingestion layer:** This layer is the aggregation layer, where data is aggregated from wearable sensors using batch processing or real-time processing techniques. The data ingestion layer collects and imports raw physiological data, such as SpO2 and RR, from various sources. In this framework, a simulated sensor generates the time-series data, which is then captured by the Kafka Producer API. The data ingestion layer ensures the seamless intake of continuous data streams, facilitating the initial step of data processing and making the data available for subsequent layers in the system. This layer is critical in ensuring that the data is efficiently and accurately ingested into the system for real-time analysis and forecasting.
- **Data lake layer:** A data lake is a centralized repository that saves significant amounts of raw data in different formats, including structured, semi-structured, and unstructured data. It provides a schema-on-read approach to help data remain flexible for data

ingestion and analysis [54]. The SMHA framework's data lake layer is a centralized repository for holding vast amounts of raw, organized, and unstructured physiological data, including RR and SpO2. The massive volumes of data gathered from the data ingestion layer can be efficiently stored, managed, and retrieved with this layer. The data lake layer's scalable and flexible data architecture supports the subsequent data transformation and visualization (analysis) processes, enabling robust AI-driven forecasting and real-time health monitoring.

- **Data transformation layer:** The raw physiological data kept in the data lake must be processed and transformed into an organized and analyzable format by the data transformation layer of the SMHA system. The transformation layer gets the data ready for AI-driven forecasting by using methods including filtering, aggregation, and standardization. For example, windowing and stream processing are carried out every three minutes using Apache Flink. The TFT model is then provided with the altered data to forecast SpO2 and RR. The transformation layer ensures that the data is synced, cleaned, and prepared for instant analysis and display.
- **Data visualization layer:** The SMHA framework's data visualization layer is in charge of presenting the predicted physiological data, such as expected SpO2 and RR values, in an understandable and user-friendly way. The data visualization layer links to InfluxDB, where the projected data is kept, using tools like Grafana to provide real-time visuals that assist healthcare professionals in properly monitoring and interpreting the data. The data visualization layer makes it possible to get precise and valuable information, which supports proactive healthcare management and well-informed decision making.

In summary, four key considerations underpin the design of the deployment pipeline. First, each component – Kafka for data ingestion, Flink for stream processing, InfluxDB for time series storage, and Grafana for visualization – was purposefully selected for its critical role in supporting real-time ICU monitoring, enabling continuous data flows, ensuring low-latency analytics, and providing an intuitive visualization of vital signs. Second, the system architecture is intentionally modular and scalable, allowing for flexible adaptation or substitution with clinical-grade components in future implementations. Third, the entire deployment operates within a simulated environment using synthetically streamed MIMIC-III data, which provides a realistic and controlled benchmark for assessing the system's responsiveness and stability before clinical integration. Fourth, this design aligns with the operational demands of ICU settings, and serves as a foundational prototype for embedding predictive models into real-time streaming infrastructures within smart healthcare ecosystems.

Importantly, while the infrastructure may appear sophisticated for a proof-of-concept study, it is essential to have all of the components to accurately emulate the streaming dynamics that would be encountered in real-world settings. The pipeline is not presented as a finalized clinical solution, but rather as a practical and extensible framework with clear pathways for integration into hospital systems and IoT-enabled environments. Future work will focus on clinical usability assessments and the system's deployment in real-world intensive care contexts.

5. Results and discussion

In this section, we discuss the results from various models for our two experiments. In Experiment 1 – univariate multi-horizon time-series forecasting – we test the performance of different models (LSTM, GRU, Bi-LSTM, Bi-GRU, CNN, TCN, and TFT) for predicting either RR or SpO2 individually. In Experiment 2 – multivariate multi-horizon time-series forecasting – we then explore the performance of various models (S2S-LSTM, S2S-GRU, S2S-A-BiLSTM, S2S-A-BiGRU, TCN, and TFT) for predicting both RR and SpO2 simultaneously. In each experiment, we noticed a significant improvement using the TFT model when compared with the other models.

5.1. Experimental setup

The dataset contains multivariate time-series data for 20 patients. This dataset is divided into 15 patients for fine-tuning and five patients for testing. The DL models are tested under different scenarios, as shown in Fig. 5. The first setting uses just five testing examples. The data for a single patient is divided into 80% for training and the remaining 20% for testing, using different lags, 3, 7 and 15 min, to forecast at 7, 15 and 25 min respectively. Results are collected to analyze the performance of the testing. In this experiment, each model with initially random weights is trained independently with each patient. These models are trained and tested for (1) univariate multi-horizon time-series forecasting (i.e., predicting RR or SpO2) using DL models such as LSTM, GRU, Bi-LSTM, GR-LSTM, CNN, TCN, and TFT, and (2) multivariate multi-horizon time-series forecasting (i.e., jointly predicting RR and SpO2) using sequence-to-sequence models (i.e., S2S-LSTM, S2S-GRU, S2S-A-BiLSTM, S2S-A-BiGRU), TCN, and TFT.

The second setting cumulatively fine-tunes every model 15 times, one for each patient, as shown in Fig. 5. Random initial weights are used to start with. Then, the model is trained using one patient from the 15. After that, the resulting weights are used as the new weights to further fine-tune the model for the second patient. The process continues in a cascading way until it reaches the 15th patient. The resulting model is then tested with the testing set of five patients.

The hardware configuration of the experimental platform is an Intel i7-6700 CPU, the graphics card is an RTX 4090, the memory is 16 GB, the operating system is Windows 11, and the model is implemented on Python 11 using Keras version 3.8.0, TensorFlow version 2.19.0, and PyTorch version 2.5.1+. The results of the RMSE and MAE metrics are reported because these two metrics are consistent with all reported results. We adopted the following model parameters: Adam as an optimizer, MSE as a loss function, a learning_rate of 0.03, 30 epochs with an early stopping patience value of 70, and a batch size of 20. Other settings are shown in Table 3.

As they are all regression models, they are best evaluated using the commonly used Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\mathrm{obs}} - y_i^{\mathrm{pred}}\right)^2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{\mathrm{obs}} - y_i^{\mathrm{pred}}\right|$$

5.2. Experiment 1: Results from the univariate multi-horizon time-series forecasting models

We carried out different experiments using various numbers of forecasting minutes (as described earlier) to arrive at the conclusion that the TFT transformer model achieved the best performance when compared to other models.

5.2.1. Results for forecasting RR

The analysis of the multi-horizon time-series forecasting results for five patients reveals a clear dominance of the TFT model across all forecasting horizons (7, 15 and 25 min), as seen in Tables 4 and 5. Moreover, these tables show the time complexity of each model. The TFT achieved the best RMSE and MAE values in every scenario, underscoring its superior ability to model temporal dependencies and dynamic feature relationships. However, although the TFT achieves the best error results, it also has the highest time complexity. The self-attention mechanism in the transformer enables it to focus on

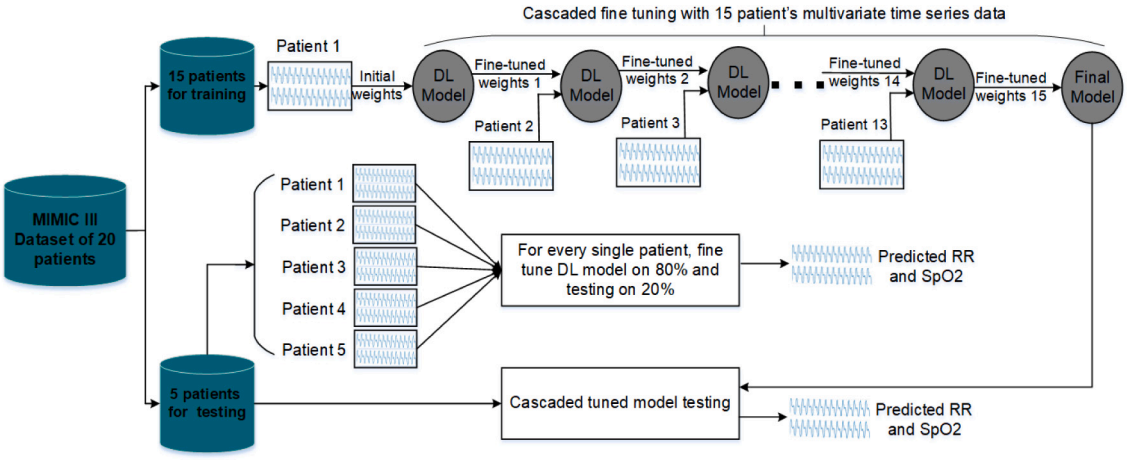

Fig. 5. Cascaded fine-tuning in the DL model using multi-patient time-series data.

Table 3
Setting of parameters.

Models	Parameters	Specifications
LSTM	Number of nodes	160
	Dropout	0.3
	Activation function	Relu
GRU	Number of nodes	160
	Dropout	0.3
	Activation function	Relu
Bi-LSTM	Number of nodes	160
	Dropout	0.3
	Activation function	Relu
Bi-GRU	Number of nodes	160
	Dropout	0.3
	Activation function	Relu
CNN	Filter size	250
	Kernel size	4
	Dropout	0.2
	Number of nodes	150
S2S-LSTM, S2S-GRU	Number of nodes in Encoder layer	200
	Dropout in Encoder layer	0.2
	Activation function in Encoder layer	Relu
	Number of nodes in Decoder layer	200
	Dropout in Decoder layer	0.2
S2S-A-BiLSTM, S2S-A-BiGRU	Number of nodes in Encoder layer	200
	Activation function in Encoder layer	Relu
	Number of nodes in Decoder layer	200
	Optimizer	Adam
TFT	hidden_size	30
	attention_head_size	6
	Dropout	0.5
	hidden_continuous_size	8
	Loss function	QuantileLoss()
TCN	Dropout	0.2
	kernel_size	5
	Optimizer	Adam

relevant segments of the time series, dynamically adapting to the temporal dependencies and feature relationships. This ability is particularly advantageous in handling the inherent complexities and variances of medical time-series data, as considered in this study. The following is a summary of the results with more details on numerical performance.

For Patient 1, with the 7 min setting, the TFT model with an RMSE of 1.8154 outperformed the Bi-GRU model (RMSE of 1.9536) and the CNN model (RMSE of 2.0633). The self-attention mechanism enabled precision for short-term forecasts. As can be seen, TCN achieves comparable results with an RMSE of 1.7879 and a MAE of 1.2379, but with lower time complexity than the TFT transformer model. For the 15 min setting, the TFT model with an RMSE of 1.9939 was better than both the Bi-GRU (RMSE of 2.0073) and CNN (RMSE of 2.1119) models, and again for the 25 min scenario, the TFT model (RMSE of 2.0284) remained the best, while the CNN model (RMSE of 2.2825) was the worst. Again, with an RMSE of 2.0820 and an MAE of 1.5542, TCN achieved the second-best results compared with the TFT, and the same result pattern was achieved for the forecasting at 25 min.

For Patient 2 and the 7 min setting, the TFT model had an RMSE of 4.3373, which was much better than either the Bi-GRU (RMSE of 5.4816) or CNN (RMSE of 6.1259) models. TCN achieved the second-best results (RMSE of 5.0356 and MAE of 3.9802) with the lowest time complexity. For 15 min, the TFT model (RMSE of 5.2232) outperformed TCN (RMSE of 5.4161), the Bi-LSTM model (RMSE of 5.7750), and the

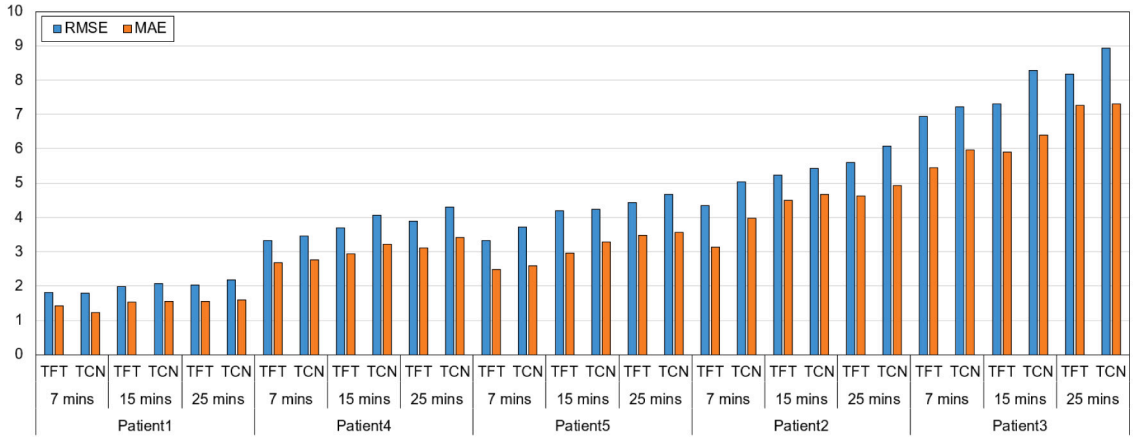

Fig. 6. The effect of problem complexity on the results of the TFT transformer model for RR forecasting.

CNN model (RMSE of 6.4177), and for 25 min, the TFT model (RMSE of 5.5968) excelled, while the CNN mode (RMSE of 6.9583) performed poorly.

The results of Patient 3 confirmed the superior results of the TFT transformer model, with TCN as the second-best model. TCN model achieves the lowest time complexity. For the 7 min experiment, the TFT transformer model (RMSE of 6.9325) surpassed the TCN (RMSE of 7.22786), Bi-LSTM (RMSE of 8.2304) and CNN (RMSE of 8.5268) models, and for 15 min, the TFT transformer (RMSE of 7.3032) surpassed TCN (RMSE of 8.2751), Bi-GRU (RMSE of 8.8400) and CNN (RMSE of 8.8670). Again, TCN achieves the lowest time complexity compared to other models. For 25 min, the transformer (RMSE of 8.1807) retained superiority, while CNN, with an RMSE of 9.9005, faltered. TCN achieved the second-best performance with the lowest time complexity.

Moreover, the TFT transformer still achieved the best results for Patient 4. For the 7 min experiment, TFT, with an RMSE of 3.3220, beat TCN (RMSE of 3.4679), Bi-GRU (RMSE of 3.5601) and CNN (RMSE of 3.6568). For the 15 min setting, TFT with an RMSE of 3.6965 led, then TCN with an RMSE of 4.0687 came next, followed by CNN with an RMSE of 4.1738. For 25 min, the TFT transformer model with an RMSE of 3.8850 surpassed TCN (RMSE of 4.3034), Bi-LSTM (RMSE of 4.1497) and CNN (RMSE of 4.4112). TCN achieved the lowest time complexity.

Finally, for Patient 5, the TFT model achieved the best RMSE and MAE for all settings. TCN had the second-best performance and the lowest time complexity. For the 7 min setting, the transformer with an RMSE of 3.3319 was better than TCN (RMSE 3.7194), Bi-GRU (RMSE of 3.9062) and CNN (RMSE of 3.9994). For the 15 min setting, the TFT model (RMSE of 4.1829) outperformed TCN (RMSE 4.2267), Bi-LSTM (RMSE of 4.2784) and CNN (RMSE of 4.6441). For the 25 min setting, the transformer model had an RMSE of 4.4274, while the TCN model (RMSE 4.6628) and the CNN model (RMSE of 5.3552) performed worse.

Fig. 6 compares TFT and the next-best model (TCN) for each forecast setting and for every patient. In summary, TFT achieved the best results but with high time complexity, and TCN achieved the second-best results but with the lowest time complexity. The TFT and TCN models specialize in time-series data analysis. As a result, they achieved the best results when compared with the other models, such as the CNN-, LSTM-, and GRU-based models.

As we have seen, the TFT transformer model consistently performed the best for multi-horizon time-series forecasting, leveraging its self-attention mechanism to dynamically capture both short-term and long-term dependencies in time-series data, with details in Tables 4 and 5. Its robust performance across all patients and forecasting horizons highlights its adaptability, accuracy, and efficiency. Even as task complexity increases, its ability to maintain low RMSE and MAE values establishes the TFT as the optimal choice for real-time healthcare applications, as shown in Fig. 7. Bidirectional models such as the Bi-GRU and Bi-LSTM were often ranked third, benefiting from their ability to process forward and backward dependencies. However, their sequential processing capabilities limit their scalability and precision when compared to the parallelized operations of the TFT.

In contrast, CNNs were consistently ranked as the worst-performing models across most forecasting horizons and patients. CNNs rely on convolutional operations, which are less effective in capturing sequential and temporal dependencies than recurrent and attention-based models. Their inability to adapt to long-term dependencies, essential for accurate multi-horizon predictions, likely contributed to their inferior performance. This limitation becomes more pronounced as the forecasting horizon increases, as seen in the significant degradation of CNN performance for longer forecasts (e.g., 25 min). The results of this experiment show a superior and stable performance of the TFT transformer model when compared with other classical models. However, all models struggled as the problem became more complex (i.e., when forecasting for increasing numbers of minutes). Fig. 8 shows the average performance for the different forecasting sizes used, highlighting a consistent increase in the error rate as the number of forecasting minutes increased.

5.2.2. *Results for forecasting SpO2*

An analysis of the SpO2 forecasting results highlights distinct patterns in the performance of the different models across the five patients and three forecasting horizons (7, 15 and 25 min), with details in Tables 6 and 7. These tables also show the time complexity of the different models. This section provides a detailed discussion of these results, focusing on RMSE as the primary evaluation metric. As shown in Fig. 9, the TFT transformer model consistently achieved the best performance in terms of minimized errors across all forecasting horizons and patients, demonstrating its robust capability in accurately modeling complex temporal dependencies. In addition, TCN achieved the second-best result for all patients. However, the TFT model had the highest time complexity in all experiments, while the TCN model had the best (shortest) times. As a result, different models dominated in terms of the various evaluation metrics.

For example, in Patient 1, the TFT achieved RMSE values of 1.5518, 1.6198, and 1.7545 for 7-, 15- and 25 min forecasts, respectively. These values are significantly lower than those of the other models, such as CNN, which recorded RMSEs of 2.2749, 2.6169, and 2.7346, and TCN, which recorded RMSEs of 1.6633, 1.7138, and 1.9019 for the same horizons. This trend underscores the TFT's ability to effectively

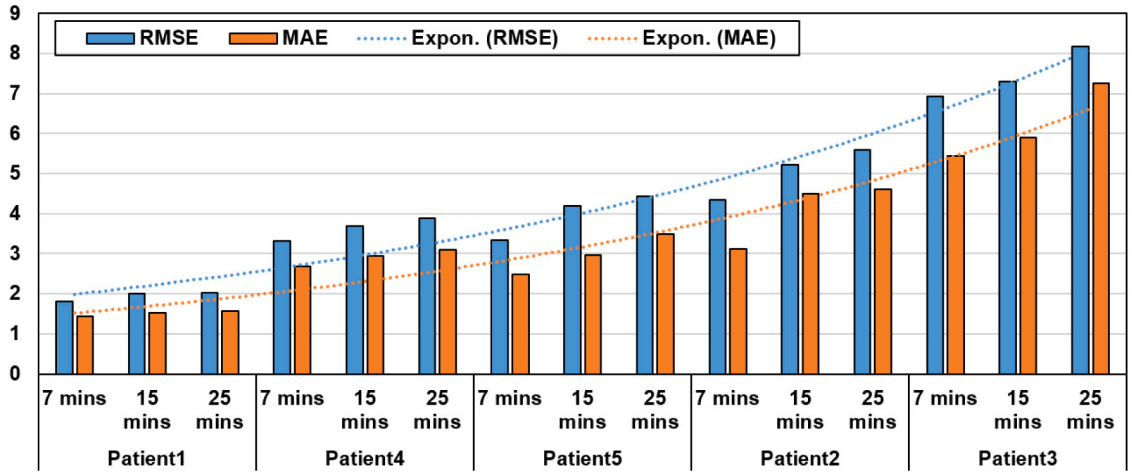

Fig. 7. Effect of the problem complexity on the results of the TFT transformer model for RR predictions.

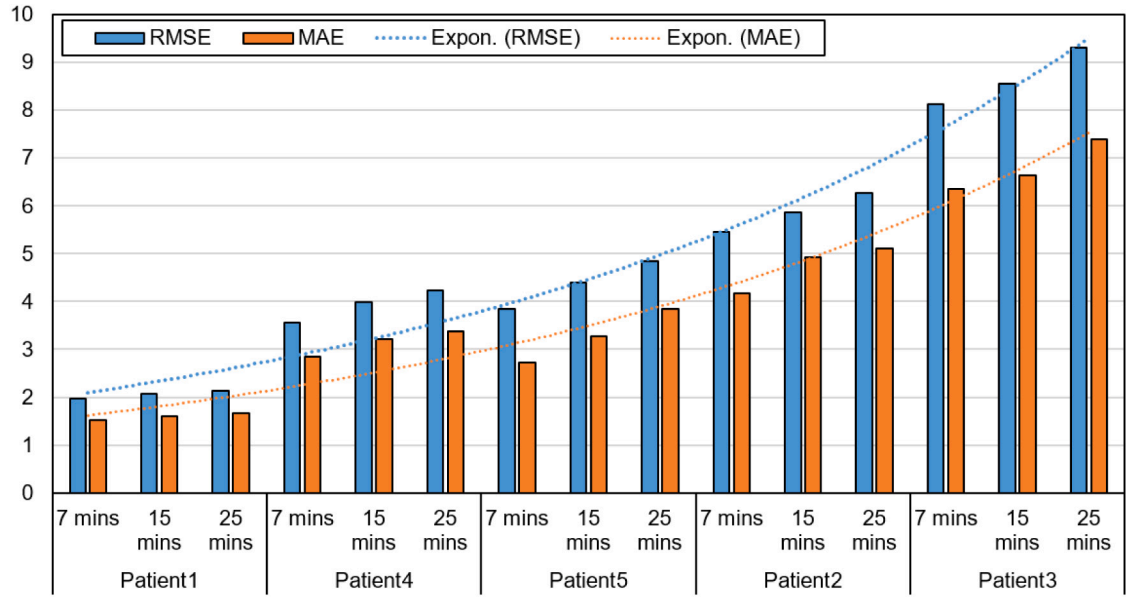

Fig. 8. Average results of different numbers of forecasting minutes for RR forecasting.

handle multi-horizon forecasting tasks by leveraging its attention mechanism to dynamically focus on the most relevant features. Conversely, CNN consistently underperformed relative to the other models, often recording the highest RMSE values. For instance, for Patient 2, CNN had RMSE values of 2.1199, 2.2582, and 2.7019 for the 7-, 15- and 25 min horizons, in contrast to TFT's significantly lower RMSEs of 1.8758, 1.9824, and 2.1654. The inability of CNNs to capture long-range dependencies and its reliance on localized convolutions likely contributed to this inferior performance.

In the 7 min forecasting scenario, the TFT displayed very strong results across all patients, as can be seen in Tables 6 and 7. For Patient 3, the TFT achieved an RMSE of 0.4501, beating TCN (0.6728), GRU (0.6767) and Bi-LSTM (0.7158). This superior performance is attributed to the TFT's advanced temporal dynamics modeling, critical for handling short-horizon forecasts with relatively high variability. The TCN model often emerged as the second-best performer in short-horizon forecasts. For Patient 1, TCN recorded an RMSE of 1.6633, closely trailing behind the TFT but significantly beating CNN (RMSE of 2.2749). Bi-GRU, on the other hand, achieved lower results than TFT and TCN. This demonstrated that while Bi-GRU effectively captures bidirectional dependencies, its sequential nature limits its efficiency compared to the parallel processing capabilities of the TFT. The TFT maintained its lead for the mid-horizon forecasts (15 min). For Patient 4, the TFT achieved an RMSE of 2.3708, defeating TCN (RMSE of 2.4228), Bi-LSTM (RMSE of 2.4903), and CNN (RMSE of 2.6477). The ability of the TFT to dynamically weigh features and adjust to shifting temporal patterns proved advantageous as the forecasting horizon extended.

The GRU model consistently performed better than CNN but lagged behind the TFT, TCN, and Bi-GRU. For instance, in Patient 5, GRU recorded an RMSE of 2.5806, compared to TFT's value of 2.2973 and TCN of 2.50891. This pattern suggests that GRU's sequential architecture struggles to compete with the advanced attention mechanisms of the TFT and TCN in capturing complex temporal relationships. As the forecasting horizon increased (i.e., to 25 min), the task complexity intensified, leading to higher RMSE values across all models. Fig. 10 shows the consistent increase in error for the TFT as the problem complexity increases. Despite this, the TFT demonstrated resilience and continued to do better than other models. For Patient 5, the TFT achieved an RMSE of 2.8280, significantly better than TCN (RMSE of

Table 4
Results of the univariate multi-horizon forecasting for RR.

Patients	Forecasting minutes	Models	RMSE	MAE	Time
Patient 1	Forecasting 7 minutes	LSTM	1.9898	1.5370	0 min 40 s
		GRU	1.9814	1.5297	0 min 41 s
		Bi-LSTM	1.9645	1.4966	0 min 40 s
		Bi-GRU	1.9536	1.4302	0 min 42 s
		CNN	2.0633	1.6659	0 min 40 s
		TCN	1.7879	1.2379	0 min 20 s
		TFT	**1.8154**	**1.4346**	**0 min 55 s**
	Forecasting 15 minutes	LSTM	2.1434	1.6521	0 min 50 s
		GRU	2.1893	1.7138	0 min 51 s
		Bi-LSTM	2.0349	1.5642	0 min 50 s
		Bi-GRU	2.0073	1.5345	0 min 51 s
		CNN	2.1119	1.6430	0 min 50 s
		TCN	2.0820	1.5542	0 min 35 s
		TFT	**1.9939**	**1.5710**	**1 min 10 s**
	Forecasting 25 minutes	LSTM	2.1269	1.6484	1 min 5 s
		GRU	2.2078	1.7120	1 min 7 s
		Bi-LSTM	2.0962	1.6255	1 min 5 s
		Bi-GRU	2.0929	1.6214	1 min 7 s
		CNN	2.2825	1.7763	1 min 6 s
		TCN	2.1804	1.6045	0 min 40 s
		TFT	**2.0284**	**1.5608**	**1 min 20 s**
Patient 2	Forecasting 7 minutes	LSTM	5.7495	4.5838	0 min 20 s
		GRU	5.5201	4.1846	0 min 20 s
		Bi-LSTM	5.4998	4.0006	0 min 25 s
		Bi-GRU	5.4816	4.0986	0 min 25 s
		CNN	6.1259	5.0511	0 min 20 s
		TCN	5.0356	3.9802	0 min 10 s
		TFT	**4.3373**	**3.1233**	**0 min 40 s**
	Forecasting 15 minutes	LSTM	6.1543	5.2518	0 min 40 s
		GRU	5.7767	4.7303	0 min 41 s
		Bi-LSTM	5.7750	4.7720	0 min 45 s
		Bi-GRU	5.8036	4.8202	0 min 40 s
		CNN	6.4177	5.5469	0 min 43 s
		TCN	5.4161	4.6783	0 min 20 s
		TFT	**5.2232**	**4.4946**	**0 min 55 s**
	Forecasting 25 minutes	LSTM	6.6448	4.8936	0 min 50 s
		GRU	6.2711	4.8572	0 min 51 s
		Bi-LSTM	6.3372	5.4823	0 min 50 s
		Bi-GRU	5.8756	4.8606	0 min 54 s
		CNN	6.9583	6.0138	0 min 50 s
		TCN	6.0741	4.9245	0 min 30 s
		TFT	**5.5968**	**4.6163**	**1 min 10 s**
Patient 3	Forecasting 7 minutes	LSTM	8.2366	6.3994	0 min 15 s
		GRU	8.6013	6.7332	0 min 15 s
		Bi-LSTM	8.2304	6.4088	0 min 18 s
		Bi-GRU	8.2777	6.4201	0 min 18 s
		CNN	8.5268	6.7213	0 min 16 s
		TCN	7.22786	5.9716	0 min 10 s
		TFT	**6.9325**	**5.4401**	**0 min 30 s**
	Forecasting 15 minutes	LSTM	8.5997	6.6233	0 min 25 s
		GRU	9.0176	6.9917	0 min 25 s
		Bi-LSTM	8.7681	6.7618	0 min 30 s
		Bi-GRU	8.8400	6.8151	0 min 30 s
		CNN	8.8670	6.8183	0 min 25 s
		TCN	8.2751	6.3957	0 min 15 s
		TFT	**7.3032**	**5.8912**	**0 min 42 s**
	Forecasting 25 minutes	LSTM	9.3403	7.3078	0 min 40 s
		GRU	9.5332	7.3956	0 min 43 s
		Bi-LSTM	9.4134	7.3551	0 min 40 s
		Bi-GRU	9.5735	7.4434	0 min 45 s
		CNN	9.9005	7.5853	0 min 40 s
		TCN	8.9264	7.3067	0 min 25 s
		TFT	**8.1807**	**7.2601**	**0 min 57 s**

3.1067), CNN (RMSE of 3.2497), and GRU (RMSE of 3.3277). This gap in results highlights the TFT's robustness in maintaining accuracy for long-term forecasts, with more details in Tables 6 and 7. The CNN model consistently ranked as the worst performer for long-horizon forecasts, as seen with Patient 2 where it recorded an RMSE of 2.7019 compared to TFT's 2.1654. This underperformance is linked to CNN's inability to effectively model temporal dependencies over extended time horizons.

As a result of these experiments, we can see that the TFT achieved the best RMSE values, and TCN was the second-best model, showcasing their strength in handling temporal complexities through attention mechanisms. Their performance advantage was most pronounced for longer horizons, where traditional models such as LSTM and GRU

Table 5
Continued results of the univariate multi-horizon forecasting for RR.

Patients	Forecasting minutes	Models	RMSE	MAE	Time
Patient 4	Forecasting 7 minutes	LSTM	3.5945	2.8722	0 min 10 s
		GRU	3.5647	2.8519	0 min 23 s
		Bi-LSTM	3.6106	2.8591	0 min 20 s
		Bi-GRU	3.5601	2.8467	0 min 21 s
		CNN	3.6568	2.9420	0 min 23 s
		TCN	3.4679	2.7571	0 min 13 s
		TFT	**3.3220**	**2.6841**	**0 min 40 s**
	Forecasting 15 minutes	LSTM	4.0077	3.2274	0 min 35 s
		GRU	4.0415	3.2809	0 min 33 s
		Bi-LSTM	4.0594	3.2773	0 min 35 s
		Bi-GRU	3.9301	3.2156	0 min 35 s
		CNN	4.1738	3.3709	0 min 36 s
		TCN	4.0687	3.2264	0 min 20 s
		TFT	**3.6965**	**2.9463**	**0 min 50 s**
	Forecasting 25 minutes	LSTM	4.3382	3.4085	0 min 50 s
		GRU	4.3841	3.4438	0 min 52 s
		Bi-LSTM	4.1497	3.2845	0 min 50 s
		Bi-GRU	4.2844	3.3493	0 min 53 s
		CNN	4.4112	3.6803	0 min 50 s
		TCN	4.3034	3.4081	0 min 35 s
		TFT	**3.8850**	**3.1029**	**1 min 5 s**
Patient 5	Forecasting 7 minutes	LSTM	3.9755	2.8572	0 min 25 s
		GRU	3.9659	2.8829	0 min 24 s
		Bi-LSTM	3.9228	2.6979	0 min 25 s
		Bi-GRU	3.9062	2.6252	0 min 27 s
		CNN	3.9994	2.8104	0 min 25 s
		TCN	3.7194	2.5895	0 min 15 s
		TFT	**3.3319**	**2.4883**	**0 min 35 s**
	Forecasting 15 minutes	LSTM	4.1753	2.9526	0 min 42 s
		GRU	4.5830	3.5630	0 min 41 s
		Bi-LSTM	4.2784	3.1955	0 min 40 s
		Bi-GRU	4.4755	3.4204	0 min 43 s
		CNN	4.6441	3.5513	0 min 42 s
		TCN	4.2267	3.2739	0 min 25 s
		TFT	**4.1829**	**2.9645**	**0 min 55 s**
	Forecasting 25 minutes	LSTM	4.7255	3.6161	0 min 55 s
		GRU	4.7643	3.8436	0 min 50 s
		Bi-LSTM	4.9156	3.8888	0 min 55 s
		Bi-GRU	4.8712	3.6105	0 min 53 s
		CNN	5.3552	4.6446	0 min 55 s
		TCN	4.6628	3.5588	0 min 35 s
		TFT	**4.4274**	**3.4854**	**1 min 10 s**

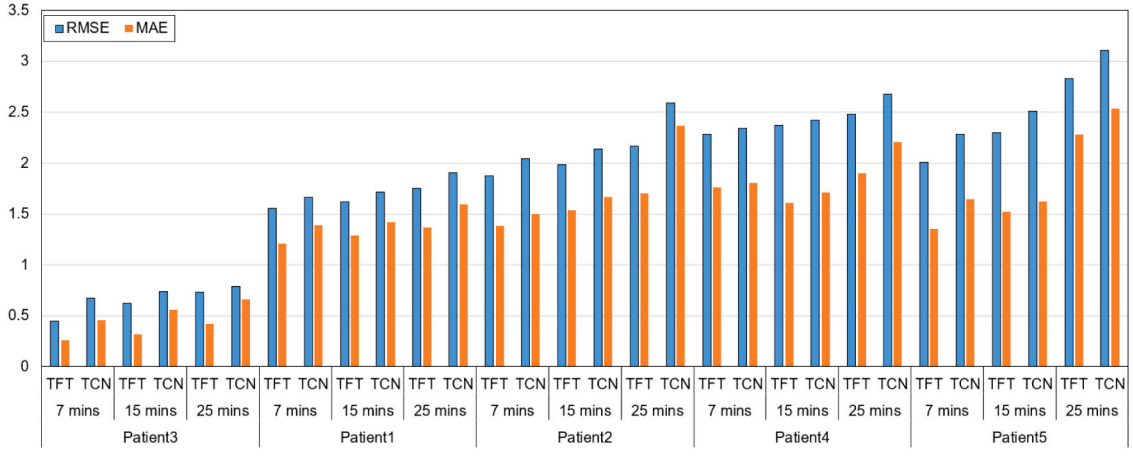

Fig. 9. Performance of the TFT transformer model for SpO2 prediction with different patients.

struggled. Regarding model scalability, bidirectional models such as Bi-GRU and Bi-LSTM performed well, particularly for short- and mid-term horizons. However, their sequential nature rendered them less effective than the TFT and TCN for long-term predictions. CNN's reliance on localized convolutions resulted in poor performance across all horizons, especially for longer forecasts where capturing global temporal relationships is crucial. The TFT is reliable for SpO2 forecasting, with

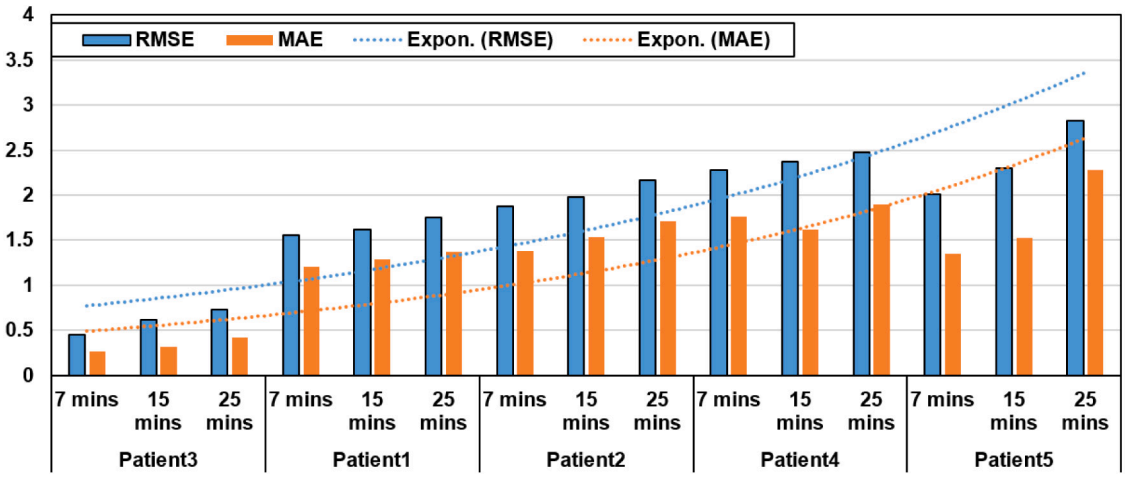

Fig. 10. Effect of the problem complexity on the results of the TFT transformer model for SpO2 predictions.

TCN being the second best model, and CNN consistently had the lowest RMSE values across all horizons and patient datasets. Temporally advanced architectures like TFT and TCN that integrate attention mechanisms and can handle multi-horizon tasks efficiently are the superior choice for time-series forecasting in medical applications. Conversely, CNN demonstrated significant limitations, reinforcing the importance of selecting models tailored to the temporal dynamics of the data. These findings validate the potential of the TFT and the TCN in improving real-time healthcare predictions, providing critical insights for timely interventions.

5.3. Experiment 2: Results from the multivariate multi-horizon time-series forecasting models

As shown in Fig. 11, the results for jointly predicting both RR and SpO2 as multivariate time series over multiple forecasting horizons (7, 15 and 25 min) underscore the robust performance of the TFT model, with more details in Tables 8 and 9. However, the TFT also had the highest time complexity. On the other hand, TCN had the second-best performance compared to TFT, but it had the lowest time complexity. For the 7 min forecasting period, the TFT model consistently outperformed all other models in the short-term forecasting horizon, as seen in Fig. 12. For Patient 1, the TFT achieved RMSE values of 1.4780 for RR and 1.5776 for SpO2, significantly lower than the second-best model, TCN, which recorded RMSEs of 2.2198 and 2.0335 for RR and SpO2, respectively, and the third-best model, S2S-A-BiGRU, which recorded RMSEs of 2.4069 for RR and 2.2165 for SpO2. The TFT's performance is attributed to its attention mechanism, which allows it to focus on the most relevant features in the data, capturing short-term temporal dynamics effectively. Conversely, the S2S-LSTM performed the worst in this category for most patients. For Patient 3, it recorded RMSEs of 8.2981 for RR and 0.6559 for SpO2, indicating its struggle to adapt to complex short-term patterns, particularly in the RR series. These results highlight traditional sequence-to-sequence architectures' limitations in handling the task's multivariate nature within a short horizon. For 15 min forecasting, the TFT model maintained its dominance across all patients. Patient 2 achieved RMSEs of 3.9396 for RR and 1.2271 for SpO2, beating S2S-A-BiGRU, which recorded RMSEs of 5.2306 for RR and 1.8606 for SpO2. It also did better than TCN, which achieved an RMSE of 4.4838 for RR and 1.6242 for SpO2.

For 15 min, the gap between TFT and the other models grew narrower when compared to the 7 min forecast, particularly for SpO2, suggesting that different models, such as S2S-A-BiGRU and S2S-GRU, began to capture more of the mid-term temporal dynamics, though not to the extent of TFT. Although not the worst, the performance of the CNN-based models was consistently suboptimal. For Patient 4, the CNN recorded RMSEs of 3.5100 for RR and 2.5221 for SpO2, falling short of capturing the nuanced temporal relationships required for accurate forecasting, particularly in the multivariate setting. As the forecasting horizon extended to 25 min, the task complexity increased, leading to higher RMSE values across all models. Despite this, the TFT demonstrated remarkable robustness, continuing to deliver superior performance. For Patient 5, the TFT achieved RMSEs of 3.0673 for RR and 1.6917 for SpO2, surpassing TCN, which recorded RMSEs of 3.4840 for RR and 1.8928 for SpO2. This resilience underscores TFT's ability to manage long-term dependencies effectively. On the other hand, traditional sequence-to-sequence models such as S2S-LSTM and S2S-GRU struggled significantly as the horizon lengthened. For Patient 1, S2S-LSTM recorded RMSEs of 6.7062 for RR and 9.8074 for SpO2, indicating its inability to maintain accuracy over extended periods. These results emphasize the limitations of recurrent architectures without attention mechanisms in capturing long-term temporal relationships.

As can be seen in Tables 8 and 9, across all horizons, the TFT consistently delivered the lowest RMSE values, reflecting its ability to adjust to both short- and long-term dependencies dynamically. Incorporating attention mechanisms enables the TFT to weigh features effectively, a critical advantage in multivariate tasks involving RR and SpO2. TCN was consistently ranked the second-best model for all patients and all time horizons. The bidirectional GRU models, particularly S2S-A-BiGRU, were consistently ranked as the third-best performers. Their ability to capture bidirectional temporal dependencies made them competitive, especially in the 7- and 15 min horizons. However, their sequential nature limited their scalability for longer horizons. The CNN models struggled to capture temporal dependencies effectively, particularly in the RR series. This limitation was most pronounced in longer horizons, where their performance lagged significantly behind attention-based models like the TFT. The increase in forecasting horizon times highlighted scalability issues in recurrent models such as S2S-LSTM and S2S-GRU, which exhibited substantial performance degradation as the task complexity grew. This trend reinforces the importance of models designed to handle both temporal depth and breadth, as demonstrated by the TFT. The increased complexity affected the TFT model, as can be seen in Fig. 11.

As a general conclusion, the TFT is the most accurate model for multivariate multi-horizon forecasting of RR and SpO2 values, achieving the best RMSE values across all patients and horizons, with full details in Tables 8 and 9. Its ability to dynamically focus on relevant features and adapt to temporal complexities makes it the superior choice for real-time healthcare applications. While models like S2S-A-BiGRU offered competitive performance for shorter horizons, their limitations

Table 6
Results of the univariate multi-horizon forecasting for SpO2.

Patients	Forecasting minutes	Models	RMSE	MAE	Time
Patient 1	Forecasting 7 minutes	LSTM	2.1791	1.8851	0 min 55 s
		GRU	2.1501	1.7700	0 min 54 s
		Bi-LSTM	1.7757	1.4873	0 min 55 s
		Bi-GRU	1.7584	1.4772	0 min 55 s
		CNN	2.2749	1.9064	0 min 54 s
		TCN	1.6633	1.3879	0 min 30 s
		TFT	**1.5518**	**1.2087**	**1 min 5 s**
	Forecasting 15 minutes	LSTM	2.2937	1.8840	1 min 10 s
		GRU	2.3240	1.9351	1 min 15 s
		Bi-LSTM	1.8651	1.4909	1 min 10 s
		Bi-GRU	1.7696	1.4122	1 min 12 s
		CNN	2.6169	2.2509	1 min 11 s
		TCN	1.7138	1.4216	0 min 45 s
		TFT	**1.6198**	**1.2923**	**1 min 34 s**
	Forecasting 25 minutes	LSTM	2.1166	1.7386	1 min 35 s
		GRU	2.5748	2.2857	1 min 37 s
		Bi-LSTM	2.0231	1.6521	1 min 35 s
		Bi-GRU	1.7925	1.4229	1 min 36 s
		CNN	2.7346	2.3324	1 min 35 s
		TCN	1.9019	1.5979	0 min 53 s
		TFT	**1.7545**	**1.3667**	**1 min 45 s**
Patient 2	Forecasting 7 minutes	LSTM	2.0409	1.5355	0 min 35 s
		GRU	2.1513	1.6053	0 min 37 s
		Bi-LSTM	2.1772	1.6384	0 min 35 s
		Bi-GRU	2.1651	1.6185	0 min 38 s
		CNN	2.1199	1.5996	0 min 40 s
		TCN	2.0453	1.5009	0 min 20 s
		TFT	**1.8758**	**1.3807**	**0 min 45 s**
	Forecasting 15 minutes	LSTM	2.2311	1.7748	0 min 50 s
		GRU	2.2146	1.7501	0 min 51 s
		Bi-LSTM	2.2328	1.7994	0 min 50 s
		Bi-GRU	2.2378	1.7835	0 min 54 s
		CNN	2.2582	1.7974	0 min 50 s
		TCN	2.1358	1.6662	0 min 30 s
		TFT	**1.9824**	**1.5393**	**1 min 7 s**
	Forecasting 25 minutes	LSTM	2.4042	2.0097	1 min 5 s
		GRU	2.6754	2.2785	1 min 7 s
		Bi-LSTM	2.5829	2.3222	1 min 10 s
		Bi-GRU	2.4980	2.2008	1 min 5 s
		CNN	2.7019	2.4391	1 min 7 s
		TCN	2.5925	2.3655	0 min 45 s
		TFT	**2.1654**	**1.7071**	**1 min 25 s**
Patient 3	Forecasting 7 minutes	LSTM	0.6921	0.5268	0 min 25 s
		GRU	0.6767	0.4649	0 min 28 s
		Bi-LSTM	0.7158	0.5858	0 min 25 s
		Bi-GRU	0.6830	0.4591	0 min 24 s
		CNN	0.7321	0.6174	0 min 20 s
		TCN	0.6728	0.4554	0 min 10 s
		TFT	**0.4501**	**0.2625**	**0 min 40 s**
	Forecasting 15 minutes	LSTM	0.8150	0.7155	0 min 45 s
		GRU	0.7288	0.5739	0 min 45 s
		Bi-LSTM	0.7489	0.6206	0 min 40 s
		Bi-GRU	0.7369	0.5815	0 min 40 s
		CNN	0.7190	0.5572	0 min 45 s
		TCN	0.7352	0.5632	0 min 25 s
		TFT	**0.6210**	**0.3169**	**1 min 5 s**
	Forecasting 25 minutes	LSTM	0.7687	0.6504	0 min 55 s
		GRU	0.7708	0.6434	0 min 58 s
		Bi-LSTM	0.7733	0.6451	0 min 55 s
		Bi-GRU	0.7935	0.6619	0 min 57 s
		CNN	0.8255	0.7216	0 min 54 s
		TCN	0.7932	0.6636	0 min 35 s
		TFT	**0.7285**	**0.4195**	**1 min 15 s**

became evident for longer-term predictions. Conversely, CNN and traditional LSTM-based architectures failed to manage this task's intricate temporal and multivariate relationships effectively. These findings validate the TFT as a pivotal tool for advancing predictive healthcare analytics.

Table 10 shows the standard deviations of different models for different forecasting tasks. TFT has the lowest standard deviation values for all forecasting tasks. As a result, it can be noticed that TFT achieves the most stable results compared to other classical S2S or S2S-A models. This leads to the observation that TFT is more generalizable when compared to the other different models. As a result, in the next experiment, we focus on fine-tuning this model and testing it with full time-series datasets.

Table 7
Continued results of the univariate multi-horizon forecasting for SpO2.

Patients	Forecasting minutes	Models	RMSE	MAE	Time
Patient 4	Forecasting 7 minutes	LSTM	2.4826	1.7990	0 min 28 s
		GRU	2.4271	1.8025	0 min 27 s
		Bi-LSTM	2.4176	1.7748	0 min 25 s
		Bi-GRU	2.4504	1.8482	0 min 25 s
		CNN	2.4878	1.9360	0 min 26 s
		TCN	2.3428	1.8048	0 min 15 s
		TFT	**2.2823**	**1.7631**	**0 min 40 s**
	Forecasting 15 minutes	LSTM	2.5521	1.6456	0 min 43 s
		GRU	2.5477	1.6622	0 min 42 s
		Bi-LSTM	2.4903	1.7427	0 min 45 s
		Bi-GRU	2.5754	1.7004	0 min 45 s
		CNN	2.6477	1.8987	0 min 36 s
		TCN	2.4228	1.7093	0 min 25 s
		TFT	**2.3708**	**1.6129**	**0 min 57 s**
	Forecasting 25 minutes	LSTM	2.8485	2.3616	0 min 55 s
		GRU	2.8310	2.3542	0 min 55 s
		Bi-LSTM	2.7147	2.2476	0 min 54 s
		Bi-GRU	2.7701	2.2839	0 min 55 s
		CNN	2.8518	2.3657	0 min 58 s
		TCN	2.6778	2.2074	0 min 30 s
		TFT	**2.4781**	**1.9014**	**1 min 8 s**
Patient 5	Forecasting 7 minutes	LSTM	2.2331	1.5531	0 min 30 s
		GRU	2.2113	1.5792	0 min 32 s
		Bi-LSTM	2.2877	1.6502	0 min 30 s
		Bi-GRU	2.2583	1.5672	0 min 33 s
		CNN	2.2461	1.6488	0 min 30 s
		TCN	2.2866	1.6448	0 min 20 s
		TFT	**2.0088**	**1.3526**	**0 min 45 s**
	Forecasting 15 minutes	LSTM	2.6495	1.7065	0 min 45 s
		GRU	2.5806	1.6878	0 min 44 s
		Bi-LSTM	2.5532	1.6561	0 min 45 s
		Bi-GRU	2.5336	1.6715	0 min 44 s
		CNN	2.6495	1.7065	0 min 45 s
		TCN	2.50891	1.6211	0 min 28 s
		TFT	**2.2973**	**1.5230**	**0 min 57 s**
	Forecasting 25 minutes	LSTM	3.1822	2.5404	0 min 60 s
		GRU	3.3277	2.7279	0 min 58 s
		Bi-LSTM	3.3319	2.7883	0 min 60 s
		Bi-GRU	3.2791	2.7278	0 min 57 s
		CNN	3.2497	2.6720	0 min 60 s
		TCN	3.1067	2.5387	0 min 35 s
		TFT	**2.8280**	**2.2784**	**1 min 5 s**

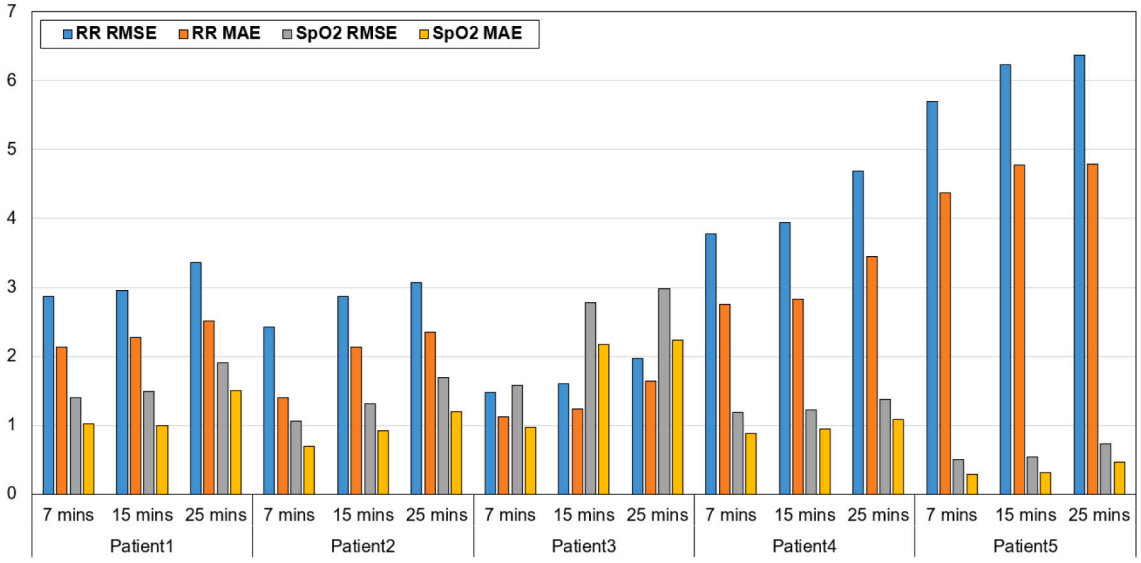

Fig. 11. Results of the TFT model for the multivariate multi-horizon task.

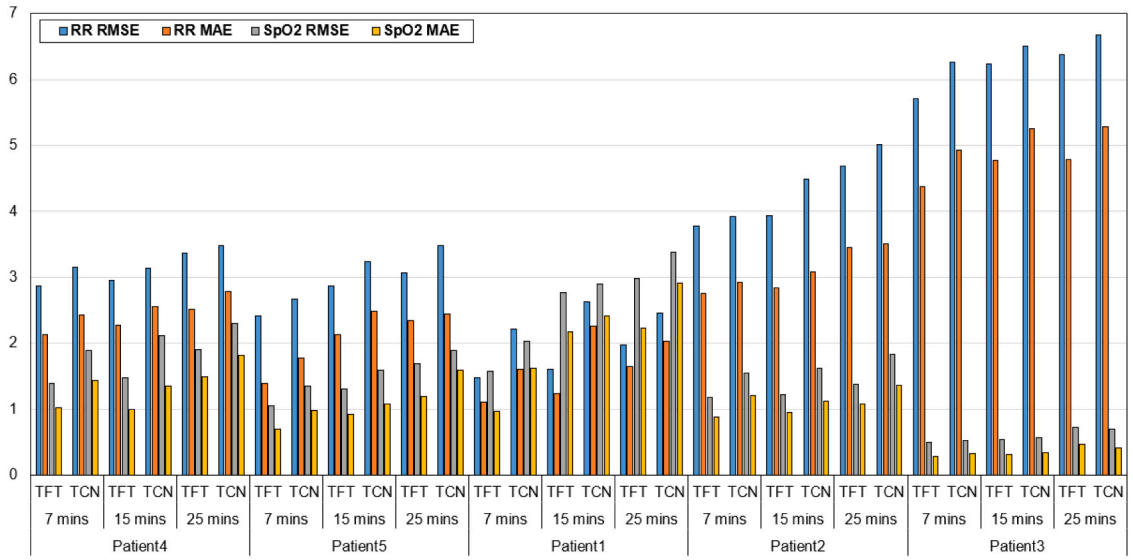

Fig. 12. Comparison of the TFT and TCN models for multivariate multi-horizon predictions across different patients.

To illustrate the quality of the predicted values of SpO2 and RR versus real values in the multivariate multi-horizon task, which is more complex than for the univariate single task models, Fig. 13 presents a comparative visualization of the predicted values generated by the TFT and TCN models against the actual ground truth values for a 7 min multi-horizon forecasting window. This figure clearly illustrates that the TFT model is closely aligned with the ground truth across both SpO2 and RR signals, and this demonstrates the model's superior capability in accurately capturing short-term fluctuations and underlying trends in physiological signals. The TFT's attention-based architecture allows it to dynamically capture subtle temporal variations and interdependencies between the multivariate inputs, particularly during periods of rapid signal change or transient fluctuations. In contrast, the TCN model tends to either over-smooth or lag during abrupt transitions, indicating its limited adaptability to high-frequency variations in physiological dynamics. Despite TCN's strengths in modeling general trends, its predictions exhibit notable discrepancies at signal peaks and troughs. These results underscore the TFT's superior temporal sensitivity and predictive precision over short-term horizons, making it more clinically reliable for early warnings in ICU monitoring scenarios. The consistent approximation to real values across different patients and signals shows the model's robustness, stability, and capacity for generalized representation under the proposed cascaded fine-tuning strategy.

Fig. 14 compares the performance of TFT and TCN for a 15 min prediction horizon, again relative to the ground truth for both physiological variables. As the forecasting window extends, the distinction between the two models becomes more pronounced. The TFT continues demonstrating strong predictive fidelity, preserving the signal morphology and trend directionality even over the longer horizon. The multi-head attention aspect of the model allows it to maintain contextual relevance over extended time steps. Conversely, the TCN model shows increased deviation from the ground truth, particularly in forecasting delayed trends and signal reversals. The limitations of fixed receptive fields and the absence of dynamic attention mechanisms in the TCN model become more evident in this setting. While TCN can approximate the general shape of the time series, its inability to model long-term dependencies with sufficient granularity leads to performance degradation. These findings affirm that TFT is more robust and effective than TCN in delivering accurate, reliable, multi-horizon forecasts for critical care variables.

5.4. Experiment 3: Results of the cascaded fine-tuning

Cascaded fine-tuning of the TFT is used as a robust approach for improving the generalizability and performance of the model for multivariate multi-horizon time-series forecasting of both RR and SpO2 values. The models are tuned using data from 15 patients and are then tested using the full time series for five different patients. As shown in Table 11, despite the increased complexity of this generalization task, where the model was sequentially fine-tuned with 15 patient datasets and tested on unseen data from five patients, the TFT achieved the best RMSE values across all forecasting horizons.

For short-term (7 min) forecasts, the TFT demonstrated robust performance, achieving RMSE values of 2.5278 and 2.1320 for RR and SpO2 respectively for Patient 1 (comparable to the previous single-patient settings with RMSEs of 1.4780 and 1.5776 for RR and SpO2), and RMSE values of 6.3006 and 1.1872 for RR and SpO2 respectively for Patient 3. For mid-term (15 min) predictions, the TFT maintained strong performance with RMSE values of 3.0571 and 3.0035 for RR and SpO2 for Patient 1, and 6.9273 and 1.2622 for Patient 3, while for long-term (25 min) forecasts, the model achieved RMSEs of 3.6328 and 3.3786 for RR and SpO2 for Patient 1, and 7.2340 and 1.5670 for Patient 3. The TFT's ability to manage long-term dependencies while maintaining accuracy highlights its robustness, particularly in unseen patient data, with small RMSE gaps between cascaded fine-tuning and single-patient training. For instance, for Patient 5 at a 7 min forecast, the TFT achieved RMSEs of 3.5019 (RR) and 1.7368 (SpO2) under cascaded fine-tuning, compared to 2.4194 (RR) and 1.0604 (SpO2) in single-patient training, illustrating that cascaded fine-tuning achieves a balance between performance and generalizability. The model's performance in SpO2 forecasting is particularly noteworthy, with RMSEs consistently lower than for RR, such as in the 7 min forecast where SpO2 RMSE values ranged between 1.1872 and 2.1320 across all patients, highlighting the TFT's capability to capture relatively stable SpO2 dynamics compared to the more variable RR signals. While cascaded fine-tuning presents increased challenges due to its focus on generalization, the TFT's sequential adaptation to diverse data patterns enhances its robustness and potential for real-time healthcare monitoring, where patient-specific training may not be feasible. These findings validate the TFT as a scalable and generalizable tool for real-time patient monitoring, achieving strong performance metrics despite the complexity of multivariate multi-horizon forecasting tasks, and demonstrating its critical role in predictive healthcare analytics.

Table 8
Results of the multivariate multi-horizon time-series forecasting models.

Patients	Forecasting minutes	Model	RR		SpO2		Time
			RMSE	MAE	RMSE	MAE	
Patient 1	Forecasting 7 minutes	S2S-LSTM	5.5770	5.1597	8.3750	8.1240	1 min 10 s
		S2S-GRU	3.4825	3.0027	3.4868	3.1259	1 min 15 s
		S2S-A-BiLSTM	4.3638	4.0595	5.1057	4.8469	1 min 15 s
		S2S-A-BiGRU	2.4069	1.9250	2.2165	1.8220	1 min 12 s
		TCN	2.2198	1.6029	2.0335	1.6293	0 min 45 s
		TFT	**1.4780**	**1.1171**	**1.5776**	**0.9699**	**1 min 40 s**
	Forecasting 15 minutes	S2S-LSTM	6.1582	6.6752	9.0949	8.8188	1 min 30 s
		S2S-GRU	4.8751	4.4165	4.1339	3.745	1 min 35 s
		S2S-A-BiLSTM	5.6078	5.2333	6.016	5.7511	1 min 32 s
		S2S-A-BiGRU	3.5796	2.7305	2.9987	2.5403	1 min 31 s
		TCN	2.6287	2.2631	2.9019	2.4110	0 min 55 s
		TFT	**1.6058**	**1.2332**	**2.7737**	**2.1730**	**1 min 55 s**
	Forecasting 25 minutes	S2S-LSTM	6.7062	6.172	9.8074	9.5064	1 min 40 s
		S2S-GRU	4.6713	4.1666	6.1856	5.8770	1 min 42 s
		S2S-A-BiLSTM	5.4694	5.9803	8.3051	8.0245	1 min 41 s
		S2S-A-BiGRU	2.6478	2.1627	3.7597	3.4095	1 min 40 s
		TCN	2.46552	2.0296	3.3884	2.9162	0 min 60 s
		TFT	**1.9724**	**1.6479**	**2.9873**	**2.2301**	**2 min 5 s**
Patient 2	Forecasting 7 minutes	S2S-LSTM	5.3750	3.5515	1.9619	1.4863	0 min 50 s
		S2S-GRU	5.2537	3.3974	1.9983	1.5161	0 min 54 s
		S2S-A-BiLSTM	5.3924	3.5320	1.9557	1.4998	1 min 5 s
		S2S-A-BiGRU	4.4362	3.1341	1.7795	1.4063	1 min 5 s
		TCN	3.9257	2.9257	1.5450	1.2082	0 min 40 s
		TFT	**3.7830**	**2.7510**	**1.1865**	**0.8837**	**1 min 20 s**
	Forecasting 15 minutes	S2S-LSTM	5.4181	3.7494	2.0894	1.5974	1 min 5 s
		S2S-GRU	5.6487	3.6957	2.0238	1.5097	1 min 7 s
		S2S-A-BiLSTM	5.5369	3.7758	2.0015	1.4977	1 min 25 s
		S2S-A-BiGRU	5.2306	3.2901	1.8606	1.3854	1 min25 s
		TCN	4.4838	3.0905	1.6242	1.12474	0 min 50 s
		TFT	**3.9396**	**2.8365**	**1.2271**	**0.9498**	**1 min 30 s**
	Forecasting 25 minutes	S2S-LSTM	5.9264	4.5052	2.5835	2.1128	1 min 15 s
		S2S-GRU	6.1330	4.9386	2.1434	1.6587	1 min17 s
		S2S-A-BiLSTM	5.3560	3.6684	1.9808	1.4998	1 min 38 s
		S2S-A-BiGRU	5.0926	3.6616	1.9722	1.4901	1 min 35 s
		TCN	5.0123	3.5120	1.8309	1.3701	0 min 55 s
		TFT	**4.6854**	**3.4482**	**1.3809**	**1.0894**	**1 min 45 s**
Patient 3	Forecasting 7 minutes	S2S-LSTM	8.2981	6.3413	0.6559	0.3841	0 min 47 s
		S2S-GRU	8.1163	6.2675	0.6479	0.3766	0 min 42 s
		S2S-A-BiLSTM	8.1500	6.3326	0.6406	0.3593	0 min 58 s
		S2S-A-BiGRU	6.6927	5.2571	0.5603	0.3424	0 min 55 s
		TCN	6.2665	4.9301	0.5235	0.3257	0 min 20 s
		TFT	**5.7043**	**4.3780**	**0.5003**	**0.2872**	**0 min 60 s**
	Forecasting 15 minutes	S2S-LSTM	8.7048	6.7639	0.6792	0.3986	0 min 60 s
		S2S-GRU	8.4319	6.4997	0.6472	0.3862	0 min 58 s
		S2S-A-BiLSTM	8.3162	6.4196	0.6935	0.3770	1 min 15 s
		S2S-A-BiGRU	7.3585	5.6338	0.5644	0.3406	1 min 13 s
		TCN	6.5046	5.2624	0.5693	0.3474	0 min 35 s
		TFT	**6.2357**	**4.7773**	**0.5377**	**0.3103**	**1 min 25 s**
	Forecasting 25 minutes	S2S-LSTM	8.9502	6.8962	1.154	0.7309	1 min 40 s
		S2S-GRU	8.5620	6.5808	1.1495	0.6639	1 min 38 s
		S2S-A-BiLSTM	8.3721	6.4139	0.9320	0.5749	1 min 55 s
		S2S-A-BiGRU	7.0485	5.3828	0.6876	0.4220	1 min 53 s
		TCN	6.6802	5.27960	0.6965	0.4173	0 min 45 s
		TFT	**6.3717**	**4.7875**	**0.7275**	**0.4703**	**1 min 45 s**

5.5. Prototype system for multivariate multi-horizon forecasting using streams of RR and SpO2 data

The results from the multivariate multi-horizon experiment showed that forecasting 7 min ahead using the TFT recorded the best performance in terms of the smallest RMSE and MAE values. Our prototype system used this optimal setup (the best transformer model for the same forecasting horizon), predicting RR and SpO2 in parallel and in real-time 7 min into the future using the TFT applied to the past 3 min of data.

In our system, the first step was to implement a simulated sensor using a Python script in order to generate time-series data for both RR and SpO2, and then send this streaming data to the storage zone to be stored in a Kafka topic. The Kafka Producer API collects data from the simulated sensor and saves it in the topic. A Flink consumer retrieves data from the topic using stream processing. Flink, a robust stream processing framework for complex computations, uses windowing to slice the retrieved data. The sliding window is defined according to the predefined interval (3 min) along the session. The session boundary covers all aggregated data, after which the aggregated data for both RR and SpO2 is sent to the TFT model in order to forecast RR and

Table 9
Continued results of the multivariate multi-horizon time-series forecasting models.

Patients	Forecasting minutes	Model	RR		SpO2		Time
			RMSE	MAE	RMSE	MAE	
Patient 4	Forecasting 7 minutes	S2S-LSTM	3.5305	2.8385	2.4429	1.9599	0 min 37 s
		S2S-GRU	3.5543	2.8468	2.3678	1.7798	0 min 36 s
		S2S-A-BiLSTM	3.4460	2.7023	2.4702	1.8398	0 min 46 s
		S2S-A-BiGRU	3.3922	2.6609	2.3200	1.6580	0 min 46 s
		TCN	3.1612	2.4350	1.8936	1.4364	0 min 35 s
		TFT	**2.8726**	**2.1320**	**1.3983**	**1.0251**	**0 min 50 s**
	Forecasting 15 minutes	S2S-LSTM	3.9215	3.1218	2.5215	1.9394	0 min 50 s
		S2S-GRU	3.5100	2.8273	2.5221	2.0232	0 min 55 s
		S2S-A-BiLSTM	3.4778	2.7903	2.5828	2.0670	0 min 60 s
		S2S-A-BiGRU	3.2368	2.6710	2.3358	1.7517	0 min 57s
		TCN	3.1425	2.5541	2.1196	1.3551	0 min 45 s
		TFT	**2.9506**	**2.2740**	**1.4856**	**1.0029**	**1 min 5 s**
	Forecasting 25 minutes	S2S-LSTM	4.3687	3.4560	3.0974	2.4871	1 min 10 s
		S2S-GRU	3.9970	3.2022	2.6101	2.0246	1 min 15 s
		S2S-A-BiLSTM	3.7032	2.9639	2.6681	2.1293	1 min 25 s
		S2S-A-BiGRU	3.6443	2.9393	2.3468	1.7653	1 min 28 s
		TCN	3.4813	2.7876	2.3026	1.8263	0 min 55 s
		TFT	**3.3647**	**2.5176**	**1.9076**	**1.5006**	**1 min 40 s**
Patient 5	Forecasting 7 minutes	S2S-LSTM	3.6247	2.3058	2.1958	1.4331	0 min 35 s
		S2S-GRU	3.5737	2.2749	2.2666	1.4822	0 min 36 s
		S2S-A-BiLSTM	3.5737	2.2989	2.1551	1.3900	0 min 46 s
		S2S-A-BiGRU	2.6105	1.9415	1.3462	0.9757	0 min 45 s
		TCN	2.6727	1.7805	1.3560	0.9833	0 min 25 s
		TFT	**2.4194**	**1.3957**	**1.0604**	**0.6963**	**0 min 50 s**
	Forecasting 15 minutes	S2S-LSTM	4.2103	2.8055	2.2547	1.5778	0 min 45 s
		S2S-GRU	4.0247	2.8753	2.0867	1.4438	0 min 47 s
		S2S-A-BiLSTM	3.8397	2.5267	1.9616	1.3740	0 min 56 s
		S2S-A-BiGRU	3.4041	2.2769	1.7544	1.1724	0 min 55 s
		TCN	3.2344	2.4918	1.5933	1.0887	0 min 40 s
		TFT	**2.8726**	**2.1320**	**1.3134**	**0.9222**	**1 min 5 s**
	Forecasting 25 minutes	S2S-LSTM	4.6894	3.4581	2.6314	1.6592	1 min 6 s
		S2S-GRU	4.5176	3.2982	2.2200	1.5184	1 min 10 s
		S2S-A-BiLSTM	4.5829	3.2671	2.3094	1.6464	1 min 25 s
		S2S-A-BiGRU	3.5367	2.7371	1.8435	1.4734	1 min 25 s
		TCN	3.4840	2.4433	1.8928	1.5947	0 min 60 s
		TFT	**3.0673**	**2.3470**	**1.6917**	**1.1952**	**1 min 30 s**

Table 10
Standard deviations for the multivariate multi-horizon models.

Forecasting minutes	Models	Multivariate multi-horizon			
		RR		SpO2	
		STD_RMSE	STD_MAE	STD_RMSE	STD_MAE
Forecasting 7 minutes	S2S-LSTM	1.9369	1.67626	3.0141	3.0986
	S2S-GRU	1.9996	1.5674	1.0157	0.9826
	S2S-A-BiLSTM	1.9319	1.5819	1.6322	1.6914
	S2S-A-BiGRU	1.7490	1.3695	0.7187	0.5951
	TCN	1.5938	1.3358	0.5940	0.5041
	TFT	1.6031	1.2992	0.4110	0.2985
Forecasting 15 minutes	S2S-LSTM	1.9169	1.9439	3.3018	3.3782
	S2S-GRU	1.9327	1.5112	1.2514	1.2288
	S2S-A-BiLSTM	1.9160	1.6544	2.003	2.0691
	S2S-A-BiGRU	1.7556	1.3426	0.8949	0.8048
	TCN	1.55796	1.2287	0.8510	0.7440
	TFT	1.7290	1.3207	0.8143	0.6772
Forecasting 25 minutes	S2S-LSTM	1.8374	1.5750	3.4064	3.5313
	S2S-GRU	1.8478	1.3912	1.9346	2.034
	S2S-A-BiLSTM	1.7564	1.6137	2.9052	2.988
	S2S-A-BiGRU	1.7234	1.2429	1.1056	1.0787
	TCN	1.05019316	1.2277	0.72087	0.6853
	TFT	1.6897	1.2114	0.8276	0.6420

Table 11
Results of the cascaded fine-tuning.

Forecasting minutes	Patients	RR		SpO2	
		RMSE	MAE	RMSE	MAE
Forecasting 7 minutes	Patient 1	2.5278	2.1245	2.1320	1.7445
	Patient 2	4.3087	3.3868	1.8297	1.2666
	Patient 3	6.3006	4.9625	1.1872	0.9616
	Patient 4	3.2345	2.5658	1.7096	1.5114
	Patient 5	3.5019	2.8819	1.7368	1.3400
Forecasting 15 minutes	Patient 1	3.0571	2.4251	3.0035	2.3423
	Patient 2	4.9959	4.0095	2.0508	1.5686
	Patient 3	6.9273	5.5321	1.2622	1.0859
	Patient 4	3.3502	2.6799	2.5278	2.1245
	Patient 5	3.8494	2.9177	2.1033	1.7138
Forecasting 25 minutes	Patient 1	3.6328	3.0129	3.3786	2.7452
	Patient 2	5.4209	4.3608	2.1542	1.5749
	Patient 3	7.2340	6.1150	1.5670	1.1272
	Patient 4	4.2720	4.0663	2.6781	2.0690
	Patient 5	4.0859	3.2310	2.2427	1.6919

SpO2 values 7 min ahead of time. Then, this data is stored in InfluxDB, designed to handle high-frequency data efficiently. InfluxDB indexes and stores the time-series data, facilitating easy retrieval. To carry out analyses and visualizations, we send the predicted and aggregated data to the Grafana platform. Grafana offers effective monitoring, analysis, and visualization tools, allowing us to gain valuable insights from the data.

5.6. Comparison with literature studies

It is challenging to compare results across various studies in the literature due to differences in sample sizes, signal lengths, data distributions, and other factors. Despite these challenges, we compare our work with related studies as shown in Table 12, based on methodologies, time-series approaches, classification/regression, regression (forecasting features), multivariate multi-horizon capabilities, transformer models, real-time analysis, datasets, and results.

Authors such as Kumar et al. [16], Lee et al. [17], and Chowdhury et al. [18] used the BIDMC dataset to train and evaluate their models. This dataset provides 8 min of data per patient, a relatively short duration for assessing model performance. These studies primarily predicted SpO2 or RR over short periods, such as one-second steps, with 30- and 60-second input windows in [18]. Erion et al. [10] applied LSTM models to predict hypoxemia using the AIMS dataset, achieving an AUPRC of 23.139 and an AUROC of 86.571, focusing on classification problems. Annapragada et al. [26] proposed their own SWIFT system to address classification problems. Shuzan et al. [15] tackled regression problems using the PPG dataset, achieving an RMSE of 1.41 for RR and an RMSE of 0.98 for SpO2. Bandopadhaya et al. [11] employed an encoder–decoder LSTM model with sensor-collected data, reporting an MAE of 1.29 and an RMSE of 1.51. For SpO2 prediction, Priem et al. [12] used deep learning with the BORA dataset, recording an RMSE of 4.4. Similarly, Zhang et al. [13] applied linear and nonlinear methods with their dataset, achieving an RMSE of 1.8. Tonmoy et al. [14] used linear regression with their dataset and reported an MAE of 0.845. Kumar et al. [16] applied a Bi-LSTM model with attention mechanisms using the BIDMC dataset, achieving an MAE of 0.70 for a one-step prediction. Soojeong et al. [17] employed gradient boosting (GB) with BIDMC, reporting an MAE of 1.94. Baker et al. [27] applied RQI with BiLSTM using the MIMIC-III dataset and recorded an MAE of 0.821. Finally, Bian et al. used ResNet and reported an MAE of 2.5.

In this study, we proposed a real-time monitoring system for an ICU patient's vital signs based on the TFT temporal transformer model. The model achieved promising results in terms of the RMSE and MAE. The metrics we reported, RMSE and MAE, provide quantitative measures of model accuracy. However, their interpretation in the context of clinical applications is indeed critical to understanding the practical benefits of our approach.

These metrics can be translated into clinical significance. RMSE reflects the standard deviation of prediction errors. A lower RMSE indicates that the predicted values are closer to the observed measurements, which is crucial for maintaining the reliability of patient monitoring. For example, an RMSE of 1.8 in terms of predicting RR translates to a deviation of less than two breaths per minute, a clinically acceptable range for early intervention. MAE provides the average magnitude of errors without considering their direction. For SpO2, an MAE of 1.5 suggests that the predicted values are, on average, within 1.5% of the actual measurements. This level of precision ensures timely detection of critical events.

The resulting system can impact the outcome of an ICU patient's monitoring process. The improved accuracy of attention-based models minimizes false alarms and missed critical events, directly impacting patient safety by enabling precise and timely alerts for abnormal trends in RR and SpO2. Accurate multi-horizon predictions provide clinicians with actionable foresight, allowing for proactive interventions. For example, a consistent prediction that SpO2 falls below 90% would prompt adjustments in oxygen therapy to prevent hypoxemia. High-fidelity predictions align with clinical guidelines, reducing the need for constant manual verification of vital sign trends, and allowing healthcare professionals to focus on critical tasks. The deployment of our proposed TFT model in ICUs can transform patient care by integrating accurate predictions into decision-making systems, leading to better resource allocation and patient management. The cascading fine-tuning approach ensures that our model generalizes well to unseen patient data, further increasing its applicability in diverse clinical environments.

6. Limitations and future work

Our study presents a significant advancement at the intersection of the medical and AI domains by proposing a robust real-time patient monitoring and forecasting framework. Integrating TFTs and cascaded fine-tuning ensures highly accurate predictions of critical physiological indicators such as SpO2 and RR. These contributions enhance clinical decision-making and demonstrate the potential of AI to address complex temporal dynamics in healthcare data. Additionally, the system's real-time data processing and multi-horizon forecasting capabilities align with real-world clinical workflows, making it a valuable tool for intensive care units (ICUs).

Despite these contributions, several limitations remain, paving the way for future research directions.

1. Integrating federated learning can enhance data privacy and facilitate collaboration across healthcare institutions without

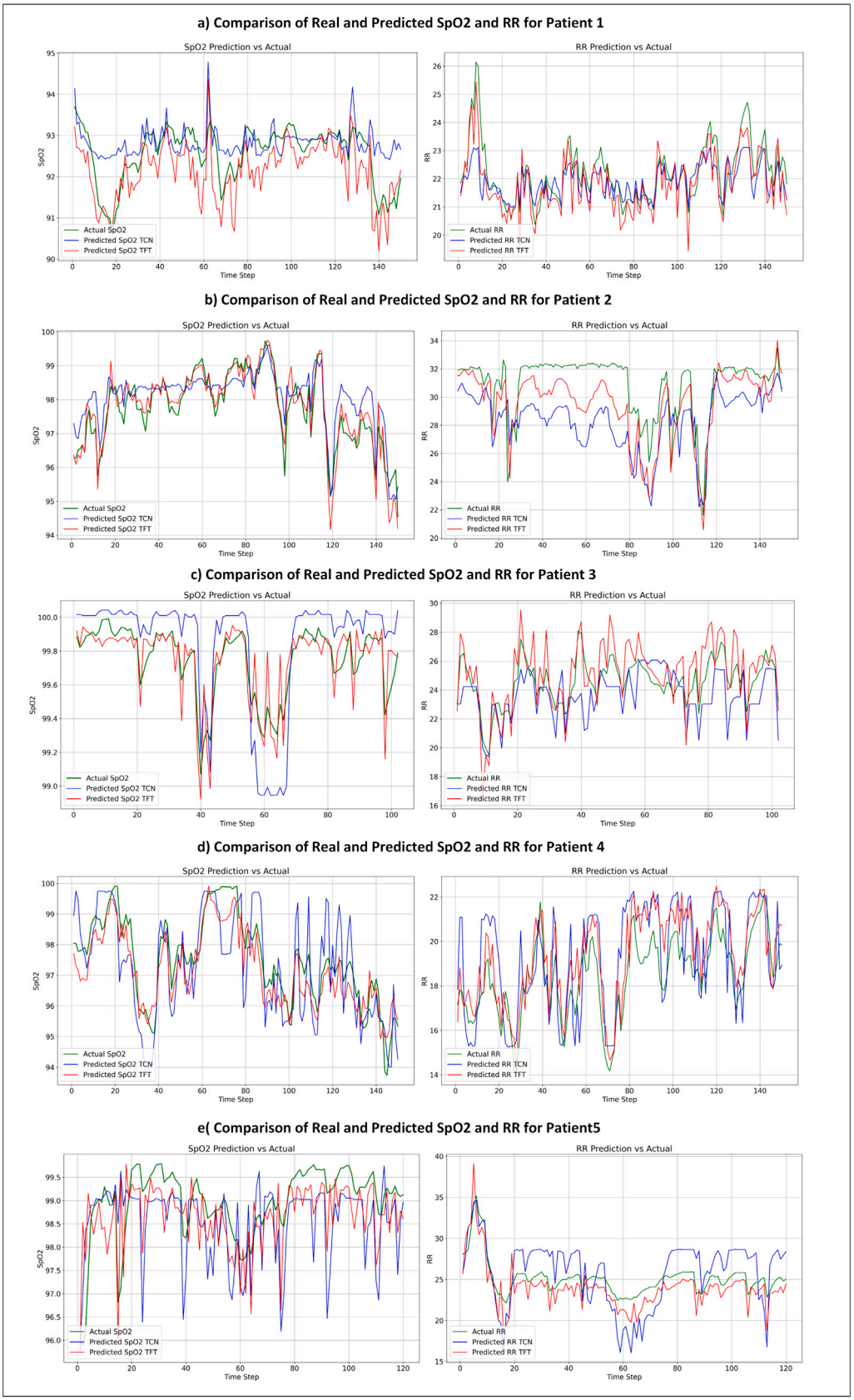

Fig. 13. SpO2 and RR real versus predicted values over time for the two best models TFT and TCN, for five patients 7 min in advance.

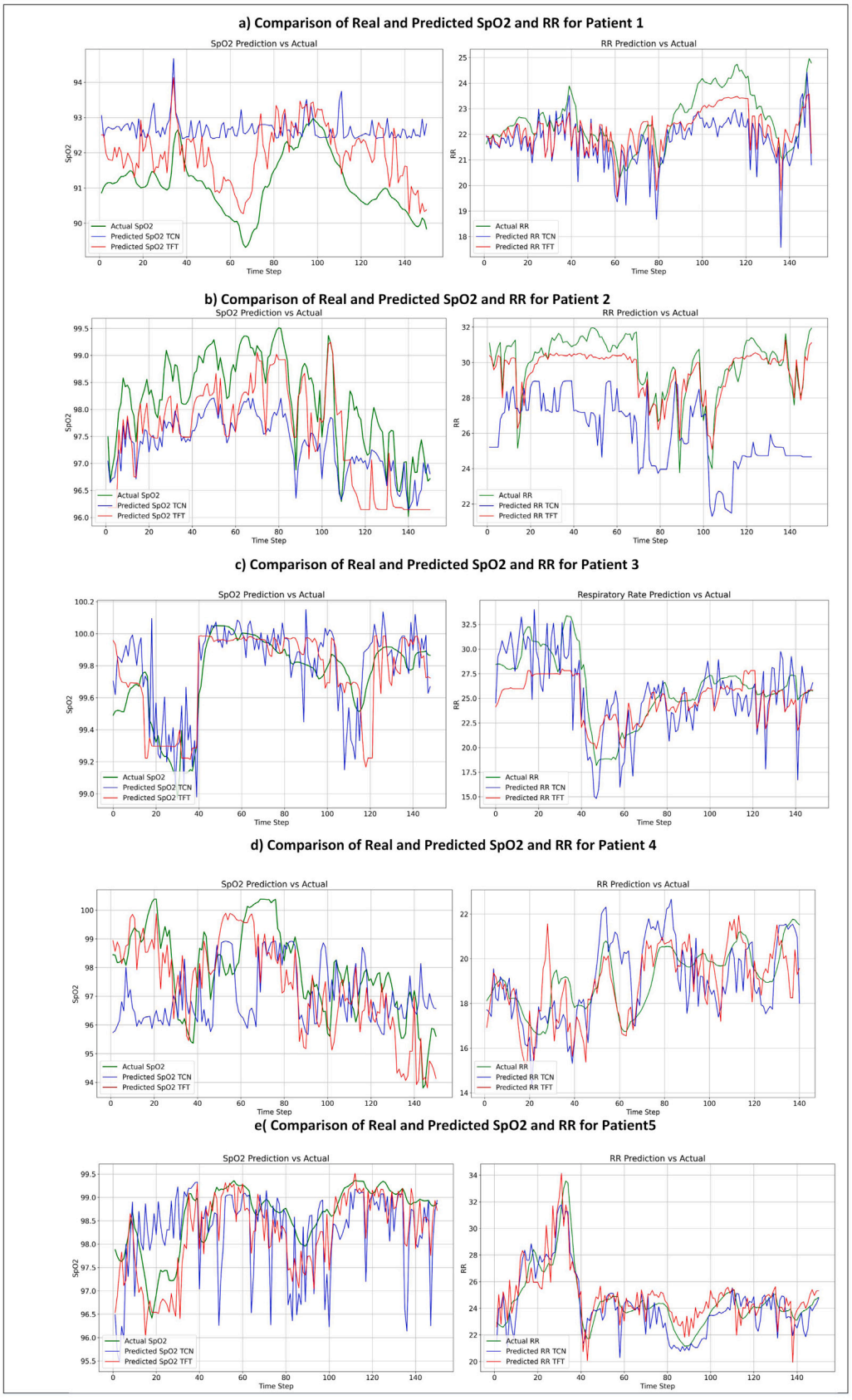

Fig. 14. SpO2 and RR real versus predicted values over time for the two best models TFT and TCN, for five patients 15 min in advance.

Table 12
Comparison with literature studies.

Papers	Years	Methods	Timeseries	Classification, Regression	Regression (Features for forecasts)	Real-Time	Multivariate multi-horizon	Transformer	Dataset	Results
Erion, Gabriel, et al. [10]	2017	LSTM	Yes	Classification	No	No	No	No	AIMS	AU-PRC=23.139 AU-ROC=86.571
Annapragada et al. [26]	2021	SWIFT	Yes	Classification	SpO2	No	No	No		–
Bandopadhaya et al. [11]	2023	Encoder–Decoder LSTM	Yes	Regression	SpO2	No	No	No	Own	MAPE=1.56 MAE=1.29 RMSE=1.51
Priem, Gurvan et al. [12]	2020	DL	Yes	Regression	SpO2	No	No	No	BORA	RMSE=4.4
Zhang, Qingxue et al. [13]	2022	Linear/Nonlinear Models	Yes	Regression	SpO2	No	No	No	Own	RMSE=1.8
Tonmoy et al. [14]	2024	LR	Yes	Regression	SpO2	No	No	No	Own	MAE=0.845
Chowdhury et al. [18]	2024	ROSE-Net	Yes	Regression	SpO2	No	No	No	BIDMC	MAE=1.20 RMSE=1.86
Kumar et al. [16]	2022	Bi-LSTM with Attention	Yes	Regression	RR	No	No	No	BIDMC	MAE=0.70
Baker et al. [27]	2021	RQI with BiLSTM	Yes	Regression	RR	No	No	No	PPG	MAE=0.821
Soojeong et al. [17]	2022	GB	Yes	Regression	RR	No	No	No	BIDMC	MAE=1.94
Bian et al. [28]	2020	ResNet	Yes	Regression	RR	No	No	No	PPG	MAE=2.5
Shuzan et al. [15]	2023	GPR with FSLib	Yes	Regression	SpO2 or RR	No	No	No	PPG	RR (RMSE=1.41, MAE=0.89) SpO2 (RMSE=0.98, MAE=0.57)
Our work	2025	TFT	Yes	Regression	SpO2 and RR	Yes	Yes	Yes	MIMIC-III	RR (RMSE=1.4780, MAE=1.1171) SpO2 (RMSE=1.5776, MSE=0.9699)

sharing sensitive patient data. Different federated learning architectures [55] can be explored to select the most suitable one for real-time monitoring in the sensitive ICU domain.

2. Further innovations in time-series forecasting algorithms can improve the handling of long-sequence dependencies and irregular sampling, addressing challenges inherent in multivariate time-series data such as missing values, multimodal data, data balancing, and data bias [56].
3. Its applicability in real-world healthcare contexts would require additional validation to ensure seamless integration with clinical workflows, analyzing any practical deployment challenges, and tackling interoperability challenges inherent in environments where many different healthcare systems are deployed [57]. Our real-time forecasting pipeline has only been validated in a simulated environment with MIMIC-III streaming data. An external validation of the model using other ICU time-series data will be handled in a future study. The model's performance in live clinical infrastructures remains untested, with various constraints to be considered, including network latency, sensor failures, and integration with electronic health record (EHR) systems. Future efforts should involve deployment in hospital testbeds, assessing system performance under real clinical loads, data irregularities, and infrastructure variability.
4. Addressing the time complexity of system deployment through optimization techniques or lightweight models can make the system more accessible for resource-constrained settings. Transformer-based models, including TFTs, can be computationally intensive and may not be suitable for deployment in resource-constrained environments like edge devices or rural clinics with poor connectivity. Exploration of model compression techniques such as pruning, knowledge distillation, or quantization may enable lightweight deployment without compromising predictive performance.
5. External validation using diverse datasets from different healthcare institutions can improve the model's generalizability and robustness.
6. Incorporating explainable AI (XAI) algorithms will enhance model transparency, helping clinicians understand the predictions and trust the system's outputs. Although the TFT model provides inherent interpretability through attention mechanisms and variable selection networks, this study did not visualize or quantify these interpretive elements. Future work will incorporate attention heat maps and feature importance visualizations to illustrate better which input variables influence predictions at different time horizons. These interpretability components are essential for clinician-facing transparency and will support more informed and trustworthy deployment in real-time critical care settings. In addition, future work could incorporate model-agnostic interpretability techniques (e.g., SHAP and LIME) and collaborations with clinicians to validate the decision pathways of the model in real-time settings.
7. The current implementation does not explicitly address the security and privacy challenges of streaming and storing sensitive patient data in real-time systems. In clinical environments, any AI-powered monitoring system must comply with strict data protection regulations such as HIPAA (USA) or GDPR (EU), particularly when integrating with cloud platforms, IoT devices, or third-party analytics tools. Future research should explore secure data transmission protocols (e.g., TLS, end-to-end encryption), privacy-preserving machine learning techniques (e.g., federated learning, differential privacy), and role-based access controls to ensure compliance with regulatory frameworks.
8. Patient conditions can change rapidly in ICU settings, potentially causing model drift if the underlying data distribution shifts. The current framework does not include mechanisms for online learning or dynamic adaptation. Future research could implement continual learning or adaptive retraining mechanisms that respond to detected concept drift or patient deterioration events, ensuring sustained model relevance.
9. The proposed model has not been evaluated under noisy, incomplete, or adversarial data conditions, which are common in real-world clinical monitoring scenarios. Robustness testing

under missing data, sensor dropout, and adversarial input conditions is necessary to ensure the reliability and trustworthiness of forecasts.

10. While the selection of prediction horizons (7, 15, and 25 min) was informed by literature, no formal user studies or clinician feedback were integrated into this research phase. As a result, the model's alignment with real-world clinical decision-making workflows still needs to be validated. Getting feedback from ICU clinicians is critical to validate and improve the model's applicability in a real environment. Future research includes structured consultations with ICU clinicians and usability testing to refine prediction intervals and interface design based on clinical priorities and operational constraints. In addition, a notable strength of the proposed framework is its modular and extensible architecture, which facilitates incremental integration into clinical environments. While the current study was conducted in a simulated setting, each system component, from data ingestion to visualization, can be independently adapted or replaced to align with existing hospital infrastructures and regulatory requirements. Future research focuses on progressively embedding the framework into real-world ICU workflows, emphasizing compliance with clinical standards for deployment in smart healthcare settings. Moreover, our study did not fully account for real-world ICU challenges such as asynchronous data arrival, sensor noise, missing values, and device integration issues. These factors are critical for evaluating any clinical deployment's robustness and fault tolerance. Future work will focus on testing the proposed framework under more realistic conditions, including asynchronous data ingestion, signal noise augmentation, and potential integration with edge-computing hardware or clinical telemetry systems to validate performance under real-world constraints.
11. Finally, establishing a direct connection to hospital EHRs will enable seamless data flow, fostering real-time, actionable insights directly within existing healthcare infrastructures. By addressing these limitations, future research can improve our framework's quality, scalability, and real-world applicability, further bridging the gap between cutting-edge AI innovations and practical clinical implementations.

7. Conclusion

This study introduced a real-time forecasting framework, StreamHealth Multi-Horizon AI, for multivariate multi-horizon prediction of critical ICU markers, SpO2 and RR, using the MIMIC-III dataset. The Temporal Fusion Transformer (TFT) demonstrated superior performance over classical and Seq2Seq-based deep learning models in univariate and multivariate settings. The cascaded fine-tuning strategy improved the model's generalizability to unseen patient data, a key advantage in heterogeneous clinical contexts.

Comprehensive experiments demonstrated the superior performance of TFT over classical methods such as LSTM, GRU, Bi-LSTM, TCN, and CNN on various forecast horizons. The cascaded fine-tuning approach further validated the robustness and generalizability of the TFT, achieving consistent accuracy when tested on unseen patient data from the MIMIC-III dataset. Additionally, integrating streaming technologies, such as Apache Kafka and Apache Flink, enabled real-time data ingestion and processing. At the same time, the visualization capabilities provided by Grafana ensured actionable insights for clinicians. However, the framework assumed idealized data conditions and did not yet address deployment constraints such as concept drift, asynchronous measurements, or integration with clinical systems. Various limitations related to dataset scope, model interpretability, and system scalability will be covered in future studies.

This work marks a critical step toward enhancing real-time ICU monitoring systems, facilitating proactive decision-making, and improving patient outcomes. By addressing key challenges in multivariate forecasting and demonstrating scalability and accuracy, the proposed framework paves the way for broader adoption in clinical applications and future research in predictive healthcare analytics. The chosen ICU setting is not an isolated case: the model applies to any other medical environment that collects temporal multivariate data and needs real-time decisions. However, the ICU environment was a perfect example for this work because medical markers are being continuously collected from sensors, and timely decisions are critical.

CRediT authorship contribution statement

Hager Saleh: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Shaker El-Sappagh:** Writing – review & editing, Writing – original draft, Methodology, Data curation. **Michael McCann:** Writing – review & editing, Writing – original draft. **Saeed Hamood Alsamhi:** Writing – review & editing, Writing – original draft, Investigation, Funding acquisition. **John G. Breslin:** Writing – review & editing, Writing – original draft, Software, Project administration.

Ethical approval

Not applicable.

Declaration of competing interest

All authors declare that they have no conflicts of interest.

Acknowledgments

This publication has emanated from research conducted with the financial support of Taighde Éireann - Research Ireland under Grant Numbers 12/RC/2289_P2 (Insight) and 21/FFP-A/9174 (SustAIn). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

Data availability

All datasets used to support the findings of this study are available from the direct link in the dataset citations.

References

[1] A. Archip, N. Botezatu, E. Şerban, P.-C. Herghelegiu, A. Zală, An IoT based system for remote patient monitoring, in: 2016 17th International Carpathian Control Conference, ICCC, IEEE, 2016, pp. 1–6.

[2] Z. Zhu, R.K. Barnette, K.M. Fussell, R.M. Rodriguez, A. Canonico, R.W. Light, Continuous oxygen monitoring—a better way to prescribe long-term oxygen therapy, Respir. Med. 99 (11) (2005) 1386–1392.

[3] P. Faverio, F. De Giacomi, G. Bonaiti, A. Stainer, L. Sardella, G. Pellegrino, G.F.S. Papa, F. Bini, B.D. Bodini, M. Carone, et al., Management of chronic respiratory failure in interstitial lung diseases: overview and clinical insights, Int. J. Med. Sci. 16 (7) (2019) 967.

[4] A.I. Siam, M.A. Almaiah, A. Al-Zahrani, A.A. Elazm, G.M. El Banby, W. El-Shafai, F.E.A. El-Samie, N.A. El-Bahnasawy, Secure health monitoring communication systems based on IoT and cloud computing for medical emergency applications, Comput. Intell. Neurosci. 2021 (1) (2021) 8016525.

[5] S. El-Sappagh, H. Saleh, R. Sahal, T. Abuhmed, S.R. Islam, F. Ali, E. Amer, Alzheimer's disease progression detection model based on an early fusion of cost-effective multimodal data, Future Gener. Comput. Syst. 115 (2021) 680–699.

[6] M. Marcos, M. Belhassen-García, A. Sánchez-Puente, J. Sampedro-Gomez, R. Azibeiro, P.-I. Dorado-Díaz, E. Marcano-Millán, C. García-Vidal, M.-T. Moreiro-Barroso, N. Cubino-Bóveda, et al., Development of a severity of disease score and classification model by machine learning for hospitalized COVID-19 patients, PLoS One 16 (4) (2021) e0240200.

[7] M.A. Al-Sheikh, I.A. Ameen, Design of mobile healthcare monitoring system using IoT technology and cloud computing, in: IOP Conference Series: Materials Science and Engineering, vol. 881, (no. 1) IOP Publishing, 2020, 012113.

[8] W. Chen, I. Ayoola, S.B. Oetomo, L. Feijs, Non-invasive blood oxygen saturation monitoring for neonates using reflectance pulse oximeter, in: 2010 Design, Automation & Test in Europe Conference & Exhibition, DATE 2010, IEEE, 2010, pp. 1530–1535.

[9] C. Rotariu, V. Manta, Wireless system for remote monitoring of oxygen saturation and heart rate, in: 2012 Federated Conference on Computer Science and Information Systems, FedCSIS, IEEE, 2012, pp. 193–196.

[10] G. Erion, H. Chen, S.M. Lundberg, S.-I. Lee, Anesthesiologist-level forecasting of hypoxemia with only SpO2 data using deep learning, 2017, arXiv preprint arXiv:1712.00563.

[11] S. Bandopadhaya, A. Roy, Early detection of silent hypoxia in COVID-19 pneumonia using deep learning and IoT, Multimedia Tools Appl. 83 (8) (2024) 24527–24539.

[12] G. Priem, C. Martinez, Q. Bodinier, G. Carrault, Clinical grade SpO2 prediction through semi-supervised learning, in: 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering, BIBE, IEEE, 2020, pp. 914–921.

[13] Q. Zhang, D. Arney, J.M. Goldman, E.M. Isselbacher, A.A. Armoundas, Design implementation and evaluation of a mobile continuous blood oxygen saturation monitoring system, Sensors 20 (22) (2020) 6581.

[14] A.S. Tonmoy, M.S. Ahmed, A. Chowdhury, M.H. Chowdhury, Estimation of oxygen saturation from PPG signal using smartphone recording, in: 2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems, ICACCESS, IEEE, 2024, pp. 1–6.

[15] M.N.I. Shuzan, M.H. Chowdhury, M.E. Chowdhury, M. Murugappan, E. Hoque Bhuiyan, M. Arslane Ayari, A. Khandakar, Machine learning-based respiration rate and blood oxygen saturation estimation using photoplethysmogram signals, Bioengineering 10 (2) (2023) 167.

[16] A.K. Kumar, M. Ritam, L. Han, S. Guo, R. Chandra, Deep learning for predicting respiratory rate from biosignals, Comput. Biol. Med. 144 (2022) 105338.

[17] S. Lee, H. Moon, C.-H. Son, G. Lee, Respiratory rate estimation combining autocorrelation function-based power spectral feature extraction with gradient boosting algorithm, Appl. Sci. 12 (16) (2022) 8355.

[18] M.H. Chowdhury, M.B.I. Reaz, S.H.M. Ali, M.S. Khan, M.E. Chowdhury, ROSE-Net: Leveraging remote photoplethysmography to estimate oxygen saturation using deep learning, Biomed. Signal Process. Control. 100 (2025) 107105.

[19] M.A. Pimentel, A.E. Johnson, P.H. Charlton, D. Birrenkott, P.J. Watkinson, L. Tarassenko, D.A. Clifton, Toward a robust estimation of respiratory rate from pulse oximeters, IEEE Trans. Biomed. Eng. 64 (8) (2016) 1914–1923.

[20] T.A. Trikalinos, D.C. Hoaglin, C.H. Schmid, An empirical comparison of univariate and multivariate meta-analyses for categorical outcomes, Stat. Med. 33 (9) (2014) 1441–1459.

[21] H. Ahmed, E.M. Younis, A. Hendawi, A.A. Ali, Heart disease identification from patients' social posts, machine learning solution on spark, Future Gener. Comput. Syst. 111 (2020) 714–722.

[22] M.T. Islam, H. Wu, S. Karunasekera, R. Buyya, SLA-based scheduling of spark jobs in hybrid cloud computing environments, IEEE Trans. Comput. 71 (5) (2021) 1117–1132.

[23] G. Elhayatmy, N. Dey, A.S. Ashour, Internet of things based wireless body area network in healthcare, Internet Things Big Data Anal. Towar. Next- Gener. Intell. (2018) 3–20.

[24] S. El-Sappagh, T. Abuhmed, S.R. Islam, K.S. Kwak, Multimodal multitask deep learning model for alzheimer's disease progression detection based on time series data, Neurocomputing 412 (2020) 197–215.

[25] B. Lim, S.Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, Int. J. Forecast. 37 (4) (2021) 1748–1764.

[26] A.V. Annapragada, J.L. Greenstein, S.N. Bose, B.D. Winters, S.V. Sarma, R.L. Winslow, SWIFT: A deep learning approach to prediction of hypoxemic events in critically-ill patients using SpO2 waveform prediction, PLoS Comput. Biol. 17 (12) (2021) e1009712.

[27] S. Baker, W. Xiang, I. Atkinson, Determining respiratory rate from photoplethysmogram and electrocardiogram signals using respiratory quality indices and neural networks, PLoS One 16 (4) (2021) e0249843.

[28] D. Bian, P. Mehta, N. Selvaraj, Respiratory rate estimation using PPG: A deep learning approach, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC, IEEE, 2020, pp. 5948–5952.

[29] W. Li, K.E. Law, Deep learning models for time series forecasting: a review, IEEE Access (2024).

[30] C. Li, Z. Shi, L. Zhou, Z. Zhang, C. Wu, X. Ren, X. Hei, M. Zhao, Y. Zhang, H. Liu, et al., Tfformer: A time frequency information fusion based cnn-transformer model for osa detection with single-lead ecg, IEEE Trans. Instrum. Meas. (2023).

[31] Z. Sun, R. Li, J. Wang, G. Chen, S. Yan, L. Ma, Static and multivariate-temporal attentive fusion transformer for readmission risk prediction, 2024, arXiv preprint arXiv:2407.11096.

[32] R.Y. He, J.N. Chiang, TFT-multi: simultaneous forecasting of vital sign trajectories in the ICU, 2024, arXiv preprint arXiv:2409.15586.

[33] B.N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-BEATS: Neural basis expansion analysis for interpretable time series forecasting, 2019, arXiv preprint arXiv:1905.10437.

[34] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 12, 2021, pp. 11106–11115.

[35] L. Hebert, L. Golab, P. Poupart, R. Cohen, Fedformer: Contextual federation with attention in reinforcement learning, 2022, arXiv preprint arXiv:2205.13697.

[36] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, M. Zhou, Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training, 2020, arXiv preprint arXiv:2001.04063.

[37] L.R. Nair, S.D. Shetty, S.D. Shetty, Applying spark based machine learning model on streaming big data for health status prediction, Comput. Electr. Eng. 65 (2018) 393–399.

[38] A. Ed-Daoudy, K. Maalmi, Application of machine learning model on streaming health data event in real-time to predict health status using spark, in: 2018 International Symposium on Advanced Electrical and Communication Technologies, Isaect, IEEE, 2018, pp. 1–4.

[39] A. Ed-daoudy, K. Maalmi, A. El Ouaazizi, A scalable and real-time system for disease prediction using big data processing, Multimedia Tools Appl. 82 (20) (2023) 30405–30434.

[40] A. Farki, E.A. Noughabi, Real-time blood pressure prediction using apache spark and kafka machine learning, in: 2023 9th International Conference on Web Research, ICWR, IEEE, 2023, pp. 161–166.

[41] H. Saleh, E.M. Younis, R. Sahal, A.A. Ali, Predicting systolic blood pressure in real-time using streaming data and deep learning, Mob. Networks Appl. 26 (2021) 326–335.

[42] L. Tan, K. Yu, A.K. Bashir, X. Cheng, F. Ming, L. Zhao, X. Zhou, Toward real-time and efficient cardiovascular monitoring for COVID-19 patients by 5G-enabled wearable medical devices: a deep learning approach, Neural Comput. Appl. (2023) 1–14.

[43] Apache Kafka, Apache kafka, 2024, https://kafka.apache.org/.

[44] ApacheFlink, Apache flink, 2024, https://nightlies.apache.org/flink/flink-docs-release-1.18/docs/deployment/overview/.

[45] S.N.Z. Naqvi, S. Yfantidou, E. Zimányi, Time series databases and influxdb, Stud. Université Libr. de Brux. 12 (2017) 1–44.

[46] M. Chakraborty, A.P. Kundan, Grafana, in: Monitoring Cloud-Native Applications: Lead Agile Operations Confidently using Open Source Software, Springer, 2021, pp. 187–240.

[47] A.E. Johnson, T.J. Pollard, L. Shen, L.-w.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (1) (2016) 1–9.

[48] Z.C. Lipton, D.C. Kale, R. Wetzel, et al., Modeling missing data in clinical time series with rnns, Mach. Learn. Heal. 56 (56) (2016) 253–270.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830, URL https://scikit-learn.org/stable/.

[50] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, Adv. Neural Inf. Process. Syst. 27 (2014).

[51] S. Bai, J.Z. Kolter, V. Koltun, Convolutional sequence modeling revisited, 2018.

[52] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, arXiv preprint arXiv:1803.01271.

[53] J. Chen, D. Chen, G. Liu, Using temporal convolution network for remaining useful lifetime prediction, Eng. Rep. 3 (3) (2021) e12305.

[54] R. Hai, C. Koutras, C. Quix, M. Jarke, Data lakes: A survey of functions and systems, IEEE Trans. Knowl. Data Eng. 35 (12) (2023) 12571–12590.

[55] A. Rauniyar, D.H. Hagos, D. Jha, J.E. Håkegård, U. Bagci, D.B. Rawat, V. Vlassov, Federated learning for medical applications: A taxonomy, current trends, challenges, and future research directions, IEEE Internet Things J. 11 (5) (2023) 7374–7398.

[56] X. Kong, Z. Chen, W. Liu, K. Ning, L. Zhang, S. Muhammad Marier, Y. Liu, Y. Chen, F. Xia, Deep learning for time series forecasting: a survey, Int. J. Mach. Learn. Cybern. (2025) 1–34.

[57] R.A. El Arab, M.S. Abu-Mahfouz, F.H. Abuadas, H. Alzghoul, M. Almari, A. Ghannam, M.M. Seweid, Bridging the gap: From AI success in clinical trials to real-world healthcare implementation—A narrative review, in: Healthcare, vol. 13, no. 7, MDPI, 2025, p. 701.