Computer Science and Information Technologies
Vol. 7, No. 3, November 2026, pp. 291~303
ISSN: 2722-3221, DOI: 10.11591/csit.v7i3.p291-303

Time series forecasting: a comparative analysis of ARIMA, LSTM, and TFT models with missing data handling

Maryam Hosseini, Mohamad Forouzanfar
Department of Systems Engineering, École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, Canada

Article Info

Article history:

Received Mar 13, 2026
Revised May 17, 2026
Accepted Jun 8, 2026

Keywords:

ARIMA
Long short-term memory
Missing data imputation
Temporal fusion transformer
Time series forecasting

ABSTRACT

Time series forecasting plays a critical role in finance, healthcare, and energy applications, where accurate predictions support decision-making and operational efficiency. Traditional approaches such as autoregressive integrated moving average (ARIMA) perform well for linear patterns but often struggle with nonlinear and complex temporal dependencies found in real-world data. Although deep learning methods such as long short-term memory (LSTM) networks and Transformer-based models have shown promise, comprehensive evaluations across multiple domains and under missing-data conditions remain limited. This study compares ARIMA, LSTM, and temporal fusion transformer (TFT) models across three applications: stock price forecasting using S&P 500 data, heart-rate prediction from electrocardiogram (ECG)-derived signals, and electricity demand forecasting using Pennsylvania–New Jersey–Maryland (PJM) power grid data. To evaluate robustness under realistic conditions, varying levels of missingness were introduced using missing completely at random (MCAR) and missing at random (MAR) mechanisms. Missing values were handled using forward fill, linear interpolation, and k-nearest neighbors (k-NN) imputation. Results show that TFT consistently achieved superior forecasting performance and demonstrated greater robustness to increasing missingness, while k-NN generally provided the most effective imputation performance across datasets.

Corresponding Author:

Maryam Hosseini
Department of Systems Engineering, École de Technologie Supérieure (ÉTS), Université du Québec
Montreal, Canada
Email: maryamy2k876@gmail.com

1. INTRODUCTION

Time-series forecasting has become increasingly important in data-driven applications; however, achieving reliable predictions in real-world settings remains challenging because temporal data frequently exhibit nonlinearity, long-range dependencies, noise, and incomplete observations. In finance, the ability to predict stock prices, market trends, and economic indicators is crucial for risk management, investment strategies, and financial planning [1], [2]. Similarly, in healthcare, forecasting vital signs like heart rate during sleep can enable early detection of abnormalities such as sleep disorders, improving patient outcomes [3]. Predicting tumor growth using imaging biomarkers can guide oncologists in assessing cancer progression and adjusting treatment plans proactively [4]. Additionally, forecasting patient responses to chemotherapy based on prior treatments and genomic data can help personalize therapy regimens, optimizing effectiveness and reducing side effects. In the energy sector, accurate predictions of electricity demand are critical for optimizing power grid operations and integrating renewable energy sources into the grid [5]–[7].

The challenge of forecasting time series data arises from its often volatile and complex nature, where non-linear patterns, noise, and long-term dependencies make accurate predictions difficult. Furthermore, real-world time series data often suffer from missing values, which can arise from various sources such as sensor failures, data entry errors, or irregular sampling. Missing data can significantly impact the accuracy and reliability of time series forecasting models, leading to biased or incomplete predictions. Therefore, effective handling of missing data is crucial to ensure the robustness and reliability of forecasting models.

Historically, the autoregressive integrated moving average (ARIMA) model has been a dominant approach in time series forecasting [8]. ARIMA is valued for its simplicity, interpretability, and efficiency in handling linear relationships. It is particularly adept at forecasting financial time series that can be made stationary through differencing, and it has also been applied in energy demand forecasting. However, ARIMA models have notable limitations, particularly in their inability to capture non-linear relationships and complex temporal dependencies. These limitations have prompted the development of more advanced models such as long short-term memory (LSTM) networks and Transformer models [9].

Deep learning has revolutionized the field of sequential data processing, offering powerful architectures capable of capturing complex patterns in diverse applications [10]–[14]. One of the key advancements in deep learning for time series forecasting is the use of recurrent neural networks (RNNs), particularly LSTM networks. LSTMs have emerged as a powerful tool for modeling long-term dependencies and capturing non-linear patterns in sequential data [15], [16]. They are especially effective in applications such as financial forecasting, where past events can have delayed effects on future market behaviors, healthcare for predicting vital signs, and energy forecasting for modeling consumption patterns influenced by factors like weather and human activity. However, despite these advantages, LSTMs struggle with capturing very long-range dependencies due to their sequential nature, which can lead to vanishing gradient issues. To address these shortcomings, Transformer models have emerged as a state-of-the-art alternative, particularly excelling in natural language processing and increasingly being applied to time series forecasting [17]. Unlike LSTMs, Transformers use self-attention mechanisms to process entire sequences in parallel, significantly improving computational efficiency and enabling the modeling of much longer-range dependencies [18], [19]. This ability to capture complex temporal patterns while scaling effectively makes Transformers highly promising for applications in financial, healthcare, and energy forecasting.

Despite significant advances in time-series forecasting, several limitations remain in the existing literature. Previous studies have often focused on evaluating forecasting models within a single application domain, such as stock markets, healthcare monitoring, or energy systems separately. Moreover, most comparative studies emphasize prediction accuracy under complete datasets and pay limited attention to the impact of missing data, despite missingness being a common issue in real-world environments due to sensor failures, communication errors, irregular sampling, and incomplete records. Existing research rarely provides a unified evaluation of traditional statistical models and recent deep learning architectures across multiple domains under controlled missing-data scenarios. In particular, comparative investigations of ARIMA, LSTM, and temporal fusion transformer (TFT) under different missing-data mechanisms and imputation strategies remain limited. This gap restricts understanding of model robustness and practical applicability in real-world forecasting settings.

To address these limitations, this study presents a comprehensive comparative analysis of ARIMA, LSTM, and TFT models across three representative domains: finance, healthcare, and energy. The novelty of this work lies in: i) evaluating model performance across multiple application domains rather than a single dataset; ii) systematically investigating model robustness under realistic missing-data conditions using both missing completely at random (MCAR) and missing at random (MAR) mechanisms; iii) comparing multiple imputation techniques, including forward fill, linear interpolation, and k-nearest neighbors (k-NN); and iv) analyzing the capability of advanced transformer-based architectures to maintain forecasting performance under increasing missingness. These contributions provide practical insights into model selection and missing-data handling strategies for real-world time-series forecasting applications.

2. METHOD

2.1. Datasets

We used three datasets across different domains: finance, healthcare, and energy.

2.1.1. Finance dataset

The financial dataset used in this study consists of daily stock prices from the S&P 500 index. This dataset covers a 10-year period, providing a rich historical context for forecasting stock price trends. The dataset includes the following variables:

- Open price: the price at which the stock market opened for each trading day.

- Close price: the final price at which the stock market closed for each trading day.
- High price: the highest price reached during the trading day.
- Low price: the lowest price reached during the trading day.
- Volume: the total number of shares traded during the trading day.

Our goal on this dataset was to predict the closing price of the S&P 500 index for the next day, based on the historical values of the aforementioned variables.

2.1.2. Healthcare dataset

The healthcare dataset, derived from the Physionet Apnea-ECG Database, comprises heart rate data derived from electrocardiogram (ECG) signals for 35 individuals during sleep, which contains annotated ECG recordings used in clinical research for sleep studies, specifically for detecting sleep apnea and other cardiovascular conditions [20]. The heart rate data in this dataset includes:

- Instantaneous heart rate (bpm): extracted from the R-R intervals in ECG signals using the Pan-Tompkins algorithm.
- Time stamp: the time corresponding to each heart rate measurement, with recordings taken at intervals during sleep.

The dataset spans several hours for each individual, providing detailed information on heart rate variability throughout the sleep cycle. The primary goal for this dataset is to forecast the heart rate for the next 2 minutes, based on the past 30 minutes of observed heart rate data.

2.1.3. Energy dataset

The energy dataset consists of hourly electricity consumption data from the Pennsylvania–New Jersey–Maryland (PJM) interconnection power grid, which serves the eastern United States. This dataset was sourced from the publicly available PJM Hourly Load Data repository and covers several years of data, making it well-suited for modeling long-term trends and seasonal patterns in electricity demand. The dataset includes the following variables:

- Total hourly load (MWh): the total electricity demand in megawatt-hours for each hour.
- Hour of day: the specific hour of the day the measurement was taken.
- Day of week: a categorical variable indicating the day of the week, as demand patterns vary significantly between weekdays and weekends.

The goal for this dataset is to predict electricity demand for the next hour based on the previous 24 hours of observed data. This type of forecasting is critical for grid operators to optimize electricity generation and balance supply with demand, especially as renewable energy sources are integrated into the power grid.

2.2. Preprocessing

Each dataset underwent specific preprocessing steps to ensure they were suitable for time series forecasting.

- Finance data preprocessing: stock prices were smoothed using a Savitzky-Golay filter, which applies a moving polynomial regression to the data. The filter used a window size of 5 data points and a polynomial order of 2, which ensured minimal distortion in the stock price trends. The data was then normalized using min-max normalization to ensure uniform scaling across all variables. Missing values were handled through forward fill techniques, where the most recent valid observation is carried forward to maintain the temporal sequence and ensure no gaps in the time series.
- Healthcare data preprocessing: the Pan-Tompkins algorithm was applied to the raw ECG signals to extract instantaneous heart rate by detecting R-peaks in the QRS complex [21]. Once the R-peaks were identified, the R-R intervals were used to calculate the instantaneous heart rate (bpm). To ensure evenly spaced time intervals, the heart rate data was interpolated using cubic spline interpolation, which maintains smooth transitions between data points. Outliers—defined as heart rate values below 30 bpm or above 200 bpm—were identified and removed to avoid physiological anomalies. Finally, the heart rate data was normalized using min-max normalization.
- Energy data preprocessing: the energy dataset was detrended using a Savitzky-Golay filter with a window size of 24 hours and a polynomial order of 2. This filter helped smooth out daily seasonality patterns while preserving important consumption trends. The data was then normalized using min-max normalization to ensure uniform scaling of electricity consumption values. Missing values were imputed using linear interpolation, where missing data points were estimated based on the values immediately before and after the gap. A rolling window technique was applied to capture seasonality and variations in demand patterns over time, allowing the models to adjust to recurring cycles in the data. Each dataset was split into training (70%), validation (15%), and test sets (15%). The validation set was used for hyperparameter tuning, while the test set was reserved for evaluating model performance.

2.3. Model implementation

Three forecasting models were implemented: ARIMA, LSTM, and TFT. Each model was fine-tuned for optimal performance based on the specific characteristics of the datasets (finance, healthcare, and energy). The hyperparameters for all models were selected based on grid search using the validation set.

2.3.1. Autoregressive integrated moving average model

The ARIMA model, a traditional statistical method for time series forecasting, has been widely used in various domains due to its simplicity and efficiency in handling linear relationships. The ARIMA model comprises three components: autoregression (AR), differencing (I), and moving average (MA). The AR component captures the linear relationship between past and current values, while the MA component models the dependency between the current value and past forecast errors. The I component involves differencing the time series to achieve stationarity.

In this study, the ARIMA model was configured by selecting the optimal parameters (p, d, and q) using the Box-Jenkins methodology. The Box-Jenkins methodology is a systematic approach used to identify, estimate, and diagnose ARIMA models. The process involves analyzing autocorrelation and partial autocorrelation functions (ACF and PACF) to determine the appropriate order of the AR and MA components, conducting stationarity tests to determine the differencing (I) order, and evaluating model diagnostics to ensure the model's adequacy. ACF and PACF plots were analyzed to determine the appropriate values for p and q, while the differencing (I) parameter d was selected based on stationarity tests. The hyperparameters were optimized by minimizing the Akaike information criterion (AIC), and the final values were selected based on the validation set, as detailed in Table 1.

2.3.2. Long short-term memory network

LSTM networks, a type of RNNs, have emerged as a powerful tool for time series forecasting due to their ability to capture long-term dependencies and non-linear patterns in sequential data. LSTMs partially address the vanishing gradient problem, a common issue in traditional RNNs, by incorporating memory cells and gating mechanisms that control the flow of information through the network. The LSTM architecture typically consists of multiple LSTM layers, each containing a set of LSTM units. Each LSTM unit comprises a memory cell, an input gate, an output gate, and a forget gate. The memory cell stores information over time, while the gates regulate the flow of information into and out of the cell. The input gate determines how much new information is stored in the cell, the output gate controls how much information is passed to the next layer, and the forget gate decides what information is discarded from the cell.

In this study, the LSTM architecture consisted of two LSTM layers followed by a dense output layer. Dropout regularization was applied to prevent overfitting, and the Adam optimizer was used to minimize the loss function. Hyperparameters such as the number of LSTM units, learning rate, dropout rate, and batch size were fine-tuned using grid search, with the final values selected based on the validation set, refer to Table 1 for details.

2.3.3. Temporal fusion transformer

We implemented the TFT for time series forecasting across finance, healthcare, and energy datasets. The model follows an encoder-decoder structure, incorporating specialized components such as variable selection networks (VSNs), gated residual networks (GRNs), and temporal self-attention [22]. The VSN dynamically selects relevant features at each time step. It processes inputs using fully connected layers, applies a sparsemax activation function to enforce sparsity, and assigns learned importance weights to each variable. This ensures that only the most relevant inputs contribute to forecasting.

The encoder processes historical inputs using LSTM networks, which capture temporal dependencies. The GRNs refine these encoded representations before passing them to subsequent layers. Each GRN consists of fully connected layers, exponential linear unit (ELU) activation functions, gating mechanisms (sigmoid activations), and residual connections to regulate information flow. GRNs are also used in the decoder to process known future covariates.

To capture long-range dependencies, the model applies masked interpretable multi-head self-attention to the encoded representations. This mechanism computes query (Q), key (K), and value (V) matrices from the LSTM outputs and learns temporal relationships without looking into the future (enforced via causal masking). The number of attention heads and dropout rates for this mechanism are listed in Table 1.

After self-attention, additional GRNs refine the learned representations before generating final forecasts. The output layer produces quantile estimates (10th, 50th, and 90th percentiles) to quantify uncertainty. Since TFT outputs quantile forecasts, the final point estimate was taken as the 50th percentile (median prediction), which represents the most likely outcome. For decision-making applications requiring risk assessment, the 10th and 90th percentiles can be used to provide lower and upper confidence bounds. The model was trained using the Adam optimizer, with a quantile loss function to optimize forecasts across different quantile levels. Hyperparameters such as the learning rate, batch size, number of GRN layers, hidden units, and dropout rate were optimized via grid search as shown in Table 1.

For each dataset, feature engineering was performed to prepare the data for the TFT model. In the finance dataset, raw stock prices (open, close, high, and low) and volume were directly used as observed historical inputs, with no static or known future inputs in this basic setup. For the healthcare dataset, the instantaneous heart rate was the primary observed historical input, while the time stamp was transformed into minute-of-day features for both observed historical and known future inputs. The energy dataset leveraged total hourly load as the core observed historical input, alongside hour-of-day and day-of-week features for both historical and future contexts. In all datasets, time stamps were converted into relevant temporal features to capture seasonality and cyclical patterns. Furthermore, categorical variables, like day-of-week, were one-hot encoded to facilitate model processing.

2.4. Model training and validation

All forecasting models were trained using the training subset, while hyperparameter optimization was performed on the validation subset. To ensure realistic time-series forecasting and prevent information leakage, rolling window cross-validation was employed across all datasets, including finance, healthcare, and energy [23]. Unlike conventional random sampling strategies, rolling window validation preserves temporal ordering by ensuring that future observations are never used during model training. This procedure more closely reflects real-world forecasting conditions, where predictions must rely solely on historical information.

For each dataset, the observations were first chronologically divided into training (70%), validation (15%), and testing (15%) subsets. The training set was used to fit the models, while the validation set supported hyperparameter optimization and model selection. The testing set remained completely unseen during model development and was reserved exclusively for final performance assessment. Maintaining chronological separation ensured preservation of temporal dependencies and prevented future observations from influencing model construction.

To further improve robustness and reduce dependence on a single temporal split, an expanding rolling window cross-validation strategy was applied during training and validation. In this approach, the training window progressively increased in size at each iteration while validation was conducted on the immediately following unseen observations. For example, for the energy dataset, the first iteration used samples T1–T1000 for training and T1001–T1100 for validation. In the next iteration, the training window expanded to T1–T1100 while the validation period shifted to T1101–T1200. This procedure was repeated across successive windows until the end of the training-validation period was reached. The same strategy was applied to the financial and healthcare datasets. The expanding-window design enabled repeated evaluation on future unseen observations and provided a more reliable estimate of model generalization performance under practical forecasting conditions.

Hyperparameter optimization was conducted using grid search across predefined parameter ranges shown in Table 1. For ARIMA, combinations of autoregressive (p), differencing (d), and moving average (q) parameters were systematically explored and evaluated using the AIC. For LSTM and TFT, hyperparameters including learning rate, hidden layer size, batch size, dropout rate, and network architecture parameters were evaluated across candidate values. The optimal configuration was selected based on validation performance, primarily minimizing MAE and MAPE while maximizing R^2.

To improve statistical robustness and reduce variability arising from stochastic training processes, each deep learning experiment (LSTM and TFT) was repeated multiple times using different random initialization seeds, and average performance metrics were reported. This approach reduced sensitivity to random weight initialization and training fluctuations. Furthermore, early stopping was implemented to prevent overfitting, terminating training if validation loss failed to improve over ten consecutive epochs. After optimal hyperparameters had been identified, models were retrained using the combined training and validation datasets to maximize data utilization. Finally, model performance was evaluated on the held-out test dataset, which constituted the final 15% of observations and was not involved in either model training or hyperparameter selection. This strict separation ensured unbiased assessment of predictive performance and enhanced the reproducibility and reliability of the reported results.

2.5. Sensitivity analysis of missing data

To assess the sensitivity of our models to missing data, we conducted a comprehensive analysis across all three datasets: financial, healthcare, and energy. We recognized that the impact of missing data can vary significantly depending on the specific characteristics of each dataset and the mechanisms by which data becomes missing. Therefore, we tailored our approach to reflect these nuances.

In the financial dataset, where missing data might arise from infrequent events like trading halts or data errors, we introduced lower levels of missingness, ranging from 1% to 5%. Conversely, in the healthcare dataset, where missing data could be more prevalent due to factors such as sensor malfunctions or patient movements, we explored a broader range of missingness, from 5% to 20%. For the energy dataset, where missing data might stem from issues like communication disruptions or meter failures, we examined a moderate range of 0.5% to 2% missingness. For each dataset, we simulated both random missingness, where data is MCAR, and missingness based on specific criteria (MAR).

For MAR, in the finance dataset, missing values were introduced on low-volume trading days based on the assumption that during low-volume days, stock price data might be incomplete due to less market activity. For the healthcare dataset, the missing heart rate values were introduced during periods of high heart rate. For the energy dataset, missing values were introduced more frequently during peak demand hours as real-world energy systems data collection errors may be correlated with high energy usage periods. This allowed us to investigate how different missingness mechanisms might affect the performance of our forecasting models. To further enhance the comprehensiveness of our analysis, we employed three distinct imputation techniques: forward fill, linear interpolation, and k-NN imputation. These techniques represent a range of complexity in handling missing data, from simple replacement with previous values to more sophisticated estimations based on neighboring data points.

2.6. Evaluation metrics

The models were evaluated on their forecasting accuracy using the following metrics:
- Mean absolute error (MAE): this measures the average magnitude of errors in the predictions.
- Mean absolute percentage error (MAPE): this measures the error as a percentage of the actual value, providing a normalized measure of accuracy.
- R-squared (R^2): this quantifies how well the model explains the variance in the dataset.

Table 1. Hyperparameter selection

Model	Hyperparameter	Range/value	Selected value (validation set)
ARIMA	p (autoregressive term)	[0, 1, 2, 3, 4, 5]	2
	d (differencing term)	[0, 1, 2]	1
	q (moving average term)	[0, 1, 2, 3, 4, 5]	2
	AIC	Optimized	-
LSTM	Number of LSTM units	[50, 100, 150, 200]	100
	Dropout rate	[0.1, 0.2, 0.3, 0.4]	0.2
	Learning rate	[0.001, 0.005, 0.01]	0.001
	Batch size	[16, 32, 64]	32
	Number of epochs	[50, 100, 150, 200]	100
	Optimizer	Adam	Adam
TFT	Number of attention heads	[2, 4, 8]	4
	Hidden layer size	[16, 32, 64, 128]	32
	Number of GRN layers	[1, 2, 3]	2
	Dropout rate	[0.1, 0.2, 0.3, 0.4, 0.5]	0.2
	Learning rate	[0.0005, 0.001, 0.005, 0.01]	0.001
	Batch size	[16, 32, 64]	32
	Number of epochs	[50, 100, 150, 200]	100
	Optimizer	Adam	Adam

3. RESULTS

3.1. Financial dataset results

In the financial domain, the objective was to forecast the closing price of the S&P 500 index using historical stock price data. The evaluation metrics for ARIMA, LSTM, and TFT models on the test set are summarized in Table 2. The ARIMA model showed reasonable performance in predicting the S&P 500 index, but its ability to capture non-linear patterns was limited. The model achieved a MAE of 15.2 USD and a MAPE of 1.9%, indicating that the predictions were within approximately 2% of the actual values. However, the R^2 value of 0.72 showed that the model explained 72% of the variance in the data.

The LSTM model, due to its ability to capture long-term dependencies and non-linear relationships, significantly outperformed ARIMA. It achieved a MAE of 10.1 USD and a MAPE of 1.3%. The R^2 value was 0.87, indicating that the LSTM model explained 87% of the variance in the stock prices. The TFT model demonstrated the best performance, leveraging attention mechanisms to capture both short- and long-term dependencies. It achieved the lowest MAE of 8.6 USD and a MAPE of 1.1%, with an R^2 of 0.91, showing that the model was able to explain 91% of the variance in stock prices, outperforming both ARIMA and LSTM.

3.2. Healthcare dataset results

For the healthcare dataset, the task was to predict future heart rate values from ECG data during sleep. Table 2 presents the performance of the ARIMA, LSTM, and TFT models on the test set. The ARIMA model was limited in capturing the non-linear variations in heart rate patterns. The model resulted in a MAE of 6.3 bpm and a MAPE of 4.7%, with an R^2 value of 0.68. This indicated that the model explained 68% of the variance in heart rate data.

The LSTM model, due to its ability to handle sequential and non-linear data, performed better than ARIMA. It achieved a MAE of 4.4 bpm and a MAPE of 3.1%, with an R^2 of 0.81. The TFT model outperformed the other models in forecasting heart rate, with a MAE of 3.8 bpm and a MAPE of 2.9%. The model achieved an R^2 value of 0.85, demonstrating that it captured more variance in the heart rate time series compared to ARIMA and LSTM.

3.3. Energy dataset results

For the energy dataset, the goal was to predict hourly electricity demand. The performance metrics for each model on the test set are provided in Table 2. The ARIMA model showed decent performance, with a MAE of 150 MWh and a MAPE of 2.7%. The R^2 value of 0.76 indicated that the model explained 76% of the variance in electricity demand.

The LSTM model demonstrated better performance, achieving a MAE of 110 MWh and a MAPE of 1.9%. The R^2 value was 0.85, showing a higher ability to capture patterns in electricity demand compared to ARIMA. The TFT model provided the best results, with a MAE of 95 MWh and a MAPE of 1.5%. The model achieved an R^2 value of 0.89, indicating that it captured almost 90% of the variance in the data, outperforming both ARIMA and LSTM.

Table 2. Model performance on financial, healthcare, and energy datasets (test set results)

Model	Dataset	MAE	MAPE (%)	R^2
ARIMA	Financial (USD)	15.2	1.9	0.72
LSTM	Financial (USD)	10.1	1.3	0.87
TFT	Financial (USD)	8.6	1.1	0.91
ARIMA	Healthcare (bpm)	6.3	4.7	0.68
LSTM	Healthcare (bpm)	4.4	3.1	0.81
TFT	Healthcare (bpm)	3.8	2.9	0.85
ARIMA	Energy (MWh)	150	2.7	0.76
LSTM	Energy (MWh)	110	1.9	0.85
TFT	Energy (MWh)	95	1.5	0.89

3.4. Overall results

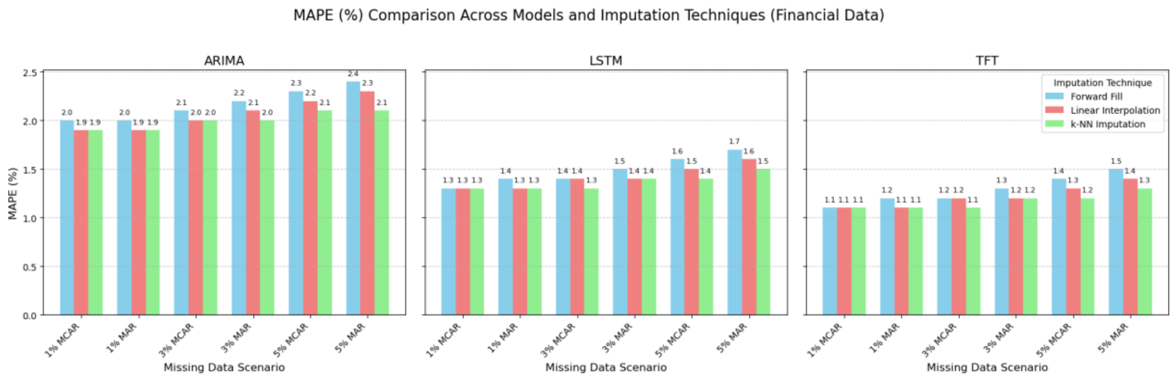
Across all datasets, the TFT consistently outperformed the ARIMA and LSTM models. This can be attributed to TFT's ability to capture both short- and long-term dependencies in the data, as well as its efficient use of attention mechanisms. The LSTM model showed better performance than ARIMA in most cases, particularly when dealing with non-linear and complex time series data, such as heart rate and electricity demand. These results suggest that while ARIMA remains a reliable model for simpler, linear time series, deep learning models such as LSTM and TFT are better suited for more complex patterns. In particular, TFT's superior performance highlights its potential as a state-of-the-art solution for time series forecasting in various domains.

3.5. Sensitivity analysis to missing data

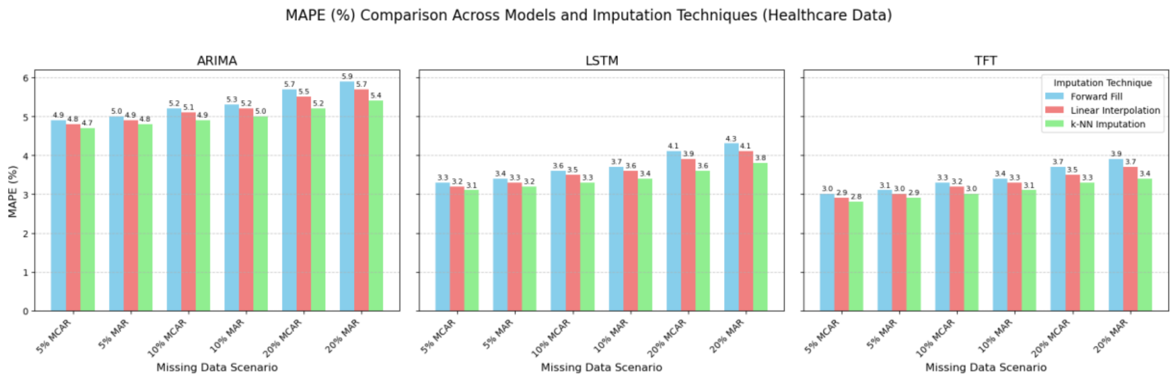
Results of the sensitivity analysis across all three datasets (financial, healthcare, and energy) are presented in Figure 1. These figures detail the MAPE performance metrics for the ARIMA, LSTM, and TFT models under different missing data scenarios and imputation techniques. As expected, the performance of all models degrades as the percentage of missing data increases. However, the extent of degradation varies across models and datasets.

The TFT model consistently demonstrates greater robustness to missing data compared to ARIMA and LSTM across all datasets, particularly in scenarios with higher levels of missingness. This suggests that TFT's ability to capture complex temporal dependencies and utilize attention mechanisms makes it less susceptible to the negative impact of missing data. The choice of imputation technique influences model performance. In general, k-NN imputation yields better results than forward fill and linear interpolation, especially for LSTM and TFT. This indicates that more sophisticated imputation techniques can help mitigate the adverse effects of missing data on forecasting accuracy.

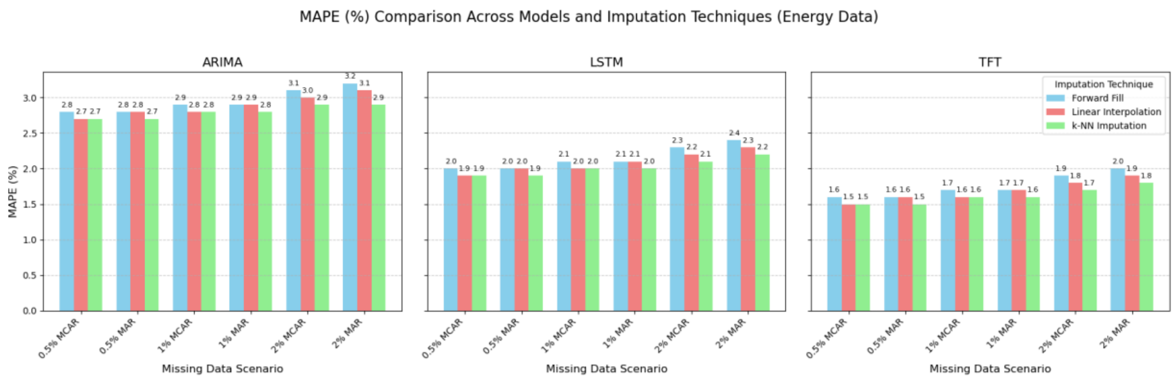
For the financial dataset as shown in Figure 1(a), the models are generally robust to low levels of missing data (up to 1%). As missingness increases, ARIMA's performance degrades more significantly than LSTM and TFT. For the healthcare dataset, in Figure 1(b), TFT exhibits the greatest robustness to missing data, maintaining relatively good performance even with 20% missingness. LSTM shows moderate sensitivity, while ARIMA is the most affected. For the energy dataset, as shown in Figure 1(c), similar to the financial dataset, the models are robust to low levels of missingness. As missingness increases, ARIMA's performance declines more rapidly compared to LSTM and TFT.



(a)



(b)



(c)

Figure 1. Performance comparison of ARIMA, LSTM, and TFT models under varying missing-data conditions and imputation techniques across three application domains in (a) financial dataset, (b) healthcare dataset, and (c) energy dataset. Forecasting performance is evaluated using MAPE under both MCAR and MAR scenarios with forward fill, linear interpolation, and k-NN imputation. Across all datasets, forecasting accuracy generally decreases as missingness increases; however, TFT demonstrates greater robustness under increasing missing-data levels, particularly under MAR conditions and higher missingness rates

When comparing the impact of MCAR versus MAR missingness mechanisms, we observe a general trend where MAR scenarios lead to slightly worse model performance across all datasets. This suggests that when missing values are dependent on observed data, the imputation process becomes more challenging, potentially leading to less reliable estimates. The TFT model remains the most robust under MAR conditions, reinforcing its ability to adapt to more complex missingness structures. Conversely, ARIMA is the most sensitive to MAR, with performance deteriorating significantly as the missing rate increases.

4. DISCUSSION

Time series forecasting is a crucial task in multiple domains, including finance, healthcare, and energy. Accurate predictions of future trends in these areas enable better decision-making, risk management, and operational efficiency. For instance, predicting stock prices helps investors make informed trading decisions, forecasting heart rate can assist in detecting anomalies related to sleep disorders, and predicting energy demand is vital for optimizing power grids. However, the inherent complexity of these datasets, characterized by non-linearity, seasonality, and long-term dependencies, makes forecasting a challenging task. Traditional linear models like ARIMA often fall short in capturing these intricate patterns, necessitating the use of more advanced deep learning methods.

4.1. Summary of findings

In this study, we compared the performance of three forecasting models: ARIMA, LSTM, and TFT, across three diverse datasets (financial, healthcare, and energy). Our findings show that while ARIMA performed adequately in simpler cases, it was significantly outperformed by the deep learning models (LSTM and TFT), particularly in datasets with non-linear dependencies. The ARIMA model yielded the highest errors and lowest explained variance, with MAE values of 15.2 USD for the financial dataset, 6.3 bpm for healthcare, and 150 MWh for energy, alongside relatively low R^2 values. LSTM, as expected, outperformed ARIMA with significantly lower errors across all datasets, demonstrating its strength in handling sequential and non-linear data. For example, LSTM reduced the MAE to 10.1 USD for financial data and 4.4 bpm for healthcare. Finally, the TFT model delivered the best performance, demonstrating superior accuracy and the ability to capture both short- and long-term dependencies. TFT achieved the lowest MAE of 8.6 USD for financial data, 3.8 bpm for healthcare, and 95 MWh for energy, with R^2 values exceeding 0.85 across all domains.

4.2. Comparison with literature

Our findings align with previous research, which consistently demonstrates the limitations of ARIMA in dealing with non-linear and complex time series. Studies comparing ARIMA with deep learning models in stock market predictions, such as those by Siami-Namini *et al*. [24], [25], report similar performance gaps, with deep learning models like LSTM and BiLSTM reducing error rates by 80% to 90% compared to ARIMA. In another study, Siami-Namini *et al*. [26] demonstrated that LSTM significantly outperformed ARIMA in financial time series forecasting, reducing errors by about 80% across various financial datasets, including stock indices and economic indicators. In the healthcare domain, LSTM has been shown to outperform traditional models in classifying physiological signals like ECG, as demonstrated in the work by Yildirim [3], where an LSTM-based model achieved over 99% accuracy in classifying ECG signals, significantly outperforming traditional methods such as support vector machines (SVM).

The TFT, a relatively recent addition to time series forecasting, has demonstrated superior performance across multiple domains. Lim *et al*. [22] showed that TFT outperforms traditional models and advanced recurrent architectures like LSTM, particularly in energy demand forecasting and multi-horizon sales prediction, due to its ability to capture both short- and long-term dependencies. Similarly, Niu *et al.* [27] confirmed TFT's exceptional performance in wind power forecasting, where it significantly outperformed LSTM and traditional models, further proving its effectiveness in handling complex, non-linear relationships in energy datasets. Our results, particularly the improved accuracy and lower MAE values in all datasets, reinforce the advantages of TFT in dealing with time series containing long-term dependencies and complex patterns.

4.3. Strengths, limitations, and future directions

This study presents several strengths, most notably its comprehensive comparison of three forecasting models (ARIMA, LSTM, and TFT) across distinct domains such as finance, healthcare, and energy. By testing these models on diverse real-world datasets, we provide valuable insights into how traditional and advanced deep learning approaches perform in different contexts. In particular, the inclusion of the TFT model, which is relatively new and has been less frequently applied in some areas, demonstrates its ability to outperform both ARIMA and LSTM in capturing complex time series patterns. The rigorous methodology, including the use of rolling window cross-validation, ensures that future data is not leaked into the training process, thus offering a realistic assessment of model performance that is highly relevant for practical applications where forecasting accuracy is critical. In addition, the impact of missing data and imputation techniques was studied.

Beyond forecasting accuracy, the observed robustness of TFT under incomplete data conditions may have important implications for large-scale information and communication technology (ICT) systems and enterprise applications. In practical environments, time-series data are increasingly generated through distributed sensors, internet-of-things (IoT) infrastructures, cloud services, and big-data platforms, where missing values frequently arise because of communication failures, latency, synchronization issues, and sensor interruptions. The ability of TFT to maintain performance under missing-data conditions suggests strong potential for deployment within cloud-based analytics pipelines and large-scale forecasting systems. Integration with distributed computing frameworks and big-data platforms could enable real-time forecasting applications in smart healthcare systems, financial monitoring environments, intelligent energy management platforms, and enterprise decision-support systems. This capability increases the practical relevance of TFT beyond experimental settings and highlights its potential role in scalable industrial forecasting solutions.

Despite these strengths, the study has certain limitations. ARIMA, while included for comparison, is somewhat outdated compared to modern deep learning models and its lower performance was expected. LSTM, though effective, lacks the flexibility and capacity of TFT, especially in capturing long-range dependencies efficiently. Furthermore, the computational cost of deep learning models, particularly LSTM and TFT, is a notable drawback, as they require significant resources, especially when fine-tuning hyperparameters through methods like grid search.

Looking ahead, there are several avenues for future research. Although the datasets used in this study span multiple domains and provide a broad evaluation framework, they primarily consist of benchmark datasets commonly used for algorithmic evaluation. While benchmark datasets facilitate reproducibility and standardized comparison, they may not fully represent the complexity and variability encountered in operational environments. Real-world time-series systems frequently exhibit domain-specific characteristics such as irregular sampling frequencies, evolving data distributions, heterogeneous sensor sources, operational disruptions, and varying missing-data patterns. Therefore, the generalizability of the proposed findings should be interpreted with caution. Future studies should validate these forecasting approaches using sector-specific real-world datasets from industrial environments, healthcare monitoring systems, smart grids, financial institutions, and large-scale enterprise platforms to better assess robustness and practical deployment potential under real operational conditions. Moreover, future work could explore the generalization of these models to more challenging scenarios, such as highly imbalanced or noisy datasets, which remain difficult for deep learning approaches [28].

While this study focused on ARIMA, LSTM, and TFT models, future research could explore a wider range of time series forecasting models. For ARIMA, this could involve investigating variations like SARIMA for seasonality and ARIMAX for incorporating external factors. Additionally, comparing ARIMA to other traditional models like Exponential Smoothing and Holt-Winters could provide valuable insights. In the context of deep learning, exploring different LSTM variations, such as Bi-LSTM and GRU, could further enhance performance. Comparing LSTM to other deep learning models like CNNs and TCNs could also reveal their relative strengths and weaknesses for time series forecasting. For transformer models, investigating variations like transformer-XL and longformer could improve efficiency and address specific challenges. Comparing transformers to other attention-based models like Informer could also be beneficial. It is important to note that our implementation of TFT was somewhat limited in its use of static and known future inputs. While the finance dataset relied solely on observed historical inputs, the healthcare and energy datasets incorporated limited known future covariates, such as time-based features. However, we did not fully leverage the potential of static variables or broader known future inputs, which could enhance TFT's interpretability and performance. Future work could investigate the impact of incorporating a richer set of these inputs on forecasting accuracy.

Another important direction is addressing the lack of interpretability in models like LSTM and TFT. While these models perform exceptionally well, their "black-box" nature makes it difficult for practitioners to understand the underlying mechanisms driving the predictions [22]. Future work could focus on enhancing model interpretability, potentially by incorporating explainability techniques that would make these models more transparent and accessible, especially in sensitive fields like healthcare [29].

Finally, exploring ensemble methods that combine the strengths of ARIMA, LSTM, and TFT could offer a promising path forward [30]. Hybrid models that utilize ARIMA's linearity for simpler patterns and leverage deep learning models' ability to capture complex non-linear trends could yield even better results. Transfer learning and domain adaptation are also areas worth exploring, especially for applications where labeled data is scarce or expensive to collect [31].

5. CONCLUSION

In this paper, we conducted a comprehensive comparison of three time series forecasting models: ARIMA, LSTM, and TFT, across three distinct datasets: finance, healthcare, and energy. Our results clearly demonstrate the limitations of traditional methods like ARIMA in capturing complex, non-linear dependencies in time series data. In contrast, deep learning models, particularly TFT, significantly outperformed ARIMA and LSTM, showcasing their ability to handle both short-term fluctuations and long-term trends. Furthermore, this study provides valuable insights into the impact of missing data on time series forecasting and the effectiveness of various imputation techniques in mitigating these effects. Our sensitivity analysis revealed that the performance of all models degrades with increasing levels of missing data, but TFT exhibited greater robustness compared to ARIMA and LSTM.

The study's findings emphasize the growing importance of advanced deep learning architectures in diverse real-world applications. The superior performance of TFT across all datasets suggests its suitability for time series forecasting tasks where both accuracy and the ability to model intricate dependencies are paramount. However, challenges remain, particularly in terms of the interpretability and computational cost of deep learning models. Future work should focus on addressing these issues and exploring the potential of hybrid approaches and ensemble methods that could further enhance forecasting accuracy.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Maryam Hosseini	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Mohamad Forouzanfar	✓	✓					✓			✓		✓	✓	✓

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no potential conflicts of interest related to this work.

INFORMED CONSENT

This research utilized de-identified data from the PhysioNet database (https://physionet.org/content/apnea-ecg/1.0.0/), which was originally collected under appropriate institutional review board (IRB) approval and adheres to all ethical guidelines for research involving human participants. As the dataset is publicly available and fully de-identified, no additional IRB approval was required for its use in this study. Informed consent was obtained from participants during the original data collection process.

DATA AVAILABILITY

This research utilized de-identified data from the PhysioNet database available at: https://physionet.org/content/apnea-ecg/1.0.0/.

REFERENCES

[1] H. Liu and Z. Long, "An improved deep learning model for predicting stock market price time series," *Digital Signal Processing: A Review Journal*, vol. 102, p. 102741, 2020, doi: 10.1016/j.dsp.2020.102741.

[2] B. Balasingam, P. Willett, G. Levchuk, and J. Freeman, "Exploratory joint and separate tracking of geographically related time series," in *Proceedings of SPIE: Signal and Data Processing of Small Targets 2012*, 2012, pp. 205–219, doi: 10.1117/12.919980.

[3] Ö. Yildirim, "A novel wavelet sequences based on deep bidirectional LSTM network model for ECG signal classification," *Computers in Biology and Medicine*, vol. 96, pp. 189–202, 2018, doi: 10.1016/j.compbiomed.2018.03.016.

[4] Q. Chen, H. Li, and X. Zheng, “A deep neural network for operator learning enhanced by attention and gating mechanisms for long-time forecasting of tumor growth,” *Engineering with Computers*, vol. 41, no. 1, pp. 423–533, 2025, doi: 10.1007/s00366-024-02003-0.

[5] A. Moradzadeh, S. Zakeri, M. Shoaran, B. Mohammadi-Ivatloo, and F. Mohammadi, “Short-term load forecasting of microgrid via hybrid support vector regression and long short-term memory algorithms,” *Sustainability*, vol. 12, no. 17, p. 7076, 2020, doi: 10.3390/su12177076.

[6] A. D. Piazza, M. C. D. Piazza, G. L. Tona, and M. Luna, “An artificial neural network-based forecasting model of energy-related time series for electrical grid management,” *Mathematics and Computers in Simulation*, vol. 184, pp. 294–305, 2021, doi: 10.1016/j.matcom.2020.05.010.

[7] G. Varelas, G. Tzimas, and P. Alefragis, “Forecasting electricity demand in Greece: a functional data approach in high dimensional hourly time series,” *SN Computer Science*, vol. 5, no. 5, p. 566, 2024, doi: 10.1007/s42979-024-02926-x.

[8] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*, 5th ed. Hoboken, NJ: John Wiley & Sons, 2015.

[9] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, “A comprehensive overview and comparative analysis on deep learning models,” *Journal on Artificial Intelligence*, vol. 6, no. 1, pp. 301–360, 2024, doi: 10.32604/jai.2024.054314.

[10] L. Benavides-Cesar, M. Á. Manso-Callejo, and C. I. Cira, “Methodology based on BERT (bidirectional encoder representations from transformers) to improve solar irradiance prediction of deep learning models trained with time series of spatiotemporal meteorological information,” *Forecasting*, vol. 7, no. 1, pp. 1–21, 2025, doi: 10.3390/forecast7010005.

[11] S. B. Dhal and D. Kar, “Transforming agricultural productivity with AI-driven forecasting: innovations in food security and supply chain optimization,” *Forecasting*, vol. 6, no. 4, pp. 925–951, 2024, doi: 10.3390/forecast6040046.

[12] M. Frikha, K. Taouil, A. Fakhfakh, and F. Derbel, “Predicting power consumption using deep learning with stationary wavelet,” *Forecasting*, vol. 6, no. 3, pp. 864–884, 2024, doi: 10.3390/forecast6030043.

[13] P. Schlieper, M. Dombrowski, A. Nguyen, D. Zanca, and B. Eskofier, “Data-centric benchmarking of neural network architectures for the univariate time series forecasting task,” *Forecasting*, vol. 6, no. 3, pp. 718–747, 2024, doi: 10.3390/forecast6030037.

[14] M. Bahrami and M. Forouzanfar, “Deep learning forecasts the occurrence of sleep apnea from single-lead ECG,” *Cardiovascular Engineering and Technology*, vol. 13, no. 6, pp. 809–815, 2022, doi: 10.1007/s13239-022-00615-5.

[15] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, “Deep learning for time series forecasting: a survey,” *Big Data*, vol. 9, no. 1, pp. 3–21, 2021, doi: 10.1089/big.2020.0159.

[16] M. Hosseini, M. de Zambotti, F. C. Baker, and M. Forouzanfar, “Multistream LSTM for artifact detection in impedance cardiography,” *IEEE Sensors Letters*, vol. 9, no. 5, p. 6001104, 2025, doi: 10.1109/LSENS.2025.3561688.

[17] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool, “Transformers in time-series analysis: a tutorial,” *Circuits, Systems, and Signal Processing*, vol. 42, no. 12, pp. 7433–7466, 2023, doi: 10.1007/s00034-023-02454-8.

[18] A. Mahmoud and A. Mohammed, “Leveraging hybrid deep learning models for enhanced multivariate time series forecasting,” *Neural Processing Letters*, vol. 56, no. 5, p. 223, 2024, doi: 10.1007/s11063-024-11656-3.

[19] H. K. Wang, X. Zhang, H. Long, S. Yao, and P. Zhu, “W-FENet: wavelet-based fourier-enhanced network model decomposition for multivariate long-term time-series forecasting,” *Neural Processing Letters*, vol. 56, no. 2, p. 43, 2024, doi: 10.1007/s11063-024-11478-3.

[20] T. Penzel and G. Moody, “The apnea-ECG database,” in *Proceedings of Computers in Cardiology 2000*, 2000, pp. 255–258.

[21] M. A. Z. Fariha, R. Ikeura, S. Hayakawa, and S. Tsutsumi, “Analysis of pan-tompkins algorithm performance with noisy ECG signals,” in *Journal of Physics: Conference Series*, vol. 1532, 2020, p. 012022, doi: 10.1088/1742-6596/1532/1/012022.

[22] B. Lim, S. Arık, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021, doi: 10.1016/j.ijforecast.2021.03.012.

[23] V. Cerqueira, L. Torgo, and I. Mozetič, “Evaluating time series forecasting models: an empirical study on performance estimation methods,” *Machine Learning*, vol. 109, no. 11, pp. 1997–2028, 2020, doi: 10.1007/s10994-020-05910-7.

[24] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BiLSTM,” *arXiv:*1911.09512, 2019.

[25] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “The performance of LSTM and BiLSTM in forecasting time series,” in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 3285–3292, doi: 10.1109/BigData47090.2019.9005997.

[26] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A comparison of ARIMA and LSTM in forecasting time series,” in *Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 1394–1401, doi: 10.1109/ICMLA.2018.00227.

[27] Z. Niu, X. Han, D. Zhang, Y. Wu, and S. Lan, “Interpretable wind power forecasting combining seasonal-trend representations learning with temporal fusion transformers architecture,” *Energy*, vol. 306, p. 132482, 2024, doi: 10.1016/j.energy.2024.132482.

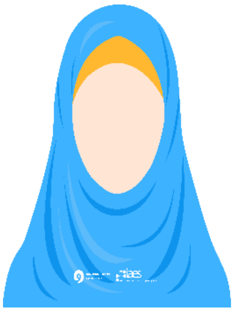
[28] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018, doi: 10.1038/s41598-018-24271-9.

[29] A. A. Ismail, M. Gunady, H. C. Bravo, and S. Feizi, “Benchmarking deep learning interpretability in time series predictions,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6441–6452, 2020.

[30] R. Kumar, P. Kumar, and Y. Kumar, “Three stage fusion for effective time series forecasting using Bi-LSTM-ARIMA and improved DE-ABC algorithm,” *Neural Computing and Applications*, vol. 34, no. 21, pp. 18421–18437, 2022, doi: 10.1007/s00521-022-07431-x.

[31] R. Ye and Q. Dai, “Implementing transfer learning across different datasets for time series forecasting,” *Pattern Recognition*, vol. 109, p. 107617, 2021, doi: 10.1016/j.patcog.2020.107617.

BIOGRAPHIES OF AUTHORS



Maryam Hosseini received her B.Sc. in Electrical Engineering from Kashan University and her M.Sc. in Biomedical Engineering from K.N. Toosi University of Technology. She has worked as a Research Intern at École de technologie supérieure (ÉTS), Montréal. Her research interests include biomedical signal processing, brain-computer interfaces (BCI), non-invasive physiological monitoring, and deep learning for time-series analysis. She can be contacted at email: maryamy2k876@gmail.com

Mohamad Forouzanfar received the Ph.D. degree in Electrical and Computer Engineering from the University of Ottawa. Following his doctoral studies, he completed Postdoctoral fellowships with Stanford University and Harvard University. He previously a Research Scientist within the Human Sleep Research Laboratory, SRI International. He is currently an Associate Professor with the École de Technologie Supérieure (ETS), Université du Québec. His research interests include development of advanced instrumentation, signal and image processing, and machine learning, specifically as they apply to non-invasive physiological monitoring for sleep and cardiovascular health. He is an Associate Editor-in-Chief for IEEE Transactions on Instrumentation and Measurement. He can be contacted at email: mohamad.forouzanfar@etsmtl.ca.