BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **25**, No 2
Sofia • 2025 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2025-0017

Improving Forecasting Performance for Abnormal Time Series Data with the TFT-TPE Integrated Model and Google Trends

Ngo Van Son[1,2], *Vo Viet Minh Nhat*[3]

[1]*University of Sciences, Hue University, Vietnam*
[2]*School of Hospitality and Tourism – Hue University, Vietnam*
[3]*Institute for Educational Testing and Quality Assurance – Hue University, Vietnam*
E-mails: ngovanson@hueuni.edu.vn vvmnhat@hueuni.edu.vn

Abstract**:** *Forecasting is essential in manufacturing and business, but is hindered by abnormal events like COVID-19. This paper proposes a model that integrates Temporal Fusion Transformer* (*TFT*) *with Tree-Structured Parzen Estimator* (*TPE*), *in which TFT is a deep neural network specifically designed for processing time series data to capture trends and model complex data variations and, at the same time, TPE is an optimization technique that uses a tree-like data structure to determine the best set of hyperparameters for TFT. The TFT-TPE integrated model, therefore, provides an effective solution to the forecasting problem, especially for abnormal data. The study proposes a combination of forecasting historical data, considering the COVID-19 period, and utilizing Google Trends to enhance forecasting accuracy. The experimental results show that the TFT-TPE integrated model achieves forecasting results better than traditional forecasting models, especially the ability to overcome the anomalies in time series data.*

Keywords**:** *Transformer, TFT-TPE integration, Abnormal time series data, Google Trends, Tourism demand forecasting.*

1. Introduction

Time series forecasting is a topic of interest in many research fields, from weather forecasting, climate change, economic growth, stock market volatility, to tourism demand. Forecasting is making short-term and long-term assumptions based on knowledge of past events. For time series data forecasting, researchers often analyze the given data series to find out the characteristics and then choose the appropriate technique for forecasting [1]. However, accurate forecasting becomes challenging to achieve when abnormal data patterns appear. Abnormal data can be sudden fluctuations or unexpected changes in trends in the data series, making traditional forecasting techniques less effective [2]. Different things, such as market shocks, disease outbreaks, data collection errors, or strange human behavior, can cause anomalies. When faced with abnormal data, traditional forecasting methods fail to capture outliers, leading to inaccurate forecasts. Traditional forecasting methods are often based on assumptions of stationarity and regularity, which do not account for anomalies in time series data. Abnormal data pose significant challenges to forecasters due to their unpredictable nature [3].

For machine learning-based time series forecasting approaches, an indispensable and crucial step during training is to initialize the initial values for parameters such as the number of epochs, classes, learning rate, etc. The selection of appropriate parameters is often based on experience [4]. For each set of parameters, we have to train the model, observe the results, evaluate them, readjust the parameters, and repeat. In order to automate the above process, some algorithms, such as Grid Search or Random Search [5], can be used. However, these algorithms only work effectively with a small set of parameters because the search space will increase rapidly when the number of parameters is significant, and it takes a lot of time to find a suitable solution. The Tree-structured Parzen Estimator (TPE) hyperparameter optimization algorithm [6] can help overcome this obstacle by building a model that estimates the conditional distribution of hyperparameters based on collected samples.

This paper proposes an integrated model to improve the forecasting accuracy of abnormal events, such as tourism demand during the COVID-19 pandemic. The integrated model is based on Temporal Fusion Transformer (TFT) [7], a powerful tool for time series forecasting with accurate forecasting ability and adaptability to abnormal data fluctuations, and TPE, a hyperparameter optimization method for TFT inputs. By integrating TFT with TPE, the TFT-TPE model can improve the forecasting performance. We evaluate the performance of the TFT-TPE model with tourism demand data, especially with abnormal data appearing during the COVID-19 pandemic. Another proposal introduced in the paper to improve forecasting performance is to combine historical tourist demand data with Google Trends. Experimental results show that the TFT-TPE model achieves accuracy higher than traditional forecasting models and can overcome the anomalies in time series data.

The main contributions of the paper include:

- Proposing an integrated TFT-TPE model and modelling the tourism demand forecasting problem as a problem of finding the TFT architecture with an optimal set of hyperparameters;
- Proposing a combination of historical data and Google Trends for forecasting, in which the collection of keyword-based search data is analyzed; This time series data is then used as a covariate to feed into the TFT-TPE model, which can learn and forecast more accurately.
- Normalizing and transforming time series data on tourism demand (international arrivals to Vietnam and the UK) before and during the COVID-19 pandemic (abnormal data) for training the TFT-TPE model; and
- Evaluating the tourism demand forecasting performance of the TFT-TPE model with the normalized time series data set.

The next sections of the paper are organized as follows: Section 2 summarizes and reviews related works in recent years, focusing on transformer-based forecasting models and their variants. Based on these analyses, Section 3 describes the TFT-TPE integrated model in terms of the structure and operation of both the TFT model and the TPE optimization method. Section 4 describes the implementation, input data processing, and evaluation of forecasting results with data before and during COVID-19. Finally, the conclusion is presented in Section 5.

2. Related works

Forecasting abnormal data is challenging in various domains, such as e-commerce, medical supplies, electricity consumption, pharmaceutical manufacturing, and tourism demand. Several different approaches have been proposed to address this problem. This section reviews recent research related to forecasting for abnormal time series data.

Liu, Ming and Huang [8] studied the demand forecasting of medical devices with sparse, temporary, and irregular data. In this study, the authors proposed to combine historical data statistics with linear regression. In order to reduce the sparse estimates, historical data transformation was added to the linear regression model. However, based on statistics and linear regression techniques, this proposal has difficulty capturing the complex nonlinear relationships contained in the demand forecasting data of medical devices. Unlike deep learning models, which can automatically learn complex patterns and features from data, the study in [8] requires manual feature extraction and model specification, which limits the adaptability to multivariate forecasting.

In Kalifa et al. [9], hypothesized that leveraging external knowledge, such as in world events, can help improve forecasting under abnormal conditions. This study exploited a 40-year archive of world events. Then, Kalifa et al. [9] proposed a new method based on the Transformer to construct daily data based on the relationship of intraday events. This data is then used to forecast future consumer behavior. An e-commerce sales dataset from eBay is experimented with using the proposed method. The results show that the proposed method predicts more accurately than statistical methods such as ARIMA or deep learning methods such as Prophet [10].

El-Hadad, Y. F. Tan and W. N. Tan [11] proposed an approach using Isolation Forest [12], Random Forest [13], and Decision Tree to predict abnormal power consumption behavior with high accuracy. The proposed method uses the Isolation Forest algorithm to label (normal or abnormal) the power consumption index of intelligent electricity meters. From this labeling, the time series data will form many time series with different lengths. Based on this data, two algorithms, Random Forest and Decision Tree, are applied to predict the possibility of abnormal power consumption. Experimental results show that the proposed method accurately predicts abnormal states 30 minutes before. The results also show no significant difference in performance between Random Forest and Decision Tree when considering different dataset sizes and data series lengths.

Luochen and Hasachoo [14] studied the case of a hospital having difficulty managing drug inventory (surplus or shortage) when there is an abnormal demand for some drugs. The study compared some methods, such as Croston [15], TSB [16], SBA [17], and Kalaya, Termsuksawad and Wasusri [18] on the same abnormal drug demand data set. The results showed that the Kalaya method gave the best results for abnormal forecasting over 130 days. This study mainly compared abnormal time series forecasting methods, but in univariate form, without considering the ability to combine data in multivariate form and exploiting the relationship between variables.

Research in [19] has shown that irregular data makes time series forecasting difficult. The authors proposed using Support Vector Machines (SVM) and Ensemble Empirical Mode Decomposition (EEMD) [20] to forecast this type of irregular data. Specifically, the time series data is decomposed into “smooth” and “continuous” sub-series using the EEMD technique. Then, SVM is applied to model each sub-series. Finally, the forecasts for the sub-series are aggregated to form the ensemble forecast. Experimental results on two artificial datasets show that the SVM-EEMD combination outperforms SVM, ARIMA, and Croston in terms of RMSE, SMAPE, MDRAE, and MASE.

The study in [21] proposed a wireless demand forecasting model with abnormal traffic values due to interference, mobility, connection requirements, etc., to improve energy efficiency and avoid network outages. Sun and Gou o proposed a Feature Embedding (FE) kernel for Gaussian Process (GP) to forecast traffic demand with extreme values. Experimental results show that the FE-GP combination reduces the performance by 32% compared to S-ARIMA and reduces the performance by 17% compared to Naive-GP. For long-term mean value forecasting, the FE-GP model has a reduction of 21% compared to S-ARIMA and reduces the performance by 12% compared to Naive-GP.

In a research on analyzing short, unstructured comments containing emotional language slang, R a n a et al. [22] proposed a combination of RoBERTa-1D-CNN-BiLSTM and Modern Aspect-Based Sentiment Analysis (ABSA). Specifically, the pre-trained Robustly Optimized BERT approach (RoBERTa) and One-Dimensional Convolutional Neural Network (1D-CNN) models are used to extract aspect-level features from the context of comments. Next, Bidirectional Long Short-Term Memory (BiLSTM) is used to perform classification. The model is then evaluated with datasets related to electronic products (e.g., MP3 Players, Canon, Apex AD 2600, Nikon, and Nokia 6610), hotels and restaurants, and movies. The results show that the combined model of RoBERTa-1D-CNN-BiLSTM and ABSA achieves an accuracy of 92.33%, outperforming other methods such as LSTM-LSTM, GRU-LSTM, and LSTM-GRU.

In summary, studies and proposed solutions have been used to forecast abnormal demand in several fields, such as medical supplies, electricity consumption, e-commerce, pharmaceuticals, and tourism. These methods include combining historical data statistics with linear regression, using world events with Transformer models, applying machine learning algorithms such as Isolation Forest and Random Forest, or combining feature embedding in Gaussian Process. These studies have significantly improved the accuracy of forecasting. However, these studies mainly focus on univariate data, requiring manual processing at some stages, and still need to incorporate multivariate data to exploit the relationship between time series data variables with abnormality. This paper uses the TFT architecture with multivariate data, which can explain the influence of data variables on forecasting results. In addition, TPE is integrated into TFT to automatically find the optimal hyperparameter set for the TFT architecture to improve forecasting efficiency. A combination of historical travel demand data and travel search data from Google Trends is also proposed and analyzed. The performance of the TFT-TPE integrated model is evaluated with international arrivals data to Vietnam and the UK, covering the COVID-19 period.

3. TFT-TPE integrated model

3.1. Temporal fusion transformer

TFT [7] is a transformer-based model used for multidimensional time series forecasting with higher explainability than "black box" models of conventional neural networks. TFT combines high forecasting performance with explainability, which is especially useful for forecasting complex problems such as time series with uniform variability, data from the known future, and temporal variables observed in the past.

TFT includes the following key components, as shown in Fig. 1:

- Gated Residual Networks (GRNs): Use skip connections to enable the model to learn which layers can be skipped, to improve generalization, and to reduce the required parameters.
- Static Feature Encoder learns contextual embeddings from static features (i.e., features that do not change over time) to enrich the temporal representations learned by the model.
- Variable Selection Networks learn to assess the importance of each input feature to ensure that only the most relevant features are used in subsequent layers.
- Sequence-to-Sequence Layer uses LSTMs to process local temporal patterns and combines them with static contextual embeddings to enrich the learned temporal representations.
- Multi-Head Attention helps the model capture long-term dependencies in time series data by allowing multiple attention heads to focus on different aspects of the data.

TFT leverages self-attention mechanisms to capture complex temporal dependencies in data. To do so, TFT uses various temporal encoding techniques that allow modeling the incorporation of time-related information into the input sequence to capture temporal patterns and trends. Model explainability is a crucial feature of TFT, as it provides insight into how forecasts are made and shows which past-time steps are most influential in forecasting future values. TFT applies attention mechanisms, which help highlight the importance of different input features and time factors. This approach makes the forecasting process more understandable and transparent to users.

TFT uses sequential layers to process the input data, such as selecting variables to remove irrelevant data and encoding contextual data. Like other neural network-based models, the performance of TFT depends on the configured hyperparameters, such as learning rate, number of layers, number of neurons in each layer, etc. Selecting a suitable set of hyperparameters for TFT is essential, and several hyperparameter optimization techniques, such as Grid Search, Random Search [5], or Tree-structured Parzen Estimator (TPE) [6], can be used to determine the optimal set of hyperparameters. TPE is focused on in this paper because of its ability to automatically search for hyperparameter sets, reduce the number and time of evaluating them, and escape local optimizations. Details of TPE are presented in the next section.

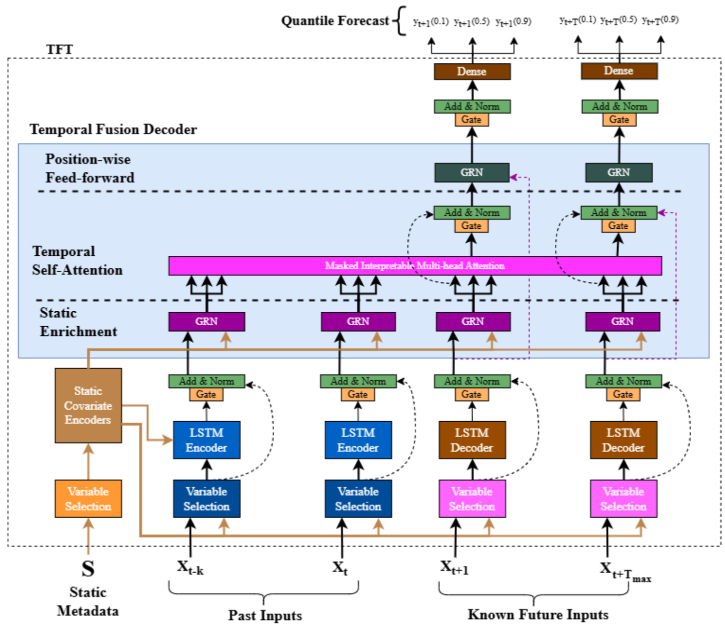


Fig. 1. TFT model [7]

3.2. Tree-structured Parzen Estimator

Hyperparameter optimization is an important requirement for neural network-based models. Given that each parameter can have its unique value range, whether continuous or discrete, it is impractical to try all possible combinations, called hyperparameter configurations, to determine the best one. TPE can help overcome this obstacle by not only learning from observed data (the tested hyperparameter configurations c and the corresponding evaluation results $v = f(c)$) but also predicting which configurations will improve the model performance, reducing the number of trials and time required for optimization.

TPE uses the Expected Improvement (EI) criterion, which represents the expectation of improving the objective function value if testing with a new hyperparameter configuration. EI considers the improvement that can be achieved compared to the current result and the probability of obtaining that improvement. Specifically, EI is calculated based on the conditional probability distribution $p(v \mid c)$, which represents the probability of obtaining the value v when evaluating the objective function with the hyperparameter configuration c, as

(1) $$\mathrm{EI}_{v^*}(c) = \int_{-\infty}^{v^*} (v^* - v) \, . \, p(v|c) dv,$$

where v^* is the threshold value selected from the top-γ quantile, which is defined in Equation (4).

To estimate $p(v \mid c)$, TPE does not directly model this distribution but uses Bayes' theorem to rewrite the conditional probability as

(2) $$p(v|c) = \frac{p(c|v)p(v)}{p(c)}.$$

The optimization problem then turns into modeling $p(c \mid v)$ as a conditional probability of a hyperparameter configuration c given the value v. To do this, TPE divides the observation dataset into two regions corresponding to the configuration with good performance $l(c)$ and the configuration with poor performance $g(c)$, as shown in the equation

(3) $$p(c|v) = \begin{cases} l(c) = p\left(c \middle| D^{(l)}\right) if\ v < v^*, \\ g(c) = p\left(c \middle| D^{(g)}\right) if\ v \geq v^*, \end{cases}$$

where:

- $l(c) = p(c \mid D^{(l)})$ is the probability density function of configurations c estimated from the set of observed points $D^{(l)}$ that performs better than the threshold v^*. In other words, this is the distribution of configurations that we "want to find more". $l(c)$, abbreviated low, represents the probability density function of a configuration c in the region with the evaluation value less than the threshold v^*, i.e., the configuration considered "good" according to the minimization criterion.

- $g(c) = p(c \mid D^{(g)})$ is the probability density function of configurations c estimated from the set of observed points $D^{(g)}$ whose performance is worse than or equal to the threshold v^*. These are the configurations that are least preferred for further testing. $g(c)$, abbreviated greater, represents the probability density function of a configuration c in the region with an evaluation value greater than or equal to the threshold v^*, i.e., the configuration is worse and is less preferred for testing.

Splitting the data based on a threshold v^* means that the set D is divided into two groups: one group consisting of γ percent of the configurations with the best performance (corresponding to $v < v^*$) and the rest with worse performance. The parameter γ is chosen by the user (usually 0.15 or 0.25), representing the proportion of "good" observations in the collected data. This value is also the probability that a configuration has a rating value that is in the region better than v^*, as in the equation

(4) $$\gamma = p(v < v^*) = \int_{v<v^*} p(v)\ dv.$$

At the same time, to standardize EI, it is necessary to determine the marginal probability $p(c)$ of a hyperparameter configuration c by fully integrating the variable v, with the conditional probability distribution $p(c \mid v)$, as in equation

(5) $$p(c) = \int_{-\infty}^{+\infty} p(c|v)p(v)dv = \int_{v<v^*} p(c|v)p(v)\ dv + \int_{v\geq v^*} p(c|v)p(v)\ dv.$$

Combining Equations (2), (3), (4), and (5) into Equation (1), EI finally is as in the equation

(6) $$EI_{v^*}(c) = \frac{\gamma v^* l(c) - l(c)\int_{-\infty}^{v^*} p(v)dv}{\gamma l(c) + (1-\gamma)g(c)} \propto \left(\gamma + \frac{g(c)}{l(c)}(1-\gamma)\right)^{-1}.$$

The point with the largest EI corresponds to the point with the smallest ratio of $\frac{g(c)}{l(c)}$. This allows the algorithm to select the optimal hyperparameter set in each iteration. After multiple iterations, TPE identifies the best hyperparameter configuration based on the point with the highest EI.

The TPE Algorithm is performed iteratively with the following seven steps.

Step 1. Determine the optimal value range for the hyperparameters, randomly generate some initial hyperparameter configurations, and evaluate the corresponding model performances.

Step 2. Based on the evaluations, divide the set of tested hyperparameter configurations into two groups according to Equation (3).

Step 3. Determine $p(c \mid v)$ through two distributions, $l(c)$ and $g(c)$, which reflect the probability of the selected configuration c given the performance v^*.

Step 4. Apply Equation (6) to calculate the expected value of EI for each candidate configuration.

Step 5. Select the configuration c_{i+1}^* with the largest EI to test in the next round, $i+1$.

Step 6. Train the model with configuration c_{i+1}^* and evaluate the performance on the validation dataset.

Step 7. Record the results in the observation data set D. If the stopping condition, such as the maximum number of iterations, is not met, return to Step 2; if the stopping condition is met, select the configuration with the best performance as the final result.

TPE builds a probabilistic model based on the results from previous tests to find a better set of hyperparameters with each run. TPE can determine the importance of each hyperparameter during the optimization process. TPE does not directly evaluate the importance of each hyperparameter but can adjust the testing frequency for each hyperparameter value. By evaluating the model's performance with different configurations, TPE adjusts the search direction so that parameters significantly impacting performance will have a higher probability in the next runs.

3.3. TFT-TPE integration

Integrating TPE into TFT leverages the power of TPE to optimize the hyperparameter set in TFT. Specifically, this process begins with using TFT to represent a part of the hyperparameter space. TFT can represent complex relationships between parameters, especially in models with sequential data. This paper uses TFT to create a model for the objective function. Updating each pass's parameters and results is a fine-tuning process through TPE to create a more stable model.

The TFT-TPE model is implemented in 12 steps (Fig. 2).

Step 1. Data collection: Collect time series and related data that may affect the forecast results.

Step 2. Data processing and feature generation: Process the data to normalize and clean it. Identify essential features from the original data, including past and future covariates.

Step 3. Initialize the hyperparameter search space: Set the range and initial values to optimize the hyperparameters. This activity helps define the hyperparameter search space boundaries. Randomly generate some initial hyperparameter value sets and evaluate their performance.

Step 4. Group $l(c)$ and $g(c)$: Based on the observation results, divide the data into two groups: the best-performing group, $l(c)$, and the remaining group, $g(c)$.

TFT-TPE uses two density distributions $l(c)$ and $g(c)$ defined in Equation (3) to model the conditional probability $p(c \mid v)$, where $v < v^*$ indicates that the value of the objective function is less than the threshold, and $v \geq v^*$ denotes that the value of the objective function is greater than or equal to the threshold. After data splitting, TPE uses the Kernel Density Estimation (KDE) method [23] to estimate the probability density distributions $l(c)$ and $g(c)$. KDE is a non-parametric density estimation method that allows for constructing smooth approximations of probability distributions from discrete data without making any assumptions about the original distribution.

Step 5. Update the probability models: Incorporate the latest observation into the historical dataset, then re-estimate the two conditional probability models $l(c)$ and $g(c)$, which characterize the probability density of configurations with objective values below and above the threshold v^*, respectively.

Step 6. Compute $c^* = \text{argmax}_i \left(\frac{l(c)}{g(c)}\right)$: Find the hyperparameter set c^* that maximizes the ratio of $\frac{l(c)}{g(c)}$.

Step 7. Choose c_{i+1} with Max(EI): Choose the new hyperparameter set c_{i+1} that maximizes the EI in (5).

Step 8. Train the model with the new hyperparameter set c_{i+1}.

Step 9. Evaluate the model performance with c_{i+1}.

Step 10. Add $c_{i+1}$to the historical hyperparameter set.

Step 11. Check stop conditions: If i equals the maximum number of iterations, go to Step 12. Otherwise, increment the value i by 1 ($i = i + 1$), and the process will return to Step 4.

Step 12. Return the result: Choose the best-performing hyperparameter set from the historical dataset.

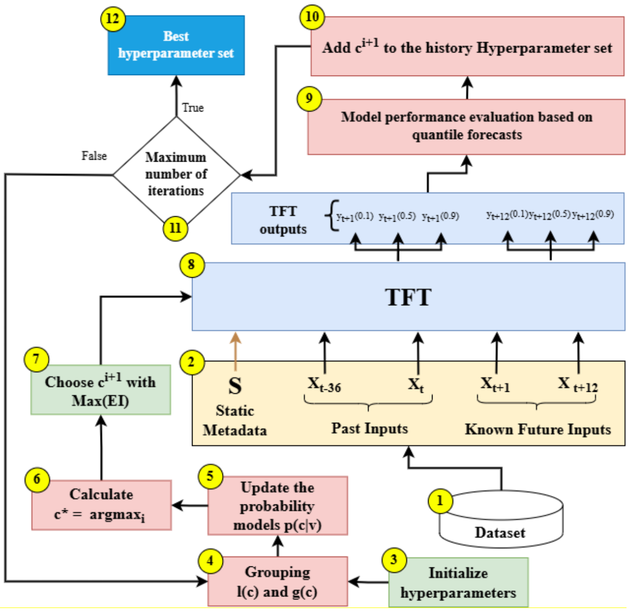


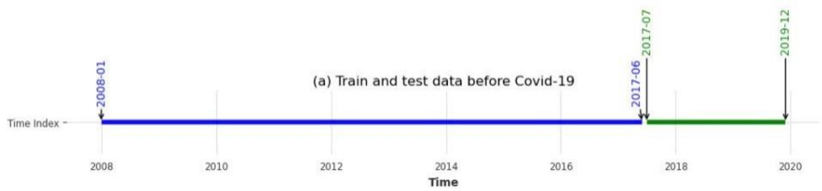
Fig. 2. The proposed TFT-TPE integrated model

The TFT-TPE integrated model offers significant benefits. Specifically, TFT allows complex relationships between parameters to be represented, while TPE provides efficient search capabilities in the hyperparameter space. This combination not only enhances the efficiency of the hyperparameter optimization process but also helps to achieve higher performance in a shorter time. This result demonstrates the outstanding potential of TFT-TPE integration in improving the efficiency and speed of predictive models.

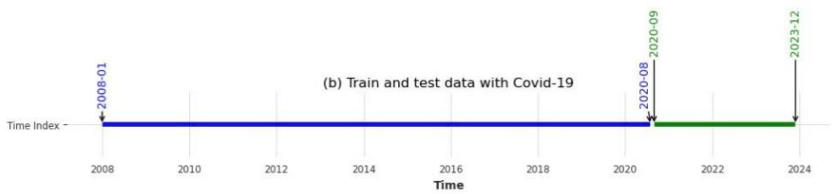
4. Simulation and analysis

4.1. Data collection

The data was collected from various sources from January 2008 to December 2023. Accordingly, abnormal data on tourism demand due to the impact of the COVID-19 pandemic are included. To evaluate the performance of the forecasting model with non-abnormal and abnormal data, we split the data into two parts: the dataset from January 2008 to December 2019 (excluding abnormal data) and the dataset from January 2008 to December 2023 (including abnormal data during the COVID-19 pandemic, from January 2020 to April 2022). These two datasets are normalized and divided into two training and testing parts with a ratio of 80:20. Specifically, for the data set of the period [01/2008, 12/2019], data from January 2008 to June 2017 is used to train the model and data from July 2017 to December 2019 is used to test and evaluate the performance of the model (Fig. 3a). For the data set covering the COVID-19 epidemic period of [01/2008, 12/2023], data from January 2008 to August 2020 is used to train the model, and from September 2020 to December 2023 is used to test and evaluate the performance (Fig. 3b). Splitting the data this way helps the model learn long-term trends and seasonality before testing it on a newer data period with both cases affected and unaffected by the COVID-19 pandemic. The goal is to determine which model can better capture and predict complex and uncertain trends in travel data.



(a)



(b)

Fig. 3. The method divides the dataset into a training set and a test set for cases of data before COVID-19 (a), and including the COVID-19 period (b)

4.2. Data analysis

The forecasting model uses monthly international arrivals data as the target variable. Monthly international arrivals data for Vietnam and the United Kingdom (the UK) for the period of [2008, 2023] are collected to be used to forecast the two countries' tourism demand. The data are then preprocessed and normalized to serve as the target variable for the TFT-TPE model (Fig. 4). To ensure comparability across different Google Trends time series, we applied Z-score normalization (Standardization), a widely used technique in time series analysis and machine learning. Given a time series $X = \{x_1, x_2, \dots, x_n\}$, the normalized value X' is computed using the Z-score formula, $X' = \frac{X-\mu}{\sigma}$, where μ is the mean of the time series, *and* σ is the standard deviation.

For Vietnam, data on monthly international arrivals were collected from the website of the Vietnam National Authority of Tourism [24], while the tourists to the UK were collected from the source [26]. As shown in Fig. 4, there were still tourists to the UK during the COVID-19 period. Meanwhile, the number of tourists to Vietnam during this time was zero due to Vietnam's complete entry ban policy.

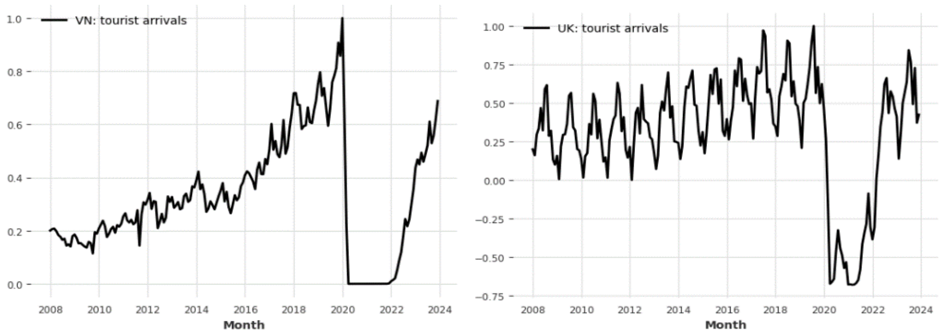


Fig. 4. Monthly international arrivals data to Vietnam and the UK covering the COVID-19 period [24, 25]

As shown in Fig. 4, the UK international arrivals data clearly shows seasonality (in the pre-COVID-19 period), while this is not evident for the Vietnam international arrivals data. The seasonality has a significant positive impact on the accuracy of the different forecasting methods.

4.3. Combine with Google Trends

Tourists often search for information about their destinations before they travel. This search information, typically expressed in the frequency of search keywords, clearly affects the accuracy of forecasting tourist arrivals to a destination. Therefore, combining Google search volume data with historical time series data on arrivals can improve forecasting accuracy.

The initial keyword list was built using reference sources, combined with Google's suggestion tool and interviews with travelers to collect popular keywords before traveling. Table 1 describes the list of keywords used. These keywords were then input into Google Trends to determine the search volume over time. Next, the time series of these keywords was compared with the target series in the training set to calculate the similarity.

Table 1. Normalized similarity of search volumes relative to keywords

VN		UK	
Keyword	Similarity	Keyword	Similarity
Visit Vietnam	0.9379	Things to do in London	0.8368
Vietnam things to do	0.9086	hand luggage	0.7786
Hanoi things to do	0.9073	Things to do in the UK	0.6352
Best time to visit Vietnam	0.8061	Liverpool Beatles Tour	0.6111
Flights to Hanoi	0.7857	Best time to visit London	0.5256
Holiday in Vietnam	0.7124	Cheap flights to anywhere	0.4994
Flights to Vietnam	0.6913	Solo travel	0.4446
Travel to Vietnam	0.6829	Flights anywhere	0.4402
Travel to Hanoi	0.6748	Cheapest places	0.4183
Hotels in Vietnam	0.6582	Visit London	0.4151

We use the Pearson Correlation Coefficient (PCC), a widely used method for identifying linear relationships between two signals, to quantify the similarity between time series. Given two normalized time series X' and Y', PCC is defined as

$$\rho(X',Y') = \frac{\sum(X_i'-\overline{X'})(Y_i'-\overline{Y'})}{\sqrt{\sum(X_i'-\overline{X'})^2}\sqrt{\sum(Y_i'-\overline{Y'})^2}} \quad (7)$$

where $\overline{X'}$ and $\overline{Y'}$ are the mean values of the respective normalized series.

A correlation coefficient close to +1 indicates a strong positive similarity, while a value near −1 indicates an inverse relationship. Based on the similarity, the most relevant keywords were selected to covariate for the past in the forecasting model.

Fig. 5 depicts the normalized similarity level to the keywords collected from Google Trends for the Vietnam and UK destinations. After identifying relevant keywords, the search volume series is normalized to ensure they are on the same scale and suitable for comparison. These normalized time series are then used as historical covariates in the TFT-TPE integrated model.

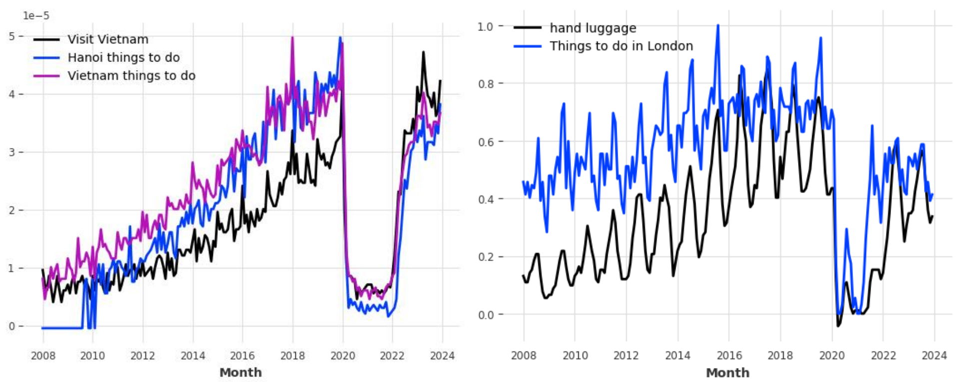


Fig. 5. Data corresponding to keywords collected from Google Trends: Vietnam and UK destinations

4.4. Future covariates

Future covariates in the TFT-TPE model are input variables outside the time series that are used to provide additional information about factors that may affect the future forecast. Future covariates are understood as data that are known at the time of forecasting. They help the model better understand the context and external factors that may affect the future target time series' value. We use time attributes such as year, month, and a linear increase series incorporating COVID-19 travel ban policy data (Linear Increase and COVID-19 Period) as future covariates (Table 2). These attributes provide detailed information about time and trends, helping the TFT model better understand the temporal characteristics of the data. The "Year" covariate helps the model recognize year-wise features. For example, if the tourism data increases over the year, the year covariate will help the model learn this trend. The "month" covariate helps the model recognize month-wise features, which aids in learning seasonal patterns. For example, if tourist arrivals increase in the summer months and decrease in the winter months, the month covariate will help the model learn this seasonality.

Table 2. Covariate and target data

Time index	2008-01	2008-02	2008-03	...	2019-12	2020-01	2020-02	…	2022-04	2022-05	2022-06	…	2023-11	2023-12
Year	2008	2008	2008	...	2019	2020	2020	…	2022	2022	2022	…	2023	2023
Month	1	2	3	...	12	1	2	…	4	5	6	…	11	12
UK-LC	0	1	2	...	143	144	145	…	0	1	2	…	20	21
G1	51	48	50	...	67	74	71	...	60	57	55	...	44	46
G2	20	18	18	...	46	48	48	...	40	50	60	...	37	39
UK-T	2349	2249	2597	...	3445	3036	2512	...	2252	2728	2977	...	2795	2931
VN-LC	0	1	2	...	143	–191	–191	...	–191	0	1	...	19	20
G3	20	15	13	...	66	84	41	...	47	46	57	...	75	85
G4	17	10	14	...	82	98	58	...	43	49	54	...	70	74
G5	0	0	0	...	84	73	84	...	25	31	41	...	67	77
VN-T	400	411	414	...	1710	1994	1243	...	101	173	237	...	1233	1371

Note:
- UK-LC: Linear Increase and COVID-19 Period of UK
- G1: Google Trends with "Things to do in London" keyword
- G2: Google Trends with "hand luggage" keyword
- UK-T: Monthly international arrivals data to the UK (in thousands)
- VN-LC: Linear Increase and COVID-19 Period of VN
- G3: Google Trends with "Visit Vietnam" keyword
- G4: Google trend with "Vietnam things to do" keyword
- G5: Google trend with "Hanoi things to do" keyword
- VN-T: Monthly international arrivals data to Vietnam (in thousands)

The Linear Increase and COVID-19 Period covariate (Fig. 6) represents a time-based index that increases gradually under normal conditions and decreases during periods of significant disruptions, such as the COVID-19 travel bans. Positive values in the "Linear Increase" indicate a gradual increase in tourism or other related factors over time, while negative values reflect a decline due to external factors like travel restrictions. For instance, the large negative values in January and February 2020 (e.g., –191 in the UK) represent the sharp decline in tourism caused by COVID-19 lockdowns. This covariate helps the model understand both the long-term trends and the abrupt shifts in the data due to such events.

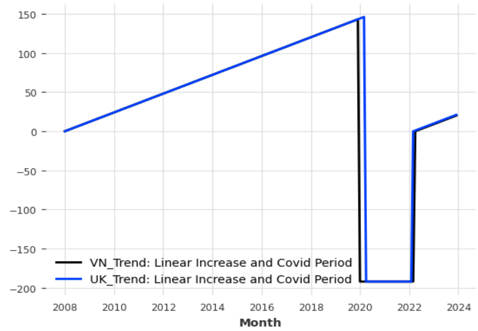


Fig. 6. Linear Increase and COVID-19 period

4.5. Evaluation metrics

The performance of the TFT-TPE integrated model is compared with that of two traditional models, Exponential Smoothing (ES) and ARIMA, and that of two machine learning models, Prophet and LSTM. Common evaluation metrics include Mean Absolute Percentage Error (MAPE), symmetric MAPE (sMAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE),

$$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{|A_t|}, \tag{8}$$

$$\text{sMAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{\frac{|A_t|+|F_t|}{2}}, \tag{9}$$

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|F_t - A_t|, \tag{10}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(F_t - A_t)^2}, \tag{11}$$

where n is the number of observations in the test sets, A_t and F_t are the actual and predicted values at step t, respectively. According to (8), (9), (10), and (11), the smaller the MAPE, sMAPE, MAE, and RMSE, the better the model's forecasting performance.

However, MAPE can become very large when the actual value is small, approaching zero, especially if there is a difference between the actual value and the forecast value, even if the difference is insignificant. sMAPE overcomes this drawback by using the average between the actual and forecast values in the denominator, making it less sensitive to small values, resulting in smaller sMAPE values in similar cases. To find the set of hyperparameter values for the TFT-TPE model, we use sMAPE instead of MAPE as a criterion to evaluate the model's performance under different hyperparameter values.

To achieve optimal forecasting performance, tuning the hyperparameters of the TFT model is extremely important. This process uses TPE to focus on the hyperparameters with the most significant impact, ensuring that the model exploits its maximum forecasting potential. The important hyperparameters that need to be optimized include the model’s main structural and training elements. These parameters directly affect the learning ability of the TFT, helping the model learn complex features and forecast more accurately for time series data. Therefore, this paper uses the following hyperparameters to be included in the TFT-TPE integrated model for optimization; the remaining hyperparameters are used by default:

- hidden_size: Represents the hidden size of the model, helping to adjust the representation ability of LSTM in TFT. Too small a size can lead to information loss, while too large can cause overfitting.
- lstm_layers: The number of LSTM layers affects the depth and learning ability of the model. Optimizing the number of layers helps the model process more complex information.
- num_attention_heads: The number of attention heads allows the model to learn more about the relationships in the data. This optimization improves the ability to pay attention to essential features.
- dropout: This parameter helps prevent overfitting by randomly ignoring some connections in the model. Optimizing the dropout level is necessary to maintain accuracy without losing information.
- batch_size: Optimizing the batch size helps adjust the training speed and stability of the model. A batch size that is too large can reduce the accuracy of learning.
- n_epochs: The number of epochs affects whether the model learns enough. This optimization ensures that the model is trained enough without being overtrained.
- random_state: Ensures reproducibility between optimizations, helping to evaluate hyperparameters objectively.
- feed_forward and norm_type: Network types and normalization are also optimized to find the best structure for the data.

The non-optimized hyperparameters include input_chunk_length and output_chunk_length, which are the lengths of the input and output sequences, respectively. Based on the characteristics of the data and the forecasting requirements, the data from the previous 2 years (input_chunk_length = 24) is used to forecast the next 6 months (output_chunk_length = 6). With sMAPE as the criterion for optimization, the set of hyperparameter values is found in Table 3.

Table 3. Hyperparameters found for the TFT-TPE integrated model corresponding to two datasets

Parameters and Domain Space	UK data before COVID-19	UK data after COVID-19	VN data before COVID-19	VN data after COVID-19
hidden_size = [16, 128]	100	79	87	117
lstm_layers = [1, 3]	1	1	2	2
dropout = [0.1, 0.5]	0.212945046655	0.100011684062	0.376140965425	0.269451232155
batch_size = [16, 64]	38	32	43	39
n_epochs = [50, 200]	176	132	105	112
num_attention_heads = [1, 8]	3	2	8	1
random_state = [0, 100]	71	54	84	17
feed_forward = [GatedResidualNetwork, GLU, Bilinear, ReGLU, GEGLU, SwiGLU, ReLU, GELU]	Bilinear	GLU	SwiGLU	ReLU
norm_type = [LayerNorm, RMSNorm, LayerNormNoBias]	LayerNormNoBias	LayerNorm	LayerNorm	RMSNorm

To ensure a fair comparison between models, we utilized the Darts (Darts 0.30.0: **https://github.com/unit8co/darts/releases/tag/0.30.0**) library for time series forecasting. Specifically, for the Prophet, we used default seasonalities (yearly) unless otherwise specified, and holidays were set based on the country of interest (Vietnam or the UK). With ARIMA, we used AutoARIMA with predefined boundaries for the values p, d, and q, allowing the model to find the best configuration automatically. In the case of deep learning models (TFT and LSTM), we tuned hyperparameters such as input and output chunk lengths, dropout rates, and the number of attention heads for TFT. These values were chosen based on prior work and experimentation to optimize forecasting accuracy. By documenting these hyperparameters, we ensure reproducibility and provide transparency in our model selection process. The details of the hyperparameters used for each model are summarized in Table 4.

Table 4. Default values of hyperparameters for the traditional models

Models	Parameters
ES	trend = ModelMode.ADDITIVE
	damped = FALSE
	Seasonal = SeasonalityMode.ADDITIVE
Prophet	add_seasonalities = None
	country_holidays = VN or UK
	add_encoders = None
	cap = None
	floor = None
LSTM	hidden_dim = 20
	dropout = 0
	batch_size = 16
	n_epochs = 300
	optimizer_kwargs = {“lr”: 1×10^{-3}}
	log_tensorboard = TRUE
	random_state = 42
	training_length = 30
	input_chunk_length = 24
	output_chunk_length = 6
	force_reset = TRUE
	save_checkpoints = TRUE

Models	Parameters
ARIMA	start_p = 8
	max_p = 12
	start_q = 1
TFT	input_chunk_length = 24
	output_chunk_length = 6
	output_chunk_shift = 0
	hidden_size = 16
	lstm_layers = 1
	num_attention_heads = 4
	full_attention = FALSE
	feed_forward = GatedResidualNetwork
	dropout = 0.1
	hidden_continuous_size = 8
	categorical_embedding_sizes = None
	loss_fn = None
	likelihood = None
	norm_type = LayerNorm
	use_static_covariates = TRUE
	n_epochs = 300
	force_reset = TRUE
	add_relative_index = TRUE

4.6. Comparing the forecasting performance with pre-COVID-19 data

The analysis of the performance of tourism forecasting models in the COVID-19 period in Vietnam and the UK provides an overview of the differences in the performance of the forecasting models. UK tourism demand data has an apparent 12-month seasonality, while the international arrivals data in Vietnam do not show this seasonality. The experimental results show that the traditional models, ES and ARIMA, and the machine learning models, Prophet and LSTM, show good forecasting ability for UK tourism demand data. However, these models give poor forecasting accuracy when applied to international arrivals data in Vietnam. This result shows that seasonality positively impacts forecasting accuracy and that the traditional models can capture the data trend with seasonality, but have difficulty with the data type whose seasonality is unclear.

Meanwhile, deep learning models such as TFT and especially the improved version, TFT-TPE, have shown superior predictive performance for both datasets. Experimental results demonstrate that TFT-TPE consistently outperforms traditional models and yields significant improvements over the standard TFT configuration with evaluation metrics, as described in Table 5.

Table 5. Comparing the forecasting models based on evaluation metrics with the data before COVID-19

Model	UK			VietNam		
	sMAPE (%)	MAE	RMSE	sMAPE (%)	MAE	RMSE
ES	23.6	898	932	22.4	324,627	341,744
Prophet	17.7	653	686	17.6	248,631	265,616
ARIMA	17.2	626	679	20.3	290,525	312,562
TFT	21.0	772	869	17.9	256,045	291,570
LSTM	21.8	804	921	17.9	258,870	310,041
TFT-TPE	**11.0**	**387**	**461**	**12.8**	**179,322**	**240,548**

Based on the results in Table 5, TFT-TPE outperforms in forecasting tourism demand in both Vietnam and the UK from July 2017 to December 2019. Specifically, for the UK, TFT-TPE has the lowest error indices: sMAPE = 11%, MAE = 387, and RMSE = 461, meaning that the model achieves the highest forecasting accuracy. ARIMA also shows good forecasting ability with sMAPE = 17.2%, while LSTM and TFT have higher sMAPE values. ES is the worst in terms of forecasting ability for both datasets.

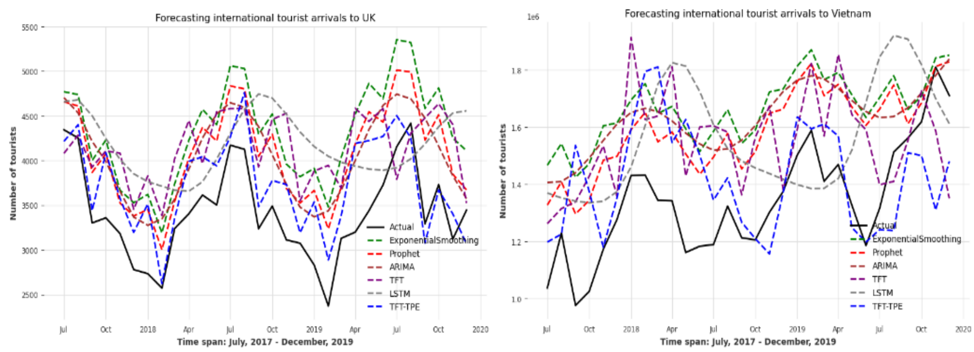


Fig. 7. Comparison of forecast results and actual international arrivals in the UK and Vietnam (from December 2017 to December 2019)

With non-seasonal data, such as the Vietnam tourism demand data, the forecasting performance of TFT-TPE is better than that of other models. With the metric values sMAPE = 12.8%, MAE = 179.322, and RMSE = 240.548, the forecast error in TFT-TPE is significantly reduced, indicating that the model is robust to non-seasonal data. Prophet shows better forecasting ability than other models, with sMAPE = 17.6%, while ARIMA and ES perform poorly.

TFT-TPE demonstrates high effectiveness due to its ability to adapt to diverse data characteristics. By leveraging the TFT and optimizing its hyperparameters with TPE, the model effectively captures both seasonal trends, as seen in UK tourism data, and irregular patterns in non-seasonal data, like Vietnam's international arrivals. Traditional models often struggle with non-seasonal data, while TFT-TPE's advanced architecture allows it to model complex variations and deliver consistent, accurate forecasts regardless of the data's underlying patterns (Fig. 7).

4.7. Comparing the forecasting performance with data including the COVID-19 period

We further compare the TFT-TPE forecasting performance with the other forecasting models mentioned above for the dataset covering the COVID-19 period. Fig. 8 depicts the forecasting results (from September 2020 to December 2023) and shows that, despite the sharp decline in international arrivals to the UK, TFT-TPE can still capture this abnormal event well and forecast pretty accurately. Particularly for Vietnam, due to the policy of banning international tourists altogether, there has been a significant increase in forecasting accuracy during the COVID-19 period.

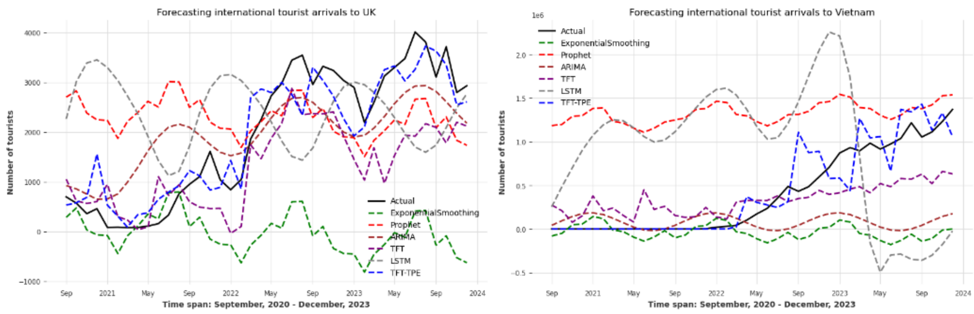


Fig. 8. Comparison of forecast results and actual international arrivals in the UK and Vietnam (from September 2020 to December 2023)

Table 6 compares the forecasting performance of the forecasting models, in which TFT-TPE outperforms ES, Prophet, ARIMA, LSTM, and TFT. TFT-TPE achieves the lowest evaluation metrics for both the UK and Vietnam travel demand datasets, especially sMAPE. Although widely used in forecasting evaluation, the ES and Prophet models give the worst forecasting results with the highest errors. ES has the highest sMAPE, 164.5% for the UK and 193.2% for Vietnam, indicating that it is unsuitable for abnormal events. ARIMA gives better forecasting results than ES and Prophet, but is still less effective than expected, with the evaluation metrics relatively high, especially sMAPE. With TFT, a recently proposed advanced machine learning model, the prediction accuracy has been significantly improved compared to traditional models, but it still cannot compete with TFT-TPE.

TFT-TPE stands out in evaluation metrics thanks to its ability to handle abnormal data well and integrate covariates such as COVID-19 cases. Experimental results show that TFT-TPE is robust to non-seasonal data and responds well to abnormal data, confirming that it is the most effective and reliable forecasting model for different data conditions. TFT-TPE achieves a relatively low sMAPE value of 35.9% for the UK and 40.7% for Vietnam. TFT-TPE's superior performance during the COVID-19 period is driven by its robustness in handling abnormal data and its capacity to incorporate external factors such as COVID-19 case numbers. These capabilities enable the model to identify and respond to sharp disruptions and irregularities in tourism trends. While traditional models are limited by their assumptions about trend stability, TFT-TPE dynamically adjusts to unexpected changes, ensuring accurate forecasts even under extreme volatility caused by unprecedented events like the pandemic.

Table 6. Comparing the forecasting models based on evaluation metrics with the data, including the COVID-19 period.

Mô hình	UK			VN		
	sMAPE (%)	MAE	RMSE	sMAPE (%)	MAE	RMSE
ES	164.5	2011	2434	193.2	47,2626	6.61×10^5
Prophet	76.1	1304	1469	128.5	921,256	1.01×10^6
ARIMA	55.7	739	849	169.5	413,774	5.71×10^5
TFT	63	824	1002	119.7	285,612	3.38×10^5
LSTM	79	1437	1692	169.9	1139,650	1.19×10^6
TFT-TPE	**35.9**	**384**	**492**	**40.7**	**130,319**	$\mathbf{2.13}\times10^5$

5. Conclusion

The paper proposed a TFT-TPE integration model in which TPE searches for the best set of hyperparameter values to fill the TFT. The paper evaluated the TFT-TPE integration model with the dataset of international arrivals to Vietnam and the UK from 2008 to 2023, combined with the data on new COVID-19 cases. The evaluation metrics include RMSE, MAE, MSE, and sMAPE. The experimental results show that TFT-TPE achieves the best performance in determining the hyperparameter set and helps optimize the performance of the TFT architecture. TFT-TPE consistently achieves better evaluation metrics for both datasets before and including the COVID-19 period than the traditional forecasting models, ES and ARIMA, and the machine learning-based forecasting models, Prophet and LSTM. Compared with TFT, TFT-TPE gives more accurate forecasting results by identifying the optimal hyperparameter set and incorporating covariates of COVID-19 cases. Therefore, TFT-TPE is suitable for non-seasonal data and particularly robust to abnormal data. This confirms the superiority of the proposed TFT-TPE integrated model.

Acknowledgement: The authors thank the School of Hospitality and Tourism – Hue University, Vietnam, for funding this project (Code DHH2024-10-38).

References

1. Kotu, V., B. Deshpande. Time Series Forecasting. – Data Sci., 2019, pp. 395-445. DOI: 10.1016/B978-0-12-814761-0.00012-5.
2. Schmidl, S., P. Wenig, T. Papenbrock. Anomaly Detection in Time Series: A Comprehensive Evaluation. – Proc. VLDB Endow., Vol. **15**, 2022, No 9, pp. 1779-1797. DOI: 10.14778/3538598.3538602.

3. C h o i, K., J. Y i, C. P a r k, S. Y o o n. Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. – IEEE Access, Vol. **9**, 2021, pp. 120043-120065. DOI: 10.1109/ACCESS.2021.3107975.
4. P r a v i n, P. S., J. Z. M. T a n, K. S. Y a p, Z. W u. Hyperparameter Optimization Strategies for Machine Learning-Based Stochastic Energy Efficient Scheduling in Cyber-Physical Production Systems. – Digit. Chem. Eng., Vol. **4**, 2022, p. 100047. DOI: 10.1016/J.DCHE.2022.100047.
5. B e r g s t r a, J., Y. B e n g i o. Random Search for Hyper-Parameter Optimization. – J. Mach. Learn. Res., Vol. **13**, 2012, No 10, pp. 281-305 (online). **http://jmlr.org/papers/v13/bergstra12a.html**
6. B e r g s t r a, J., R. B a r d e n e t, Y. B e n g i o, B. K é g l. Algorithms for Hyper-Parameter Optimization. – In: Proc. of Adv. Neural Inf. Process. Syst. 24: 25th Annu. Conf. Neural Inf. Process. Syst. 2011 (NIPS'11), 2011, pp. 1-9.
7. L i m, B., S. A r i k, N. L o e f f, T. P f i s t e r. Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. – Int. J. Forecast., Vol. **37**, 2021, No 4, pp. 1748-1764. DOI: 10.1016/j.ijforecast.2021.03.012.
8. L i u, P., W. M i n g, C. H u a n g. Intelligent Modeling of Abnormal Demand Forecasting for Medical Consumables in Smart City. – Environ. Technol. Innov., Vol. **20**, 2020, p. 101069. DOI: 10.1016/j.eti.2020.101069.
9. K a l i f a, D., U. S i n g e r, I. G u y, G. D. R o s i n, K. R a d i n s k y. Leveraging World Events to Predict e-Commerce Consumer Demand under Anomaly. – Association for Computing Machinery, Vol. **1**, 2022, No 1. DOI: 10.1145/3488560.3498452.
10. S h a r m a, K., R. B h a l l a, G. G a n e s a n. Time Series Forecasting Using FB-Prophet. – CEUR Workshop Proc., Vol. **3445**, 2022, pp. 59-65.
11. E l-H a d a d, R., Y. F. T a n, W. N. T a n. Anomaly Prediction in Electricity Consumption Using a Combination of Machine Learning Techniques. – Int. J. Technol., Vol. **13**, 2022, No 6, pp. 1317-1325. DOI: 10.14716/ijtech.v13i6.5931.
12. L i u, F. T., K. M. T i n g, Z. H. Z h o u. Isolation Forest. – In: Proc. of IEEE Int. Conf. Data Mining, ICDM, 2008, pp. 413-422. DOI: 10.1109/ICDM.2008.17.
13. C u t l e r, A., D. R. C u t l e r, J. R. S t e v e n s. Random Forests. – In: C. Zhang, Y. Ma, Eds. Ensemble Machine Learning. New York, NY, Springer New York, 2012, pp. 157-175. DOI: 10.1007/978-1-4419-9326-7_5.
14. L u o c h e n, X., N. H a s a c h o o. The Study of Irregular Demand Forecasting for Medicines: The Case Study of ABC Medical Center Hospital. – In: Proc. of 10th Int. Conf. Ind. Technol. Manag. (ICITM'21), 2021, pp. 115-120. DOI: 10.1109/ICITM52822.2021.00028.
15. C r o s t o n, J. D. Forecasting and Stock Control for Intermittent Demands. – J. Oper. Res. Soc., Vol. **23**, 1972, No 3, pp. 289-303. DOI: 10.1057/JORS.1972.50.
16. T e u n t e r, R. H., A. A. S y n t e t o s, M. Z. B a b a i. Intermittent Demand: Linking Forecasting to Inventory Obsolescence. – Eur. J. Oper. Res., Vol. **214**, 2011, No 3, pp. 606-615. DOI: 10.1016/j.ejor.2011.05.018.
17. B a b a i, M. Z., A. S y n t e t o s, R. T e u n t e r. Intermittent Demand Forecasting: An Empirical Study on Accuracy and the Risk of Obsolescence. – Int. J. Prod. Econ., Vol. **157**, 2014, No 1, pp. 212-219. DOI: 10.1016/J.IJPE.2014.08.019.
18. K a l a y a, P., P. T e r m s u k s a w a d, T. W a s u s r i. Forecasting Lumpy Demand for Planning Inventory: The Case of Community Hospitals in Thailand. – In: Proc. of IEEE Int. Conf. Ind. Eng. Eng. Manag., December 2019, pp. 686-690. DOI: 10.1109/IEEM44572.2019.8978541.
19. B a o, Y., R. Z h a n g, T. X i o n g, Z. H u. Forecasting Non-Normal Demand by Support Vector Machines with Ensemble Empirical Mode Decomposition. – Adv. Inf. Sci. Serv. Sci., Vol. **3**, 2011, No 3, pp. 81-91. DOI: 10.4156/aiss.vol3.issue3.11.
20. W u, Z., N. E. H u a n g. Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method. – Adv. Adapt. Data Anal., Vol. **1**, 2009, No 1, pp. 1-41. DOI: 10.1142/S1793536909000047.
21. S u n, S. C., W. G u o. Forecasting Wireless Demand with Extreme Values Using Feature Embedding in Gaussian Processes. – In: Proc. of IEEE Veh. Technol. Conf., Vol. **2021**, 2021, pp. 1-8. DOI: 10.1109/VTC2021-Spring51267.2021.9449040.

22. R a n a, M. R. R, A. N a w a z, T. A l i, A. S. A l a t t a s, A. S. A b d E l m i n a a m. Sentiment Analysis of Product Reviews Using Transformer Enhanced 1D-CNN and BiLSTM. – Cybernetics and Information Technologies, Vol. **24**, 2024, No 3, pp. 112-131.
23. S i l v e r m a n, B. W. Monographs on Statistics and Applied Probability: Density Estimation for Statistics and Data Analysis. – Appl. Stat., Vol. **37**, 1986, No 1, p. 120.
24. Vietnam National Authority of Tourism (Accessed 18 November 2024). **https://vietnamtourism.gov.vn/en**
25. Vietnam: Coronavirus Pandemic Country Profile – Our World in Data (Accessed May 18, 2024). **https://ourworldindata.org/coronavirus/country/vietnam?country=~VNM**
26. Overseas Travel and Tourism Time Series – Office for National Statistics (Accessed 21 August, 2024). **https://www.ons.gov.uk/peoplepopulationandcommunity/leisureandtourism/datasets/internationalpassengersurveytimeseriesspreadsheet**

Received: 20.11.2024, First Revision: 27.03.2025, Second Revision: 07.04.2025, Accepted: 23.04.2025