Tikrit Journal of Administrative and Economic Sciences, Vol. 21, No. 72, Part (1): 466-487
Doi: www.doi.org/10.25130/tjaes.21.72.1.24

Tikrit Journal of Administrative and Economics Sciences
مجلة تكريت للعلوم الإدارية والاقتصادية
EISSN: 3006-9149
PISSN: 1813-1719

A Hybrid Forecasting Framework for Iraq's Life Expectancy Prediction: Integrating Prophet's Time Series Decomposition with XGBoost's Machine Learning Capabilities for Improved Accuracy

Aras Jalal Mhamad*, Huda Mohammed Saeed, Shaima Nuradin Yaeqb
College of Administration & Economic/University of Sulaimani,-Sulaymaniyah

Keywords:
Time Series Forecasting, Hybrid Model, Prophet Model, XGBoost Model, Life Expectancy Prediction

A R T I C L E I N F O

Article history:

Received	23 Mar. 2025
Received in revised form	09 Apr. 2025
Accepted	17 Apr. 2025
Available online	31 Dec. 2025

*Corresponding author:
Aras Jalal Mhamad
College of Administration & Economic/University of Sulaimani,-Sulaymaniyah

Abstract: Time series forecasting is a critical task in various fields such as economics, healthcare, and supply chain management, where accurate predictions are essential for informed decision-making. This study focuses on forecasting life expectancy in Iraq, a complex task influenced by socio-economic, healthcare, and demographic factors. Traditional statistical methods often struggle to capture the non-linear relationships inherent in such data, prompting the exploration of advanced forecasting models. This research employs a hybrid approach, integrating Prophet, a time series decomposition tool developed by Facebook, with XGBoost, a powerful machine learning algorithm known for its ability to model non-linear relationships. Despite previous research has explored hybrid models, to the best of the researchers' knowledge, this is the first study to apply such a hybrid model to this specific dataset. The hybrid model aims to combine Prophet's strength in capturing trends and seasonality with XGBoost's capability to model complex interactions and residuals. The hybrid approach addresses the limitations of traditional models and machine learning techniques when used individually. This research fills a gap in the literature by applying this innovative framework to Iraq, providing policymakers with more accurate forecasts for healthcare and social service planning. Results indicate that the hybrid model provides more accurate and realistic predictions compared to the individual Prophet and XGBoost models. Although, this study shows hybrid models improve life expectancy predictions in Iraq. Future research should factor in healthcare, socio-political, and environmental influences. These models can guide better health policies in Iraq.

Tikrit Journal of Administrative and Economic Sciences, Vol. 21, No. 72, Part (1): 466-487
Doi: www.doi.org/10.25130/tjaes.21.72.1.24

إطار تنبؤ هجين لتوقع متوسط العمر المتوقع في العراق: دمج تحليل السلاسل الزمنية لبرنامج Prophet مع قدرات التعلم الآلي لـ XGBoost لتحسين الدقة

أرس جلال محمد هدى محمد سعيد شيماء نور الدين يعقوب
كلية الإدارة والاقتصاد/جامعة السليمانية ـ السليمانية

المستخلص

يُعدّ التنبؤ بالسلاسل الزمنية مهمة بالغة الأهمية في مجالات متنوعة كالاقتصاد والرعاية الصحية وإدارة سلاسل التوريد، حيث تُعدّ التنبؤات الدقيقة ضرورية لاتخاذ قرارات مدروسة. تركز هذه الدراسة على التنبؤ بمتوسط العمر المتوقع في العراق، وهي مهمة معقدة تتأثر بعوامل اجتماعية واقتصادية وصحية وديموغرافية. غالبًا ما تواجه الأساليب الإحصائية التقليدية صعوبة في رصد العلاقات غير الخطية الكامنة في هذه البيانات، مما دفع إلى البحث عن نماذج تنبؤ متقدمة. يستخدم هذا البحث منهجًا هجينًا يدمج بين Prophet، وهي أداة لتحليل السلاسل الزمنية طورتها شركة فيسبوك، وXGBoost، وهي خوارزمية تعلم آلي قوية معروفة بقدرتها على نمذجة العلاقات غير الخطية. على الرغم من أن أبحاثًا سابقة قد استكشفت نماذج هجينة، إلا أن هذه الدراسة، على حد علم الباحثين، هي الأولى التي تُطبّق نموذجًا هجينًا على هذه المجموعة من البيانات تحديدًا. يهدف النموذج الهجين إلى الجمع بين قوة Prophet في رصد الاتجاهات والموسمية وقدرة XGBoost على نمذجة التفاعلات المعقدة والبواقي. يُعالج المنهج الهجين قيود النماذج التقليدية وتقنيات التعلم الآلي عند استخدامها بشكل منفرد. يُسدّ هذا البحث ثغرةً في الدراسات السابقة من خلال تطبيق هذا الإطار المبتكر على العراق، مُزوّداً صانعي السياسات بتوقعاتٍ أكثر دقةً لتخطيط الرعاية الصحية والخدمات الاجتماعية. تُشير النتائج إلى أن النموذج الهجين يُقدّم تنبؤاتٍ أكثر دقةً وواقعيةً مقارنةً بنموذجي Prophet وXGBoost المُنفصلين. مع ذلك، تُظهر هذه الدراسة أن النماذج الهجينة تُحسّن من توقعات متوسط العمر المتوقع في العراق. ينبغي أن تُراعي الأبحاث المستقبلية تأثيرات الرعاية الصحية والاجتماعية والسياسية والبيئية. يُمكن لهذه النماذج أن تُسهم في وضع سياسات صحية أفضل في العراق.

الكلمات المفتاحية: التنبؤ بالسلاسل الزمنية، النموذج الهجين، نموذج Prophet، نموذج XGBoost، التنبؤ بمتوسط العمر المتوقع

1. Introduction

Time series forecasting has become a fundamental task across various fields such as finance, economics, healthcare, and supply chain management, where the accurate prediction of future trends is essential for effective decision-making. In these domains, reliable forecasting models are critical for predicting demand, identifying market trends, managing resources, and preparing for potential disruptions. As a result, the development of more sophisticated forecasting models, combining classical statistical techniques and machine learning algorithms, has gained significant attention. Among the most notable models are Prophet, a time series forecasting tool developed

by Facebook, and XGBoost, an advanced gradient boosting algorithm that has proven highly effective in handling complex datasets. Prophet, by (Taylor and Letham 2018: 41), is a time series tool that handles trends, seasonality, and holidays. It's easy to use and adapts well to different patterns, including non-linear. On the other hand, XGBoost, introduced by (Chen and Guestrin, 2016: 791), is an ensemble learning method based on gradient boosting that creates decision trees sequentially. XGBoost excels in modeling non-linear relationships and interactions within data, making it highly effective for regression and classification tasks, including time series forecasting. While both Prophet and XGBoost have demonstrated strong individual performances, recent studies have explored the potential benefits of hybridizing these two methods to improve forecasting accuracy. By combining Prophet's ability to model seasonality and trends with XGBoost's capacity to capture residual errors and complex non-linearities, hybrid models aim to address the limitations of each individual method. For example, (wang et al., 2022: 150) show that a hybrid Prophet-XGBoost model can outperform each model in isolation. (Odhiambo et al., 2024: 111) propose a hybrid ARIMA-XGBoost model for time series forecasting, highlighting the improvement in accuracy when integrating statistical and machine learning approaches. Similarly, (Zeng et al., 2025: 5) demonstrate the effectiveness of combining Prophet – BO – XGBoost for energy consumption forecasting, achieving superior performance over traditional models.

Life expectancy is a critical indicator of a nation's overall health and well-being, reflecting the interplay of socioeconomic, environmental, and healthcare factors. Accurate prediction of life expectancy is essential for policymakers to design effective public health interventions, allocate resources efficiently, and plan for future demographic changes. Traditional statistical methods for life expectancy forecasting often struggle to capture the complex, non-linear relationships inherent in the data, particularly in regions with volatile socioeconomic conditions, such as Iraq. To address these challenges, this study applies a hybrid forecasting framework that integrates Prophet, a time series decomposition tool developed by Facebook, with XGBoost, a powerful machine learning algorithm known for its ability to model non-linear relationships. Forecasting life expectancy at birth in Iraq is complex due to the influence of various socio-economic, healthcare, and

demographic factors, including age dependency ratio, infant mortality rate, population growth, and GDP per capita. Traditional forecasting models like Prophet may struggle to capture the complex relationships between these factors, while machine learning models like XGBoost may overlook temporal dynamics. This study aims to develop a hybrid forecasting model that integrates Prophet and XGBoost to more accurately predict life expectancy by accounting for both temporal trends and the influence of multiple variables. This study proposes a hybrid model combining Prophet for trend and seasonality modelling with XGBoost for capturing non-linear relationships and residuals. It incorporates key socio-economic and demographic variables influencing life expectancy in Iraq, this study contributes to the literature by applying this hybrid model to life expectancy forecasting in Iraq, a region where such an approach has not been extensively explored. Although, the current provides policymakers with more accurate life expectancy predictions, which are crucial for planning healthcare and social services. It also offers a flexible forecasting approach that can be applied in other developing countries facing similar socio-economic challenges. The research fills a gap in the literature on life expectancy forecasting in Iraq and provides a framework for future studies in this area. So that the primary objective of this study is to develop a robust and accurate hybrid forecasting model for predicting life expectancy at birth in Iraq and integrate the time series decomposition capabilities of Prophet with the non-linear modeling strength of XGBoost to improve forecasting accuracy for life expectancy. Additionally, it seeks to incorporate key socio-economic and demographic variables, such as age dependency ratio, infant mortality rate, fertility rate, GDP per capita, and population growth, to capture their impact on life expectancy. This study combines statistical and machine learning methods to improve the accuracy and reliability of life expectancy predictions, overcoming the weaknesses of using either approach alone.

2. Methodology

2-1. Prophet's Additive Model: Prophet is a robust time series forecasting tool developed by Facebook’s Core Data Science team. It is designed to handle time series data that exhibit strong seasonal effects, trends, and holidays. Prophet is built on an additive model that allows for flexibility in modelling complex seasonalises and trend changes, making it suitable for a

variety of forecasting tasks in different domains, such as finance, retail, and healthcare (Taylor, & Letham, 2018: 39).

Prophet is an additive model, meaning that it decomposes the time series into several components: trend, seasonal, and holiday effects. It assumes that the observed time series y_t at time t can be modelled as a sum of three components:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t \quad y_t = g(t) + s(t) + h(t) + \backslash epsilon_t \qquad (1)$$

where:

- $g(t)$ represents the trend component, which models the underlying trend of the data.
- $s(t)$ represents the seasonal component, which captures the periodic fluctuations due to seasonality.
- $h(t)$ represents the holiday component, which captures the effects of special events such as holidays or promotions.
- ϵ_t is the noise or error term at time t, assumed to be normally distributed with zero mean and constant variance.

A. **Trend Component $g(t)$**: The trend component of Prophet is modelled using piecewise linear or logistic growth curves. The choice of growth type depends on whether the time series exhibits a linear trend or has a saturating growth pattern (e.g., logistic growth) (Taylor, & Letham, 2018:39).

Linear Trend: For linear growth, the model assumes that the trend at time t, $g(t)$, follows a linear function of time (Taylor, & Letham, 2018:39):

$$g(t) = \beta_0 + \beta_1 \, tg(t) \qquad (2)$$

where:

- β_0 is the intercept (the starting value of the time series).
- β_1 is the slope of the trend (the rate of change in the time series over time).
- t is the time.

Logistic Growth:

For logistic growth, the model assumes that the trend follows a logistic curve, which is particularly useful for modelling processes that exhibit saturation:

$$g(t) = \frac{C}{1 + \exp\left(-k(t - t_0)\right)} \qquad (3)$$

where:

- C is the carrying capacity (the maximum value that the series can reach).

- k is the growth rate.
- t_0 is the inflection point (the time at which the growth rate is maximized)?

In practice, Prophet automatically selects between these two growth types based on the input data (Taylor & Letham, 2018: 39).

B. Seasonal Component $s(t)$: The seasonal component models the periodic fluctuations in the time series. Prophet supports both yearly and weekly seasonalities, and it allows the user to specify custom seasonalities if needed. Prophet uses Fourier series to capture the periodic patterns in the data (Taylor, & Letham, 2018:39; Karim, & Ahmed, 2023: 683).

Yearly Seasonality: The yearly seasonal component is modelled using a Fourier series expansion:

$$s(t) = \sum_{i=1}^{k} \left[a_i \cos\left(\frac{2\pi it}{365}\right) + b_i \sin\left(\frac{2\pi it}{365}\right) \right] \quad (4)$$

where:

- a_i and b_i are the Fourier coefficients that determine the amplitude of the seasonal fluctuations.
- k is the number of Fourier terms used (determines the complexity of the seasonal pattern).
- the factor $\frac{2\pi it}{365}$ normalizes the time to account for yearly periodicity, assuming there are 365 days in a year (Taylor, & Letham, 2018:39).

Weekly Seasonality:

Similarly, weekly seasonality can be modelled using a Fourier series expansion with a period of 7 days (Zhang, 2003:162):

$$s(t) = \sum_{i=1}^{k} \left[a_i \cos\left(\frac{2\pi it}{7}\right) + b_i \sin\left(\frac{2\pi it}{7}\right) \right]$$

C. Holiday Component $h(t)$

The holiday component models the effect of special events, such as holidays or specific promotions, on the time series. The holiday effect is modelled as a set of binary indicator variables that indicate whether a holiday occurs at time t:

$$h(t) = \sum_{i=1}^{m} \delta_i \,.\, 1(t \in \text{Holiday i}) \quad (5)$$

where:

- δ_i is the impact of holiday i on the time series.
- $1(t \in Holiday\ i)$ is an indicator function that takes the value 1 if t corresponds to holiday i, and 0 otherwise.

Users can specify a list of holidays and their corresponding effects, and Prophet will automatically incorporate these effects into the model (Zhang, 2003:162)

Error Term ϵ_t: The error term ϵ_t represents the unexplained variance in the data after accounting for the trend, seasonality, and holidays. It is assumed to be independently and identically distributed (i.i.d.) with zero mean and constant variance (Zhang, 2003: 170):

$$\epsilon_t \sim N(0, \sigma^2)$$

2-2. Model Estimation and Fitting: Prophet uses a Bayesian approach to estimate the parameters of the model, allowing for uncertainty quantification. The parameters include the trend growth rates, seasonalities, and holiday effects. The model's parameters are estimated by maximizing the posterior distribution of the model, using Markov Chain Monte Carlo (MCMC) or optimization algorithms.

The process involves:

A. Defining the likelihood function for the time series based on the model structure.
B. Estimating the parameters using MCMC methods or optimization techniques.
C. Generating forecasts with uncertainty intervals.

$$\hat{y}_t = g(t) + s(t) + h(t) \qquad (6)$$

where the future values of $g(t)$, $s(t)$ and $h(t)$ are predicted based on the historical data and the estimated model parameters (Taylor, & Letham, 2018:39).

2-3. Uncertainty and Forecast Intervals: One of the key features of Prophet is the ability to generate uncertainty intervals for the forecasts. The uncertainty is derived from the model parameters, which are estimated with a Bayesian approach. Prophet typically uses a 95% confidence interval for its forecasts, providing a range of possible future values (Taylor, & Letham, 2018: 42). The uncertainty intervals are generated by sampling from the posterior distribution of the model parameters, and the forecast at time tt is represented as:

Doi: www.doi.org/10.25130/tjaes.21.72.1.24

$$\hat{y}_t \pm 1.96 \, . \sigma_{\hat{y}_t} \qquad (7)$$

where $\sigma_{\hat{y}_t}$ is the standard deviation of the forecast at time t.

2-4. Visualizing Prophet Forecasts: Prophet provides a built-in visualization tool that allows users to visualize the individual components of the forecast (trend, seasonality, and holiday effects), along with the overall forecast and uncertainty intervals. These visualizations help in understanding the contributions of different components to the final forecast (Shmueli et al., 2016:133).

2-5. XGBoost (EXtreme Gradient Boosting): XGBoost, short for eXtreme Gradient Boosting, is a highly effective and popular machine learning algorithm designed for both regression and classification tasks. It is based on the gradient boosting framework and is widely recognized for its performance in terms of accuracy and scalability. XGBoost is particularly effective in handling large datasets with complex relationships and is commonly employed in various machine learning competitions and real-world applications. This detailed theory discusses the underlying principles of XGBoost, its ensemble approach, and the key mathematical concepts involved (Chen & Guestrin, 2016: 761).

Gradient boosting is a machine learning technique that builds an ensemble of decision trees sequentially, with each subsequent tree attempting to correct the errors made by its predecessor. The general idea behind gradient boosting is to minimize the residual error of a model by iteratively improving the predictions. The core concept of gradient boosting involves fitting new models to the residual errors of previous models (Friedman, 2001: 1195).

In gradient boosting, the model prediction $\hat{y}_t \backslash hat \{y\}\_t$ is the sum of the predictions of all trees up to the current stage t:

$$\hat{y}_t = \sum_{i=1}^{t} f_i(x) \qquad (8)$$

where:

- $\hat{y}_t$ is the prediction for the t^{th} iteration.
- $f_i(x)$ represents the prediction of the i^{th} tree.
- x is the input feature vector.

The objective is to minimize a loss function that measures the difference between the true values y and the predicted values $\hat{y}_t$. The key idea is to minimize the residual errors at each stage (Chen & Guestrin, 2016:761).

2-6. Boosting and Ensemble Learning: The boosting process is sequential, where each new model is trained to correct the errors of the combined model of previous trees. A key distinction of boosting is its ability to adaptively focus on the examples that are difficult to predict. The core principle of boosting is that weak learners (typically shallow decision trees) can be combined to create a strong learner.

In the case of XGBoost, the individual trees are trained using a gradient descent optimization method to minimize the loss function. The sequential process ensures that new trees focus on the residuals or errors made by the previous trees. This approach leads to improved predictive performance, especially when the data is complex or noisy (Chen & Guestrin, 2016:761; Friedman, 2001:1195).

2-7. Mathematics Behind XGBoost: XGBoost enhances the gradient boosting algorithm by introducing regularization techniques, parallelization, and a more efficient optimization framework. The primary goal in XGBoost is to minimize the following objective function (Chen & Guestrin, 2016:761):

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (9)$$

where:

- $\mathcal{L}(\theta)$ is the objective function.
- $L(y_i, \hat{y}_i)$ is the loss function that measures the error between the value y_i and the predicted value $\hat{y}_i$ for the i^{th} instance.
- $\Omega(f_k)$ is a regularization term that penalizes the complexity of the k^{th} tree, ensuring that the trees are not overly complex and reducing over fitting.
- N is the number of data points, and K is the number of trees in the ensemble.

2-8. Loss Function and Gradient Descent: In gradient boosting, the goal is to minimize the loss function LL. Common loss functions for regression include mean squared error (MSE), and for classification tasks, the logarithmic loss is used. The key idea in XGBoost is to use gradient descent to minimize this loss function (Friedman, 2001: 1195).

The loss function L is often differentiated with respect to the predicted value $\hat{y}_i$, and the gradient of the loss is used to update the predictions. if ${\hat{y}_i}^{(t)}$ represents the prediction from the t^{th} iteration, the gradient of the loss with respect to ${\hat{y}_i}^{(t)}$ is computed (Chen & Guestrin, 2016:761):

$$g_i{}^{(t)} = \frac{\partial L\left(y_i, {\hat{y}_i}^{(t)}\right)}{\partial {\hat{y}_i}^{(t)}} \qquad (10)$$

where $g_i{}^{(t)}$ is the gradient of the loss function.

The gradient represents the direction of the steepest increase in the loss function, and the optimization process involves moving in the opposite direction to minimize the loss.

2-9. Additive Model and Tree Construction: XGBoost constructs trees in a sequential manner, with each tree $f_k(x)$ attempting to correct the errors made by the previous trees. The model is an additive model, and the final prediction is the sum of the predictions from all trees (Chen & Guestrin, 2016:761):

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \qquad (11)$$

where:

- $\hat{y}_i$ is the final prediction for the i^{th} data point.
- $f_k(x_i)$ is the prediction from the k^{th} tree for the i^{th} data point.

The key idea in XGBoost is that each tree is learned in a way that minimizes the residual errors from the previous trees. XGBoost employs a sophisticated algorithm to learn these trees efficiently by using the gradient of the loss function at each stage (Chen & Guestrin, 2016:761)

2-10. Regularization Term: XGBoost introduces a regularization term Ωf_k to prevent overfitting and control the complexity of the model. The regularization term typically takes the form of:

$$\Omega f_k = \gamma T + \frac{1}{2}\lambda \sum_{i=1}^{T} w_j^2 \qquad (12)$$

where:

- T is the number of leaves in the tree.
- w_j is the weight of the j^{th} leaf in the tree.
- γ and λ are regularization parameters that control the complexity of the tree.

The regularization term penalizes trees with many leaves or large leaf weights, preventing the model from fitting noise and improving generalization (Friedman, 2001:1195).

2-11. Gradient Boosting with Second-Order Approximation: In addition to first-order gradients, XGBoost also uses second-order derivatives (Hessians) of the loss function to improve the efficiency of the optimization process. The second-order approximation allows for more accurate updates, leading to faster convergence. The second-order Taylor expansion of the loss function around the current model is:

$$L\left(y_i, \hat{y}_i^{(t)}\right) \approx L\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i^{(t)}\left(\hat{y}_i^{(t)} - \hat{y}_i^{(t-1)}\right) + \frac{1}{2} h_i^{(t)}\left(\hat{y}_i^{(t)} - \hat{y}_i^{(t-1)}\right)^2 \quad (13)$$

where:

- $g_i^{(t)}$ is the gradient (first derivative).
- $h_i^{(t)}$ is the Hessian (second derivative).

The inclusion of the Hessian allows XGBoost to optimize the model more efficiently, improving convergence speed and predictive accuracy (Friedman, 2001: 1195).

XGBoost is known for its scalability and speed, making it suitable for large datasets. It can be implemented in parallel, meaning that it can efficiently handle multi-core processors during training. XGBoost also supports various hyperparameters that can be fine-tuned to improve performance, such as the learning rate, tree depth, and number of estimators (Chen & Guestrin, 2016: 761; Friedman, 2001: 1195).

2-12. Hybrid Approach: Combining Prophet and XGBoost: The hybrid Prophet-XGBoost model combines the strengths of both approaches by using Prophet to model the seasonality and trend components of the time series, and XGBoost to capture any residual patterns, non-linearities, or complex relationships that Prophet may not fully account for. The process typically involves two main steps: decomposition using Prophet and then correction of residuals using XGBoost (Zhang, 2003:169).

2-12-1 Step 1: Decompose the Time Series Using Prophet: In the first step, Prophet decomposes the time series into its trend, seasonal, and holiday components. The goal here is to extract these components in such a way that the residual series, which consists of noise or unexplained variation, can be

modelled effectively by XGBoost (Cleveland et al., 190:61; Zhang, 2003: 169).

Let the time series be represented as:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t \quad (14)$$

where $g(t)$, $s(t)$, and $h(t)$ are the trend, seasonal, and holiday components, respectively. the residual component is:

$$\epsilon_t = y_t - \big(g(t) + s(t) + h(t)\big) \quad (15)$$

Prophet models (t), $s(t)$, and $h(t)$, while the residual ϵ_t represents the part of the time series that is unexplained by these components (Zhang, 2003:169).

2-12-2. Step 2: Model Residuals Using XGBoost: Once the residuals ϵt are obtained from the Prophet decomposition, XGBoost is applied to model these residuals. XGBoost can capture non-linearities and interactions between features that may not have been modelled by Prophet, such as complex seasonal variations or other factors contributing to the error term. This can improve the accuracy of the overall model by addressing patterns not accounted for by Prophet's additive components (Makridakis, 2018:82). The residual model can be expressed as:

$$\epsilon_t = \sum_{k=1}^{K} f_k(x_t) \quad (16)$$

where x_t represents the input features for XGBoost, which may include time-related features (e.g., day of the week, month, holidays), and $f_k(x_t)$ represents the prediction from the k^{th} tree (SHMUELI et al., 2016:135).

The final prediction is the combination of the predictions from Prophet and XGBoost:

$$\hat{y}_t = g(t) + s(t) + h(t) + \sum_{k=1}^{K} f_k(x_t) \quad (17)$$

where $g(t) + s(t) + h(t)$ is the prediction from Prophet and $\sum_{k=1}^{K} f_k(x_t)$ is the correction made by XGBoost on the residuals (SHMUELI et al., 2016: 135).

2-13. Advantages of the Hybrid Model: The hybrid approach of Prophet and XGBoost offers several advantages (Ahmed et al. 2010: 601; Ali et al., 2022: 441):

- **Capturing Trend and Seasonality**: Prophet effectively handles seasonal patterns and trend changes in the data. It allows for flexible trend modelling, including logistic growth, which is often crucial for datasets with saturating growth.
- **Handling Residuals with XGBoost**: After Prophet captures the trend and seasonality, XGBoost models the residuals, which may include complex non-linear relationships that Prophet cannot capture. XGBoost's ability to model interactions between features allows it to handle more intricate patterns in the residuals.
- **Improved Forecasting Accuracy**: By combining both models, the hybrid approach results in a more accurate forecast compared to using either method alone, especially in cases where the time series contains complex non-linear relationships that are difficult for additive models like Prophet to capture.
- **Scalability and Flexibility**: The XGBoost model is highly scalable and efficient, capable of handling large datasets, making it a good fit for time series with a high volume of data points. Additionally, Prophet is flexible in terms of adding custom seasonalities and holidays, making the model adaptable to different forecasting scenarios.

2-14 Mathematics of the Hybrid Model: The mathematical formulation of the hybrid model can be summarized as follows (Hyndman, & Athanasopoulos, 2018: 1120):

1. Decompose the Time Series using Prophet:
   $y_t = g(t) + s(t) + h(t) + \epsilon_t$
2. Fit XGBoost to the Residuals:
   $$\epsilon_t = \sum_{k=1}^{K} f_k(x_t)$$
3. Final Prediction:
   $$\hat{y}_t = g(t) + s(t) + h(t) + \sum_{k=1}^{K} f_k(x_t) \qquad (18)$$

where $g(t)$, $s(t)$, and $h(t)$ are the components predicted by the Prophet, and $\sum_{k=1}^{K} f_k(x_t)$ is the correction made by XGBoost to the residuals (Hyndman, & Athanasopoulos, 2018:1120).

2-15. Hyper parameter Tuning in the Hybrid Model: In the hybrid Prophet-XGBoost model, there are several hyper parameters that need to be tuned to optimize the performance. For Prophet, key hyperparameters include:

❖ Seasonality Prior Scale: Controls the flexibility of the seasonal component.
❖ Trend Changepoint Prior Scale: Controls the flexibility of the trend component.
❖ Holidays: Custom holidays can be added to capture special events.

For XGBoost, key hyperparameters include:

- Number of Trees (n_estimators): Controls the number of decision trees in the ensemble.
- Learning Rate: Determines the contribution of each tree to the final prediction.
- Maximum Depth of Trees: Controls the complexity of individual decision trees.
- Regularization Parameters lambda, and alpha which are ($g(t)$, $s(t)$, and $h(t)$ are the components predicted by the Prophet, and $\sum_{k=1}^{K} f_k(x_t)$ is the correction made by XGBoost to the residuals): Control the regularization of the model to prevent overfitting.

The hybrid model requires careful tuning of both Prophet and XGBoost parameters to achieve the best predictive performance [5], [9].

The hybrid Prophet-XGBoost model combines the strengths of both Prophet and XGBoost to improve forecasting accuracy for time series data with complex seasonalities, trends, and non-linearities. By decomposing the time series using Prophet and modeling the residuals with XGBoost, this hybrid approach can achieve better performance than either model used independently. Its flexibility, scalability, and ability to handle various types of time series data make it a powerful tool for time series forecasting tasks across different domains (Wolpert, 1992).

3. Application Part

3-1. Data Description: This study utilizes key socio-economic and demographic variables to analyse and forecast life expectancy at birth in Iraq. These variables provide a comprehensive understanding of health trends, economic conditions, and population dynamics. The data is sourced from global economic and health databases. The variables are; Life expectancy at birth, measured in years, which represents the average number of years a newborn is expected to live under current mortality conditions. It serves as a key indicator of a country's overall health, socio-economic development, and quality of life. Higher life expectancy generally reflects improvements in healthcare, economic stability, and living standards. The other variable is crude death rate, expressed as the number of deaths per 1,000 people,

provides an overview of a population's overall mortality trends. Monitoring this variable helps assess the general health status of the population. Fertility rate is another variable, that measured as the average number of births per woman over her lifetime, is a critical demographic factor influencing population growth and social policies. While the variable infant mortality rate, particularly among males, is an essential health indicator measuring the number of male infant deaths per 1,000 live births. This variable is crucial in assessing gender-based disparities in child survival and healthcare access. Although, the variable GDP per capita, measured in current U.S. dollars, represents the total economic output per person in a country without adjusting for inflation. This variable is a key determinant of living standards. The last variable in this study is proportion of the population aged 65 and above, which expressed as a percentage of the total population, highlights demographic aging trends. The data utilized in this study was obtained from the World Bank, specifically through its comprehensive online platform available at www.albankaldawli.org. This website provides access to a wide range of global development indicators. These indicators are compiled and regularly updated by the World Bank to support research.

3-2. Results of Prophet Model: The Prophet model summary indicates a linear growth trend with 25 changepoints detected between 1962 and 1999, suggesting structural changes in the data at these points. The changepoint prior scale (0.05) controls the model's flexibility in detecting these changes, while the seasonality mode is additive, with yearly seasonality automatically determined and modelled using a Fourier order of 10. No holiday effects were included. The model incorporates extra regressors (Death Rate, Fertility Rate, Mortality Rate, GDP per Capita, and Population Ages 65+) all standardized and influencing the trend in an additive manner. The estimated trend parameter ($k = -0.539$) suggests a declining long-term trajectory, while $m = 0.106$ serves as an intercept term. The Stan optimization output confirms a successful fit (return_code = 0) with low observation noise (sigma_obs = 0.0023). The beta coefficients indicate the impact of seasonality and regressors, showing mixed positive and negative effects. Overall, this model captures long-term trends, structural shifts, and the influence of key socioeconomic factors on the target variable.

3-3. Results of XGBoost Model: The XGBoost model indicates that it was trained using the reg:squarederror objective, meaning it is designed for regression tasks by minimizing the squared error loss. The training process

involved 100 iterations (niter = 100) with 5 features, and the model's performance was tracked using an evaluation log that recorded the Root Mean Squared Error (RMSE) at each iteration. The RMSE started at 44.66 in the first iteration and progressively decreased to 0.0013 at the final iteration, suggesting that the model achieved an exceptionally strong fit to the training data. Enabling parameter validation (validate parameters = TRUE) guarantees that the supplied parameters are verified for accuracy, while the inclusion of callbacks indicates that the model's performance was actively tracked during training, after using 20.3% of the testing set data. However, the extremely low RMSE raises a potential concern of overfitting, as such a drastic reduction in error might indicate that the model has memorized the training data rather than learning a generalizable pattern. To confirm whether this model generalizes well, it is essential to evaluate its performance on an unseen validation or test dataset. If the RMSE remains consistently low on new data, the model is likely robust, but if the error increases significantly, regularization techniques, additional feature selection, or hyperparameter tuning might be required to improve its generalization ability.

3-4. Results of Meta Model (Hybrid Model): The output from table 1, provides detailed insights into the performance and significance of the meta-model, which combines predictions from Prophet and XGBoost to improve forecasting accuracy, as shown below:

Table (1): Results of Meta Model

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	200.44516	76.434334	2.622	0.0255 *
Prophet	0.022964	0.009358	2.454	0.0340 *
XGBoost	-2.003793	1.141234	-1.756	0.1096
Residual standard error: 0.8406			10 degrees of freedom	
Multiple R-squared: 0.6936			Adjusted R-squared: 0.6324	
F-statistic: 11.32, on 2 and 10 DF			p-value: 0.002699	
Residuals:				
Min = -1.403	1Q = -0.437	Median = -0.085	3Q = 0.331	Max = 1.414

Table 1 explains the output from the linear regression model fitted using R provides a detailed summary of how well the predictors Prophet and XGBoost explain the variability in the Life expectancy variable. The residuals, which represent the differences between the observed and predicted values, range from a minimum of -1.40266 to a maximum of

1.41411, with a median value close to zero at -0.08531. The spread of the residuals suggests that the model performs well for most predictions but struggles with some observations, especially those further from the mean. The model's intercept is significant, with an estimated value of 200.44516 and a p-value of 0.0255, indicating that when both Prophet and XGBoost are zero, the expected value of Actual is significantly different from zero. The Prophet prediction is also statistically significant with a p-value of 0.0340, indicating that for each unit increase in Prophet prediction, there is a significant positive change in meta-model's prediction. In contrast, the XGBoost prediction has a negative coefficient of -2.00379, but its p-value of 0.1096 suggests that it is not statistically significant in meta-model's prediction at the 5% level. The meta-model assigns a significant positive weight to Prophet's predictions but a non-significant negative weight to XGBoost's predictions. This suggests that Prophet's predictions are more reliable in this context, while XGBoost's predictions may not add significant value. The residual standard error of 0.8406 indicates that the meta-model's predictions are relatively close to the actual values, but there is still some variability in the residuals. The multiple R-squared value of 0.6936 indicates that 69.36% of the variance in the actual values is explained by the meta-model. This is a relatively high value, suggesting that the model performs well. The F-statistic of 11.32, along with its p-value of 0.002699, this tests the overall significance of the model. A higher F-statistic indicates that the model is a better fit for the data, and the p-value is less than 0.05, confirming that the meta-model is statistically significant. meaning at least one of the predictors (Prophet or XGBoost) has a meaningful relationship with life expectancy.

Table (2): presents the values of prediction of Prophet, XGBoost, and Hybrid Models

Year	Actual	Prophet Predict	XGBoost Predict	Hybrid Predict
2011	67.659	68.42596	66.8639	68.03512
2012	68.023	70.64498	67.06063	67.69187
2013	68.253	73.8696	66.77498	68.33831
2014	68.914	78.67688	66.30568	69.38908
2015	69.44	84.42417	66.30568	69.52106
2016	68.988	90.69618	66.30568	69.6651

Doi: www.doi.org/10.25130/tjaes.21.72.1.24

Year	Actual	Prophet Predict	XGBoost Predict	Hybrid Predict
2017	70.413	99.71661	66.30568	69.87224
2018	71.514	109.62957	66.30568	70.09989
2019	71.576	119.57223	66.30568	70.32822
2020	69.123	128.17026	66.30568	70.52566
2021	70.378	140.7638	66.30568	70.81487
2022	71.336	153.26548	66.30568	71.10196
2023	71.13	164.6605	66.30568	71.36364

Table 2 compares actual life expectancy values with predictions from Prophet, XGBoost, and the hybrid model for the years 2011 to 2023. Prophet's predictions show a strong upward trend, starting at 68.43 in 2011 and reaching 164.66 by 2023, but they significantly overestimate life expectancy in later years. In contrast, XGBoost's predictions remain stable around 66.30 for all years, failing to capture the upward trend and providing little variation. The hybrid model, which combines the strengths of Prophet and XGBoost, produces more accurate and realistic predictions. For example, in 2011, the hybrid prediction is 68.04, closer to the actual value of 67.66 than Prophet's 68.43 or XGBoost's 66.86. By 2023, the hybrid prediction is 71.36, much more reasonable than Prophet's 164.66 and closer to the actual value of 71.13. Overall, the hybrid model effectively balances Prophet's trend-capturing ability with XGBoost's stability, resulting in predictions that are consistently closer to the actual values and demonstrating the value of a hybrid approach for life expectancy forecasting.

Table (3): Models Evaluation

Models	RMSE Value
Prophet	47.51224
XGBoost	3.642033
Hybrid	0.7372452

Table 3 evaluates three models (Prophet, XGBoost, and the Hybrid model) based on their Root Mean Squared Error (RMSE) values, which measure prediction accuracy. Prophet has the highest RMSE (47.51224), indicating significant deviations from actual values, likely due to its over-reliance on trend and seasonality without effectively incorporating other factors. XGBoost performs better with an RMSE of 3.642033, reflecting its

Doi: www.doi.org/10.25130/tjaes.21.72.1.24

ability to model non-linear relationships, but it is still outperformed by the hybrid model. The Hybrid model achieves the lowest RMSE (0.7372452), demonstrating superior accuracy by combining the strengths of Prophet (trend and seasonality) and XGBoost (non-linear relationships). This highlights the hybrid model's effectiveness in minimizing prediction errors and producing reliable forecasts, making it the best choice for life expectancy prediction.

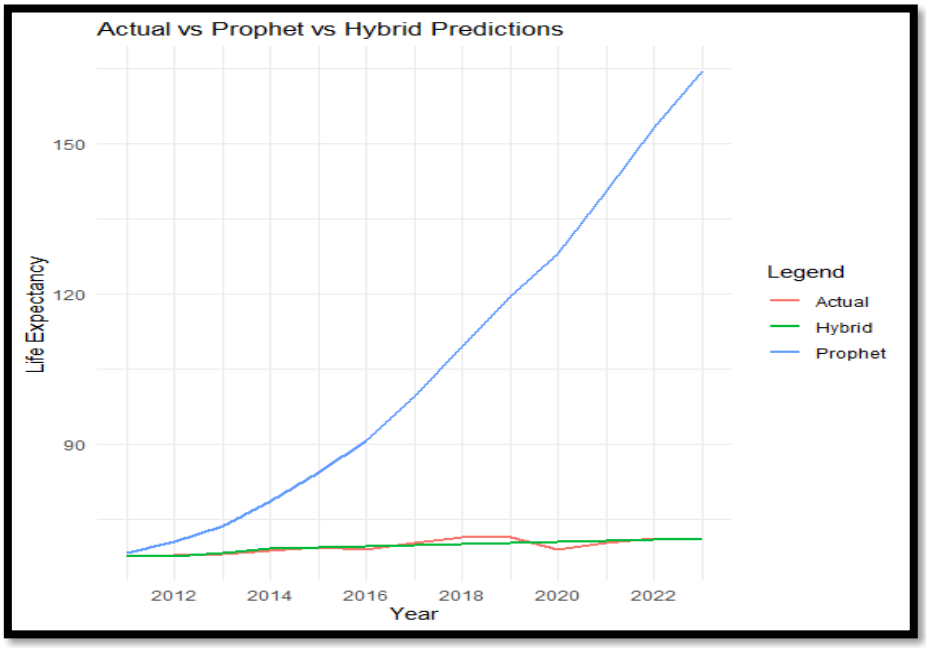

Figure (1): Comparison between Actual, Prophet, and Hybrid Prediction Values

The figure 1 compares three sets of data: Actual life expectancy values, predictions made by the Prophet model, and predictions made by a Hybrid model. The figure likely demonstrates how closely the Prophet and Hybrid models' predictions align with the actual life expectancy data. The Hybrid model's predictions may be closer to the actual values, suggesting that it performs better than the Prophet model alone in this context. This comparison can help in understanding the effectiveness of different forecasting approaches as shown in table 4.

Table (4): Presents Future Forecast (2024-2033)

Year	Prophet Forecast	XGBoost Forecast	Hybrid Forecast
2024	71.33482	71.12963	71.12962
2025	71.48803	71.12963	71.12957
2026	71.60714	71.12963	71.12954

Doi: www.doi.org/10.25130/tjaes.21.72.1.24

Year	Prophet Forecast	XGBoost Forecast	Hybrid Forecast
2027	71.719	71.12963	71.1295
2028	71.82363	71.12963	71.12947
2029	71.97684	71.12963	71.12943
2030	72.09596	71.12963	71.12939
2031	72.20781	71.12963	71.12936
2032	72.31244	71.12963	71.12933
2033	72.46565	71.12963	71.12928

Table 4 forecasts life expectancy from 2024 to 2033 using Prophet, XGBoost, and Hybrid models. Prophet predicts a steady increase, starting at 71.33 in 2024 and reaching 72.47 by 2033. XGBoost forecasts a constant 71.13 for all years, indicating no expected change. The Hybrid model closely mirrors XGBoost but shows a minimal decrease, ending at 71.13 in 2033. Hybrid is optimistic about rising life expectancy, while XGBoost and Prophet are more conservative, with Hybrid slightly adjusting XGBoost's predictions. These highlights differing approaches to forecasting future trends.

4. **Conclusion:** This study demonstrates the effectiveness of a hybrid forecasting model that integrates Prophet and XGBoost for predicting life expectancy in Iraq. By combining Prophet's ability to capture trends and seasonality with XGBoost's strength in modeling non-linear relationships, the hybrid model achieves optimal accuracy compared to the individual models. The results show that the hybrid model significantly reduces prediction errors, as evidenced by its lower Root Mean Squared Error (RMSE) of 0.7372452, outperforming both Prophet (RMSE: 47.51224) and XGBoost (RMSE: 3.642033). The hybrid approach effectively balances the strengths of each model, providing more reliable and realistic forecasts that align closely with actual life expectancy values. The findings highlight the importance of integrating statistical and machine learning techniques to address the limitations of traditional forecasting methods, particularly in complex scenarios influenced by socio-economic and demographic factors. This research not only fills a gap in the literature on life expectancy forecasting in Iraq but also offers a flexible framework that can be adapted to other developing countries facing similar challenges.

Despite the promising results, the hybrid model has several limitations. Its accuracy depends on the quality and completeness of the data,

and incomplete or inaccurate data could impact predictions. The model's generalizability to other regions with different socio-economic conditions may also be limited. Additionally, there is a risk of overfitting, especially with XGBoost, which could reduce its ability to generalize. The complexity of the hybrid model makes it harder to interpret the contributions of each component. Moreover, the study did not explore hyperparameter optimization, which could improve performance. Lastly, the model may not account for unpredictable events, such as political changes or pandemics, that affect life expectancy. Future research should address these limitations to enhance the model's robustness and applicability.

References

1. Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. Econometric reviews, 29(5-6), 594-621.
2. Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition. J. off. Stat, 6(1), 3-73.
3. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.
4. Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
5. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. PloS one, 13(3), e0194889.
6. Shmueli, G., & Lichtendahl Jr, K. C. (2016). Time Series Forecasting With R.
7. Taylor, S. J., & Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37-45.
8. Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.
9. Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.
10. Ali, T. H., Mahmood, A. P. D. S. H., & Wahdi, A. P. D. A. S. (2022). Using Proposed Hybrid method for neural networks and wavelet to estimate time series model. Tikrit Journal of Administrative and Economic Sciences, 18(57 part 3).
11. Karim, A. J. M., & Ahmed, N. M. (2023). Multivariate time series analysis of COVID-19 Pandemic and gold price by using Error Corrections Model. Tikrit Journal of Administrative and Economic Sciences, 19(64 part 1).
12. Wang, J., Du, X., & Qi, X. (2022). Strain prediction for historical timber buildings with a hybrid Prophet-XGBoost model. Mechanical Systems and Signal Processing, 179, 109316.
13. Odhiambo, S. O., Nyakundi, C., & Waititu, H. (2024). Developing a Hybrid ARIMA-XGBOOST Model for Analysing Mobile Money Transaction Data in Kenya. Asian Journal of Probability and Statistics, 26(10), 10-9734.

14. Zeng, S., Liu, C., Zhang, H., Zhang, B., & Zhao, Y. (2025). Short-Term Load Forecasting in Power Systems Based on the Prophet–BO–XGBoost Model. Energies, 18(2), 227.
15. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).