axioms

MDPI

Article

Mortality Forecasting Using LSTM-CNN Model

Ning Zhang [1,2,*], Jingyang Chen [1], Hao Chen [3] and Jingzhen Liu [3,4]

[1] School of Finance, Central University of Finance and Economics, Beijing 102206, China; 2024210375@email.cufe.edu.cn
[2] Central University of Finance and Economics Chinese Fintech Research Center, School of Finance, Central University of Finance and Economics, Beijing 102206, China
[3] School of Insurance, Central University of Finance and Economics, Beijing 102206, China; 2022110072@email.cufe.edu.cn (H.C.); jzliu@cufe.edu.cn (J.L.)
[4] China Institute for Actuarial Science, Central University of Finance and Economics, Beijing 102206, China
* Correspondence:zntop@cufe.edu.cn

Abstract

Accurate mortality prediction is essential to actuarial practice as it is directly linked to insurance pricing, reserving, and the management of longevity risk. This study proposes a deep neural network (DNN) model for the mortality rates of multiple populations; it is composed of long short-term memory (LSTM) and convolutional neural network (CNN) components. As mortality trends evolve over long time horizons, and as capturing the complex dependencies among mortality rates across countries or regions with a linear model is challenging, the LSTM and CNN were applied to mortality modeling. The former can automatically learn long-term dependencies of sequential data, whereas the latter can extract local features from grid or sequential data. Formulated as a nonlinear generalization of the Lee–Carter decomposition, the model maps the log-mortality matrix $\log M$ to future $\log m(x,t)$ end-to-end and generates multi-step forecasts through dynamic recursive prediction. Then, the DNN and baseline models were used to fit mortality data of 21 countries from the Human Mortality Database (HMD), which were divided into training and test sets with the year 2000 as the split point. Extensive numerical experiments from the perspectives of accuracy, stability, and reliability of long-term forecasting revealed that DNN models yield better predictive performance, particularly the LSTM-CNN model. It combines the LSTM, CNN, and fully connected network (FCN) layers and thus exploits each deep neural network to fit nonlinear age, period, and cohort effects as well as their interactive terms to achieve better predictive performance. However, the CNN still outperformed other models for certain groups. In addition, the conclusions hold for remaining life expectancy.

Keywords: longevity; mortality forecasting for multiple populations; deep learning; long short-term memory (LSTM); convolutional neural network (CNN)

MSC: 91G05

Check for updates

Academic Editors: Maria P. Beccar-Varela, Maria C. Mariani and Osei K. Tweneboah

Received: 30 January 2026
Revised: 15 April 2026
Accepted: 22 April 2026
Published: 29 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the rapid progress of medicine, mortality rates have continued to decline and life expectancy to increase. Thus, longevity risk has intensified. However, longevity risk results in increased government expenditure and higher costs of certain insurance products such as endowment insurance and health insurance, which adversely impacts social and economic development and threatens sustainable management of life insurance companies.

Consequently, accurate quantification is the key to managing longevity risk and building an accurate mortality forecasting model is the most critical step.

Currently, there are three stages of time-series mortality forecasting: static mortality, dynamic mortality, and machine learning modeling. Static models only establish the relationship between mortality rates and ages, such as the De Moivre model, the Gompertz model, the Makeham model, the Weibull model, and the Heligman–Pollard model, as presented in Table 1.

Table 1. Summary of expressions and parameters used in static models.

Model	Expression	Parameters
De Moivre	$m(x) = (\omega - x)^{-1}$	ω
Compertz	$m(x) = BC^x$	B, C
Makeham	$m(x) = A + Bc^x$	A(constant), B, C
Weibull	$m(x) = Ax^\gamma$	A, γ
Helligman–Pollard	$\frac{q(x)}{p(x)} = A^{(x+B)^C} + De^{-E(\ln x - \ln F)^2} + GH^x$	A, B, C, D, E, F, G, H, γ

A drawback of the static mortality forecasting model is that it only considers the age effect and cannot capture the dynamic changes in mortality rates over time. Later, Lee and Carter [1] introduced the L-C model to decompose mortality rates into an age-related static life-table component and a time-related dynamic index component by principal component analysis. This is expressed as follows:

$$\log m(x,t) = \alpha_x + \beta_x k_t + \varepsilon_{x,t}. \quad (1)$$

The L-C model is estimated through a two-step procedure, which is well documented in the literature. The first step involves ignoring the latent factor structure and estimating α_x and β_x. In the second step, the latent factor k_t is extrapolated with a time-series model, such as a random walk. In addition, to ensure the identifiability of the model, there are two constraints on the parameters:

$$\sum_x b(x) = 1, \quad \sum_t k(t) = 0. \quad (2)$$

However, errors in the L-C model are normally distributed, which is an unrealistic assumption. This is because the logarithm of the observed force of mortality is much more variable at older ages than at younger ages. Brouhns et al. [2] embedded the L-C model into the Poisson regression model to relax the parametric hypothesis instead of applying the Singular Value Decomposition (SVD) introduced by Lee and Carter [1]; they estimated parameters of the L-C model considering the maximum likelihood. The results showed that the extension yielded more accurate predictions than the initial L-C model. In contrast to the L-C model based on the log-transformation of mortality, Cairns et al. [3] proposed a Cairns–Blake–Dowd (CBD) model based on logistic transformation, which was initially applied to fit the empirical male mortality in England and Wales. The CBD model has been verified as being suitable for forecasting mortality of the elderly population in many countries through a large empirical analysis.

Mortality rates vary over age and time and are also affected by health care, economy, and country; that is, mortality differs depending on age, time, cohort, and country. Lee [4] summarized various extensions of L-C models and their applications by considering gender, region, and race. Further, Renshaw and Haberman [5] also extended the L-C model to a two-factor form, which is as follows:

$$\log m(x,t) = a(x) + \sum_{i=1}^{d} b^i(x)k^i(t) + \varepsilon(x,t). \quad (3)$$

The parameter estimation method of the Renshaw–Haberman model is similar to that of the L-C model. There are also two constraints on the parameters:

$$\sum_x b^i(x) = 1, for\ i\ = 1,2,\ldots,d;\ \sum_t k^i(t) = 0, for\ i\ = 1,2,\ldots,d. \quad (4)$$

Hobcraft et al. [6] discussed the age-period-cohort model (APC), which is widely used in medicine and epidemiology. Later, Willets [7] introduced the APC model in the insurance context to measure the longevity risk. Currie [8] applied the APC model to the mortality modeling, employing the P-spline method to smoothen and reduce the number of free parameters. Renshaw and Haberman [9] extended the L-C model with cohort effect to forecast the different mortality rates of different cohorts, which is estimated by the maximum likelihood. In contrast to the APC model, the Renshaw–Haberman (R-H) model considered the cross effects of time effect and age effect. In a similar manner, Cairns et al. [10] also extended the CBD model to a two-factor form with a cohort effect to enhance its prediction performance. In addition, Plat [11] proposed the Plat model, which summarized the CBD, L-C, and APC models. It can distinguish the improvement effect of mortality over time between the young and the old, and it fits better for all ages than CBD, L-C, and APC models.

For most mortality models mentioned above, a single population has been considered. With the convergence of global mortality levels, Wilson [12] suggested that it may be improper to prepare mortality forecasts for individual national populations in isolation from one another. In addition, Li and Lee [13] argued that mortality patterns and trajectories in closely related populations are likely to be similar in certain respects, and differences are unlikely to increase in the long run. Therefore, the Lee–Carter model should be applied to a group of populations, allowing each its own age pattern and level of mortality, while imposing shared rates of change based on age. They extended the Lee–Carter method and proposed the augmented common factor (ACF) model by considering the augmented common factor to predict the mortality of multiple groups, which can be estimated by the maximum likelihood method. The results showed that the ACF model can improve the prediction of mortality. Further, the common age effect (CAE) model proposed by Kleinow [14] also predicted the mortality of multiple groups using the common principal components method, where different populations had the same age effect yet different period effect. The CAE model can also be used for hedging the longevity risk in an annuity portfolio.

With the rapid development of artificial intelligence and computing power, nonlinear models represented by the BP neural network and the support vector machine have been applied to insurance in recent years. They can provide more accurate prediction performance than linear mortality models, and are well suited for modeling mortality for multiple populations. Puddu and Menotti [15] used a multilayer perceptron to predict the mortality of coronary heart disease patients across seven countries. Abdulkarim and Garko [16] constructed a fully connected neural network with a particle swarm optimization algorithm, which obtained the optimal parameters to predict the maternal mortality rate in Nigeria. Deprez et al. [17] found that machine learning algorithms can effectively evaluate the goodness-of-fit of standard mortality prediction models (such as L-C and R-H models). Levantesi and Pizzorusso [18] fitted and predicted mortality data based on decision tree, random forest, gradient boosting tree, and other tree methods, thereby significantly improving forecasting of mortality. Tuominen et al. [19] evaluated a variety of advanced

time series models, demonstrating that DeepAR, N-BEATS, Temporal Fusion Transformer (TFT), and LightGBM could improve 24-h prediction accuracy by up to 15% over base methods by integrating multi-source data. Li et al. [20] combined a gradient boosting framework with specialized stochastic mortality models as weak learners, introducing age- and spatial-shrinkage penalties to exploit structural similarities in the data, thereby enhancing the accuracy and interpretability of mortality forecasts at various levels in the United States.

Classical machine learning algorithms cannot directly accept multi-dimensional sequence data or grid data. However, if the multi-dimensional time-series mortality data is flattened to input data, the sequential structure of the data is lost. The deep learning networks represented by the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) can better adapt to the grid data. Hainaut [21] proposed a two-stage nonlinear semi-parametric mortality prediction method. First, the time effect in mortality data was extracted using an auto-encoder, which was then fitted and extrapolated using the random walk model with a drift term. Nigri et al. [22] also used the two-stage method by combining the RNN with the L-C model, and long short-term memory (LSTM) overcame the limitations of ARIMA and extrapolated k_t. In contrast to the two-stage method, Perla et al. [23] explored a mortality forecasting framework with multiple neural network components, where the best-performing model uses a one-dimensional convolutional network as its backbone, which is combined with country and gender embeddings and a fully connected output layer to jointly forecast age-specific mortality rates; an LSTM-based alternative was also analyzed. Their results showed that the shallower architecture performed better on the Human Mortality Database. Nigri et al. [24] used the LSTM model to directly predict life expectancy and lifespan disparity instead of the two-stage method and found that it provided more accurate and reasonable results compared with the double gap, ARIMA, CoDa, and LC models. Richman and Wüthrich [25] applied a multilayer neural network to the Lee–Carter model, exploiting its capacity of automatically learning features of data, which can directly map input vector, featuret, x, i, j, t, age, x, region_i, gender_j, to mortality rates. Perla et al. [26] proposed an interpretable framework centered on LocalGLMnet, which uses two-dimensional locally connected layers to learn input-dependent local regression coefficients from historical mortality matrices and interprets them as age-specific varying-coefficient autoregressive models, achieving both high accuracy and transparency on public databases. For fine-grained mortality forecasting, Zheng et al. [27] designed a deep learning architecture called MortFCNet, which learns directly from raw time series and weather data, outperforming feature-engineering-dependent Serfling-type methods and XGBoost in weekly predictions across multiple European regions.

Although several studies have applied deep learning to mortality forecasting, most rely on either a two-stage procedure that inherits the linear constraints of classical models or a single-component network that captures only one type of pattern in the data. This study proposes an LSTM-CNN model whose cascade architecture is driven by the structure of mortality data: the LSTM layer models long-term temporal dependencies, the CNN layer extracts local age-period-cohort patterns and their interactions, and the FCN layer with BN and Dropout maps the learned features to future mortality rates. The model maps log-mortality matrices directly to forecasts in an end-to-end manner, avoiding the inter-stage error propagation of two-stage methods, and it uses a population encoding vector to process 21 countries simultaneously. Conceptually, the architecture serves as a nonlinear generalization of the Lee–Carter decomposition. The LSTM learns nonlinear period effects, and the CNN captures nonlinear age-sensitivity patterns together with cohort and interaction terms, relaxing the linearity and single-factor constraints of classical stochastic mortality models.

The remainder of the paper is organized as follows. Section 2 reviews the concepts of mortality modeling and discusses how the model proposed in this study differs from the classical L-C model in the literature. Section 3 presents a review of several neural networks and discusses the FCN, LSTM, and CNN layers. In Section 4, the models used in this study are described. In Section 5, the proposed models are used to fit data from the Human Mortality Database (HMD) and perform numerical results analysis. In addition, how DNN models perform in mortality forecasting is discussed. Finally, Section 6 summarizes the results of this study.

2. Extension of the LC Model for Multiple Populations

Li and Lee [13] proposed the ACF model to avoid long-run divergence in mean mortality forecasts for a group, which is an extension of the L-C model with an augmented common factor. All populations in the group have the same $B(x)$ and $K(t)$, which are used to describe the mortality change of the entire group. Meanwhile, $K(t)$ is modeled as a random walk with drift (RWD) to forecast the common trend in future mortality change.

The ACF model is described as:

$$\log m(x,t,i) = a(x,i) + B(x)K(t) + b(x,i)k(t,i) + \varepsilon(x,t,i)\,, \tag{5}$$

where $a(x,i)$ is the average over time of $\log m(x,t,i)$, $B(x)$ and $b(x,i)$ are common age factor and age factor of the i-th population, respectively, $K(t)$ and $k(t,i)$ are common period factor and period factor of the i-th population, respectively, and $a(x,i) + B(x)K(t)$ represents the common factor of the i-th population.

This study used constrained weighted least squares (CWLS) algorithms to estimate $B(x)$, $K(t)$, $b(x,i)$, and $k(t,i)$. Then, RWD was used to fit $K(t)$ and AR(1) to fit $k(t,i)$.

3. Deep Learning Models

3.1. Deep Feedforward Neural Network

The deep feedforward neural network, also known as a multilayer perceptron, is a classical deep learning model. It acts as a complex nonlinear function that maps an input vector to an output, which can be a scalar or a vector. The neural network is related to a directed acyclic graph combining different functions, whose architecture is shown in Figure 1.

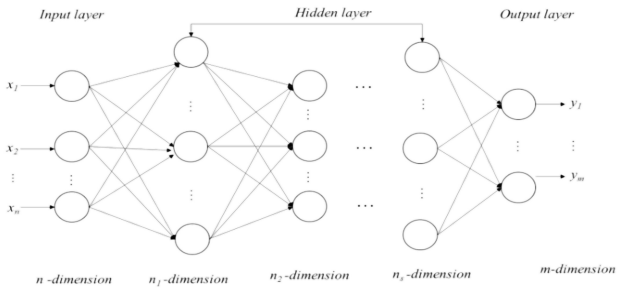


Figure 1. Architecture of deep feedforward neural network.

The first layer of the neural network model, referred to as the input layer, receives the input data X. The middle layers are the hidden layers, which apply an affine transformation to the output of the previous layer, followed by the activation function. The last layer is the output layer, where units are fully connected to produce the output $f(x)$. Therefore, the neural network maps the input data x to $f(x)$:

$$\widehat{y} = f(x),\ f : \mathbb{R}^n \to \mathbb{R}^m. \tag{6}$$

The neural network, $f(x)$, approximates the label value y to the best extent possible. The observed label y is a noisy approximation of the true function value, and the training data provide the supervision signal for learning the output mapping. The algorithm determines how the best approximation can be obtained. In addition, the nonlinear activation function added to the hidden layer introduces the nonlinear property to the neural network, which enhances prediction performance. According to the universal approximation theorem (Hornik et al., 1989; Cybenko, 1989), a feedforward neural network with a linear output layer and at least one hidden layer with squashing properties, such as the sigmoid, can approximate any Borel measurable function from one finite-dimensional space to another to arbitrary precision provided it has a sufficient number of hidden units [28,29].

A neural network is mathematically expressed as:

$$\begin{aligned} h^{(1)} &= a^{(1)}\left(W^{(1)\,T}x + b^{(1)}\right), \quad (x \in \mathbb{R}^n) \\ h^{(i)} &= a^{(i)}\left(W^{(i)\,T}h^{(i-1)} + b^{(i)}\right), \quad i \geq 2, \ (h^{(i)} \in \mathbb{R}^{n_i}) \\ \widehat{y} &= o\left(W^{(s)\,T}h^{(s-1)} + b^{(s)}\right), \quad s \geq 2, \ (\widehat{y} \in \mathbb{R}^{n_s}). \end{aligned} \tag{7}$$

Let s be the depth of the neural network, n_i be the number of hidden units in the layer i, $W^{(i)}$ be the weight of the layer i, $a^{(i)}(.)$ be the activation function of the i-th hidden layer, $h^{(i)}$ be the output of the i-th hidden layer, and $o(.)$ be the output function. Here, depth and number of units are most important among all parameters.

The nonlinear activation functions can help neural networks address complex problems. Several popular activation functions are as follows:

$$\begin{aligned} ReLU(x) &= max(x, 0), \\ Sigmoid(x) &= \frac{1}{1+e^{-x}}, \\ Tanh(x) &= \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \end{aligned} \tag{8}$$

ReLU is a piecewise linear function with unilateral inhibition mapping the negative value to 0 (i.e., not activated), which implies that only certain units are activated. ReLU units make the neural network sparse, increase computational efficiency, and alleviate overfitting, particularly in a CNN, which weakens the interdependence of parameters. However, the output of the sigmoid function is between [0, 1] and is not centered on 0. In addition, the exponential operation makes the computation slow. The derivative of the sigmoid function is significantly non-zero only near 0, making parameter updates challenging. In addition, this becomes more serious with the deepening of network layers, which results in gradient vanishing. The output of the tanh function lies between $[-1, 1]$ with centralization; thus, theoretically, tanh is superior to the sigmoid function. It also converges faster than sigmoid, which can reduce the number of iterations. However, tanh also causes the gradient to vanish.

Although several activation functions have been described, there is no definitive guideline for selection; the choice should be made according to the specific application.

3.2. Convolutional Neural Network

LeCun et al. [30] proposed the CNN, specifically designed to process grid data, where there is at least one convolution (Convolution is a special mathematical operation on two real-valued functions, and the continuous form is expressed as $y(t) = (x \circledast w)(t) = \int x(s)w(t-s)ds$. The discrete form is expressed as $y(t) = (x \circledast w)(t) = \sum_{s=-\infty}^{+\infty} x(s)w(t-s)$, $(\circledast)$ represents convolution, $w(s)$ represents the kernel function, $x(s)$ represents input data, and $y(t)$ represents the feature layer). Image data and multi-

dimensional time-series data can be regarded as grid structure data. The CNN effectively and automatically extracts local features of the input data through convolution operation, freeing deep learning models from the complex artificial feature engineering.

In fact, "convolution layer" is a misnomer, because the operation in the CNN is actually cross-correlation (2D cross-correlation is: $S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$.), not convolution. Specifically, multiple filters of fixed size slide across the input matrix. For each position, the inner product between the windowed matrix and the convolution kernel is computed, that is, product cumulative summation. In addition to the convolution layer, the classical structure of the convolution network also includes a pooling layer and a full connection layer. The convolution layer uses convolution kernels to extract local features, whereas the pooling layer highlights and reduces the dimension of the features. Further, the full connection layer maps to the output. The basic structure of the general CNN is shown in Figure 2.

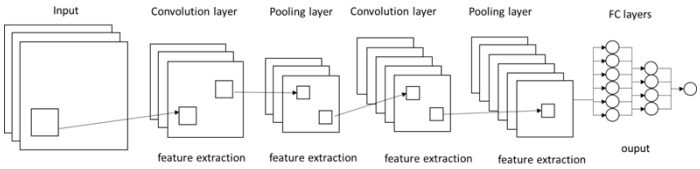


Figure 2. Architecture of a classical two-dimensional CNN model with convolution layers, pooling layers, and FCN layers.

Convolution enhances machine learning performance through the following three properties: sparse interaction, parameter sharing, and equivariance. Sparse interaction implies that the size of the convolution kernel is much smaller than that of the data, which reduces the number of model parameters and improves the computational efficiency. Parameter sharing indicates that the parameters of a CNN no longer correspond one-to-one with the data. In a fully connected neural network, the weight parameters correspond one-to-one with the data. However, weight parameters (convolution kernel) are used more than once in the convolution neural network, which greatly reduces the number of parameters. The parameter sharing mechanism also results in CNN equivariance. For other transformations such as scaling and rotation, equivariance does not hold.

If $f(x)$ and $g(x)$ satisfy $f(g(x)) = g(f(x))$, $g(.)$ is transform function, then $f(.)$ is equivariant to $g(.)$; that is, the result of translation before convolution is equal to that of convolution before translation.

If $f(x)$ and $g(x)$ satisfy $f(x) = g(f(x))$, $g(.)$ is transform function, then $f(x)$ is invariant to $g(f(x))$.

Intuitively, CNNs are more sensitive to the presence of a feature than to its position. For example, when processing time-series data, a time axis composed of different characteristics in the input can be obtained by convolution. If an event is postponed in the input, it will still have the same representation in the output; however, the time is postponed. It is similar to the grid data in that convolution generates a two-dimensional map to indicate the position of some features in the input. If an object is moved in the input, its feature representation will move the same distance.

3.3. Long Short-Term Memory

Hochreiter and Schmidhuber [31] proposed the LSTM model as an extension of the RNN. Its gate mechanism can be used to solve the gradient vanishing (Gradient vanishing in an FCN implies that the derivatives of nearby layers, computed via the back-propagation algorithm, are a product of multiple derivatives, which is so close to 0 and is difficult to update parameters. However, derivatives of parameters are the cumulative multiplication

of parameter matrices for different sequences that share the same weights in the RNN. The total derivative of weights will always exist; however, the distant derivative will vanish. Thus, derivatives in the RNN are dominated by close derivatives, which makes it difficult for the RNN model to learn long-term dependencies.) in the RNN model (Figure 3). Further, memory cells of the LSTM hidden layers can be used to process time-series data. However, although the LSTM greatly enhances the performance of the RNN, it still incurs high computational cost and long training times.

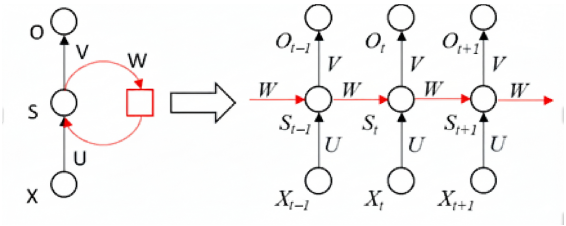


Figure 3. Architecture of RNN where historical information is passed on in hidden units.

The LSTM introduces three gating mechanisms—forget, input, and output gates—into the recurrent neural network to selectively retain or discard information, thereby enabling the capture of long-term dependencies. At each time step t, the gate activations and state updates are computed as follows:

$$\begin{aligned} f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\ i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o). \end{aligned} \tag{9}$$

where f_t, i_t, and o_t denote the forget, input, and output gate vectors, respectively; $\sigma(\cdot)$ is the logistic sigmoid function; x_t is the input vector at time t; h_{t-1} is the hidden state from the previous step; $W_{(\cdot)}$, $U_{(\cdot)}$, and $b_{(\cdot)}$ are the learnable weight matrices and bias vectors for each gate. The cell state c_t and hidden state h_t are then updated by:

$$\begin{aligned} \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{10}$$

where $\tilde{c}_t$ is the candidate cell state, $\odot$ denotes the Hadamard (element-wise) product, $f_t \odot c_{t-1}$ controls how much of the previous cell state is retained, and $i_t \odot \tilde{c}_t$ determines how much new information is written into the cell (Figure 4).

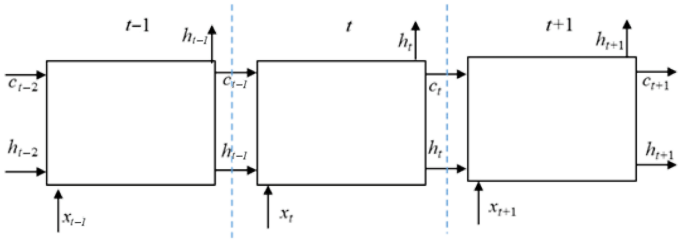


Figure 4. External architecture of LSTM; h_t represents hidden layer output of time t; c_t represents cell state of time t; x_t represents input of time t.

The LSTM differs from the RNN in three ways (Figure 5): (1) three gates are added to the LSTM; (2) in addition to the hidden units h_t, the memory cells c_t are passed on to the next step; (3) there is a recurrent connection inside and outside the LSTM. These advantages enable the LSTM to capture the long-term dependencies in sequential data.

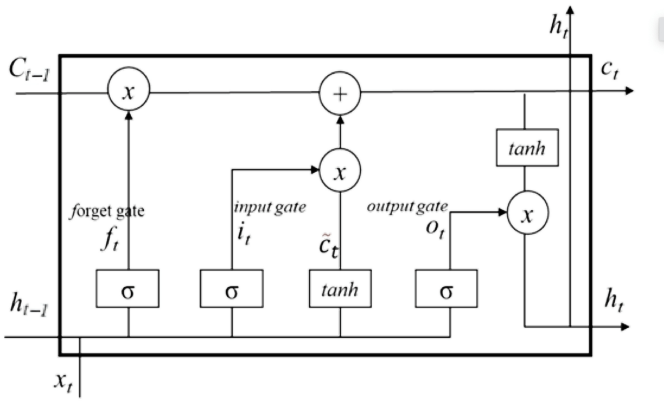


Figure 5. Internal architecture of LSTM.

3.4. Dropout

Hinton et al. [32] proposed the dropout algorithm to prevent overfitting in neural networks. Further, Krizhevsky et al. [33] applied it to a deep convolution neural network for image classification. During the training, dropout sets neural units to zero temporarily with a given probability; that is, when the neural network propagates forward, it makes the hidden units stop working with a certain probability P. Hence, the dropout algorithm produces a more generalizable model because it does not rely excessively on certain local features, which is an important tool for preventing overfitting and enhancing performance.

Dropout can be performed in the following steps:

1. Training stage

Hidden neurons are randomly deleted (temporarily) with probability p, while input and output neurons remain unchanged (neurons with a cross are deleted in Figure 6).

The data propagates forward through the modified network, and then the loss function propagates back through the modified network. The parameters of neurons that are not deleted are updated, while the parameters of deleted neurons are not.

Repeat the above process.

2. Test stage

All deleted neurons of the neural network are restored. When predicting, the weight parameters of each neural unit are multiplied by the probability p.

When dropout is not used, the neural network is a fully connected network (FCN), which is expressed as follows:

$$\begin{aligned} z_i^{(l+1)} &= W_i^{(l+1)} y^{(l)} + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f\left(z_i^{(l+1)}\right). \end{aligned} \tag{11}$$

However, when dropout is used, the neural network is expressed as follows (Figure 7):

$$\begin{aligned} r_j^{(l)} &\sim Bernoulli(p), \\ \widetilde{y}^l &= r^{(l)} * y^{(l)}, \\ z_i^{(l+1)} &= W_i^{(l+1)} \widetilde{y}^l + b_i^{(l+1)}, \\ y^{(l+1)} &= f\left(z_i^{(l+1)}\right). \end{aligned} \tag{12}$$

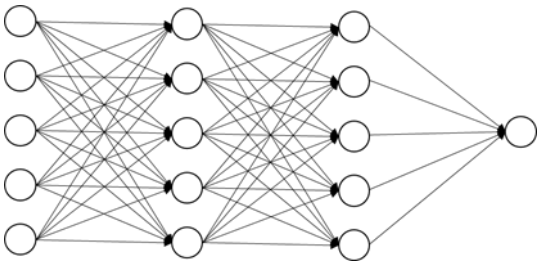


Figure 6. Fully connected neural network where all neurons are connected to each other and participate in training to be updated together.

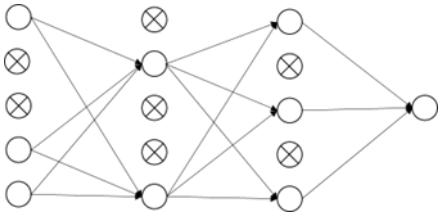


Figure 7. Fully connected neural network with Dropout; only some neurons are activated to participate in training and to be updated.

3.5. Batch Normalization

Ioffe and Szegedy [34] proposed Batch Normalization (BN), which has been widely used in deep learning. In the training stage, maintaining a constant distribution of hidden layer output is challenging, which results in Internal Covariate Shift. However, BN causes the middle layers to maintain fixed mean and variance and helps to handle the gradient vanishing or gradient explosion problem. In addition, BN can accelerate the training process and produce more stable training results. BN also acts as a form of regularization, which can also reduce overfitting and improve prediction performance.

BN is performed in two steps. During the first stage, x is standardized to $\widehat{x}$, which obeys the distribution of zero mean and unit variance. In the second step, x is scaled and shifted to y, which obeys the distribution of mean β and variance γ. Hence, BN receives x and outputs y, which is expressed as follows:

Standardization: $\widehat{x} = \frac{x - E(x)}{\sqrt{Var(x) + \varepsilon}}$, ε prevents denominator from approaching 0, scale and shift: $y = \widehat{x} \times \gamma + \beta$, γ and β are parameters to be trained.

4. LSTM-CNN Model

This study proposes a new LSTM-CNN deep learning model to forecast mortality rates, composed of the LSTM, CNN, and FCN layers. As explained in Section 3, the LSTM layer can learn long-term dependencies of mortality data, the CNN layer can extract and highlight nonlinear features, and the FCN layer performs predictions by mapping information learned from the LSTM and CNN to future mortality. In addition, the CNN and LSTM models are also used to perform predictions.

Figure 8 displays the framework of the LSTM-CNN model for prediction. The LSTM layer is the first layer, receiving the log-mortality matrix and outputting a new matrix: $\log M \in \mathbb{R}^{T \times (x_{max} - x_{\min} + 1)}$:

$$f^{LSTM} : \mathbb{R}^{T \times (x_{\max} - x_{\min} + 1)} \to \mathbb{R}^{T \times h}, z^1 = f^{LSTM}(\log M). \tag{13}$$

The input mortality matrix is a multi-dimensional time-series dataset with a grid structure ranging from year t to t + 10 and from age $x_{\min}$ to $x_{\max}$. After receiving the input matrix, the LSTM provides a $T \times h$ matrix, where the time-series structure of mortality

data is retained and the age effect is refined. However, the LSTM does not consider the interactive terms between age and period.

After the mortality matrix is processed by the LSTM, the CNN is used to further extract local features of grid data. While both convolution and pooling layers are components of a CNN, neither alone constitutes the CNN layer. The details of the CNN operation are as follows: the kernel will continuously slide windows of the same size on the mortality matrix to conduct convolution; that is, the inner product of the kernel and mortality data on the window. For each slide, data on the window includes mortality rates at the same age in different years and mortality rates at different ages in the same year, such that the architecture is designed to capture patterns that correspond to age, period, and cohort dimensions as well as their interactions in mortality data by convolution of kernel and data on the window is convenient. Subsequently, the pooling layer highlights the extracted information from the CNN layer and reduces noise:

$$f^{CNN}: \mathbb{R}^{T\times h} \to \mathbb{R}^{m\times n}, z^2 = f^{CNN}\left(z^1\right). \tag{14}$$

z^2, output of the CNN, is a p-dimensional vector of the FCN after the ReLU, BN, and flatten layers. Meanwhile, individual static information vector c is also input into the FCN. The last part is the fully connected layer, also known as a feedforward neural network or multilayer perceptron. Further, in the LSTM-CNN, the FCN also contains the BN and dropout layers, which can map learned features to future mortality directly. As mentioned in Section 3.1, the multilayer perceptron is close to a complex nonlinear function, which maps the extracted feature flattened to a vector. The output of the FCN can be expressed as follows:

$$\begin{aligned} &\widehat{\log m} = W_1^T z_i^{(t)} + W_2^T c + b, W_2 \in \mathbb{R}^{23\times 100}, W_1 \in \mathbb{R}^{p\times 100}, b \in \mathbb{R}^{100}, \\ &\widehat{\log m} \in \mathbb{R}^{100}. \end{aligned} \tag{15}$$

The LSTM-CNN model requires several hyperparameters, such as hidden unit size, filter size, filter numbers, pooling size, and learning rate. This study used a random search algorithm to determine the best hyperparameters. Meanwhile, the Adam algorithm [35] is used to optimize neural network models. Adam, a type of adaptive learning rate algorithm, is among the most widely used optimization algorithms at present. Compared with other adaptive learning rate algorithms, Adam can aid in reaching convergence faster and thus achieve more effective learning. The final hyperparameter configuration is as follows: LSTM hidden layer size is 40, CNN output channels is 10, kernel size is 6, number of fully connected layers is 2, dropout rate is 0.3, learning rate is 0.001, and batch size is 32.

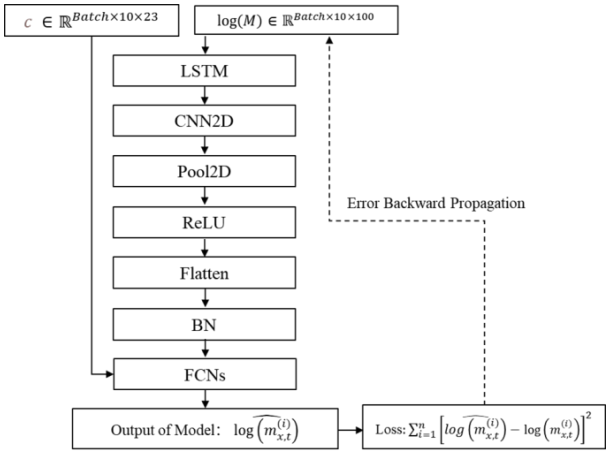


Figure 8. Architecture of LSTM-CNN.

The data used in this study were of two types: dynamic and static data. As mortality rates vary over time, mortality data is dynamic. However, population information vector $c \in \mathbb{R}^{23}$, is static. Here, c includes the country code and the gender code, which are always constant. Because deep learning models can automatically learn data features, the need for complex feature engineering is consequently eliminated. To compare the results obtained in this study with those yielded by ACF, the log-forces of mortality were used as input for the neural network, which are denoted by $\log M$.

Let $\log M \in \mathbb{R}^{1992 \times T \times (x_{max} - x_{\min} + 1)}$ denote mortality input matrix, where $x_{\min}$ is 0, $x_{\max}$ is 99, and T is 10.

The batch gradient descent algorithm was used to optimize parameters; thus, only a batch of data was processed in each iteration rather than all data. In this study, batch size N was set to 32, that is, $\log M \in \mathbb{R}^{N \times T \times (x_{\max} - x_{\min} + 1)}$.

5. Numerical Results and Analysis

This section applies the proposed deep learning models and traditional actuarial baselines to mortality data from 21 countries in the Human Mortality Database (HMD). Model performance is evaluated along five dimensions: prediction accuracy, stability, long-term forecasting reliability, remaining life expectancy, and prediction uncertainty. Three error metrics are used throughout, where $y = \log m(x,t)$ denotes the observed log-mortality and $\hat{y}$ its predicted value:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|, \tag{16}$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{17}$$

$$\text{MAPE} = \frac{100\%}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \tag{18}$$

The data span ages 0–99 and calendar years from 1950 to 2024 for 21 countries. Years up to 2000 serve as the training set, with 15% reserved for validation and hyperparameter tuning; the test set covers 2011 to 2024. During testing, mortality predictions are generated via dynamic recursive forecasting, where the model's own predictions serve as inputs for subsequent forecast steps, simulating a realistic out-of-sample scenario. Consider the prediction for year t as an example:

$$\widehat{\log m(x,t,i)} = f(\log m(x,t-10,i), \cdots, \log m(x,t-1,i), c), \tag{19}$$

where $c \in \mathbb{R}^{23}$ is the static population information vector: country (21 categories) and gender (2 categories) are each one-hot encoded and concatenated to form a 23-dimensional vector, which is then appended to the 100-dimensional log-mortality vector $\log m(x,t,i) \in \mathbb{R}^{100}$ to yield a 123-dimensional input feature. All deep learning models are independently trained 10 times to reduce randomness from weight initialization; traditional actuarial models (Lee–Carter, CBD, APC, ACF) are executed once owing to their deterministic estimation procedures. The deep learning framework is implemented in PyTorch 2.1 .

5.1. NN Lee–Carter Baseline

Richman and Wüthrich [25] extend the Lee–Carter model to multiple populations by replacing its linear structure with a deep feedforward network. Age, Country, and Gender are each mapped to learned 5-dimensional embedding vectors, concatenated with the numerical Year input, and passed through four hidden layers (128 neurons, ReLU,

BatchNorm, Dropout $p = 0.05$) with a skip connection from the input features to the output layer. The embeddings generalize the Lee–Carter age and population parameters, while the deep layers automatically learn nonlinear interactions without manual specification. This architecture is denoted DEEP5 in the original paper; all hyperparameters follow the original specification.

5.2. Prediction Accuracy

This section evaluates predictive performance from multiple perspectives: overall accuracy, training stability, gender-specific accuracy, age-group-specific accuracy, and residual distributions. The test set spans 14 calendar years (2011–2024), ages 0–99, and covers multi-population mortality data from 21 countries.

5.2.1. Overall Accuracy

Table 2 summarizes the test-set MSE, MAE, and MAPE (mean $\pm$ standard deviation), along with trainable parameter counts. On the test set, the LSTM-CNN achieves the best performance, with an MSE of 0.078 ± 0.011, MAE of 0.188 ± 0.018, and MAPE of 0.037 ± 0.003. Compared with Lee–Carter, the LSTM-CNN reduces test MSE by 25.0%, MAE by 18.6%, and MAPE by 19.6%, confirming the advantage of the hybrid architecture for long-horizon multi-population mortality forecasting. Among deep learning models, the LSTM ranks second with a test MSE of 0.082 ± 0.005, and CNN ranks third at 0.108 ± 0.029; although the CNN falls behind the first two, it still outperforms all traditional actuarial models.

As shown in Table 2, the LSTM-CNN achieves the lowest test MSE (0.078) among all models, including the reproduced NN Lee–Carter [25], whose MSE (0.101) is 29.5% higher with larger cross-run variance. Although NN Lee–Carter records lower MAE and MAPE, MSE penalizes large deviations quadratically and is more relevant to actuarial tail-risk applications such as reserve calculation and annuity pricing.

Table 2. Test-set prediction accuracy across 21 countries and model parameter counts. Deep learning models report the mean $\pm$ standard deviation over 10 independent training runs; traditional actuarial models report a single run. **Bold** indicates the best value in each column.

Model	MAE	MSE	MAPE	Params
Deep learning models				
LSTM-CNN	0.188 ± 0.018	**0.078** ± 0.011	0.037 ± 0.003	67,456
LSTM	0.196 ± 0.008	0.082 ± 0.005	0.039 ± 0.002	39,420
CNN	0.230 ± 0.035	0.108 ± 0.029	0.046 ± 0.008	19,710
LSTM-CNN (*w/o* FCN)	0.216 ± 0.019	0.097 ± 0.012	0.041 ± 0.003	56,490
NN Lee–Carter	**0.180** ± 0.016	0.101 ± 0.026	**0.033** ± 0.002	55,539
Traditional actuarial models				
Lee–Carter	0.231	0.104	0.046	10,560
CBD	0.256	0.242	0.046	6564
APC	0.276	0.142	0.082	12,678
ACF	0.598	0.545	0.112	17,088

Among traditional models, Lee–Carter performs best, with a test MSE of 0.104. CBD, constrained by its parametric structure, yields a test MSE of 0.242. APC has a test MSE of 0.142 and MAPE of 8.2%, suggesting systematic bias in certain age ranges. ACF records a test MSE of 0.545, far exceeding other models and indicating severe deterioration in extrapolation ability over long forecast horizons.

Deep learning models have more trainable parameters, enabling more flexible feature representations. The LSTM-CNN has the largest parameter count (67,456), followed by the

LSTM (39,420) and the CNN (19,710). Despite having fewer parameters, the CNN achieves test-set performance comparable to the LSTM, indicating that convolutional structures are effective at extracting local features from mortality data. The LSTM-CNN benefits from both architectures and achieves the best test-set MSE among all models. A detailed ablation analysis is provided in Section 5.3.

5.2.2. Training Stability

Figure 9 displays box plots of performance distributions for each deep learning model across 10 independent training runs. On the validation set, boxes for all models are positioned at similar levels, indicating comparable validation performance; the CNN, however, has the widest span and clear outliers, suggesting greater sensitivity to initialization conditions. On the test set, inter-model differences widen. The LSTM-CNN occupies the lowest and most compact box, with the best accuracy and the least variation across runs. The LSTM is positioned slightly higher but remains similarly compact, reflecting good consistency. The CNN is positioned noticeably higher with the largest dispersion, particularly in MSE and MAE, where prominent outliers appear, indicating that CNN prediction quality is highly sensitive to stochastic factors during training. NN Lee–Carter achieves competitive MAE and MAPE but shows wider MSE dispersion than the LSTM-CNN, reflecting occasional large errors in certain training runs. This contrast is consistent with the point-estimate comparison in Table 2: despite its MAE and MAPE advantages, the higher MSE variance of NN Lee–Carter indicates less reliable control of extreme mispredictions across independent runs. Lee–Carter, being deterministic, appears as a single-valued line segment, providing a stable but less accurate baseline. Overall, the LSTM-CNN achieves both the best mean performance and the smallest variance, making it the most reliable choice for practical use.

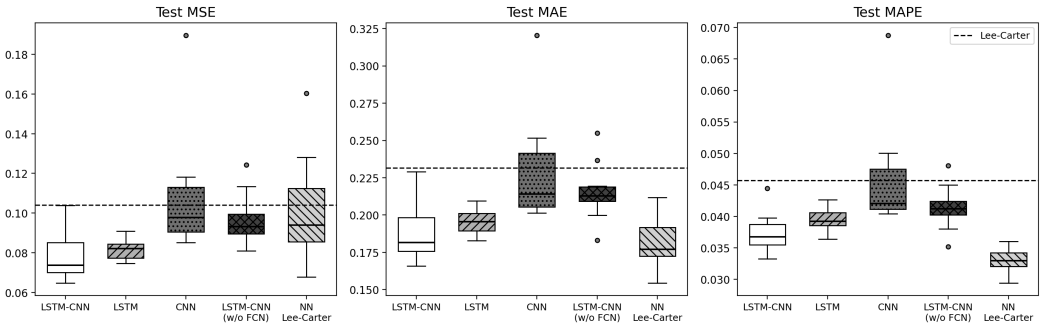


Figure 9. Box plots of MSE, MAE, and MAPE on the test set. Deep learning models are based on 10 independent training runs; Lee–Carter is deterministic (single value).

Gender-Specific Accuracy

Table 3 presents prediction errors on the test set grouped by gender. Across all models, male MAE and MSE are consistently lower than the corresponding female values. For the LSTM-CNN, the male MSE is 0.0635 while the female MSE is 0.0832, a gap of approximately 23.7%. This pattern may reflect the greater complexity of female mortality dynamics: in most developed countries, female life expectancy is higher and heterogeneity in improvement trends greater, thus increasing prediction difficulty.

The LSTM-CNN achieves the best performance for both genders. The LSTM matches the LSTM-CNN on male MSE (both 0.0635), yet falls behind on female MSE (0.0947 vs. 0.0832), suggesting that the convolutional module provides an additional contribution to capturing cross-age features of female mortality. Lee–Carter performs particularly poorly for males, with an MAE of 0.2410 and MSE of 0.1076, both much higher than those of all deep learning models, reflecting the limitations of a single-factor linear model in handling nonlinear structural changes in male mortality.

Table 3. Gender- specific test-set prediction accuracy (21 countries, 2011–2024). Each metric is averaged over years, ages, and countries.

Model	Female			Male		
	MAE	**MSE**	**MAPE**	**MAE**	**MSE**	**MAPE**
LSTM-CNN	0.1945	0.0832	0.0356	0.1639	0.0635	0.0353
LSTM	0.2185	0.0947	0.0408	0.1652	**0.0635**	0.0361
CNN	0.2194	0.0987	0.0416	0.2103	0.0906	0.0440
LSTM-CNN (*w*/*o* FCN)	0.2228	0.1017	0.0400	0.1964	0.0829	0.0399
Lee–Carter	0.2220	0.1006	0.0403	0.2410	0.1076	0.0511

Age-Group-Specific Accuracy

Figure 10 shows prediction errors of each model across age groups, plotted separately by gender. MAE and MSE exhibit a clear U-shaped pattern: the youngest groups (0–14) have the largest errors, middle-aged groups (45–64) the smallest, and elderly groups (75–99) rise again. For the LSTM-CNN, female MAE is 0.379 for ages 5–14, drops to 0.096 for ages 55–64, and rises to 0.097 for ages 85–99; the male pattern is similar, with MAEs of 0.360, 0.094, and 0.087 for the same respective groups. The elevated errors at young ages arise because absolute mortality rates are extremely low, so small random fluctuations are amplified on the log scale. The upturn at old ages is associated with heterogeneity in elderly mortality and cohort effects. For MAPE, all models show a marked increase above age 75, because the absolute value of log-mortality is larger for the elderly; even when the absolute residual is comparable to that of middle-aged groups, the relative percentage error is amplified.

The CNN displays a clear advantage in the infant age group (age 0), with a female MAE of 0.209 and a male MAE of 0.220, both lower than the LSTM-CNN (0.270 and 0.294), suggesting that convolutional filters are well suited to the local structure of infant mortality. Lee–Carter records an MAE of 0.456 for males aged 1–4, far exceeding all deep learning models and reinforcing that traditional linear models struggle in low-mortality, high-volatility age segments. Overall, the LSTM-CNN maintains the best or near-best accuracy across the vast majority of age groups, showing strong adaptability to mortality patterns across most age groups.

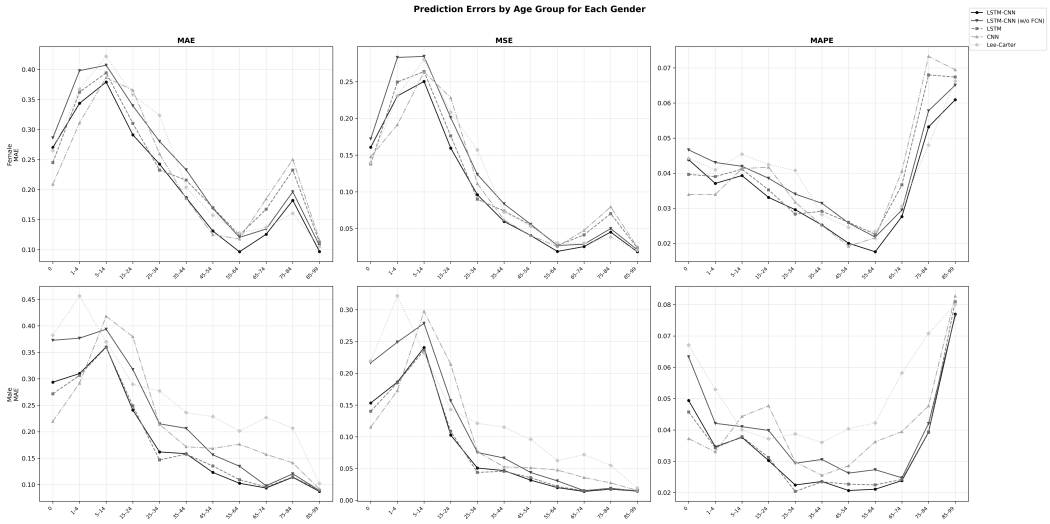


Figure 10. Test-set prediction errors by age group. **Top row**: female; **bottom row**: male; from **left** to **right**: MAE, MSE, MAPE. Age groups: 0, 1–4, 5–14, 15–24, 25–34, 35–44, 45–54, 55–64, 65–74, 75–84, 85–99.

Residual Analysis

Figure 11 presents heatmaps of prediction residuals (predicted minus actual) on the test set, with age (0–99) on the horizontal axis and year (2011–2024) on the vertical axis, shown separately by gender. The LSTM-CNN heatmap shows the lightest overall coloring, with residuals closest to zero, consistent with its top ranking on the aggregate accuracy metrics. All models display noticeable residual patches in the young age range (0–20), consistent with the left-hand peak of the U-shaped error distribution described above.

Residuals for all models increase visibly after 2020, forming a clear band along the time axis, most likely reflecting the impact of the COVID-19 pandemic on mortality patterns. The atypical mortality fluctuations during the pandemic deviated from historical trends, posing a systematic challenge to all models trained on historical data. Regarding residual sign, positive residuals predominate, indicating that models tend to overestimate future mortality levels, i.e., underestimate the actual pace of mortality improvement. The Lee–Carter heatmap shows the deepest color patches, especially at young ages where large dark regions appear, indicating the largest prediction bias. The CNN also shows deeper patches in the male 40–60 age region, suggesting limited feature extraction capacity in that range. By contrast, the LSTM-CNN maintains a relatively uniform and mild residual distribution across all ages and years, consistent with its predictive advantage.

5.3. Ablation Study

To verify the contribution of each component in the LSTM-CNN, this section presents ablation experiments. The complete LSTM-CNN consists of an LSTM module, a CNN module, and a two-layer fully connected network (FCN) with inter-layer Batch Normalization and Dropout regularization. Eight ablation variants are organized into three groups (Table 4): component ablation removes the CNN, LSTM, or FCN module; FCN depth ablation varies the number of fully connected layers (1 FC, 3 FC); and regularization ablation removes BN, Dropout, or both from the FCN.

Table 4. Ablation study results on the test set (mean ± std over 10 runs). **Bold** indicates the best value in each column.

Model	MAE	MSE	MAPE
LSTM-CNN	**0.188** ± 0.018	**0.078** ± 0.011	**0.037** ± 0.003
Component ablation			
LSTM (*w/o* CNN)	0.196 ± 0.008	0.082 ± 0.005	0.039 ± 0.002
CNN (*w/o* LSTM)	0.230 ± 0.035	0.108 ± 0.029	0.046 ± 0.008
LSTM-CNN (*w/o* FCN)	0.216 ± 0.019	0.097 ± 0.012	0.041 ± 0.003
FCN depth ablation			
LSTM-CNN (1 FC)	0.215 ± 0.019	0.098 ± 0.016	0.041 ± 0.003
LSTM-CNN (3 FC)	0.216 ± 0.024	0.095 ± 0.016	0.043 ± 0.004
Regularization ablation			
LSTM-CNN (*w/o* BN)	0.218 ± 0.018	0.098 ± 0.013	0.042 ± 0.003
LSTM-CNN (*w/o* Dropout)	0.355 ± 0.029	0.208 ± 0.031	0.066 ± 0.005
LSTM-CNN (*w/o* BN&Dropout)	0.333 ± 0.023	0.190 ± 0.019	0.064 ± 0.005

The complete LSTM-CNN (test MSE 0.078) clearly outperforms all ablation variants. Removing the CNN module (LSTM only, MSE 0.082) or the LSTM module (CNN only, MSE 0.108) both degrade performance, confirming that the two feature extractors are complementary: the LSTM captures temporal dependencies while the CNN captures local cross-age patterns. Removing the FCN module (*w/o* FCN, MSE 0.097) also leads to clear regression.

The three FCN depth variants yield test MSEs of 0.097 (w/o FCN), 0.098 (1 FC), and 0.095 (3 FC), all considerably worse than the full model.

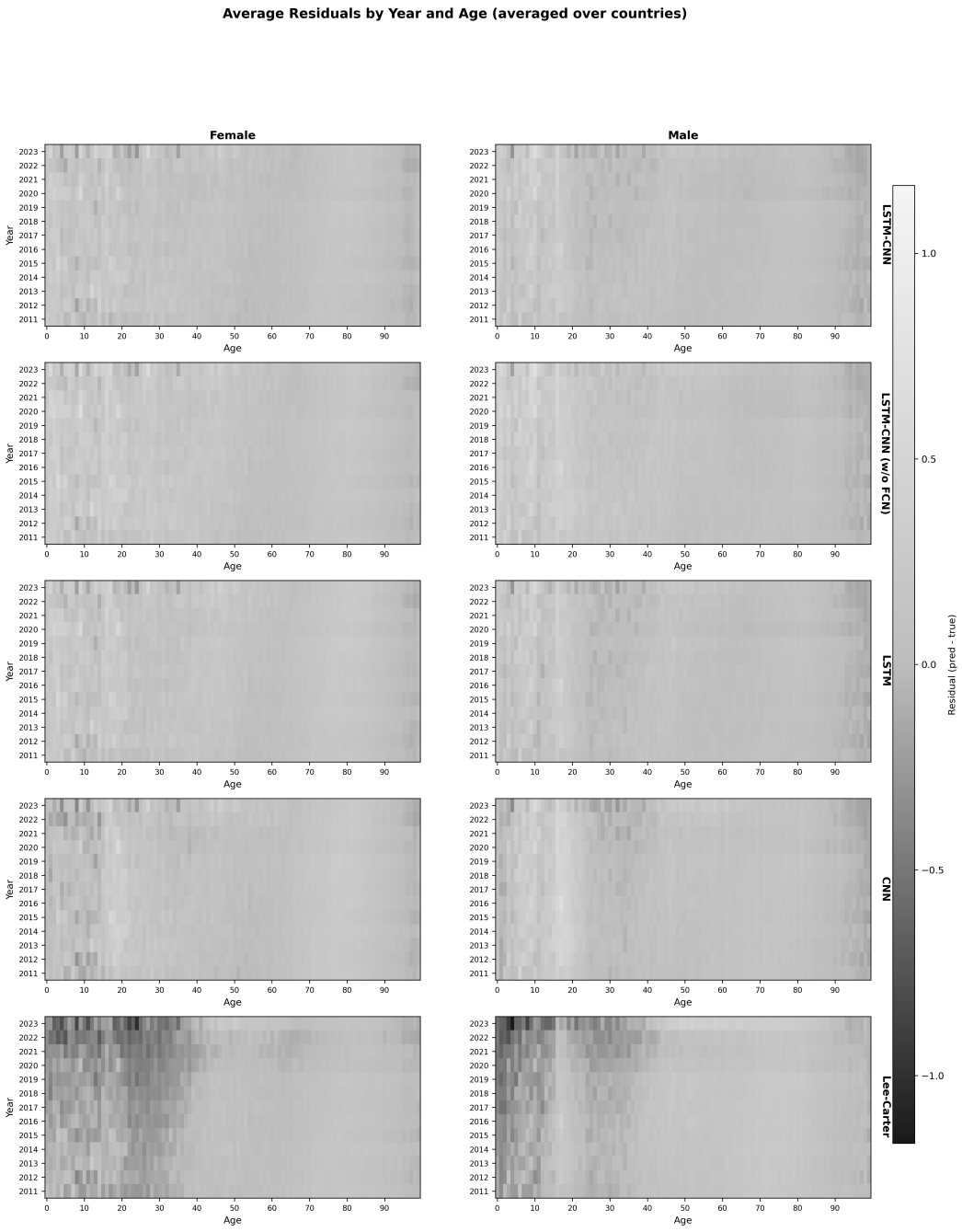


Figure 11. Mean residual heatmaps on the test set (predicted−actual), averaged across countries. Each row corresponds to a model (**top** to **bottom**: LSTM-CNN, LSTM-CNN w/o FCN, LSTM, CNN, Lee–Carter); the left and right columns represent female and male, respectively.

Comparing w/o FCN and 1 FC reveals the nonlinear mapping capacity of the fully connected network. Neither variant includes inter-layer regularization, yet 1 FC contains an additional fully connected layer. Their test MSEs are nearly identical (0.098 vs. 0.097), indicating that merely increasing fully connected network depth yields minimal improvement in prediction accuracy.

Comparing 1 FC with the complete LSTM-CNN (2 FC) highlights the importance of inter-layer regularization. Moving from 1 FC to 2 FC, one fully connected layer is added with Batch Normalization and Dropout, reducing test MSE from 0.098 to 0.078, an

improvement of 20.4%. Combined with the preceding finding that depth alone contributes little, this gain can be primarily attributed to inter-layer regularization. Batch Normalization stabilizes intermediate layer feature distributions, while Dropout suppresses overfitting through random neuron deactivation; their combination enables stronger generalization while preserving expressive capacity.

Comparing 2 FC and 3 FC further shows that a deeper fully connected network is not necessarily better. Although 3 FC has the largest parameter count (77,756), its test MSE is 0.095, which not only fails to surpass 2 FC but actually falls below it. This indicates that, given the current data scale and feature complexity, a two-layer fully connected network already fulfills its role in feature fusion and nonlinear mapping, and additional depth merely introduces extra optimization difficulty.

In summary, the ablation experiments show that the optimal performance of the LSTM-CNN arises from the combination of three components: the LSTM captures long-range temporal dependencies in mortality, the CNN extracts local features across ages and periods, and the two-layer fully connected network with inter-layer regularization achieves effective feature fusion and generalization control. Removing or simplifying any component leads to a clear decline in test-set performance.

To further isolate the individual contributions of Batch Normalization (BN) and Dropout within the fully connected module, three additional regularization configurations are evaluated (Table 4, regularization ablation). Removing BN while retaining Dropout increases MSE to 0.098, a 25.6% degradation, indicating that BN provides a meaningful but secondary contribution. In contrast, removing Dropout while retaining BN causes the most severe degradation: MSE rises to 0.208, a 166.7% increase over the full model. Removing both BN and Dropout yields MSE 0.190. However, the variant without Dropout (MSE 0.208) performs worse than the variant without either regularizer (MSE 0.190), suggesting that BN without Dropout may amplify overfitting by stabilizing feature distributions in a way that allows the network to memorize training patterns more effectively. This unexpected result shows that Dropout is the primary regularizer in the fully connected module, and BN contributes positively only when combined with Dropout.

5.4. Statistical Significance Tests

This section applies the Diebold–Mariano (DM) test in a pairwise fashion across all models, to statistically verify the reliability of the performance ranking. Given the panel structure of the data (42 country $\times$ gender populations), observations within the same population may be correlated, and a naïve DM test could underestimate standard errors. To address this, a cluster-robust DM test is employed: sample-level errors are first aggregated into population-level means across the 42 groups, and a t-test is then conducted on these 42 independent observations (degrees of freedom $df = 41$).

Table 5 presents the pairwise DM test results based on the best iteration. Each entry is a cluster-robust t-statistic; a positive value indicates that the column model outperforms the row model, and a negative value indicates the reverse.

As shown in Table 5, the pairwise comparisons between the LSTM-CNN and every other model reach $p < 0.001$ significance across all three metrics, with absolute t-statistics ranging from 4.38 to 10.24. Most pairwise comparisons among the remaining models also reach significance, with a few exceptions: the difference between the LSTM and the LSTM-CNN (w/o FCN) is not significant on MAE and MAPE, although it is significant on MSE, where the LSTM outperforms the LSTM-CNN (w/o FCN). In addition, the significance level between the CNN and Lee–Carter on MAPE is relatively low.

To assess the robustness of these conclusions to training randomness, the DM test is conducted for all $10 \times 10 = 100$ iteration combinations of the deep learning models, and

the proportion of significant wins is recorded. Taking MSE as an example, the LSTM-CNN wins 86 out of 100 pairings against the LSTM-CNN (*w/o* FCN), 63 against the LSTM, 89 against the CNN, and 9 out of 10 against Lee–Carter. A similar pattern is observed for MAE. The win rate of the LSTM-CNN against the LSTM is comparatively lower, consistent with the small gap in their point estimates, yet the LSTM-CNN still wins in the majority of iteration combinations. This indicates that the advantage of the LSTM-CNN does not depend on a particular random training seed and is consistent across different iteration pairings.

Table 5. Pairwise DM test based on the best iteration. Top panel: MSE; middle panel: MAE; bottom panel: MAPE.

	LSTM-CNN	**LSTM-CNN (*w/o* FCN)**	**LSTM**	**CNN**
MSE				
LSTM-CNN (*w/o* FCN)	9.70 ***			
LSTM	6.37 ***	−4.02 ***		
CNN	9.65 ***	2.72 **	4.47 ***	
Lee–Carter	6.98 ***	3.77 **	4.54 ***	3.09 **
MAE				
LSTM-CNN (*w/o* FCN)	6.89 ***			
LSTM	5.75 ***	−0.25		
CNN	9.45 ***	6.23 ***	3.90 **	
Lee–Carter	8.22 ***	5.63 ***	4.93 ***	3.79 **
MAPE				
LSTM-CNN (*w/o* FCN)	4.38 ***			
LSTM	5.77 ***	1.63		
CNN	10.24 ***	8.69 ***	5.27 ***	
Lee–Carter	6.32 ***	5.49 ***	4.19 ***	2.42 *

Note: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$. The t-statistics are computed using cluster-robust standard errors (clustered by country × gender population, $G = 42$, $df = 41$). A positive value indicates that the column model outperforms the row model; a negative value indicates the reverse.

5.5. Prediction Stability

The preceding sections evaluated model performance from the perspective of average prediction error. This section complements that analysis by examining prediction stability, defined as the degree of fluctuation in prediction errors across populations for a given model. In actuarial practice, high average accuracy alone is insufficient. If a model predicts well for certain populations but poorly for others, the uneven error distribution poses risks to life insurance companies even when the overall average error is low. Specifically, individuals whose mortality is underestimated tend to purchase more whole-life insurance, whereas those whose mortality is overestimated tend to avoid survival-contingent products, generating adverse selection effects that undermine sound operation and risk diversification for insurers. An effective mortality forecasting model should therefore show not only high accuracy but also strong stability, meaning prediction error fluctuations across populations are as small as possible.

Following the definitions in the original work, accuracy and stability are defined as follows:

$$\text{Accuracy}_i = E_{x \in D_i}[\text{Loss}(x; \Theta)], \tag{20}$$

$$\text{Stability}_i = \sqrt{\frac{\sum_{x \in D_i} [\text{Loss}(x; \Theta) - \text{Accuracy}_i]^2}{N_i}}, \tag{21}$$

where D_i denotes the i-th population (country $\times$ gender), $\text{Loss}(x;\Theta)$ is the MAE of sample x across all ages, and N_i is the number of samples in population i. A lower stability value indicates smaller year-to-year fluctuations in prediction errors within that population.

Figure 12 displays the stability metric for each model across all 42 populations (21 countries $\times$ 2 genders) in heatmap form. Rows correspond to model–gender combinations and columns to countries; color intensity reflects the magnitude of the stability value.

Overall, stability values of the deep learning models are generally concentrated in the low range of 0.01–0.04, with predominantly light shading, indicating small year-to-year fluctuations in prediction errors. By contrast, Lee–Carter shows noticeably dark regions in several countries, particularly the United States (female 0.10, male 0.09), Japan (female 0.06), France (female 0.05), and the Netherlands (female 0.05). These values far exceed those of the deep learning models, reflecting large inter-annual volatility during long-horizon extrapolation.

Along the country dimension, model stability shows considerable consistency. The United Kingdom (GBR) is the most stable country for all models, with stability values uniformly at 0.01 or 0.02. Iceland (ISL) and Australia (AUS), on the other hand, are countries where deep learning models generally show larger fluctuations, possibly attributable to Iceland's small-sample characteristics and Australia's distinctive mortality change patterns.

Along the gender dimension, female populations generally show slightly weaker stability than male populations. For the LSTM-CNN, five countries in the female group have stability values of 0.03 or above, compared with only three in the male group. This finding is consistent with the observation in Section 5.1 that prediction accuracy for females is lower than for males, further indicating that heterogeneous female mortality improvement patterns pose a greater challenge for stable prediction.

Among the deep learning models, inter-model stability differences are smaller than the corresponding accuracy differences. The CNN yields relatively uniform stability values with few extreme cases. The LSTM-CNN and the LSTM perform similarly in most countries but have elevated fluctuations in a few populations (e.g., Japanese females and Australian females). The LSTM-CNN (w/o FCN) achieves stability comparable to the full model, suggesting that the fully connected layers exert limited influence on stability.

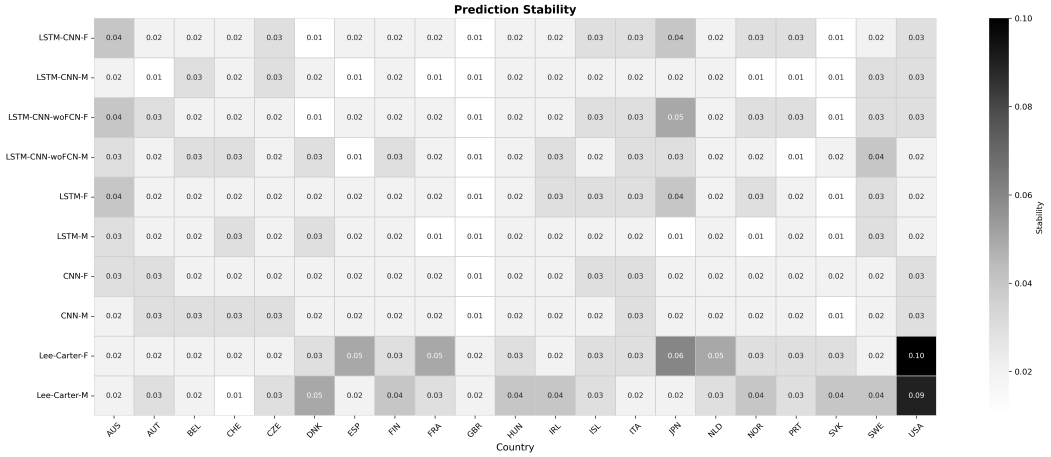


Figure 12. Prediction stability heatmap for each model across 42 populations (21 countries $\times$ 2 genders). Rows correspond to model–gender combinations and columns to countries. Values represent the stability metric (Equation (21)); darker colors indicate larger year-to-year fluctuations in prediction errors.

Best-Stability Win Counts

To provide a more intuitive comparison of stability performance, Table 6 tallies the number of populations (out of 42) in which each model achieves the best stability (i.e., the

lowest stability value), broken down by gender. For total wins, the LSTM-CNN achieves the best stability in 13 of 42 populations, ranking first. The CNN follows closely with 12, the LSTM ranks third with 8, and Lee–Carter ranks last with only 5.

The results also reveal a gender-specific pattern. The CNN excels in female populations, winning in 9 countries and outperforming all other models, yet wins in only 3 among males. Conversely, the LSTM-CNN performs markedly better in male populations (8 country wins) while winning 5 among females. The LSTM likewise performs better for males (5 countries) than for females (3 countries). This pattern suggests that the convolutional structure of the CNN is better suited to capturing local cross-age fluctuations in female mortality, stabilizing predictions for females, whereas the LSTM-CNN hybrid architecture achieves stronger prediction stability for males.

Table 6. Number of populations (out of 42) in which each model achieves the best stability, by gender. The model with the lowest stability value within each population is counted as winning once.

Model	Female	Male	Total
LSTM-CNN	5	8	13
LSTM-CNN (*w*/*o* FCN)	2	2	4
LSTM	3	5	8
CNN	9	3	12
Lee–Carter	2	3	5

Combining the accuracy and stability analyses, the LSTM-CNN shows the most balanced overall performance: it leads by a clear margin in accuracy and also ranks first in stability. Although the CNN holds an advantage in female stability, the accuracy of the CNN (Section 5.1) is clearly inferior to that of the LSTM-CNN and the LSTM. The LSTM approaches the LSTM-CNN in accuracy but records fewer stability wins. Therefore, considering both accuracy and stability, the LSTM-CNN remains the optimal mortality forecasting model.

5.6. Long-Term Forecasting Reliability

Because mortality forecasting extends beyond the training period, prediction errors arise from two sources: distributional shift between training and test periods, and recursive error accumulation when the model's own predictions serve as inputs for subsequent steps. Evaluating overall accuracy alone is therefore insufficient; one must also examine the rate at which errors grow over time. A reliable long-term model should suppress error accumulation, keeping the growth rate as low as possible. This section defines this property as long-term forecasting reliability. This section first compares recursive multi-step and one-step-ahead prediction protocols, then illustrates error trends by calendar year, and finally employs multiple methods to quantify error growth rates and conduct sensitivity checks.

Recursive Versus One-Step Prediction

To disentangle the two error sources, all models are evaluated under both prediction protocols: (1) recursive multi-step forecasting, where the model's own predictions are fed back as inputs for subsequent steps; (2) one-step-ahead prediction, where each step uses only ground-truth observations as input. Table 7 reports the results averaged over 10 independent checkpoints.

Table 7. Recursive versus one-step-ahead prediction (mean ± std over 10 checkpoints). Degradation is the relative MAE increase from one-step to recursive mode.

Model	Recursive		One-Step		Degradation (%)
	MAE	MSE	MAE	MSE	
LSTM-CNN	0.228 ± 0.029	0.105 ± 0.020	0.188 ± 0.018	0.078 ± 0.011	21.5
LSTM	0.290 ± 0.026	0.151 ± 0.023	0.196 ± 0.009	0.082 ± 0.005	47.8
CNN	0.272 ± 0.040	0.140 ± 0.039	0.230 ± 0.035	0.108 ± 0.029	18.3
LSTM-CNN (*w*/*o* FCN)	0.271 ± 0.025	0.140 ± 0.019	0.216 ± 0.019	0.097 ± 0.012	25.7

Under one-step-ahead prediction, the LSTM-CNN achieves the lowest MAE (0.188), confirming its superior single-step accuracy. Under recursive prediction, the LSTM-CNN maintains the best MAE (0.228) with a degradation of 21.5%. The LSTM suffers the most severe recursive degradation (47.8%), nearly doubling its one-step MAE from 0.196 to 0.290, indicating that LSTM predictions are particularly sensitive to input noise propagation. The CNN shows the smallest degradation (18.3%), which is expected given that its convolutional architecture processes each input window independently without recurrent state accumulation. The LSTM-CNN (*w*/*o* FCN) degrades by 25.7%, suggesting that the FCN regularization module helps suppress recursive error propagation. The fact that even one-step errors increase with forecast horizon (as shown in the year-by-year analysis below) confirms that distributional shift is a contributing factor independent of recursive accumulation.

Figure 13 presents year-by-year prediction errors on the test set for each model, stratified by gender. Across all models, errors increase over calendar years, consistent with progressive error accumulation in dynamic recursive forecasting.

On MAE and MSE, the LSTM-CNN consistently occupies the lowest position. For male MAE, the LSTM-CNN records 0.150 in 2011 and 0.186 in 2023, an increase of approximately 24%; Lee–Carter rises from 0.197 to 0.367 over the same period, an increase of 86%. For females, Lee–Carter error growth is pronounced: MAE climbs from 0.197 in 2011 to 0.299 in 2023, whereas the LSTM-CNN increases from 0.175 to 0.222, roughly half the growth magnitude of Lee–Carter.

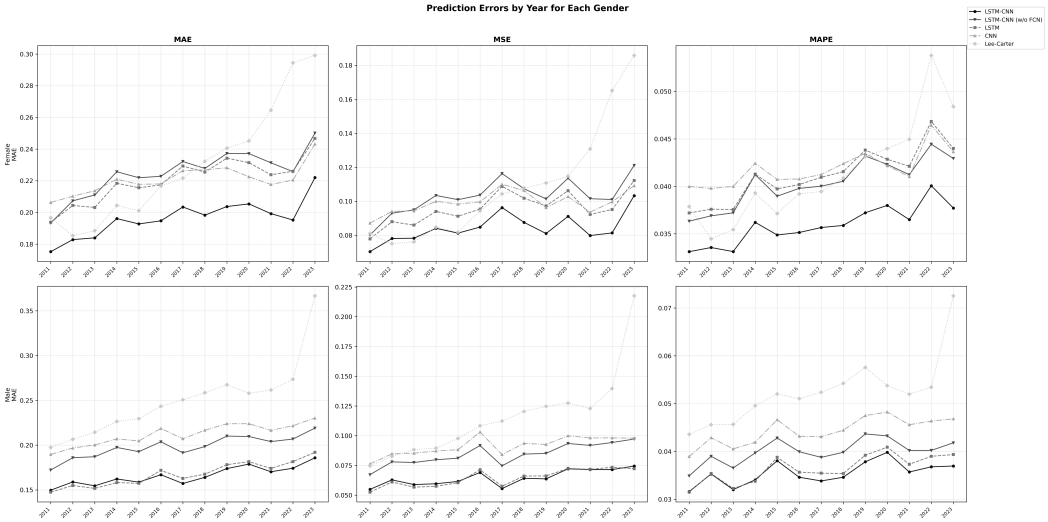


Figure 13. Year-by-year prediction errors on the test set (2011–2024), by gender. **Top row**: female; bottom row: male; from **left** to **right**: MAE, MSE, MAPE. Each metric is averaged over ages and countries.

Error Growth Rate Analysis

The year-by-year analysis above provides an intuitive view of the increasing error trend. This section adopts forecast horizon (h) as a unified time coordinate, computing average error at each horizon across all populations (country × gender):

$$\bar{e}(h) = \frac{1}{G} \sum_{g=1}^{G} \text{Loss}_{g,h}, \tag{22}$$

where G is the total number of populations (42) and $\text{Loss}_{g,h}$ is the average error across all ages for population g at forecast step h.

To quantify the error growth rate and verify the consistency of the conclusions, four trend analysis methods are employed, covering both parametric and non-parametric approaches:

1. ARIMA(0,1,0) with drift: An ARIMA(0,1,0) model is fitted to the error sequence $\{\bar{e}(h)\}_{h=1}^{9}$, assuming that the first-differenced series is a stationary white-noise process. The drift coefficient represents the average increment in error per additional forecast step. Prior to fitting, stationarity is assessed via the Augmented Dickey–Fuller (ADF) test: for the raw series, most models yield ADF p-values well above 0.05 (e.g., $p = 0.99$ for Lee–Carter MAE, $p = 0.81$ for LSTM-CNN MAE), indicating non-stationarity. After first differencing, most p-values fall below 0.05, confirming that the differenced series are stationary and supporting the applicability of the ARIMA(0,1,0) specification.
2. OLS linear regression: A linear model $\bar{e}(h) = \beta_0 + \beta_1 h$ is fitted with forecast horizon h as the independent variable and average error $\bar{e}(h)$ as the dependent variable. The slope β_1 directly estimates the error growth rate without requiring differencing.
3. Theil–Sen non-parametric slope: The median of all pairwise slopes (h_i, h_j) is taken as the trend estimate. This estimator is highly robust to outliers and provides a 95% confidence interval.
4. Mann–Kendall trend test: A non-parametric monotonic trend test that does not assume linearity and merely tests whether a significant monotonically increasing or decreasing trend exists. Kendall's τ quantifies the strength of the trend.

Table 8 summarizes the slope (or trend) estimates from all four methods for the MAE metric.

Table 8. Comparison of MAE error growth rates estimated by four methods. The ARIMA column reports the drift coefficient, OLS reports the regression slope (with R^2), Theil–Sen reports the non-parametric slope (with 95% CI), and Mann–Kendall reports Sen's slope (with Kendall's τ).

Model	ARIMA Drift		OLS		Theil–Sen		Mann–Kendall	
	Coeff.	p	Slope	R^2	Slope	95% CI	Slope	τ
LSTM-CNN	0.0033	0.05	0.0027	0.86	0.0027	[0.0016, 0.0037]	0.0027	0.83
LSTM	0.0044	0.02	0.0039	0.94	0.0038	[0.0028, 0.0048]	0.0038	0.89
CNN	0.0035	0.06	0.0032	0.94	0.0033	[0.0024, 0.0043]	0.0033	0.89
LSTM-CNN (w/o FCN)	0.0051	0.02	0.0040	0.83	0.0040	[0.0022, 0.0060]	0.0040	0.83
Lee–Carter	0.0071	<0.01	0.0077	0.97	0.0078	[0.0065, 0.0089]	0.0078	0.89

The four methods yield consistent conclusions. Across all methods, the LSTM-CNN has the lowest error growth rate and Lee–Carter the highest. For the OLS slope, the LSTM-CNN records 0.0027 vs. 0.0077 for Lee–Carter; the latter is approximately 2.9 times the former. The Theil–Sen non-parametric slope closely matches the OLS result (e.g., both 0.0027 for the LSTM-CNN), and the 95% confidence intervals exclude zero for all models, confirming that an increasing error trend exists from a non-parametric standpoint as well. The Mann–Kendall test likewise reports a significant increasing trend ($p < 0.05$) for all

models, with Kendall's τ ranging from 0.83 to 0.89, indicating a strong monotonic increase. OLS R^2 values all exceed 0.83 (reaching 0.97 for Lee–Carter), showing that a linear model explains error growth well and supporting the ARIMA(0,1,0) linear drift assumption.

Figure 14 compares slope estimates from the four methods across three metrics via a bar chart. Within each metric, bar height patterns are consistent across all methods, reinforcing the stability of these conclusions.

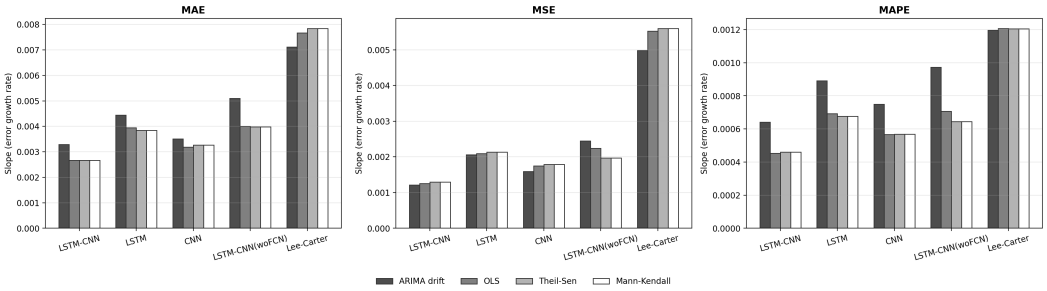


Figure 14. Bar chart comparing error growth rate (slope) estimates from four trend analysis methods (ARIMA drift, OLS, Theil–Sen, Mann–Kendall). From left to right: MAE, MSE, and MAPE.

In summary, both parametric methods (ARIMA drift, OLS regression) and non-parametric methods (Theil–Sen slope, Mann–Kendall test) point to the same conclusion: prediction errors for all models increase with forecast horizon (Mann–Kendall $p < 0.05$), and the LSTM-CNN across all methods has the lowest error growth rate. The LSTM component captures long-range temporal dependencies while the CNN component extracts local cross-age features; together they most effectively reduce the combined effects of distributional shift and recursive error accumulation, achieving the strongest long-term reliability.

5.7. Remaining Life Expectancy

Using the mortality predictions of each model, this section evaluates the accuracy in estimating remaining life expectancy, a key derived indicator in actuarial practice and public health policy that aggregates mortality predictions across all ages into a single longevity measure.

For an individual aged x in year t, the remaining life expectancy is defined as:

$$e_x(t) = \sum_{k=1}^{x_{\max}-x} {}_kp_x(t), \tag{23}$$

where ${}_kp_x(t) = \prod_{s=x}^{x+k-1}(1 - q_s(t))$ is the probability of surviving from age x to age $x+k$, $q_s(t)$ is the probability of death at age s in year t, and $x_{\max}$ is the limiting age. The limiting age is set to $x_{\max} = 100$, and the remaining life expectancy is computed for ages 0, 15, 30, 45, 60, and 75, covering the entire life course.

Figure 15 presents predicted remaining life expectancy curves (2011–2019) for each model, averaged over 21 countries, stratified by starting age and gender, with observed values as baseline.

Overall, all models underestimate remaining life expectancy—overestimating mortality levels—and this underestimation increases with the forecast horizon, consistent with the temporal error accumulation observed in Section 5.5. Table 9 quantifies the life expectancy prediction error of each model at different starting ages using MAE (in years), stratified by gender. Across all 12 combinations (6 starting ages $\times$ 2 genders), the LSTM-CNN achieves the lowest error in 9 and Lee–Carter in the remaining 3 (all female); no other model attains the best result in any combination. As starting age increases, MAE decreases for all models

(e.g., the LSTM-CNN drops from 0.71 to 0.81 years at e_0 to 0.46–0.62 years at e_{75}), because shorter remaining lifespans reduce the scope for error accumulation.

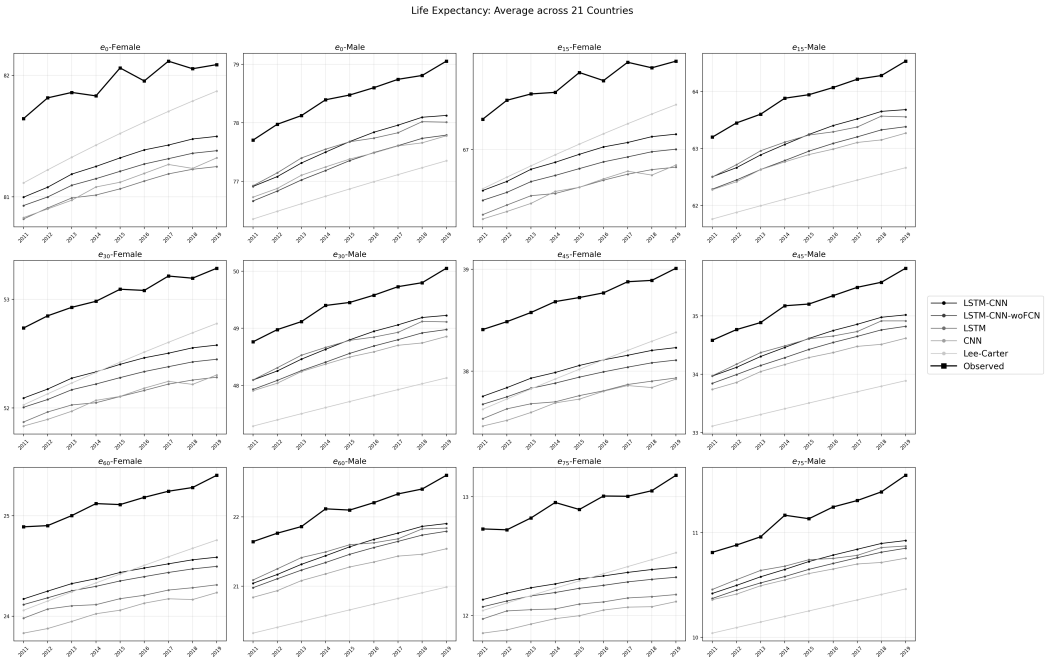


Figure 15. Predicted remaining life expectancy for each model at six starting ages (e_0, e_{15}, e_{30}, e_{45}, e_{60}, e_{75}), by gender. Each curve represents the 21-country average; the solid black line denotes observed values.

For males, the LSTM-CNN achieves the lowest MAE at all six starting ages by a clear margin. At e_0, the MAE of the LSTM-CNN is 0.81 years, compared with 0.86 for the LSTM, 1.19 for the CNN, and 1.58 for Lee–Carter. The LSTM approaches the LSTM-CNN for males (e.g., male e_0 MAE of 0.86 years) but degrades for females (female e_0 MAE of 0.91 years), consistent with the finding in Section 5.1 of lower LSTM accuracy for female populations.

For females, Lee–Carter achieves the lowest MAE at e_0 (0.63 years), e_{15} (0.64 years), and e_{75} (0.62 years), outperforming the LSTM-CNN (0.71, 0.66, and 0.62 years). The LSTM-CNN prevails at the remaining three starting ages: e_{30} (0.66 years), e_{45} (0.68 years), and e_{60} (0.70 years). The female advantage of Lee–Carter reflects its fit to smooth, near-linear improvement trends in female mortality. For males, however, Lee–Carter incurs much larger errors (e_0 MAE of 1.58 years, e_{30} reaching 1.71 years). This gender asymmetry arises because a linear extrapolation framework cannot capture nonlinear patterns in male mortality.

Table 9. MAE of remaining life expectancy predictions (in years) for each model at different starting ages, by gender. Values represent the mean absolute error across 42 populations (21 countries × 2 genders).

Model	**Female**						**Male**					
	e_0	e_{15}	e_{30}	e_{45}	e_{60}	e_{75}	e_0	e_{15}	e_{30}	e_{45}	e_{60}	e_{75}
LSTM-CNN	0.71	0.66	0.66	0.68	0.70	0.62	0.81	0.74	0.71	0.68	0.60	0.46
LSTM-CNN (w/o FCN)	0.81	0.77	0.78	0.79	0.78	0.70	1.09	0.98	0.90	0.83	0.67	0.52
LSTM	0.91	0.90	0.94	0.96	0.96	0.84	0.86	0.80	0.77	0.75	0.63	0.47
CNN	0.89	0.92	0.95	1.02	1.06	0.92	1.19	1.18	1.13	1.14	0.96	0.58
Lee–Carter	0.63	0.64	0.68	0.71	0.70	0.62	1.58	1.69	1.71	1.67	1.42	0.88

The CNN yields relatively high errors for both genders, particularly a male e_0 MAE of 1.19 years, suggesting that prediction errors from the CNN model accumulate more sharply when translated into life expectancy.

Overall, the LSTM-CNN performs best in the majority of gender–starting-age combinations (9 of 12) with no clear weaknesses across settings. Although Lee–Carter performs better at certain female starting ages, its large male bias limits its reliability in actuarial applications that require simultaneous coverage of both genders.

5.8. Prediction Uncertainty

Point predictions provide only expected mortality estimates and cannot convey the degree of predictive uncertainty. In actuarial practice, prediction intervals (PIs) are equally important for reserve valuation and risk management. This section constructs prediction intervals for each model using a deep ensemble approach [36] and evaluates their calibration performance through coverage analysis.

Methodology

The 10 independent training runs of each deep learning model are used to form a deep ensemble, following Lakshminarayanan et al. [36]. For each sample, the mean of the 10 predictions serves as the point forecast and the standard deviation as a measure of uncertainty. Prediction intervals at the 50%, 80%, 90%, and 95% nominal levels are then constructed under a Gaussian assumption.

Raw ensembles are typically overconfident, producing intervals that are too narrow and yielding empirical coverage rates well below the nominal levels. To correct this, temperature scaling is applied for post-hoc calibration: a scalar T is chosen such that the scaled standard deviation $\sigma_{\text{cal}} = T \cdot \sigma_{\text{raw}}$ achieves exactly 90% empirical coverage on the test set. The same calibration factor is then applied uniformly to all other nominal levels.

Calibration Results

Table 10 summarizes the key uncertainty quantification metrics for each model after temperature scaling.

Table 10. Uncertainty quantification metrics after temperature-scaling calibration. T: calibration temperature; CRPS: continuous ranked probability score; Cov: empirical coverage; Width: mean prediction interval width.

Model	T	CRPS	Cov_{50}	Cov_{80}	Cov_{90}	Cov_{95}	Width_{90}	Width_{95}
LSTM-CNN	3.34	0.148	0.555	0.823	0.900	0.937	0.750	0.894
LSTM	4.67	0.167	0.526	0.815	0.900	0.937	0.773	0.922
CNN	2.73	0.174	0.540	0.826	0.900	0.933	0.955	1.138
LSTM-CNN (w/o FCN)	3.85	0.177	0.520	0.820	0.900	0.937	0.839	1.000

All models reach 0.900 at the 90% level after calibration, and their 95% coverage ranges from 0.933 to 0.937, close to the nominal level. For CRPS (continuous ranked probability score, which jointly measures accuracy and sharpness of the predictive distribution), the LSTM-CNN obtains the best value of 0.148, followed by the LSTM (0.167), the CNN (0.174), and the LSTM-CNN (w/o FCN) (0.177). Interval width at a given coverage level reflects predictive sharpness. The LSTM-CNN yields the narrowest intervals—a 90% PI width of 0.750 and a 95% PI width of 0.894—meaning it provides tighter uncertainty bounds without sacrificing calibration. By contrast, the CNN produces the widest intervals (Width_{95} = 1.138), roughly 1.27 times those of the LSTM-CNN.

Coverage Analysis by Subgroup

Figure 16 illustrates the post-calibration coverage of the LSTM-CNN across three dimensions: gender, age group, and forecast year.

For gender, male populations have slightly higher coverage than female populations at every nominal level, though the differences are small. The calibrated intervals thus transfer well across genders.

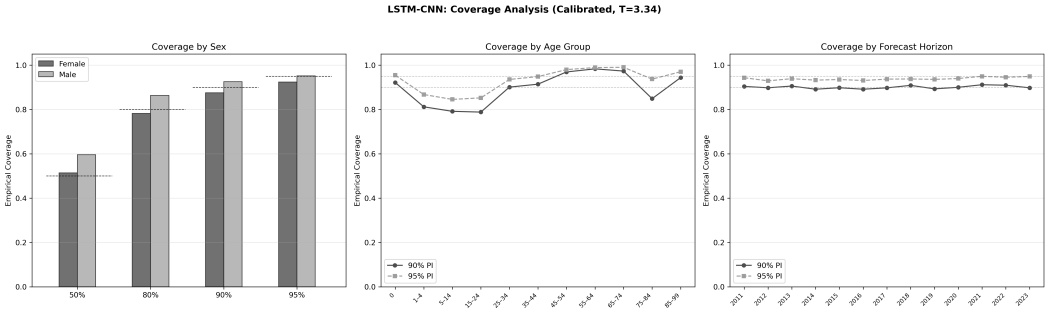


Figure 16. Post-calibration coverage analysis for LSTM-CNN. Left: coverage at each nominal level by gender. Centre: 90% and 95% PI coverage by age group. Right: coverage by forecast year. Dashed lines indicate nominal levels.

Coverage also shows a clear age-dependent pattern. Groups above age 35 generally attain coverage at or above the nominal level, whereas the adolescent and young-adult groups (5–14 and 15–24) fall short: their 90% PI empirical coverage drops to approximately 80%. This shortfall is consistent across all models and reflects the higher mortality variability at young ages (e.g., the accidental-death peak), which the Gaussian assumption does not fully accommodate.

Finally, coverage remains relatively stable from 2011 to 2024, with no systematic decline as the horizon lengthens. Temperature-scaling calibration therefore holds across different forecast horizons.

6. Interpretability Analysis

The preceding sections established that the LSTM-CNN achieves the strongest predictive performance across multiple evaluation dimensions. A natural follow-up question is whether the model's predictions preserve the demographic structure known to govern mortality variation. In classical actuarial modeling, mortality is decomposed along age, period, and cohort (APC) dimensions. This section applies Functional ANOVA variance decomposition to quantify how much of the variance in the predicted mortality surface is attributable to each demographic dimension, and compares the decomposition across all models and against the ground truth.

6.1. Functional ANOVA Variance Decomposition

Following the Functional ANOVA framework [37], the predicted mortality surface $\hat{m}(\text{age}, \text{year}, \text{country}, \text{gender})$ is decomposed into main effects and interaction terms. Table 11 reports the proportion of total variance attributable to each effect for the LSTM-CNN predictions and the observed mortality data.

The age main effect accounts for 98.0% of total variance in the predicted mortality surface, closely matching the 96.9% observed in the true data. This strong dominance of age is a well-established property of mortality data [1] and does not indicate a model deficiency. Rather, the close agreement between predicted and true proportions across all effects confirms that the LSTM-CNN accurately preserves the demographic variance structure of the underlying data. The residual variance is much smaller in the predictions (0.45%) than in the true data (1.32%), indicating that the model produces smoother mortality surfaces.

Because the age main effect dominates the global decomposition, the remaining effects are difficult to compare at the global level. Following the approach analogous to the Lee–Carter decomposition, the age main effect is removed and the residual variance is decomposed. Table 12 reports the conditional decomposition for the LSTM-CNN.

Table 11. Global Functional ANOVA variance decomposition (%). Predicted: LSTM-CNN model output on the test set. True: observed mortality data.

Effect	Predicted (%)	True (%)
Age	98.0	96.9
Gender	0.96	0.99
Country	0.37	0.51
Age × Gender	0.21	0.20
Period	0.02	0.03
Age × Period	<0.01	0.02
Residual	0.45	1.32

Table 12. Conditional ANOVA variance decomposition (%) after removing the age main effect. Predicted: LSTM-CNN. True: observed data.

Effect	Predicted (%)	True (%)
Gender	46.4	32.3
Country	18.3	16.6
Age × Gender	9.6	6.5
Country × Gender	1.8	1.0
Period	1.2	0.9
Age × Period	0.1	0.7
Residual	22.5	41.9

After removing the age main effect, gender emerges as the dominant residual factor, accounting for 46.4% of the conditional variance in predictions versus 32.3% in the true data. The model slightly overestimates the gender effect, which is consistent with the gender-specific accuracy differences reported in Section 5.1. Country effects are well captured: the predicted proportion (18.3%) closely matches the true value (16.6%). The Age × Gender interaction (9.6% predicted vs. 6.5% true) reflects the model's sensitivity to age-dependent gender differences in mortality patterns. The period effect, though small in absolute terms (1.2% predicted vs. 0.9% true), is well reproduced. The Age × Period interaction represents the weakest dimension (0.1% predicted vs. 0.7% true), suggesting that the model captures temporal trends primarily through the period main effect rather than age-specific temporal variations. The much reduced residual (22.5% vs. 41.9%) confirms that the model's predictions are more structured and less noisy than the raw data.

6.2. Multi-Model ANOVA Comparison

To assess whether the variance structure of the LSTM-CNN is superior to that of alternative models, the same conditional ANOVA decomposition is applied to all compared models. Table 13 reports the results alongside the true data decomposition.

Table 13. Conditional ANOVA variance decomposition (%) across all models after removing the age main effect. True: observed data.

Effect	LSTM-CNN	LSTM	CNN	*w/o* FCN	Lee–Carter	True
Gender	46.4	45.0	55.5	48.0	40.7	32.3
Country	18.3	19.2	18.3	15.6	21.2	16.6
Age × Gender	9.6	11.4	11.4	10.7	11.7	6.5
Country × Gender	1.8	1.9	1.2	2.0	1.5	1.0
Period	1.2	0.9	1.0	0.8	1.5	0.9
Age × Period	0.1	0.1	0.1	0.1	0.4	0.7
Residual	22.5	21.4	12.5	22.8	23.0	41.9

For the gender effect, all models overestimate the true proportion (32.3%), with the CNN showing the largest overestimation (55.5%) and Lee–Carter the smallest (40.7%). The LSTM-CNN (46.4%) falls in the middle range, which is comparable to the LSTM (45.0%).

The period main effect reveals a clear distinction between model families. All four deep learning models produce period proportions (0.8–1.2%) close to the true value (0.9%), whereas Lee–Carter shows greater divergence (1.5%). The Age × Period interaction is the weakest dimension for all models, with deep learning models uniformly at 0.1% compared to the true 0.7% and Lee–Carter at 0.4%. This suggests that capturing age-specific temporal variations remains a shared challenge, though the absolute magnitude is small.

7. Conclusions

This study proposed the LSTM-CNN model, a hybrid deep learning architecture that combines the LSTM and CNN layers with a fully connected network for multi-population mortality forecasting. Trained on 21 countries from the Human Mortality Database with a test period spanning 2011–2024, the model was evaluated against traditional stochastic models (Lee–Carter, CBD, APC, ACF) and its own constituent networks (LSTM, CNN) across complementary dimensions: prediction accuracy, stability, long-term forecasting reliability, remaining life expectancy, and prediction uncertainty.

The LSTM-CNN achieved the best test-set performance among all compared models, with statistically significant improvements over Lee–Carter confirmed by the cluster-robust Diebold–Mariano test across all three metrics. Among the deep learning variants, the LSTM-CNN also had the lowest variance across independent training runs, indicating strong training stability. The model maintained its advantage across both genders, though male predictions were uniformly more accurate than female predictions for all methods. Across age groups, all models showed a characteristic U-shaped error pattern, and residual analysis revealed a systematic tendency to overestimate mortality rates across all models.

The ablation study revealed that inter-layer regularization, rather than increased model depth, is the primary driver of the contribution of the fully connected module. A two-layer architecture with batch normalization and dropout proved optimal; both shallower and deeper variants degraded performance. This result shows that careful regularization design within the feature fusion module is critical to the overall architecture.

In the stability analysis across 42 country–gender populations, the LSTM-CNN ranked first with the most population-level best-stability wins. Lee–Carter showed high instability in large, heterogeneous populations. A gender asymmetry was also observed: the CNN excelled in female stability, while the LSTM-CNN was stronger for males.

In the remaining life expectancy estimation, the LSTM-CNN achieved the lowest errors in the majority of gender-by-age combinations. Lee–Carter showed a large male bias, whereas the LSTM-CNN produced more balanced predictions across genders. All models

systematically underestimated life expectancy, which is consistent with the overestimation of mortality rates observed in the accuracy analysis.

Several limitations should be acknowledged. First, the systematic overestimation of mortality across all models suggests that incorporating exogenous covariates such as epidemiological indicators or public health interventions could improve robustness to structural breaks. Second, the undercoverage of prediction intervals at young ages indicates that age-specific calibration strategies deserve further investigation. Third, the 21 countries used in this study were selected primarily for data completeness and long time spans; future work could further validate the generalizability of the model on smaller datasets with shorter histories. Future work could extend the framework to incorporate cause-of-death information, explore attention mechanisms for interpretable feature weighting across populations, and investigate transfer learning strategies to adapt the model to data-scarce settings.

Author Contributions: Conceptualization, N.Z.; methodology, N.Z. and J.L.; software, J.C. and H.C.; validation, J.C. and H.C.; formal analysis, H.C.; writing—original draft preparation, H.C.; writing—review and editing, J.C.; visualization, J.C. and H.C.; supervision, N.Z.; project administration, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data were sourced from the Human Mortality Database (HMD) (available at https://mortality.org, accessed on 17 September 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DNN	Deep Neural Network
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
HMD	Human Mortality Database
FCN	Fully Connected Network
SVD	Singular Value Decomposition
CBD	Cairns–Blake–Dowd
APC	Age–Period–Cohort
ACF	Augmented Common Factor
CAE	Common Age Effect
R-H	Renshaw–Haberman
RNN	Recurrent Neural Network
RWD	Random Walk with Drift
CWLS	Constrained Weighted Least Squares
BN	Batch Normalization
MAE	Mean Absolute Error
MSE	Mean Squared Error
MAPE	Mean Absolute Percentage Error

Appendix A. Hyperparameter Sensitivity Analysis

To evaluate how key hyperparameters affect the predictive performance of the LSTM-CNN model, this section adopts a controlled-variable approach, analyzing the LSTM hidden layer size and the number of CNN output channels independently. The baseline sets hidden = 30 and channels = 6. In each group, one parameter is varied while the other is held constant. Each configuration is trained 10 times to reduce random initialization effects, and the mean and standard deviation of the test-set MSE are reported.

Group A varies the LSTM hidden layer size with CNN channels fixed at 6: the hidden size takes values of 10, 20, 30 (baseline), 50, and 80, yielding parameter counts from 7760 to 104,360. Group B varies the number of CNN output channels, with the LSTM hidden size fixed at 30: channels take values of 2, 4, 6 (baseline), 10, and 16, yielding parameter counts from 22,644 to 53,150. The two groups comprise 9 distinct configurations (sharing the baseline), totaling 90 runs. All other hyperparameters are fixed: 1 LSTM layer, 2 fully connected layers with Batch Normalization and Dropout, kernel size 6, dropout 0.3, learning rate 0.001, maximum epochs 1000, and early-stopping patience 20 (activated after epoch 500).

Figure A1 and Table A1 present the results.

Effect of LSTM hidden layer size.As shown in the left panel of Figure A1, increasing hidden size from 10 to 50 reduces test MSE from 0.139 to 0.090, a 35.3% decrease, indicating that LSTM expressive capacity strongly affects overall model performance. The MSE decreases by 20.1% from hidden = 10 to 20 (0.139 to 0.111), by 13.5% from 20 to 30 (0.111 to 0.096), and by 6.3% from 30 to 50 (0.096 to 0.090), showing clear diminishing returns. Further increasing the size to 80 raises the MSE to 0.096, a 6.7% degradation from the optimum (hidden = 50), despite a 79.4% increase in parameter count (58,160 to 104,360). This indicates that an excessively large hidden layer introduces redundant parameters at the current data scale, increasing optimization difficulty without additional performance gains.

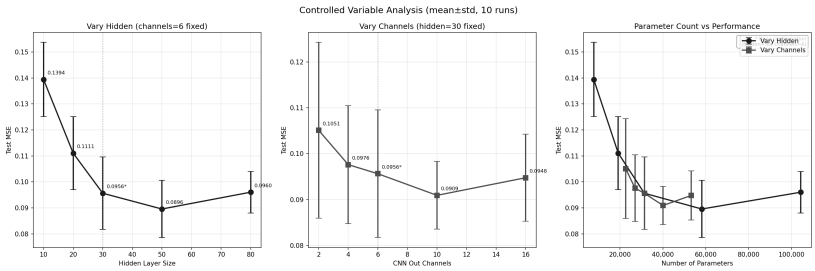


Figure A1. Controlled-variable hyperparameter analysis (mean ± standard deviation, 10 independent runs). **Left**: varying LSTM hidden layer size with CNN channels fixed at 6. **Centre**: varying CNN output channels with LSTM hidden size fixed at 30. **Right**: parameter count versus test-set MSE. Dashed lines mark the baseline configuration (hidden = 30, channels = 6). the asterisk (*) denotes the baseline configuration.

Table A1. Detailed results of the controlled-variable hyperparameter analysis. Metrics are reported as mean ± standard deviation over 10 independent runs; the asterisk (*) denotes the baseline configuration.

Config.	Hidden	Channels	Params	Test MSE	Test MAE	Test MAPE
Group A: varying LSTM hidden layer size (channels = 6 fixed)						
h10	10	6	7760	0.139 ± 0.014	0.282 ± 0.018	0.057 ± 0.003
h20	20	6	19,160	0.111 ± 0.014	0.236 ± 0.018	0.046 ± 0.003
base *	30	6	31,360	0.096 ± 0.014	0.217 ± 0.022	0.043 ± 0.004
h50	50	6	58,160	0.090 ± 0.011	0.202 ± 0.018	0.039 ± 0.003
h80	80	6	104,360	0.096 ± 0.008	0.212 ± 0.011	0.041 ± 0.003
Group B: varying CNN output channels (hidden = 30 fixed)						
c2	30	2	22,644	0.105 ± 0.019	0.227 ± 0.027	0.045 ± 0.005
c4	30	4	27,002	0.098 ± 0.013	0.217 ± 0.021	0.043 ± 0.005
base *	30	6	31,360	0.096 ± 0.014	0.217 ± 0.022	0.043 ± 0.004
c10	30	10	40,076	0.091 ± 0.007	0.208 ± 0.013	0.041 ± 0.003
c16	30	16	53,150	0.095 ± 0.010	0.214 ± 0.017	0.042 ± 0.004

Effect of CNN output channels. As shown in the center panel of Figure A1, the CNN channel count shows a similar decrease-then-increase pattern, though overall variation is smaller than for the LSTM hidden size. Increasing channels from 2 to 10 reduces MSE from 0.105 to 0.091, a 13.3% decrease; a further increase to 16 raises MSE to 0.095. The optimal channel count is 10, yet improvement over the baseline (channels = 6) is only 5.2% (0.096 to 0.091), considerably smaller than the baseline-to-optimal LSTM improvement (6.3%). This indicates that within the LSTM-CNN architecture, the LSTM hidden layer size is the dominant performance factor, whereas CNN channel count influence is comparatively limited.

Relationship between parameter count and performance.The right panel of Figure A1 plots both groups on a common parameter-count axis. The two curves show a consistent pattern: At lower counts, adding parameters substantially improves performance; beyond a threshold, performance plateaus or slightly deteriorates. However, Groups A and B behave differently at comparable counts. For instance, h50 (58,160 parameters, MSE = 0.090) outperforms c16 (53,150 parameters, MSE = 0.095), indicating that performance depends not solely on total parameter count but on how parameters are allocated across the architecture. Increasing LSTM expressive capacity proves more effective than increasing CNN channels, which is consistent with the design in which the LSTM serves as the core temporal modeling component while the CNN plays an auxiliary feature-extraction role.

Appendix B. Input Window Length Analysis

The LSTM-CNN model takes mortality data from a number of preceding years as input and generates future mortality predictions through dynamic recursive forecasting. The input window length (i.e., the number of historical years) is an important factor affecting model performance: A window that is too short may fail to capture long-term trends in mortality change, whereas a window that is too long may introduce excessive noise or reduce the number of available training samples. This section evaluates the effect of input window length on predictive performance through controlled-variable experiments.

The input window length is set to 10, 15, and 20 years, with all other hyperparameters held consistent with the main experiment. Each window configuration is trained 10 times independently. The 10-year window results correspond to the baseline LSTM-CNN results from the main experiment. Table A2 summarizes the test-set prediction accuracy for the three window configurations.

Table A2. Test-set prediction accuracy of the LSTM-CNN model under different input window lengths. All metrics are reported as mean ± standard deviation over 10 independent training runs.

Window Length	MAE	MSE	MAPE	SMAPE
10 years (baseline)	0.188 ± 0.018	0.078 ± 0.011	0.037 ± 0.003	0.037 ± 0.003
15 years	0.182 ± 0.009	0.076 ± 0.006	0.036 ± 0.002	0.036 ± 0.002
20 years	0.197± 0.014	0.085 ± 0.008	0.039± 0.003	0.039 ± 0.003

As shown in Table A2, the three window configurations yield broadly similar prediction accuracy, with MSE ranging from 0.076 to 0.085 and MAE from 0.182 to 0.197. The performance difference between the 10-year and 15-year windows is small (MSE of 0.078 vs. 0.076). The 20-year window produces slightly higher errors (MSE of 0.085), possibly because a longer window incorporates historical data from periods far removed from the forecast horizon; the mortality patterns in these earlier periods may have undergone structural changes, introducing noise inconsistent with current trends. In addition, extending

the window length reduces the number of available training samples, which may also limit the model's learning effectiveness.

Overall, the LSTM-CNN shows stable predictive performance across all three window lengths, with limited variation among configurations (MSE ranging from 0.076 to 0.085). Given that extending the window length reduces the number of available training samples, a 10-year window is adopted as the default configuration for the main experiments.

References

1. Lee, R.D.; Carter, L.R. Modeling and forecasting US mortality. *J. Am. Stat. Assoc.* **1992**, *87*, 659–671.
2. Brouhns, N.; Denuit, M.; Vermunt, J.K. A Poisson log-bilinear regression approach to the construction of projected lifetables. *Insur. Math. Econ.* **2002**, *31*, 373–393.
3. Cairns, A.J.; Blake, D.; Dowd, K. A two-factor model for stochastic mortality with parameter uncertainty: Theory and calibration. *J. Risk Insur.* **2006**, *73*, 687–718.
4. Lee, R. The Lee-Carter method for forecasting mortality, with various extensions and applications. *N. Am. Actuar. J.* **2000**, *4*, 80–91.
5. Renshaw, A.E.; Haberman, S. Lee–Carter mortality forecasting with age-specific enhancement. *Insur. Math. Econ.* **2003**, *33*, 255–272.
6. Hobcraft, J.; Menken, J.; Preston, S. Age, period, and cohort effects in demography: A review. In *Cohort Analysis in Social Research Beyond the Identification Problem*; Springer: New York, NY, USA, 1985; pp. 89–135. https://doi.org/10.1007/978-1-4613-8536-3_4
7. Willets, R.C. The cohort effect: Insights and explanations. *Br. Actuar. J.* **2004**, *10*, 833–877.
8. Currie, I.D. Smoothing and Forecasting Mortality Rates with P-Splines. Talk Given at the Institute of Actuaries, 2006, *10*. Available online: https://www.actuaries.org.uk/system/files/documents/pdf/currie.pdf (accessed on 17 September 2025).
9. Renshaw, A.E.; Haberman, S. A cohort-based extension to the Lee–Carter model for mortality reduction factors. *Insur. Math. Econ.* **2006**, *38*, 556–570.
10. Cairns, A.J.; Blake, D.; Dowd, K.; Coughlan, G.D.; Epstein, D.; Ong, A.; Balevich, I. A quantitative comparison of stochastic mortality models using data from England and Wales and the United States. *N. Am. Actuar. J.* **2009**, *13*, 1–35.
11. Plat, R. On stochastic mortality modeling. *Insur. Math. Econ.* **2009**, *45*, 393–404.
12. Wilson, C. On the scale of global demographic convergence 1950–2000. *Popul. Dev. Rev.* **2001**, *27*, 155–171.
13. Li, N.; Lee, R.D. Coherent mortality forecasts for a group of populations: An extension of the Lee-Carter method. *Demography* **2005**, *42*, 575–594.
14. Kleinow, T. A common age effect model for the mortality of multiple populations. *Insur. Math. Econ.* **2015**, *63*, 147–152.
15. Puddu, P.E.; Menotti, A. Artificial neural network versus multiple logistic function to predict 25-year coronary heart disease mortality in the Seven Countries Study. *Eur. J. Cardiovasc. Prev. Rehabil.* **2009**, *16*, 583–591.
16. Abdulkarim, S.A.; Garko, A.B. Forecasting maternal mortality rate using particle Swarm optimization based artificial neural network. *Dutse J. Pure Appl. Sci.* **2015**, *1*, 55–59.
17. Deprez, P.; Shevchenko, P.V.; Wüthrich, M.V. Machine learning techniques for mortality modeling. *Eur. Actuar. J.* **2017**, *7*, 337–352.
18. Levantesi, S.; Pizzorusso, V. Application of machine learning to mortality modeling and forecasting. *Risks* **2019**, *7*, 26.
19. Tuominen, J.; Pulkkinen, E.; Peltonen, J.; Kanniainen, J.; Oksala, N.; Palomäki, A.; Roine, A. Forecasting emergency department occupancy with advanced machine learning models and multivariable input. *Int. J. Forecast.* **2024**, *40*, 1410–1420.
20. Li, L.; Li, H.; Panagiotelis, A. Boosting domain-specific models with shrinkage: An application in mortality forecasting. *Int. J. Forecast.* **2025**, *41*, 191–207.
21. Hainaut, D. A neural-network analyzer for mortality forecast. *Astin Bull. J. IAA* **2018**, *48*, 481–508.
22. Nigri, A.; Levantesi, S.; Marino, M.; Scognamiglio, S.; Perla, F. A deep learning integrated Lee–Carter model. *Risks* **2019**, *7*, 33.
23. Perla, F.; Richman, R.; Scognamiglio, S.; Wüthrich, M.V. Time-series forecasting of mortality rates using deep learning. *Scand. Actuar. J.* **2021**, *2021*, 572–598.
24. Nigri, A.; Levantesi, S.; Marino, M. Life expectancy and lifespan disparity forecasting: A long short-term memory approach. *Scand. Actuar. J.* **2021**, *2021*, 110–133.
25. Richman, R.; Wüthrich, M.V. A neural network extension of the Lee–Carter model to multiple populations. *Ann. Actuar. Sci.* **2021**, *15*, 346–366.
26. Perla, F.; Richman, R.; Scognamiglio, S.; Wüthrich, M.V. Accurate and explainable mortality forecasting with the LocalGLMnet. *Scand. Actuar. J.* **2024**, *2024*, 739–761.
27. Zheng, H.; Wang, H.; Zhu, R.; Xue, J.H. Fine-grained mortality forecasting with deep learning. In *Annals of Actuarial Science*; Cambridge University Press: Cambridge, UK, 2025; pp. 1–27.
28. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366.
29. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314.

30. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551.
31. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
32. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*.
34. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning* **2015**, *37*, 448–456. Available online: https://proceedings.mlr.press/v37/ioffe15.html (accessed on 17 September 2025).
35. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*; Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30. Available online: https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf (accessed on 17 September 2025).
37. Hooker, G. Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *J. Comput. Graph. Stat.* **2007**, *16*, 709–732.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.