Scandinavian Actuarial Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/sact20

Time-series forecasting of mortality rates using transformer

Jun Wang, Lihong Wen, Lu Xiao & Chaojie Wang

To cite this article: Jun Wang, Lihong Wen, Lu Xiao & Chaojie Wang (2023): Time-series forecasting of mortality rates using transformer, Scandinavian Actuarial Journal, DOI: 10.1080/03461238.2023.2218859

To link to this article: https://doi.org/10.1080/03461238.2023.2218859

Published online: 30 May 2023.

SCANDINAVIAN ACTUARIAL JOURNAL
https://doi.org/10.1080/03461238.2023.2218859

Taylor & Francis
Taylor & Francis Group

Time-series forecasting of mortality rates using transformer

Jun Wang[a], Lihong Wen[b], Lu Xiao[c] and Chaojie Wang[b]

[a]Institute of Applied System Analysis, Jiangsu University, Zhenjiang, People's Republic of China; [b]School of Mathematical Science, Jiangsu University, Zhenjiang, People's Republic of China; [c]School of Management, Jiangsu University, Zhenjiang, People's Republic of China

ABSTRACT

Predicting mortality rates is a crucial issue in life insurance pricing and demographic statistics. Traditional approaches, such as the Lee-Carter model and its variants, predict the trends of mortality rates using factor models, which explain the variations of mortality rates from the perspective of ages, gender, regions, and other factors. Recently, deep learning techniques have achieved great success in various tasks and shown strong potential for time-series forecasting. In this paper, we propose a modified Transformer architecture for predicting mortality rates in major countries around the world. Through the multi-head attention mechanism and positional encoding, the proposed Transformer model extracts key features effectively and thus achieves better performance in time-series forecasting. By using empirical data from the Human Mortality Database, we demonstrate that our Transformer model has higher prediction accuracy of mortality rates than the Lee-Carter model and other classic neural networks. Our model provides a powerful forecasting tool for insurance companies and policy makers.

ARTICLE HISTORY
Received 17 December 2022
Accepted 23 May 2023

KEYWORDS
Mortality rates; time series forecasting; deep learning; transformer

1. Introduction

With the improvement of living environment and medical care, the overall human mortality shows a significant downward trend after the Second World War (Klenk et al. 2016). The increase in life expectancy raises a number of social issues, such as the aging population. It also poses challenges for insurance companies to determine life insurance premiums since the pricing of life insurance must take dynamic changes in mortality into account (Pitacco et al. 2009). Thus, predicting mortality rates is a crucial issue in life insurance pricing and demographic statistics.

In 1992, Lee & Carter (1992) proposed a one-factor model to forecast the time series of mortality in the United States. The Lee-Carter (LC) model explained the variation of mortality rates in terms of age effects and the force of mortality. This seminal work provided a good fit to the empirical data by extrapolating historical rates of mortality, and then established itself as the benchmark of mortality forecasting for a long period. Cairns et al. (2009) commented that the LC model effectively combined statistical time series methods with demographic modeling. After that, many variants of the LC model were proposed to improve the prediction accuracy further. Brouhns et al. (2002) adopted a Poisson distribution to approximate death counts instead of the Gaussian assumption. In the Poisson-LC framework, parameter estimation is derived by maximum likelihood estimation

CONTACT Chaojie Wang cjwang@ujs.edu.cn, wang910930@163.com School of Mathematical Science, Jiangsu University, Zhenjiang 212013, People's Republic of China

(MLE) with the Newton-Raphson updating algorithm. Booth et al. (2002) found that the mortality index k_t in the LC model did not hold linearity over the whole fitting period. They proposed a modification called the Booth-Maindonald-Smith (BMS) model to choose the optimal fitting period automatically. Cairns et al. (2006) proposed a two-factor stochastic mortality model, denoted the Cairns-Blake-Dowd (CBD) model. They extended the LC model by introducing a factor to characterize the difference in mortality at higher and lower ages. Li & Wong (2020) proposed a modified LC method to allow for structural changes in mortality, in which the entire data period was divided into more homogeneous subperiods and a unique set of age-specific parameters was incorporated for each subperiod.

The LC model and its extensions established a simple framework to forecast and interpret mortality rates. However, the structures of factor models limit prediction accuracy. In recent years, deep learning techniques have achieved great breakthroughs in various tasks, e.g. image recognition (Krizhevsky et al. 2017), machine translation (Devlin et al. 2018), and protein structure prediction (Jumper et al. 2021). Due to their superior ability to capture the non-linear characteristics of data sets, deep neural networks also show strong potential for applications in time-series forecasting (Lim & Zohren 2021). Thus, it is an intuitive idea to consider predicting mortality rates with deep neural networks. Nigri et al. (2019) applied the long short-term memory (LSTM) within the usual two-step procedure to fit the LC model, instead of the traditional statistical time-series ARIMA model. They found that the resulting forecasts were more accurate and flexible than the classical approach. Richman & Wüthrich (2021) proposed a multi-population extension of the LC model using fully connected neural networks (FCNs) and embedding layers, which are a specialized method of incorporating categorical data into neural networks. Inspired by this method, Lindholm & Palmborg (2022) extended the Poisson Lee-Carter model using a LSTM neural network, with a particular focus on different procedures for how to use training data efficiently, combined with ensembling to stabilize the predictive performance. In order to characterize the spatial structure of the data, Scognamiglio (2022) proposed the Locally-Connected Networks (LCNs) for more robust fitting of Lee-Carter and Poisson Lee-Carter models associated with multiple populations.

Recently, more and more studies have gotten rid of the LC framework and fitted mortality rates with neural networks directly. Perla et al. (2021) incorporated recurrent neural networks (RNNs) and convolutional neural networks (CNNs) into a network model to predict mortality rates. Furthermore, Perla & Scognamiglio (2023) employed the multi-layer perceptron (MLP) for large-scale mortality modeling and forecasting with the assumption of locally-coherence among multiple populations. Besides, Wang et al. (2021) introduced a novel neighbouring prediction model using 2D CNN, which can capture the intricate nonlinear structure in the mortality data. Schnürch & Korn (2022) considered the prediction uncertainty of CNNs on mortality rates and implemented a bootstrapping-based technique to yield an interval estiamte. Although CNNs and RNNs have been shown in recent studies to be capable of predicting mortality rates, these traditional architectures still have some drawbacks. For example, the pooling layers in CNNs cause the loss of valuable information by ignoring the part-whole relationships (Xi et al. 2017); RNNs have been prone to result in vanishing and exploding gradients in the back-propagation process (Huang et al. 2019). In 2017, Vaswani et al. (2017) proposed an innovative deep learning architecture called Transformer, which uses the attention mechanism instead of traditional CNN and RNN architectures, to address natural language processing (NLP) problems. This seminal work provided a brand-new perspective on how to construct neural networks with attention mechanisms. Inspired by the success in NLP tasks, Transformer has been widely applied in various fields recently (Parmar et al. 2018, Tetko et al. 2020), including time series forecasting (Wen et al. 2022). Zhou et al. (2021) proposed an improved Transformer model called Informer for long-sequence time series forecasting. Wang et al. (2022) applied Transformer to predict the stock market index. They demonstrated that Transformer outperformed other classic methods significantly and could gain excess earnings for investors.

In this paper, we consider predicting mortality rates using the Transformer architecture. As far as we know, it is the first time that Transformer is evaluated in the task of mortality rates forecasting.

Through the multi-head attention mechanism and positional encoding, the proposed Transformer model extracts key features effectively and thus achieves better performance in time-series forecasting. Compared with the sequential structure of RNN and LSTM, the attention mechanism in Transformer can be trained in parallel, and it is easier to obtain global information. This paper explores the mortality rates of eight major countries on the Human Mortality Database (HMD). Empirically, we demonstrate that our Transformer model has higher prediction accuracy of mortality rates than the LC model and other classic neural networks. Our model provides a powerful forecasting tool for insurance companies and policy makers.

This paper is organized as follows: Section 2 introduces the background knowledge of the LC model and traditional deep learning models used in the experiments. Section 3 details the architecture of Transformer used for mortaliry rates prediction. Section 4 describes the process of empirical experiments, including data processing, parameter settings, evaluation criteria, and error analysis. Section 5 concludes this paper.

2. Background

2.1. Lee-carter model

In 1992, Lee & Carter (1992) proposed the classic LC model for mortality forecasting in the United States. Much of the attention on mortality was first stimulated by this seminal work. The LC model decomposed the log-mortality into an age-specific parameter and a time-varying component. Specifically, the LC model can be expressed as follows:

$$\ln(m_{x,t}) = \alpha_x + \beta_x \kappa_t + \epsilon_{x,t}, \tag{1}$$

where $m_{x,t}$ is the mortality rate at age x in year t, α_x is the average of age-specific mortality over time, κ_t is a time index of the general level of mortality, and β_x describes the relative mortality at each age. The error term $\epsilon_{x,t}$ reflects residual age-specific temporal influences not captured by the model, which is assumed to have zero mean and variance σ_ϵ^2.

Given the observed mortality $m_{x,t}$, the LC model first computes α_x by averaging log-mortality rates at age x over time. After subtracting α_x from each column of $\ln(m_{x,t})$ to produce a centred matrix, it performs the singular value decomposition (SVD) to derive β_x and κ_t. Specifically, β_x is obtained from the first left singular vector under the constrain $\sum_x \beta_x = 1$, and κ_t is the product of the leading singular value, the first right singular vector and the sum of the first left singular vector. Here κ_t is a univariate time series vector that captures most of the mortality trend. Under the assumption that β_x is invariant over time, the LC model predicts future mortality by extrapolating the trend of κ_t linearly.

In general, the LC model provided a simple yet powerful stochastic method to predict and interpret the variants of mortality rates. However, the structure of factor models also limits the further improvement of prediction accuracy. Thus, we consider deep learning techniques to predict mortality rates in this paper.

2.2. CNN

CNN is one of the most important deep learning architectures. Through the combination of convolutional layers and pooling layers, CNN achieves great success in image processing (Egmont-Petersen et al. 2002) and thus motivates the interest of researchers in deep learning. For time series forecasting, CNN employs a one-dimensional convolution operator to extract features from sequential data (Tang et al. 2022).

Assume that the mortality data is $\mathbf{X} = \{\boldsymbol{x}_t, t = 1, \ldots, T\}$, where $\boldsymbol{x}_t \in \mathbb{R}^d$. Here d is the dimension of ages and $\boldsymbol{x}_t$ denotes the mortality rates in year t at each age. The mortality data $\mathbf{X}$ as an input is fed into the one-dimensional convolutional layer with K filters, where the kernel size is m and the stride is s. Let $\boldsymbol{w}_j \in \mathbb{R}^d$, $j = 1, \ldots, m$, and $b_k \in \mathbb{R}$, $k = 1, \ldots, K$, be weights and biases of the convolutional

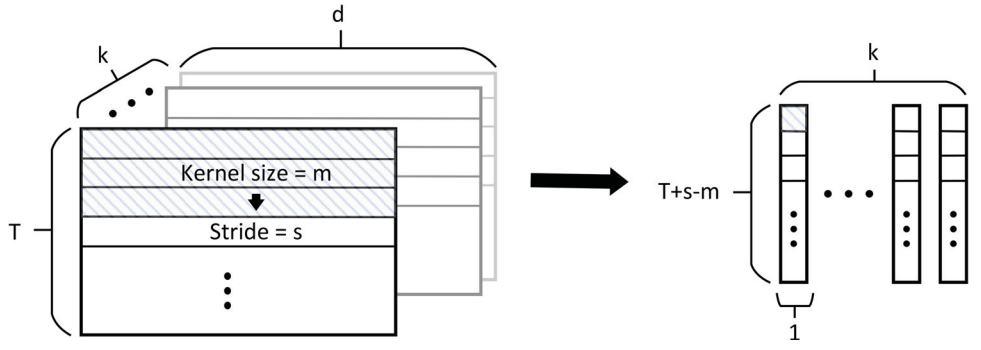


Figure 1. An example of one-dimensional convolution layer in CNN.

layer, respectively. The output of the convolutional layer is written as $\mathbf{Z} = \{\boldsymbol{z}_k, k = 1, \ldots, K\}$, where $\boldsymbol{z}_k \in \mathbb{R}^{T+s-m}$ and

$$z_{k,i} = \phi\left(b_k + \sum_{j=1}^{m} \langle \boldsymbol{w}_j, \boldsymbol{x}_{m+i-j} \rangle\right), \quad 1 \le i \le T + s - m.$$

Here $\langle \cdot, \cdot \rangle$ represents the inner product in $\mathbb{R}^d$, and $\phi : \mathbb{R} \to \mathbb{R}$ is an activation function. Figure 1 presents an example of the one-dimensional convolution layer. By using convolutional layers and pooling layers, CNN extracts the key features of time series and then outputs the final prediction through a fully connected layer.

2.3. RNN-LSTM

Compared with CNN, RNN is specifically designed for sequential data. To model the interdependent information in the context, RNN introduces an internal state called the memory cell to store past information (Hüsken & Stagge 2003). RNN provides an effective architecture to model time series, but still has some limitations. Long-term dependence in RNN, for example, may result in vanishing and exploding gradients in the back-propagation process (Huang et al. 2019).

Hochreiter & Schmidhuber (1997) proposed a variant of RNN, called LSTM, to tackle the long-term dependence problem. LSTM inherits the memory cell in RNN and adds a specific gate structure, i.e. input gate i_t, forget gate f_t, and output gate o_t. Through the gate structure, key features are stored in the memory cells and passed to the next neuron, while obsolete information can be forgotten to save memory space. Figure 2 presents the details of the LSTM architecture.

Given the mortality data is $\mathbf{X} = \{\boldsymbol{x}_t, t = 1, \ldots, T\}$, where $\boldsymbol{x}_t \in \mathbb{R}^d$ denotes the mortality rates in year t at each age, the feedforward steps in LSTM can be expressed as follows:

$$\begin{aligned}
\boldsymbol{f}_t &= \sigma(\mathbf{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f), \\
\boldsymbol{i}_t &= \sigma(\mathbf{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i), \\
\boldsymbol{o}_t &= \sigma(\mathbf{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o), \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tanh(\mathbf{W}_c \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_c), \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t),
\end{aligned}$$

where $\boldsymbol{h}_t \in \mathbb{R}^h$ denotes the hidden state with h hidden units, $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_c \in \mathbb{R}^{h \times (h+d)}$ and $\boldsymbol{b}_f, \boldsymbol{b}_i, \boldsymbol{b}_o, \boldsymbol{b}_c \in \mathbb{R}^h$ are weight matrices and bias vectors for the corresponding connection, $\sigma(\cdot)$ and $\tanh(\cdot)$ represent the sigmoid function and the tanh function, $[\cdot, \cdot]$ denotes the concat operation which merges the two vectors together, and $\odot$ denotes the Hadamard product of two vectors.

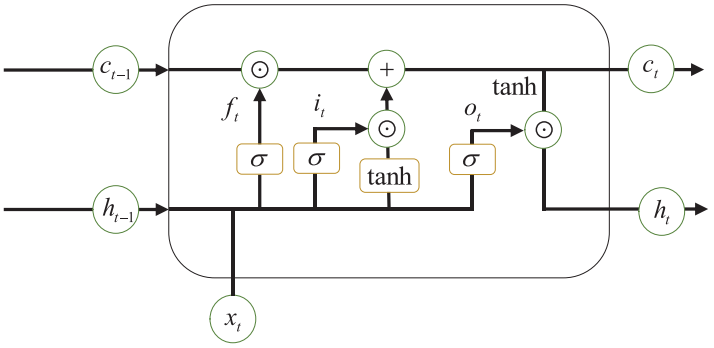


Figure 2. The structure of Long Short-Term Memory network.

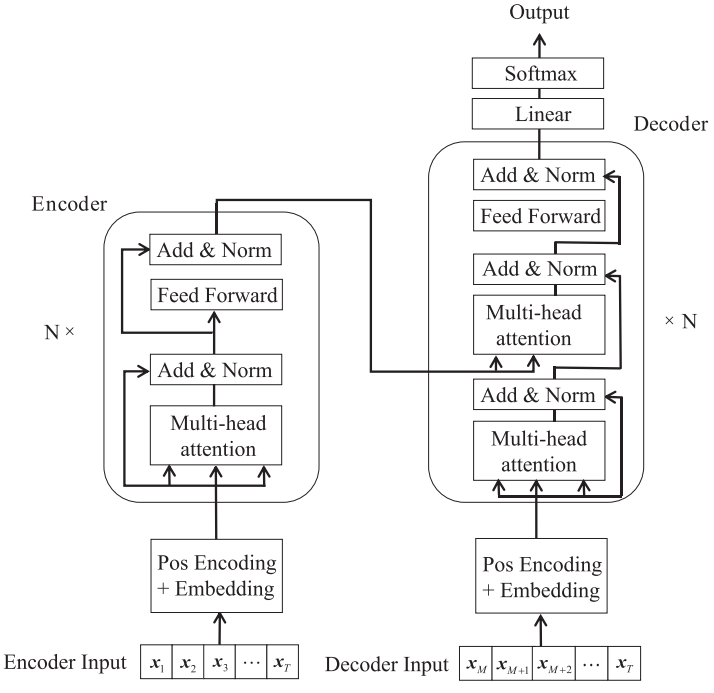


Figure 3. The architecture of Transformer for mortality prediction.

3. Transformer

In 2017, Vaswani et al. (2017) proposed a seminal deep learning architecture called Transformer for NLP problems. Transformer uses the self-attention mechanism instead of traditional CNN and RNN architectures to extract key features from contexts. Just as they said, 'attention is all you need.' The success of Transformer in machine translation also piqued the interest of applications on time series forecasting. For example, Zhou et al. (2021) proposed an improved Transformer model called Informer for long-sequence time series forecasting. Wang et al. (2022) applied Transformer to predict the stock market index. In this paper, we consider using Transformer to predict mortality rates. Figure 3 presents the architecture of Transformer briefly.

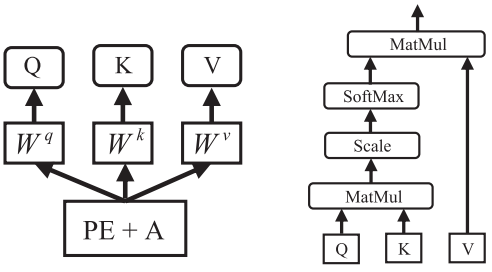


Figure 4. The structure of self-attention.

3.1. Embeddings and positional encoding

At the beginning, the mortality data $\mathbf{X}$ are fed into the embedding layer and positional encoding layer to extract spatial and temporal features. Embedding is a commonly used technique in NLP problems, which maps the sparse and high-dimensional word vectors into a low-dimensional space (Mikolov et al. 2013). Through a fully connected network, the embedding layer extracts spatial information on the age structure from the input and outputs a matrix $A \in \mathbb{R}^{T\times d_m}$, where d_m is the dimension of the model. Besides, the positional encoding layer characterizes the sequential information of time series by using the sine and cosine functions of different frequencies:

$$\mathrm{PE}_{t,2s} = \sin(t/10,000^{2s/d_m}),$$
$$\mathrm{PE}_{t,2s+1} = \cos(t/10,000^{2s/d_m}),$$

where $1 \le 2s \le d_m$. Then, the embedded input and positional encoding are summed together and fed into the encoder module.

3.2. Encoder-decoder

Transformer takes the encoder–decoder architecture, which was first proposed in the variational autoencoder (VAE) method (Kingma & Welling 2013). The encoder expresses the key information into a fixed-length vector, and then the decoder converts it into an output. The encoder-decoder architecture has been shown to be effective for modeling sequential data (Bahdanau et al. 2014).

Specifically, both the encoder and decoder modules are composed of a stack of N layers with identical structures: the multi-head self-attention layers and a feed-forward network, which is fully connected. The residual connection and normalization operators are added in each sub-layer to improve the performance (He et al. 2016). Different from the original decoder in Vaswani et al. (2017), this paper omits the mask attention mechanism by referring to the idea in Wang et al. (2022).

3.3. Multi-head attention

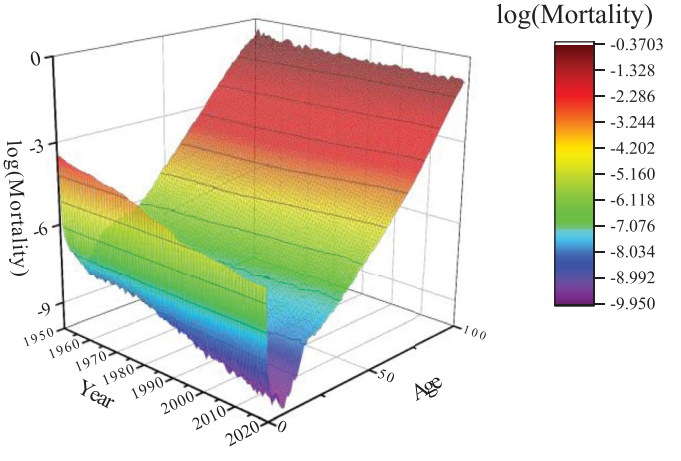
The attention mechanism is the core of Transformer and may be the most exciting innovation in deep learning in the past few years (Mnih et al. 2014). The attention mechanism is inspired by the mechanism of human vision that our eyes often focus limited attention on the important local areas rather than the whole scope. Thus, it can save computational resources and capture the essential information quickly.

The self-attention mechanism in Vaswani et al. (2017) is defined as follows:

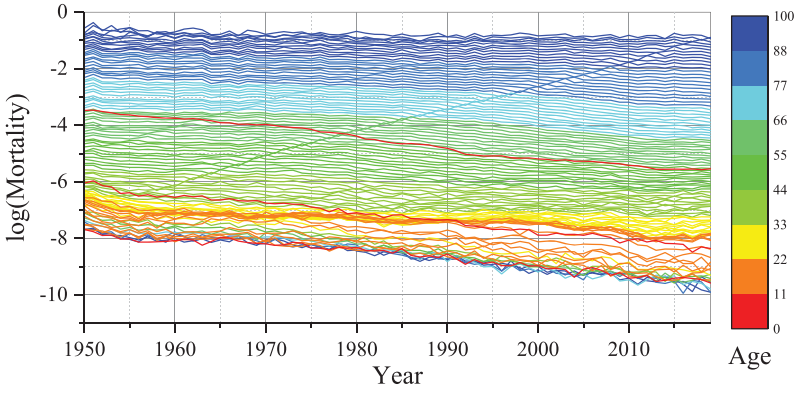
$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_m}}\right)V, \tag{2}$$

Table 1. Optimal hyperparamerters in our transformer model.

Hyperparameter	Optimal Value	Hyperparameter	Optimal Value
Encoder/Decoder Layer	1	Learning Rate	0.001
Attention Head	2	Batch Size	32
Embedding Dimension	32	Drop Rate	0.1
Hidden Size	16		



(a) 3D-distribution of mortality rates in the UK.



(b) Log-mortality rates over time for each age.

Figure 5. Mortality rates in the UK: $\mathbf{X} = \{x_{t,a} : 1950 \leq t \leq 2019, 0 \leq a \leq 100\}$. (a) 3D-distribution of mortality rates in the UK. and (b) Log-mortality rates over time for each age.

where $Q \in \mathbb{R}^{T \times d_m}$, $K \in \mathbb{R}^{T \times d_m}$ and $V \in \mathbb{R}^{T \times d_m}$ are query, key and value matrices respectively, which are outputs of three different linear layers with the same input. The dot product of Q and K^T measures the similarity between query and key. Then, the attention is calculated by the weighted average of the corresponding value V. Figure 4 presents the structure of the self-attention mechanism.

In practice, Transformer concatenates multiple self-attention together, called multi-head attention, to improve the performance further. Each attention function is executed in parallel with the respective projected versions of the query, key, and value matrices. The multi-head attention can be expressed

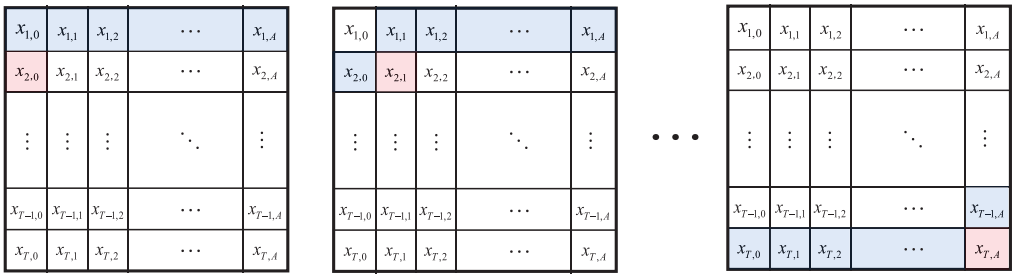


Figure 6. The data generation process of inputs and targets.

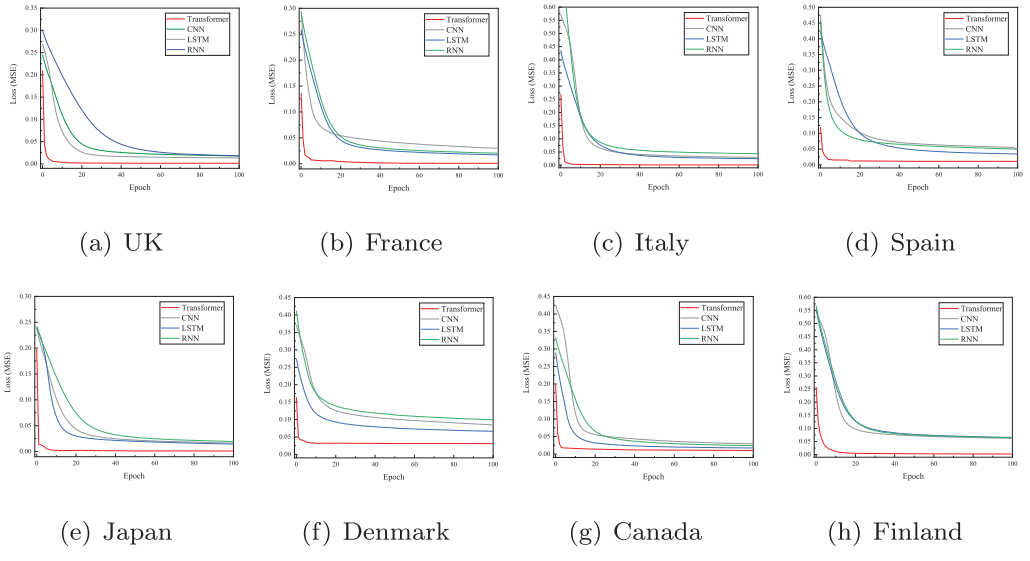


Figure 7. The MSE loss of different deep learning methods within 100 epochs. (a) UK. (b) France. (c) Italy. (d) Spain. (e) Japan. (f) Denmark. (g) Canada and (h) Finland.

as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O,$$
$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

where $i = 1, \ldots, h$ and W_i^Q, W_i^K, W_i^V are weights of corresponding networks.

4. Experiments

4.1. Data description

In this section, we evaluate the performance of Transformer on mortality prediction. Here, we consider mortality datasets for eight countries, namely the United Kingdom (UK), France, Italy, Spain, Japan, Denmark, Canada, and Finland, as referred to in Nigri et al. (2019), Perla et al. (2021), Schnürch & Korn (2022) and Shi (2021). The data covers the period from 1950 to 2019, which is determined by the availability of data in the HMD. This paper ignores the mortality data of centenarians due to the limited observations. Figure 5 presents the distribution of mortality rates in the UK as an example.

Table 2. The means and variances of three indicators on the training sets for eight countries.

Country	Models	MAE	RMSE	MAPE
UK	RNN	0.1288(3.66)	0.1992(5.26)	2.55%(2.83)
	LSTM	0.1110(2.15)	0.1552(3.84)	2.53%(1.42)
	CNN	0.0988(3.65)	0.1375(5.33)	2.44%(2.34)
	Transformer	**0.0600**(**0.95**)	**0.0854**(**1.04**)	**1.22**%(**0.22**)
France	RNN	0.1202(3.01)	0.1598(6.48)	2.28%(1.85)
	LSTM	0.1089(1.11)	0.1178(1.31)	3.12%(0.95)
	CNN	0.1399(3.02)	0.1772(4.11)	2.67%(1.54)
	Transformer	**0.0339**(**0.62**)	**0.0671**(**0.74**)	**0.98**%(**0.20**)
Italy	RNN	0.1427(4.19)	0.2202(4.02)	2.87%(2.54)
	LSTM	0.1178(2.29)	0.1458(2.33)	2.39%(1.93)
	CNN	0.1320(3.13)	0.1871(3.46)	3.21%(2.35)
	Transformer	**0.0749**(**0.79**)	**0.0974**(**0.82**)	**1.28**%(**0.31**)
Spain	RNN	0.1358(3.23)	0.2045(4.02)	2.59%(2.42)
	LSTM	0.1070(2.10)	0.1561(2.00)	2.21%(1.86)
	CNN	0.1592(2.22)	0.2119(2.75)	3.22%(2.99)
	Transformer	**0.0882**(**1.19**)	**0.1398**(**1.98**)	**1.54**%(**0.54**)
Japan	RNN	0.1329(3.24)	0.1682(4.21)	2.08%(2.98)
	LSTM	0.0942(1.68)	0.1119(1.90)	1.99%(0.82)
	CNN	0.1100(3.32)	0.1301(5.32)	2.30%(3.64)
	Transformer	**0.0641**(**0.15**)	**0.0860**(**0.09**)	**1.26**%(**0.04**)
Denmark	RNN	0.1897(4.11)	0.3132(3.75)	3.54%(2.65)
	LSTM	0.1898(1.53)	0.2927(1.75)	3.01%(2.03)
	CNN	0.1622(1.07)	0.2397(1.64)	3.23%(1.53)
	Transformer	**0.1243**(**0.77**)	**0.1934**(**0.85**)	**2.24**%(**0.33**)
Canada	RNN	0.0798(3.53)	0.1211(3.75)	2.17%(2.53)
	LSTM	0.0977(1.44)	0.1314(1.86)	1.98%(1.07)
	CNN	0.1119(2.45)	0.1622(3.67)	2.44%(3.44)
	Transformer	**0.0601**(**0.91**)	**0.0838**(**0.76**)	**1.34**%(**1.02**)
Finland	RNN	0.1692(7.24)	0.2652(7.42)	3.51%(5.96)
	LSTM	0.1618(3.82)	0.0273(2.64)	2.97%(4.10)
	CNN	0.1733(3.64)	0.2372(3.66)	4.11%(2.64)
	Transformer	**0.0622**(**0.41**)	**0.0982**(**0.67**)	**1.97**%(**0.13**)

Notes: The variances of MSE, RMSE, and MAPE are on the order of 10^{-5}.

4.2. Hyperparameter setting

Prior to training our model, we carefully consider the selection of hyperparameters. These can be divided into two categories: some are unique to our Transformer models, such as the number of attention heads, encoder and decoder layers, while others are common to deep learning models, such as learning rate and batch size.

Figure 3 presents the structure of the Transformer model. Given the limited sample size of mortality rates in HMD, we aim to avoid overfitting by constraining the complexity of the model. To achieve a simple structure, we set the number of encoder and decoder layers to $N = 1$, and the number of attention heads in the multi-attention heads module to 2. Additionally, we set the embedding dimension to 32 and the hidden size in the feedforward network to 16 by referring to existing Transformer models in the literature (Vaswani et al. 2017, Wang et al. 2022).

For common hyperparameters in deep learning models, we explore a range of values for each parameter. Specifically, we consider learning rates ranging from 0.001 to 0.1, and batch sizes of 16, 32, 64, or 128. Besides, we make use of the dropout technique to prevent overfitting (Hinton et al. 2012) and test drop rates within the interval of 0.1 to 0.5. The optimal hyperparameters are selected based on their performance on the training set, with preference given to the set that results in the lowest loss function. Table 1 displays the optimal hyperparameters selected for our Transformer model.

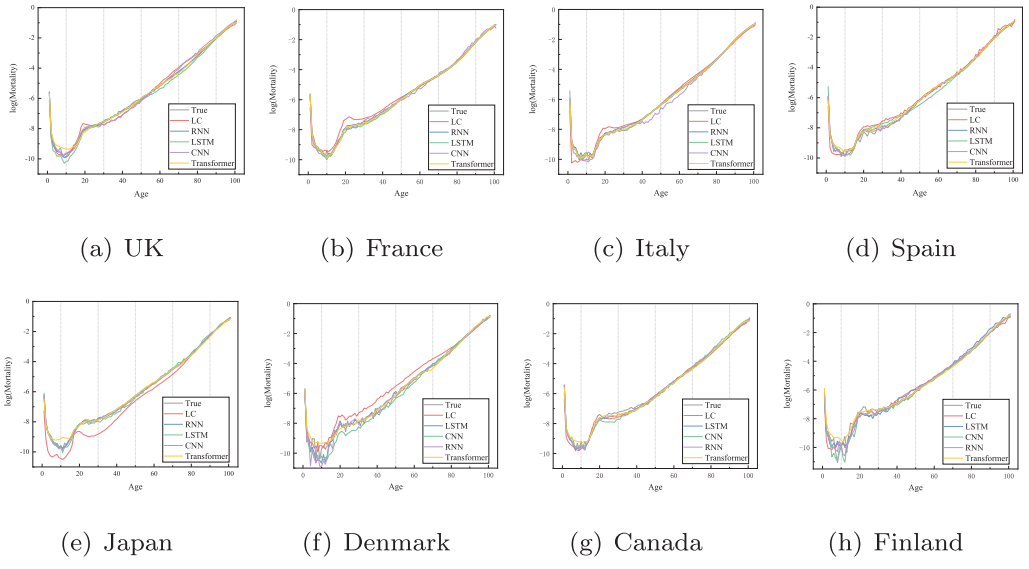


Figure 8. The predition of log-mortality for eight countries in 2019. (a) UK. (b) France. (c) Italy. (d) Spain. (e) Japan. (f) Denmark. (g) Canada and (h) Finland.

4.3. Training

By referring to Shi (2021), we split the dataset into the training and testing sets. The training set ranges from 1950 to 2000 and is used to train the prediction model. The testing set ranges from 2001 to 2019 and is used to evaluate the performance of forecasting.

Let $\mathbf{X}_{\text{train}} = (x_{t,a})_{T\times A}$ be the training set, where $T = 51$ and $A = 101$. Here $x_{t,a}$ is the log-mortality rate at age a in the calendar year t. We consider a moving window technique to construct the inputs and targets. Figure 6 presents the data generation process of inputs (marked with blue) and targets (marked with red). We use the sequence that contains information of all ages to predict the mortality at the next state.

The loss function calculates the mean squared error (MSE) between the predicted and true values to evaluate the fitness of models. Figure 7 presents the variation of MSE within 100 epochs. In all eight countries, the MSE of the Transformer model exhibits significant downward trends and converges faster than other methods. This can be attributed to its self-attention mechanism (Vaswani et al. 2017), which can be trained in parallel and makes it easier to obtain global information. In contrast, traditional sequence-based models such as LSTMs and CNNs require sequential processing of each time step, resulting in slower convergence and less effective modeling of long-term dependencies.

To account for the uncertainty in the training process of deep learning models, we performed 10 independent runs of learning and validation for each dataset. Three common indicators, i.e. root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), are used to evaluate the prediction performance of each model:

$$\text{MAE} = \frac{1}{T(A+1)} \sum_{t=1}^{T} \sum_{a=0}^{A} |\hat{y}_{t,a} - y_{t,a}|,$$

$$\text{MAPE} = \frac{1}{T(A+1)} \sum_{t=1}^{T} \sum_{a=0}^{A} \left|\frac{\hat{y}_{t,a} - y_{t,a}}{y_{t,a}}\right|,$$

Table 3. The means and variances of three indicators on the testing sets for eight countries.

Country	Models	MAE	RMSE	MAPE
UK	LC	0.1312	0.1627	3.01%
	RNN	0.1293(4.01)	0.2035(6.77)	2.71%(2.94)
	LSTM	0.1126(3.44)	0.1587(4.22)	2.36%(1.12)
	CNN	0.1080(5.98)	0.1390(6.48)	2.87%(2.64)
	Transformer	**0.0716**(**0.89**)	**0.1029**(**1.09**)	**1.63**%(**0.23**)
France	LC	0.1508	0.2153	2.65%
	RNN	0.1209(3.96)	0.1605(5.77)	2.38%(2.03)
	LSTM	0.0997(1.60)	0.1290(1.19)	1.95%(0.94)
	CNN	0.1406(6.36)	0.1860(7.65)	2.70%(1.11)
	Transformer	**0.0684**(**0.82**)	**0.0941**(**0.80**)	**1.38**%(**0.27**)
Italy	LC	0.1422	0.2132	2.66%
	RNN	0.1453(6.28)	0.2212(4.52)	2.51%(2.32)
	LSTM	0.1149(2.41)	0.1618(1.10)	2.12%(2.01)
	CNN	0.1509(2.79)	0.1996(3.67)	2.71%(2.19)
	Transformer	**0.8585**(**0.83**)	**0.1285**(**0.78**)	**1.58**%(**0.28**)
Spain	LC	0.1547	0.2174	3.17%
	RNN	0.1362(4.21)	0.2051(4.23)	2.40%(2.32)
	LSTM	0.1125(1.92)	0.1657(2.05)	2.01%(1.98)
	CNN	0.1781(2.02)	0.2594(2.58)	3.19%(3.10)
	Transformer	**0.1074**(**1.52**)	**0.1575**(**2.15**)	**1.96**%(**0.52**)
Japan	LC	0.3060	0.3968	5.06%
	RNN	0.1331(3.98)	0.1692(5.00)	2.10%(2.88)
	LSTM	0.0950(1.55)	0.1238(2.06)	2.19%(0.67)
	CNN	0.1124(4.90)	0.1399(5.64)	2.43%(3.73)
	Transformer	**0.0667**(**0.19**)	**0.1101**(**0.06**)	**1.31**%(**0.06**)
Denmark	LC	0.3074	0.4086	5.29%
	RNN	0.2082(3.90)	0.3362(3.56)	3.65%(2.75)
	LSTM	0.1902(1.27)	0.2934(2.03)	3.26%(1.52)
	CNN	0.1883(1.23)	0.2755(1.86)	3.37%(1.75)
	Transformer	**0.1410**(**0.45**)	**0.2502**(**0.83**)	**2.44**%(**0.43**)
Canada	LC	0.1101	0.1587	2.36%
	RNN	0.0974(3.64)	0.1351(3.64)	2.11%(2.53)
	LSTM	0.0982(1.46)	0.1412(1.75)	1.98%(1.04)
	CNN	0.1216(2.86)	0.1638(3.54)	2.48%(3.75)
	Transformer	**0.0815**(**0.92**)	**0.1086**(**0.64**)	**1.64**%(**1.32**)
Finland	LC	0.1745	0.2684	3.63%
	RNN	0.1899(10.24)	0.2868(8.23)	3.53%(6.79)
	LSTM	0.1691(3.92)	0.2889(3.79)	2.91%(6.09)
	CNN	0.1843(3.54)	0.2584(3.13)	4.09%(2.53)
	Transformer	**0.0853**(**0.24**)	**0.1229**(**0.13**)	**2.57**%(**0.10**)

Notes: The variances of MSE, RMSE, and MAPE are on the order of 10^{-5}.

$$\text{RMSE} = \sqrt{\frac{1}{T(A+1)}\sum_{t=1}^{T}\sum_{a=0}^{A}(\hat{y}_{t,a} - y_{t,a})^2},$$

where $\hat{y}_i$ is the predicted value of log-mortality, and y_i is the corresponding true value. Here RMSE and MAE measure the absolute errors between the predicted and true values, while MAPE is a relative indicator to measure the percentage of errors. Table 2 summarizes the means and variances of these indicators on the training sets for the eight countries. It demonstrates that our Transformer model has lower prediction errors and more robust outputs compared to other deep learning methods.

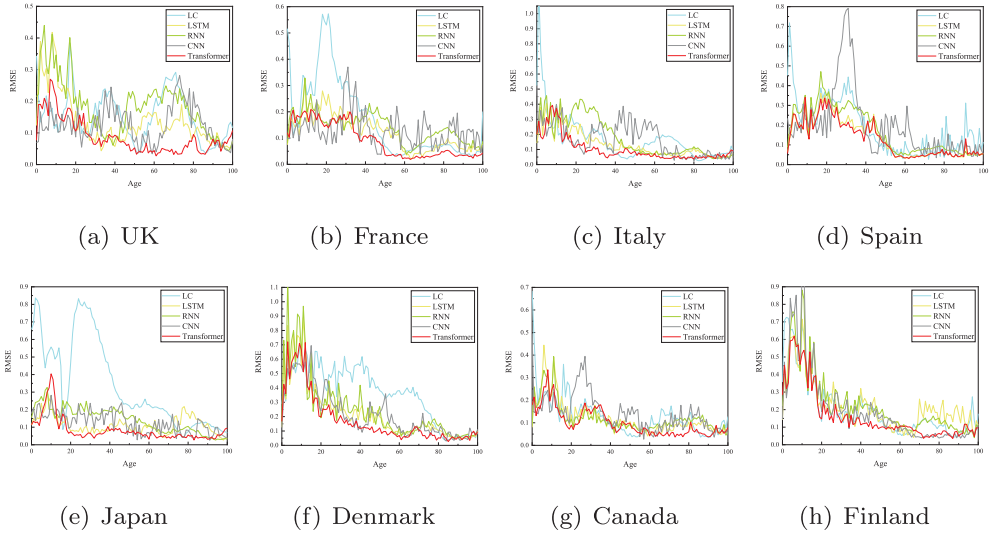


Figure 9. RMSE(*a*) calculated on age for predicted data. (a) UK. (b) France. (c) Italy. (d) Spain. (e) Japan. (f) Denmark. (g) Canada and (h) Finland.

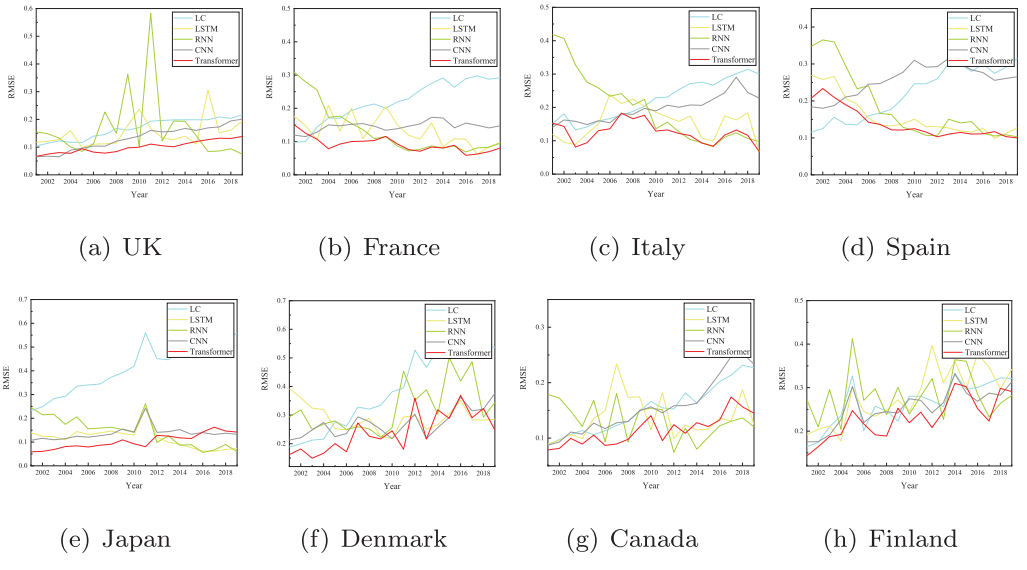


Figure 10. RMSE(*t*) calculated on year for predicted data. (a) UK. (b) France. (c) Italy. (d) Spain. (e) Japan. (f) Denmark. (g) Canada and (h) Finland.

4.4. Forecasting and evaluation

Based on the trained models, we predict mortality rates from 2001 to 2019 in eight countries. As an example, Figure 8 presents the prediction of log-mortality in 2019. They demonstrate that our Transformer model provides smoother predictions that are closer to the true values compared with traditional methods.

Similarly, taking into account the uncertainty of deep learning models, Table 3 summarizes the means and variances of RMSE, MAE, and MAPE on the testing sets based on 10 independent runs of model training. We include the classic LC model in the scope of model comparison in the testing sets.

Table 4. The average of RMSE(a) and RMSE(t) for each model.

Country	Indicators	LC	RNN	LSTM	CNN	Transformer
UK	RMSE(a)	0.1537	0.1834	0.1369	0.1264	**0.0870**
	RMSE(t)	0.1669	0.1509	0.1325	0.1656	**0.1006**
France	RMSE(a)	0.1736	0.1406	0.1218	0.1314	**0.0911**
	RMSE(t)	0.2206	0.1326	0.1454	0.1330	**0.0915**
Italy	RMSE(a)	0.1690	0.1789	0.1348	0.1666	**0.1025**
	RMSE(t)	0.2256	0.1546	0.1960	0.1963	**0.1247**
Spain	RMSE(a)	0.2210	0.1702	0.1347	0.1969	**0.1296**
	RMSE(t)	0.2208	0.1571	0.2560	0.1839	**0.1507**
Japan	RMSE(a)	0.3395	0.1412	0.1123	0.1258	**0.0833**
	RMSE(t)	0.4083	0.1167	0.1371	0.1455	**0.1038**
Denmark	RMSE(a)	0.3609	0.2515	0.2232	0.2212	**0.1792**
	RMSE(t)	0.3862	0.2901	0.2901	0.2715	**0.2413**
Canada	RMSE(a)	0.1291	0.1120	0.1197	0.1419	**0.1002**
	RMSE(t)	0.1526	0.1313	0.1362	0.1563	**0.1143**
Finland	RMSE(a)	0.2069	0.2273	0.2173	0.2001	**0.1714**
	RMSE(t)	0.2637	0.2819	0.2548	0.2821	**0.2303**

Since the parameter estimation of the LC model is deterministic, it produces unique predictions in all 10 independent runs, and therefore there is no variance. Table 3 demonstrates that our Transformer model also outperforms the classic LC model and other deep learning models in the testing sets. The fact that the proposed model performs well on both the training and testing sets indicates that there is no overfitting problem during model training.

Besides, we evaluate the prediction performance from the perspectives of age and year, respectively. By referring to Li & Lu (2017), we define the RMSE at age a, RMSE(a), and the RMSE in year t, RMSE(t), as follows:

$$\text{RMSE}(a) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{y}_{t,a} - y_{t,a})^2},$$

$$\text{RMSE}(t) = \sqrt{\frac{1}{A+1}\sum_{a=0}^{A}(\hat{y}_{t,a} - y_{t,a})^2}.$$

Figures 9 & 10 present the RMSE at each a and each t, respectively. Figure 9 illustrates that the Transformer model exhibited lower prediction errors at most ages, especially in the high age group ($a \geq 40$). Meanwhile, Figure 10 demonstrates that the Transformer model has a significant advantage in short-term predictions. However, in long-term predictions, it was outperformed by the RNN in a few countries, including the UK, Japan, and Canada. Overall, Figures 9 and 10 indicate that our Transformer model performed well across most ages and years. Moreover, Table 4 reveals that the Transformer model has the lowest average RMSE(a) and RMSE(t) values among all the models, indicating clear advantages in terms of prediction accuracy.

5. Conclusions

This paper evaluates the performance of Transformer in predicting mortality rates. It is a new application of Transformer in time series forecasting. Compared to the classical LC model and other traditional deep learning models, Transformer exhibits a strong ability to capture the underlying spatial and temporal features in the age-year structure of mortality rates. Through empirical experiments

across eight countries, we demonstrate that Transformer has higher prediction accuracy than traditional methods, especially in the high age group. This is particularly relevant for policymakers, given the growing concern over the aging population and its impact on public health. Accurately predicting mortality rates in this demographic is essential for developing effective health policies and allocation of resources. This paper provides a powerful forecasting tool for insurance companies and policy makers.

The COVID-19 has significantly altered the world in many ways over the last few years. It has a far-reaching influence on the trend of mortality rates. The last similar event may have been the 1918 Spain Flu. During this global pandemic, we found that, in addition to the disease itself, the public health policies and disease control measures implemented by governments also had significant impacts on population mortality rates. How to incorporate these unstructured information into mortality rate prediction remains a challenge. In future work, we plan to investigate the use of deep learning frameworks to integrate both numeric data and unstructured information, in order to make more precise and accurate predictions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was funded by the National Natural Science Foundation of China [12271217, 72174076, 11971202], the Outstanding Young Foundation of Jiangsu Province [BK20200042] and a scientific grant by Jiangsu Commission of Health [ZD2021009].

References

Bahdanau, D., Cho, K. & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Booth, H., Maindonald, J. & Smith, L. (2002). Applying Lee-Carter under conditions of variable mortality decline. *Population Studies* 56(3), 325–336.

Brouhns, N., Denuit, M. & Vermunt, J. K. (2002). A Poisson log-bilinear regression approach to the construction of projected lifetables. *Insurance: Mathematics and Economics* 31(3), 373–393.

Cairns, A. J., Blake, D. & Dowd, K. (2006). A two-factor model for stochastic mortality with parameter uncertainty: theory and calibration. *Journal of Risk and Insurance* 73(4), 687–718.

Cairns, A. J., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A. & Balevich, I. (2009). A quantitative comparison of stochastic mortality models using data from England and Wales and the United States. *North American Actuarial Journal* 13(1), 1–35.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Egmont-Petersen, M., de Ridder, D. & Handels, H. (2002). Image processing with neural networks–a review. *Pattern Recognition* 35(10), 2279–2301.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. P. 770–778. The Conference (CVPR 2016) was held in Las Vegas, USA

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation* 9(8), 1735–1780.

Huang, Y., Shen, L. & Liu, H. (2019). Grey relational analysis, principal component analysis and forecasting of carbon emissions based on long short-term memory in China. *Journal of Cleaner Production* 209, 415–423.

Hüsken, M. & Stagge, P. (2003). Recurrent neural networks for time series classification. *Neurocomputing* 50, 223–235.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A. & Bridgland, A. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature* 596(7873), 583–589.

Kingma, D. P. & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Klenk, J., Keil, U., Jaensch, A., Christiansen, M. C. & Nagel, G. (2016). Changes in life expectancy 1950–2010: contributions from age-and disease-specific mortality in selected countries. *Population Health Metrics* 14(1), 1–11.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60(6), 84–90.
Lee, R. D. & Carter, L. R. (1992). Modeling and forecasting US mortality. *Journal of the American Statistical Association* 87(419), 659–671.
Li, H. & Lu, Y. (2017). Coherent forecasting of mortality rates: A sparse vector-autoregression approach. *ASTIN Bulletin: The Journal of the IAA* 47(2), 563–600.
Li, J. & Wong, K. (2020). Incorporating structural changes in mortality improvements for mortality forecasting. *Scandinavian Actuarial Journal* 2020(9), 776–791.
Lim, B. & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A* 379(2194), 20200209.
Lindholm, M. & Palmborg, L. (2022). Efficient use of data for LSTM mortality forecasting. *European Actuarial Journal* 12, 749–778.
Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
Mnih, V., Heess, N., Graves, A. & Kavukcuoglu, K. (2014). Recurrent models of visual attention. *Advances in Neural Information Processing Systems* 27. 2204 to 2212.
Nigri, A., Levantesi, S., Marino, M., Scognamiglio, S. & Perla, F. (2019). A deep learning integrated Lee–Carter model. *Risks* 7(1), 33.
Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A. & Tran, D. (2018). Image transformer. In International Conference on Machine Learning. P. 4055–4064.The Conference (ICML 2018) was held in Stockholm, Sweden
Perla, F., Richman, R., Scognamiglio, S. & Wüthrich, M. V. (2021). Time-series forecasting of mortality rates using deep learning. *Scandinavian Actuarial Journal* 2021(7), 572–598.
Perla, F. & Scognamiglio, S. (2023). Locally-coherent multi-population mortality modelling via neural networks. *Decisions in Economics and Finance* **46**, 157–176.
Pitacco, E., Denuit, M., Haberman, S. & Olivieri, A. (2009). *Modelling longevity dynamics for pensions and annuity business*. Oxford University Press.
Richman, R. & Wüthrich, M. V. (2021). A neural network extension of the Lee–Carter model to multiple populations. *Annals of Actuarial Science* 15(2), 346–366.
Schnürch, S. & Korn, R. (2022). Point and interval forecasts of death rates using neural networks. *ASTIN Bulletin: The Journal of the IAA* 52(1), 333–360.
Scognamiglio, S. (2022). Calibrating the Lee-Carter and the Poisson Lee-Carter models via Neural networks. *ASTIN Bulletin: The Journal of the IAA* 52(2), 519–561.
Shi, Y. (2021). Forecasting mortality rates with the adaptive spatial temporal autoregressive model. *Journal of Forecasting* 40(3), 528–546.
Tang, W., Long, G., Liu, L., Zhou, T., Blumenstein, M. & Jiang, J. (2022). Omni-Scale CNNs: a simple and effective kernel size configuration for time series classification. In International Conference on Learning Representations. Chicago, USA.
Tetko, I. V., Karpov, P., Van Deursen, R. & Godin, G. (2020). State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature Communications* 11(1), 1–11.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems* 30, 5998–6008.
Wang, C., Chen, Y., Zhang, S. & Zhang, Q. (2022). Stock market index prediction using deep Transformer model. *Expert Systems with Applications* 208, 118128.
Wang, C.-W., Zhang, J. & Zhu, W. (2021). Neighbouring prediction for mortality. *ASTIN Bulletin: The Journal of the IAA* 51(3), 689–718.
Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J. & Sun, L. (2022). Transformers in time series: A survey. arXiv preprint arXiv:2202.07125.
Xi, E., Bing, S. & Jin, Y. (2017). Capsule network performance on complex data. arXiv preprint arXiv:1712.03480.
Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H. & Zhang, W. (2021). Informer: beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. P. 11106–11115. The Conference (AAAI 2021) was held in Vancouver, Canada.