Hybrid Lee-Carter Model: An Analysis of Mortality Rate Prediction Using ARIMA and Deep Learning Techniques

---

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mathematics

Middle Tennessee State University

By

Yuan Chen

July 2023

Thesis Committee:

Dr. Abdul Khaliq

Dr. Donglin Wang

Dr. Vajira Manathunga

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to the individuals who have contributed to the completion of this thesis. Their unwavering support, encouragement, and friendship have made this journey truly memorable.

I am immensely grateful to Dr. Khaliq, for his invaluable guidance. His expertise and constructive feedback have been instrumental in shaping the direction of this research.

I extend my sincere appreciation to my thesis committee, Dr. Abdul Khaliq, Dr. Donglin Wang, and Dr. Vajira Manathunga, for their valuable insights and recommendations. Their expertise has greatly enriched the quality of this work.

I would also like to thank my dear friends, Rihao Ruan, Tian Yuan and Wilson Musyoka for their unwavering support throughout this endeavor. Specially, I would like to thank Jinming Gao for her encouragement throughout the entire journey. Their friendship, encouragement, and countless conversations have provided me with the motivation and inspiration to overcome challenges and persevere.

Additionally, I am deeply grateful to my family for their unconditional love and belief in my abilities. Their constant encouragement and sacrifices have been the bedrock of my academic achievements.

In conclusion, I acknowledge the contributions of all those who have played a role in this thesis. Their support has been instrumental in bringing this work to fruition.

ABSTRACT

The Lee-Carter model, is a fundamental approach in mortality forecasting that estimates and projects mortality rates based on historical data. While deep learning techniques have been extensively explored in mortality prediction, most of the studies have focused on Long Short-term Memory within the family of recurrent neural networks. This thesis investigates the application of the four different Deep Learning models, namely Long Short-Term Memory, Gated Recurrent Unit, Bidirectional Long Short-Term Memory and Bidirectional Gated Recurrent Unit in mortality rate prediction, both independently and in hybrid combinations with Lee-carter model. The proposed models are evaluated using yearly mortality data categorized by genders from nine census divisions in the United States. Through the out-of-sample testing, our finding indicates that all the model outperform the traditional Lee-Carter model in terms of two selected error metrics, mean absolute error (MAE) and root mean square error (RMSE). The Gated Recurrent Unit hybrid model produces the most accurate estimations among the hybrid models, while the Bidirectional Gated Recurrent Unit model has the best overall performance.

CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Mortality prediction plays a vital role in the field of actuarial science, demography, public health, and social policy. Among the various stochastic models developed for forecasting mortality rates, the Lee-Carter (LC) model [1] stands out as the most commonly utilized. Over the years, the field of mortality prediction has witnessed significant developments, leading to notable improvements in the Lee-Carter model. One well-known example is Cairns-Blake-Dowd model (CBD model) which is proposed by Cairns et al. [2]. Additionally, the other extension, such as the Poisson extension by Brouhns et al. [3] and a functional data method using penalized regression by Hyndman and Ullah [4] have been applied to the field of mortality rate predicting. Other models for mortality rate forecasting include the studies by Renshaw and Haberman [5], Deprez et al.[6] and Levantesi and Pizzorusso[7].

The development of deep learning techniques has brought significant advancements in the field of mortality rate forecasting. Specially, recurrent neural networks have been proven to be effective tools for recognizing unidentifiable patterns from a large data set with many features. The application of deep learning models in mortality forecasting has shown the accuracy and precision of predictions. For instance, the study by Hainaut [8] showed that neural networks effectively capture valuable information from known mortality data, enabling the replication of nonlinear trends in predictions. A comparison study was conducted by Perla et al.[9] contrasting the Lee-Carter (LC) model with deep learning approaches using data sourced from the Human Mortality Database (HMD) [10].

Several studies have focused on hybridizing the deep learning models or machine learning models with the Lee-Carter (LC) model to improve the mortality rate predic-

tion. For example, Hong et al. [11] proposed the random forest and artificial neural networks hybrid Lee-Carter models, comparing them with the traditional Lee-Carter (LC) model on mortality data across multiple countries. Nigri et al.[12], [13] applied the recurrent networks with Long Short-Term Memory (LSTM) architecture to predict the time index of the Lee–Carter (LC) model and indicated that neural networks forecasting could provide higher accuracy trend compared to the ARIMA models in several countries and for both genders. Marino and Levantesi [14] extended the neural networks approach by Nigri et al.[12] and derived the related predict interval representing the Long Short-Term Memory model's parameter uncertainty. Ronald et al.[15] successfully applied the long-short-term memory (LSTM) and gated recurrent unit (GRU) in the recurrent neural network (RNN) to the Swiss population mortality time series data modeling. Other relevant applications of machine learning and deep learning studies in the mortality field can be found in Castellani et al. [16], Petnehazi and Gall [17],[20] , Richman and Wuthrich [18], Gabrielli and Wuthrich [19] and Chen and Kahliq[21]

In the previous studies, it was observed that a majority of the research on neural network applications in mortality forecasting focused on Long Short-Term Memory (LSTM) models. However, it is worth to investigate the predictive performance by Bidirectional Long Short-Term Memory (Bi-LSTM), Gated Recurrent Unit (GRU) and Bidirectional Gated Recurrent Unit (Bi-GRU) in the same task, these models have been shown to be powerful tools in time series forecasting tasks. In this thesis, the numerical experiments are running on the nine census divisions mortality data in the United States, while most of the studies explore mortality rates of the different countries around the world.

Therefore, the key contributions of this thesis are summarized below:

(i) Several new deep learning hybrid Lee-Carter models are developed to the task of mortality rate forecasting, including Bidirectional Long Short-Term Memory hybrid Lee-Carter model, Gated Recurrent Unit hybrid Lee-Carter model and Bidirectional Gated Recurrent Unit hybrid Lee-Carter model.

(ii) Four deep learning models, namely Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), Gated Recurrent Unit (GRU) and Bidirectional Gated Recurrent Unit (Bi-GRU) are applied to predict the mortality for United States.

(iii) One hybrid model which is Long Short-Term Memory hybrid Lee-Carter model (LC-LSTM) is applied to predict the time index, and utilizing it to estimate future mortality rates in the United States.

(iv) The forecasting results are measured by Mean Average Error (MAE) and Root Mean Square Error (RSME) in an out of sample test.

CHAPTER 2

MODEL INTRODUCTION

2.1 Lee-Carter Model

In this section, we discuss some important concepts regarding the Lee–Carter model by Lee and Carter[1]. The Lee–Carter model is a demographic model that is widely used in mortality prediction and life expectancy forecasting for different countries. The Lee–Carter model implies a linear relationship between the central mortality rate of an age group $m_{x,t}$ and age interval x and year t. Equation 1 describes the model:

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

It can be rewritten as Equation 2:

$$\ln(m_{x,t}) = \alpha_x + \beta_x \kappa_t \tag{2}$$

where α_x represents the specific average log-mortality rate, β_x means the deviation in mortality due to age profile κ_t variations, and κ_t is the time index to the year t. Another is that the Lee–Carter model is subject to the constraints on the parameters, so we have

$$\sum_{x=x_1}^{x_p} \tilde{\beta_x} = 1 \tag{3}$$

$$\sum_{t=t_1}^{t_n} \tilde{\kappa_t} = 0 \tag{4}$$

In practice, Singular Value Decomposition (SVD), Maximum Likelihood estimation (MLE), and Least Square (LS) are the three classical methods to estimate the parameters of the Lee–Carter model. In this paper, we applied the Singular Value Decomposition (SVD) approach to the Lee–Carter model and use the AutoRegressive Integrated Moving Average model (ARIMA) process to estimate the time index κ_t.

The first step calculates the parameter α_x, which is just the average values of raw $ln(m_{x,t})$ (observation) over time, shown in Equation 5.

$$\tilde{\alpha_x} = \frac{1}{t}\sum_{t=t_1}^{t_n} \ln m_{x,t} \tag{5}$$

The estimation of β_x and κ_t is obtained by the singular value deposition of the matrix of $ln(m_{x,t}) - \alpha_x$. Here, we present a quick introduction of the singular value decomposition: first, we denote a matrix $ln(m_{x,t}) - \alpha_x = A$. Supposing that matrix A is a n×m matrix, then A could be computed as:

$$A_{n\times m} = U_{n\times n} D_{n\times p} V'_{p\times p} \tag{6}$$

where the U and V are orthogonal, V' represents the V transposed, and D has the same dimensions as A and has singular value. We calculated the $\tilde{\beta_x}$ according to the following Equations and :

$$\tilde{\beta_x} = \frac{V_{x,1}}{\sum V_{x,1}} \tag{7}$$

$$\tilde{\kappa_t} = U_{t,1} D_{1,1} \sum_{x=x_1}^{x_p} V_{x,1} \tag{8}$$

All the SVD process can be achieved in R with the function SVD. For the forecasting of the time index κ_t, we chose to use the traditional time series model AutoRegressive Integrated Moving Average model (ARIMA) model with drift, which is discussed in the upcoming section.

2.2 ARIMA

After predicting the parameters $\tilde{\alpha_x}$, $\tilde{\beta}_x$, and $\tilde{\kappa_t}$ in the Lee–Carter model with the Singular Value Decomposition (SVD) method, we used the AutoRegressive Integrated Moving Average model (ARIMA) to predict the future $\tilde{\kappa_t}$. The process of finding the best ARIMA model to univariate time series was achieved on the R vision 4.2.1 with the auto ARIMA in the forecast package. This technique is based on the Hyndman–Khandakar algorithm by Hyndman and Khandakar [22]. The idea is using a unit root test to test the stationarity of the time series and choose the degree of differencing d, and then select the best degree of auto-regressive p and moving average order q by the 2 criterion Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC)

2.3 Long Short-term Memory

The Long Short-Term Memory (LSTM) model is an enhanced version of recurrent neural networks (RNN) that addresses the issue of capturing long-term dependencies in sequential data. Traditional RNNs suffer from the vanishing or exploding gradient problem, which hinders their ability to retain information over long sequences. As a result, Hochreiter and Schmidhuber introduced LSTM as a solution to this problem [23].

LSTM incorporates special units, which allow the network to selectively store and discard useless information over time. These units have gating mechanisms that regulate the flow of information, making LSTM well-suited for various tasks, including classification, regression, and particularly time-series analysis.In what follows, we investigated some mathematical functions in LSTM.

The sigmoid (σ), the hyperbolic tangent ($tanh$) and rectified linear unit ($ReLU$) are the most frequently nonlinear activation functions in neural networks, as shown in the following equations.

$$Sigmoid : \sigma(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

$$Hyperbolictangent : \tanh(x) = \frac{e^{x} - e^{x}}{e^{x} + e^{x}} \tag{10}$$

$$ReLU : f(x) = x^{+} = \frac{x + |x|}{2} \tag{11}$$

The sigmoid (σ) values range between 0 and 1, and it is used to decide if the information received should be uploaded and retained or be discarded and forgotten. Because any number that multiplies with 0 will be equal to 0, values may disappear or be considered as "forgotten". Any number that multiplies with 1 will remain the same, so it is "kept" or retained.

The activation function tangent hyperbolic (tanh) ranges between 1 and 1, and it is used to regulate the values flowing through the neural network.

The rectified linear unit (ReLU) outputs the input value if it is positive, and zero otherwise.

A common LSTM unit has a 3-gate mechanism as will be discussed in the following section. Forget gate

The forget gate controls the degree of information loss from the previous cell state; in another words, it decides what information is dropped or kept. The previous information passes through the sigmoid function, and the numbers coming out of the forget gate are between 0 and 1. Values closer to 0 are more likely to be forgotten, and values closer to 1 are more likely to be kept. We denoted the forget gate as f_t. Weight matrices for the forget gate are identified as W_f and U_f, and the hidden units as ht. The input variable $x = x_1, x-2, \ldots, x_T$ is a time series sequence at time $t = 1, 2, \ldots, T$, from which we could obtain Equation :

$$f_t = \sigma(W_f x_t + U_f h_{t-1}) \tag{12}$$

Input gate

The input gate controls the degree of new information to store in the current cell. It uses a sigmoid layer to decide what information would be updated in the current cell state and then uses a tanh layer to create new vector for the cell state. We denoted the input gate as i_t, the weight matrices for input gate as W_i and U_i, and the hidden units as h_t. Thus, we have Equation:

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) \tag{13}$$

Cell State

After the previous works, the old cell state is updated with the information collected from the gates. This step is achieved by multiplying the old cell state with

the weight matrix generated by the forget gate and filter the original information to decide the kept and dropped parts. Then, the results are multiplied in the input gate to obtain the new information, which is added to the cell state. We used C_t to represent the current cell state, $\tilde{C}_t$ represents the candidate cell state, W_c and U_c represent the weight matrices for cell statement, and the hidden units are h_t. The symbol $\odot$ is the Hadamard product. Here, we also provide a quick introduction of Hadamard product; for two matrices A and B with the same dimension, the Hadamard product is:

$$(A \odot B)_{ij} = A_{ij} B_{ij} \tag{14}$$

In the example of 3×3 matrix A and B, we have

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11} \cdot b_{11} & a_{12} \cdot b_{12} & a_{13} \cdot b_{13} \\ a_{21} \cdot b_{21} & a_{22} \cdot b_{22} & a_{23} \cdot b_{23} \\ a_{31} \cdot b_{31} & a_{32} \cdot b_{32} & a_{33} \cdot b_{33} \end{bmatrix}$$

Hence, we have Equations and for cell state:

$$\tilde{C}_t = \tanh(U_c x_t + W_c h_{t-1}) \tag{15}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{16}$$

Output gate

The output gate calculates the new output value of the current cell. The sigmoid layer is used again to generate the weight matrix and we used this matrix to decide what would be the output of the cell state. Then, the weight matrix is multiplied with the input results to output the part of the cell state. We denoted the output gate as o_t, and the weight matrices for output gate as W_o and U_o, and we generated the following Equations:

$$o_t = \sigma(U_o x_t + W_o h_{t-1}) \tag{17}$$

$$h_t = o_t \odot \tanh(C_t) \tag{18}$$

The U_s and the W_s are the weight matrices that help in applying the model to different length of input data. A key point of the weight matrices is that they do not change over time. The same weight matrices are used in every time steps. The weight matrices W_f, W_i, W_o, W_c have the same dimension (input dimension$\times$output dimension) and the weight matrices U_f, U_i, U_o, U_c have the same dimension (output dimension$\times$output dimension). These matrices are learned using a variant of the gradient descent algorithm. Moreover, we can calculate the network parameters for a single layer with the following equation:

$$parameters = 4\times \quad output \quad dimension\times(output \quad dimension+ \quad input \quad dimension+1) \tag{19}$$

An single unite LSTM structure is shown in the following figure:

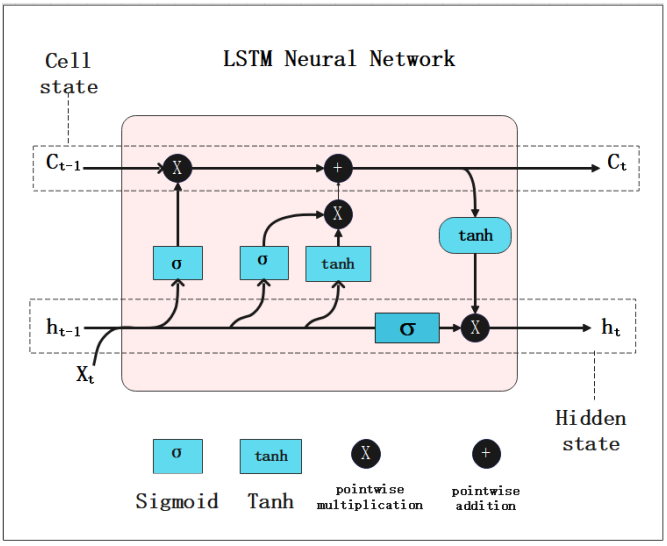

Figure 1: LSTM unit structure.

2.4 Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is a recurrent neural network architecture that was introduced by Kyunghyun Cho et al. [24]. It shares similarities with the LSTM model but has a simpler structure with fewer parameters, gates, and equations.

In comparing the LSTM and GRU models, it can be challenging to determine which one will be the best fit for a given task. Several studies, such as Chung et al. [25], have conducted the comparisons between LSTM and GRU by evaluating their performance in various applications.

The GRU model combines the forget gate and input gate of LSTM into a single update gate. This simplification helps to reduce the complexity of the model without the lost of effectively capture and model temporal dependencies.

The functioning of a Gated Recurrent Unit is governed by a set of equations, which define how information is updated, passed through the network, and outputted. These equations enable the GRU model to learn and adapt to sequential data, making it a valuable tool in tasks such as time-series forecasting, including mortality rate prediction. These equations are shown in the following:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{20}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{21}$$

$$\tilde{h_t} = \tanh(W_h x_t + U_h(r_t \odot h_{t-1})) \tag{22}$$

$$h_t = z_t \odot \tilde{h_t} + (1 - z_t) \odot h_{t-1} \tag{23}$$

z_t denotes the update gate and r_t denotes the reset gate; W_s and U_s are the weight matrices; h_t is the output information to the next unit; $\tilde{h}_t$ is the current cell state; x_t denotes the input vector; and the Hadamard product is $\odot$.

A single GRU unit structure is shown in the following figure:

2.5 Bidirectional Structure

Bidirectional deep learning models have emerged as powerful tools in various domains by capturing contextual dependencies in sequential data. Two widely used variants of bidirectional models are Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Gated Recurrent Unit (Bi-GRU).

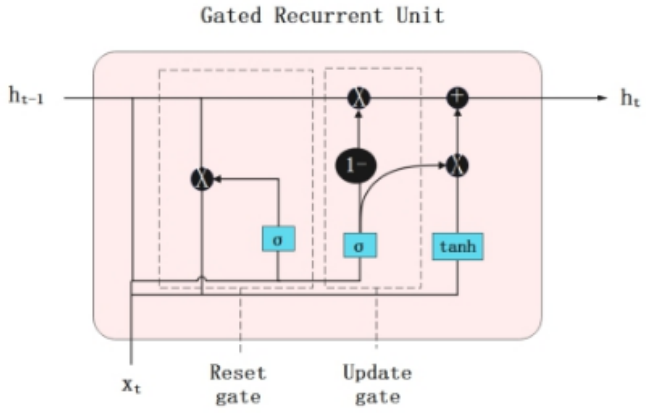

Figure 2: GRU unit structure.

Traditional recurrent neural networks (RNNs) process data in a unidirectional manner, where information flows from past to future. This unidirectional nature can limit their ability to capture dependencies that span both past and future contexts. Bidirectional models address this limitation by introducing two separate layers, one processing the sequence in the forward direction and the other in the backward direction.

Bidirectional Long Short-term Memory (Bi-LSTM), was proposed by as proposed by Schuster and Paliwa [26], is a variant of the LSTM architecture, incorporates two LSTM layers working in opposite directions. The forward layer processes the input sequence from the beginning to the end, while the backward layer processes it from the end to the beginning. By merging the outputs of both layers, Bi-LSTM captures dependencies in both directions, enabling the model to capture long-term relationships and make more informed predictions.

Similarly, Bidirectional Gated Recurrent Unit (Bi-GRU) by yunghyun Cho et al. [24], is a bidirectional extension of the Gated Recurrent Unit (GRU) architecture.

It employs two GRU layers that process the input sequence in opposite directions. The GRU units incorporate gating mechanisms to selectively update and reset the memory, allowing the model to effectively capture and propagate information through time in both directions

Bidirectional models performs well in natural language-processing problems, such as sentence classification and translation. It could also be applied in handwritten recognition problem, sequence problems, and similar fields.

A Bidirectional unit is the same as the original deep learning unit, but the architecture is different. To show the difference, the architectures of the two models are shown in the following figure:

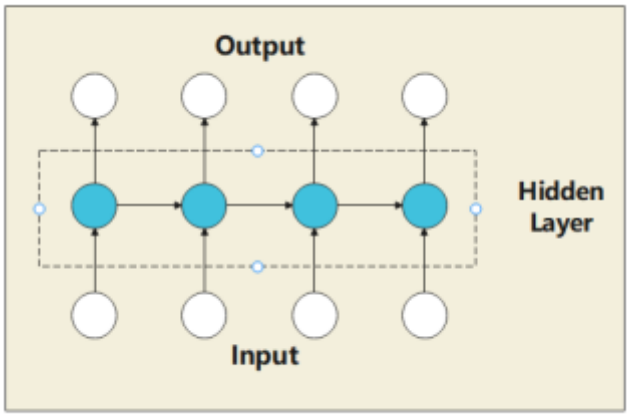

Figure 3: The normal RNN architecture.

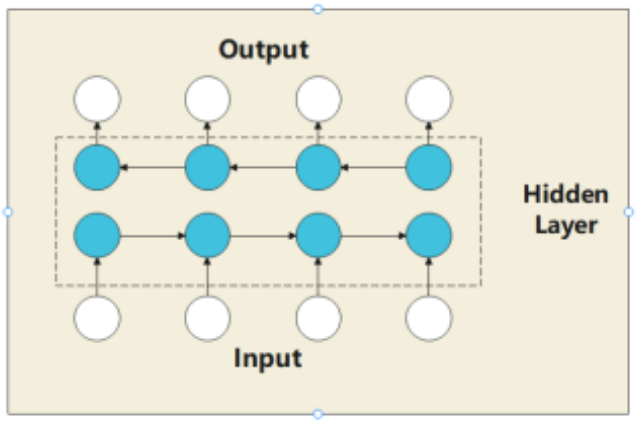

Figure 4: The Bidirectional RNN architecture.

2.6 Hybrid Lee-Carter Model

In the traditional Lee-Carter model, the Autoregressive Integrated Moving Average model (ARIMA) is used to predict the future time index. Here we denoted the time index calculated by ARIMA model, LSTM model, Bi-LSTM, GRU and Bi-GRU

model as $\tilde{\kappa}_t^{ARIMA}$, $\tilde{\kappa}_t^{LSTM}$, $\tilde{\kappa}_t^{Bi-LSTM}$, $\tilde{\kappa}_t^{GRU}$ and $\tilde{\kappa}_t^{LSTM}$. Therefore, the forecasting mortality rates calculated by these models could be described as:

$$\tilde{m}_{x,t}^{ARIMA} = \exp(\tilde{\alpha}_x + \tilde{\beta}_x \tilde{\kappa}_t^{ARIMA}) \tag{24}$$

$$\tilde{m}_{x,t}^{LSTM} = \exp(\alpha_x + \tilde{\beta}_x \tilde{\kappa}_t^{LSTM}) \tag{25}$$

$$\tilde{m}_{x,t}^{GRU} = \exp(\tilde{\alpha}_x + \tilde{\beta}_x \tilde{\kappa}_t^{GRU}) \tag{26}$$

$$\tilde{m}_{x,t}^{Bi-LSTM} = \exp(\tilde{\alpha}_x + \tilde{\beta}_x \tilde{\kappa}_t^{Bi-LSTM}) \tag{27}$$

$$\tilde{m}_{x,t}^{Bi-GRU} = \exp(\tilde{\alpha}_x + \tilde{\beta}_x \tilde{\kappa}_t^{Bi-GRU}) \tag{28}$$

In the end of this section, we define some notations for the hybrid models might appear in the forthcoming sections.

ARIMA hybrid Lee-Carter (traditional Lee-Carter): LC-ARIMA

Long Short-Term Memory hybrid Lee-Carter: LC-LSTM

Bidirectional Long Short-Term Memory hybrid Lee-Carter: LC-Bi-LSTM

Gated Recurrent Unit hybrid Lee-Carter: LC-GRU

Bidirectional Gated Recurrent Unit hybrid Lee-Carter: LC-Bi-GRU

CHAPTER 3

DATA

This study focused on mortality rates for the nine census divisions in the US: New England, Middle Atlantic, East North Central, West North Central, South Atlantic, East South Central, West South Central, Mountain, and Pacific. The data for the numerical experiment were collected from usa.mortality.org (the United States mortality database)[27]. This study was obtained on the life tables for nine census divisions in the US. This database provides the central mortality rates for 24 age groups (from 0 to 110+) by gender. These datasets were split into training and test sets with the rules of 80% training and 20% test. Due to the time series data, we could not randomly split the data set, so we picked the historical data as the training set and predicted the future mortality rates based on that. The total years and the corresponding testing set years of the data are shown in Table 1 and the average mortality rates by age groups and gender are shown in the Table 2.

We can see that the average mortality rates increase with age for both genders. In the older age groups, particularly above 85, the mortality rates exhibit a significant increase for both genders.

Table 1: Census Division and Testing Set Years

Census Division	**Total Years**	**Testing Set Years**
New England	1966-2005	2006-2015
Middle Atlantic	1966-2005	2006-2015
East North Central	1966-2005	2006-2015
West North Central	1966-2005	2006-2015
South Atlantic	1966-2005	2006-2015
East South Central	1966-2005	2006-2015
West South Central	1966-2005	2006-2015
Mountain	1966-2005	2006-2015
Pacific	1966-2005	2006-2015

Table 2: Average Mortality Rates by Genders and Age groups

Age group	**Male**	**Female**
0	0.0103184	0.008126
1–4	0.0003978	0.0003192
5–9	0.000224	0.0001586
10–14	0.00024	0.0001582
15–19	0.0008736	0.0003356
20–24	0.0012676	0.0004144
25–29	0.0012794	0.0004938
30–34	0.0014682	0.0006756
35–39	0.0019484	0.0010024
40–44	0.0028846	0.0015864
45–49	0.004514	0.0025628
50–54	0.0071644	0.0039904
55–59	0.011396	0.0061842
60–64	0.0179772	0.0097272
65–69	0.0277216	0.0151632
70–74	0.0425512	0.0242644
75–79	0.065019	0.0394208
80–84	0.100258	0.065534
85–89	0.1565726	0.111995
90–94	0.2401684	0.1854848
95–99	0.3470992	0.289526
100–104	0.4723096	0.4209162
105–109	0.601285	0.5649978
110+	0.7013268	0.680411

CHAPTER 4

NUMERICAL EXPERIMENTS

4.1 ARIMA model Selection

The process of finding the best ARIMA model to univariate time series was achieved on the R vision 4.2.1 with the auto ARIMA in the forecast package. The best ARIMA (p,d,q) with drift for males and females are shown in the following Tables 3 and 4, respectively.

Table 3: Best ARIMA(p,d,q) for females

Census Division	**Best ARIMA(p,d,q)**
New England	ARIMA(1,1,1)
Middle Atlantic	ARIMA(2,1,0)
East North Central	ARIMA(0,1,0)
West North Central	ARIMA(1,1,2)
South Atlantic	ARIMA(0,1,0)
East South Central	ARIMA(2,2,1)
West South Central	ARIMA(0,1,1)
Mountain	ARIMA(1,2,1)
Pacific	ARIMA(0,1,0)

Table 4: Best ARIMA(p,d,q) for males

Census Division	**Best ARIMA(p,d,q)**
New England	ARIMA(0,1,0)
Middle Atlantic	ARIMA(0,1,0)
East North Central	ARIMA(0,1,0)
West North Central	ARIMA(1,1,0)
South Atlantic	ARIMA(0,1,0)
East South Central	ARIMA(0,1,0)
West South Central	ARIMA(0,1,0)
Mountain	ARIMA(0,1,0)
Pacific	ARIMA(0,1,0)

4.2 Selection of Deep Learning Hyperparameters

In order to find the model with the best performance, several parameter values are discussed and selected to improve the predictive performance. We summarized these parameters below:

(i) Number of hidden layers.

(ii) Number of neurons.

(iii) Selection of activation function.

(iv) Selection of optimizer.

(v) Selection of loss function.

(vi) number of epochs.

(vii) batch size.

(viii) dropout rate.

We only fixed the number of hidden layers, activation functions, optimizer, loss function for all the models, the number of neurons, number of epochs, batch size and dropout rate would vary between the data set.

We firstly discuss the parameter selections for neural networks applied on the data set directly. A simple neural networks architecture was used, which contains two hidden layers and a dense layer with a single unit. The activation function *tanh* is used. In the model compilation part, we proposed the optimizer Adam and the loss function is mean absolute error (MAE). The recurrent neural networks predictions were run in Python with the TensorFlow package and the neurons, batch size, epochs, and the dropout percent for neural networks for females and males by divisions are shown in Table 5 and Table 6, respectively.

About the hybrid model part, a single hidden layer model shows better performance than two or three layers model, so we proposed a one-layer neural networks architecture with tanh as activation function in the hidden layer. In the model com-

pilation part, the optimizer Adam is used and the loss function is mean average error (MSE). The recurrent neural networks predictions were run in Python with the TensorFlow package. The epochs and dropout percentages for each division are shown by genders in the following Tables, respectively.

Table 5: Neural Network Configurations for Female Database by Census Division

Census Division	**Models**	**Neurons**	**Batch Size**	**Epochs**	**Dropout**
New England	LSTM	128	32	50	20%
	Bi-LSTM	128	32	50	20%
	GRU	128	32	50	20%
	Bi-GRU	128	32	50	20%
Middle Atlantic	LSTM	64	32	100	10%
	Bi-LSTM	64	32	100	10%
	GRU	64	32	100	10%
	Bi-GRU	64	32	100	10%
East North Central	LSTM	64	16	150	10%
	Bi-LSTM	64	16	150	10%
	GRU	64	16	150	10%
	Bi-GRU	64	16	150	10%
West North Central	LSTM	128	32	50	20%
	Bi-LSTM	128	32	50	20%
	GRU	128	32	50	20%
	Bi-GRU	128	32	50	20%
South Atlantic	LSTM	128	16	150	10%
	Bi-LSTM	128	16	150	10%
	GRU	128	16	150	20%
	Bi-GRU	128	16	150	10%
East South Central	LSTM	128	32	100	30%
	Bi-LSTM	128	32	100	30%
	GRU	128	32	100	10%
	Bi-GRU	128	32	100	10%
West South Central	LSTM	128	64	100	10%
	Bi-LSTM	64	64	100	10%
	GRU	128	16	100	10%
	Bi-GRU	128	16	100	10%
Mountain	LSTM	128	32	50	20%
	Bi-LSTM	128	32	50	20%
	GRU	128	32	50	20%
	Bi-GRU	128	32	50	20%
Pacific	LSTM	128	32	100	20%
	Bi-LSTM	128	32	100	20%
	GRU	128	32	100	20%
	Bi-GRU	128	32	100	20%

Table 6: Neural Network Configurations for Male Database by Census Division

Census Division	**Models**	**Neurons**	**Batch Size**	**Epochs**	**Dropout**
New England	LSTM	128	16	100	10%
	Bi-LSTM	128	16	100	10%
	GRU	128	32	100	20%
	Bi-GRU	64	16	100	10%
Middle Atlantic	LSTM	128	32	50	30%
	Bi-LSTM	64	32	50	30%
	GRU	64	32	100	30%
	Bi-GRU	64	32	100	30%
East North Central	LSTM	128	64	150	10%
	Bi-LSTM	128	64	150	10%
	GRU	128	64	150	10%
	Bi-GRU	128	64	150	10%
West North Central	LSTM	64	64	150	30%
	Bi-LSTM	128	32	150	30%
	GRU	64	32	150	30%
	Bi-GRU	64	32	150	30%
South Atlantic	LSTM	128	16	150	10%
	Bi-LSTM	128	16	150	10%
	GRU	128	16	50	10%
	Bi-GRU	128	16	50	10%
East South Central	LSTM	128	16	150	20%
	Bi-LSTM	128	32	50	20%
	GRU	64	32	50	20%
	Bi-GRU	64	32	50	20%
West South Central	LSTM	64	16	150	20%
	Bi-LSTM	64	64	150	20%
	GRU	64	16	150	20%
	Bi-GRU	64	16	100	20%
Mountain	LSTM	128	32	100	20%
	Bi-LSTM	128	32	100	20%
	GRU	128	32	100	20%
	Bi-GRU	128	32	100	20%
Pacific	LSTM	128	32	100	20%
	Bi-LSTM	128	32	100	20%
	GRU	128	32	100	20%
	Bi-GRU	128	32	100	20%

Table 7: Hyperparameters for Neural Networks in the Hybrid Lee-Carter Model: Female Database by Census Division

Census Division	**Models**	**Neurons**	**Batch Size**	**Epochs**	**Dropout**
New England	LSTM	64	32	75	20%
	Bi-LSTM	32	32	75	20%
	GRU	32	32	75	20%
	Bi-GRU	32	32	75	20%
Middle Atlantic	LSTM	32	64	100	20%
	Bi-LSTM	32	64	100	20%
	GRU	32	64	100	10%
	Bi-GRU	32	64	100	20%
East North Central	LSTM	64	32	50	20%
	Bi-LSTM	64	32	50	20%
	GRU	64	32	50	20%
	Bi-GRU	64	32	50	20%
West North Central	LSTM	64	32	75	10%
	Bi-LSTM	64	32	75	10%
	GRU	64	32	75	10%
	Bi-GRU	64	32	75	10%
South Atlantic	LSTM	64	16	100	20%
	Bi-LSTM	64	16	100	20%
	GRU	64	16	100	20%
	Bi-GRU	64	16	100	20%
East South Central	LSTM	32	32	100	20%
	Bi-LSTM	64	32	100	5%
	GRU	32	32	100	10%
	Bi-GRU	64	32	100	5%
West South Central	LSTM	64	16	200	10%
	Bi-LSTM	64	16	50	10%
	GRU	64	16	200	10%
	Bi-GRU	64	16	50	10%
Mountain	LSTM	64	32	100	20%
	Bi-LSTM	64	32	75	10%
	GRU	64	32	100	10%
	Bi-GRU	64	32	75	10%
Pacific	LSTM	64	32	50	20%
	Bi-LSTM	64	32	50	10%
	GRU	64	32	50	10%
	Bi-GRU	64	32	50	10%

Table 8: Hyperparameters for Neural Networks in the Hybrid Lee-Carter Model: Male Database by Census Division

Census Division	**Models**	**Neurons**	**Batch Size**	**Epochs**	**Dropout**
New England	LSTM	64	32	75	20%
	Bi-LSTM	32	32	75	20%
	GRU	32	32	75	20%
	Bi-GRU	32	32	75	20%
Middle Atlantic	LSTM	64	32	75	10%
	Bi-LSTM	64	32	75	10%
	GRU	64	32	75	10%
	Bi-GRU	64	32	75	10%
East North Central	LSTM	64	32	75	20%
	Bi-LSTM	64	32	100	10%
	GRU	64	32	75	10%
	Bi-GRU	64	32	100	10%
West North Central	LSTM	64	32	100	20%
	Bi-LSTM	32	32	100	20%
	GRU	32	32	100	20%
	Bi-GRU	64	32	100	10%
South Atlantic	LSTM	64	32	75	20%
	Bi-LSTM	64	32	100	10%
	GRU	64	32	75	10%
	Bi-GRU	64	32	100	10%
East South Central	LSTM	64	64	150	20%
	Bi-LSTM	64	64	50	20%
	GRU	64	64	50	5%
	Bi-GRU	64	64	50	10%
West South Central	LSTM	64	32	50	5%
	Bi-LSTM	32	32	50	5%
	GRU	64	32	50	5%
	Bi-GRU	32	32	50	5%
Mountain	LSTM	32	16	50	5%
	Bi-LSTM	32	16	50	5%
	GRU	32	16	50	5%
	Bi-GRU	32	16	50	5%
Pacific	LSTM	64	32	75	10%
	Bi-LSTM	64	32	75	10%
	GRU	64	32	75	10%
	Bi-GRU	64	32	75	10%

CHAPTER 5

RESULTS

5.1 Error Metric

To measure the prediction performance, we selected two error criteria for the out-of-sample test, mean absolute error (MAE) and root mean square error (RMSE), the equations of the MAE and RMSE are presented as equations (29-30). The total amount of data in the test data is denoted by n, the predicted mortality rate is denoted by $\tilde{m}_{x,t}$ and the actual mortality rate is denoted by $m_{x,t}$.

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\tilde{m}_{x,t} - m_{x,t}| \tag{29}$$

Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\tilde{m}_{x,t} - m_{x,t})^2} \tag{30}$$

5.2 Results of the Deep Learning models

The MAE and RMSE values for deep learning models applied on the data set directly are shown in the following tables. From the results, we can see that the Bi-GRU model showed better MAE value and RMSE value than Lee-Carter model on all the data set, which the LSTM, Bi-LSTM and GRU provided 66.7%/55.6% (MAE/RMSE), 77.8%/77.8% (MAE/RMSE) and 88.9%/77.8%.

The bidirectional models show better performance in terms of MAE and RMSE compared to the other models. Especially in East North Central female data set,

East South Central data sets for both genders, Mountain data sets for both genders and Pacific data sets for both genders.

Table 9: Summary of MAE for LC, LSTM, Bi-LSTM, GRU, and Bi-GRU, the best MAE value for each row in Bold

Census Division	**Gender**	**LC**	**LSTM**	**Bi-LSTM**	**GRU**	**Bi-GRU**
New England	female	0.003580	0.003333	**0.001915**	0.002554	0.002901
New England	male	0.003814	0.003602	0.004280	0.004250	**0.002809**
Middle Atlantic	female	0.002296	0.004322	0.003066	0.003321	**0.002158**
Middle Atlantic	male	0.003419	0.003642	0.004539	0.002458	**0.001680**
East North Central	female	0.004458	0.002742	0.002667	**0.001737**	0.002854
East North Central	male	0.004280	0.004202	0.002965	0.003735	**0.002577**
West North Central	female	0.006313	0.003326	**0.001873**	0.002378	0.002094
West North Central	male	0.005871	0.003410	0.003193	**0.002437**	0.002842
South Atlantic	female	0.004249	0.004079	0.003187	**0.002117**	0.003102
South Atlantic	male	0.004342	0.002981	0.004144	0.004228	**0.002680**
East South Central	female	0.005919	0.006182	0.004247	0.003618	**0.001605**
East South Central	male	0.006056	0.006060	0.002955	0.002968	**0.002519**
West South Central	female	0.003881	0.002977	0.002770	0.002103	**0.001793**
West South Central	male	0.004401	0.003565	0.004671	**0.003043**	0.003114
Mountain	female	0.005875	0.003609	0.002195	0.002961	**0.001781**
Mountain	male	0.005863	0.005482	0.003146	0.002909	**0.001501**
Pacific	female	0.003030	0.003108	0.001592	0.001743	**0.001179**
Pacific	male	0.003856	0.004156	0.002447	0.003850	**0.001937**

Table 10: Summary of RMSE for LC, LSTM, Bi-LSTM, GRU, and Bi-GRU, the best RMSE value for each row in bold

Census Division	**Gender**	**LC**	**LSTM**	**Bi-LSTM**	**GRU**	**Bi-GRU**
New England	female	0.008577	**0.004150**	0.008452	0.005656	0.005387
New England	male	0.007061	0.007446	0.008178	0.009505	**0.006773**
Middle Atlantic	female	0.005549	0.009855	0.007108	0.008056	**0.004276**
Middle Atlantic	male	0.006418	0.007088	0.009344	0.004819	**0.003413**
East North Central	female	0.010601	0.005402	0.005603	**0.003966**	0.006533
East North Central	male	0.008134	0.009267	0.006617	0.009019	**0.005878**
West North Central	female	0.014708	0.007639	**0.004243**	0.005635	0.004777
West North Central	male	0.012320	0.006971	0.006893	**0.005193**	0.005933
South Atlantic	female	0.010067	0.009416	0.007428	**0.004476**	0.007250
South Atlantic	male	0.007902	0.006224	0.007778	0.008747	**0.005250**
East South Central	female	0.013795	0.015090	0.010324	0.008584	**0.003085**
East South Central	male	0.012114	0.013909	0.006451	0.006333	**0.005613**
West South Central	female	0.009499	0.006708	0.006103	0.004496	**0.003553**
West South Central	male	0.008112	0.007457	0.009865	0.006260	**0.006021**
Mountain	female	0.013608	0.008673	0.005162	0.006747	**0.004486**
Mountain	male	0.011635	0.012597	0.006623	0.006447	**0.002989**
Pacific	female	0.006329	0.003649	0.005435	0.004081	**0.002598**
Pacific	male	0.007343	0.008044	0.004553	0.006967	**0.004095**

5.3 Hybrid Models

First part of this section, we show the prediction results of time index by ARIMA model. The following figures are showing the κ_t parameter estimation by ARIMA and the 80%, 85% and 95% confidence interval. Based on the figures presented, the kappa values exhibit decreasing pattern over time on all of the data sets.

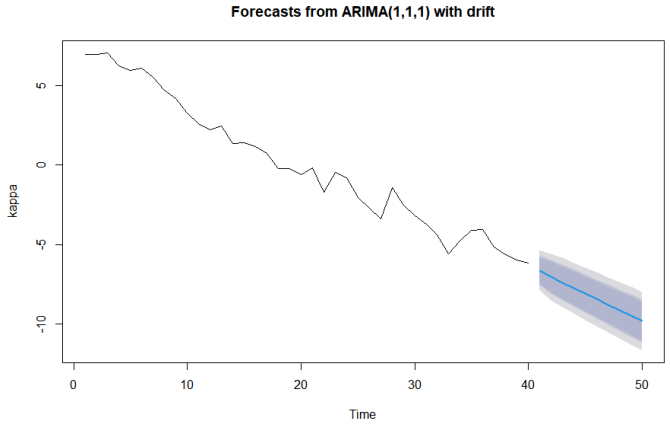

Figure 5: κ_t estimation by ARIMA, New England, female.

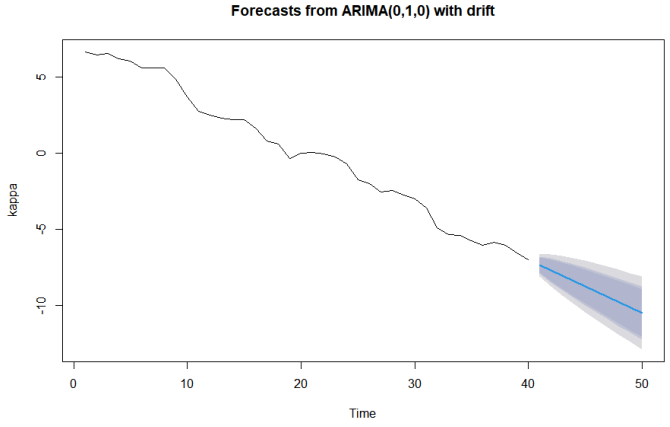

Figure 6: κ_t estimation by ARIMA, New England, male.

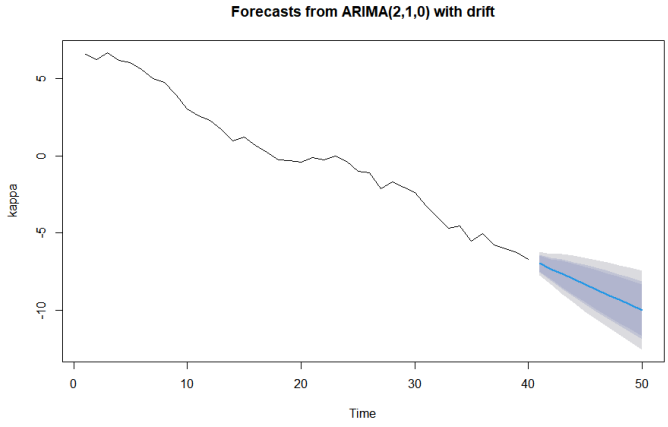

Figure 7: κ_t estimation by ARIMA, Middle Atlantic, female.

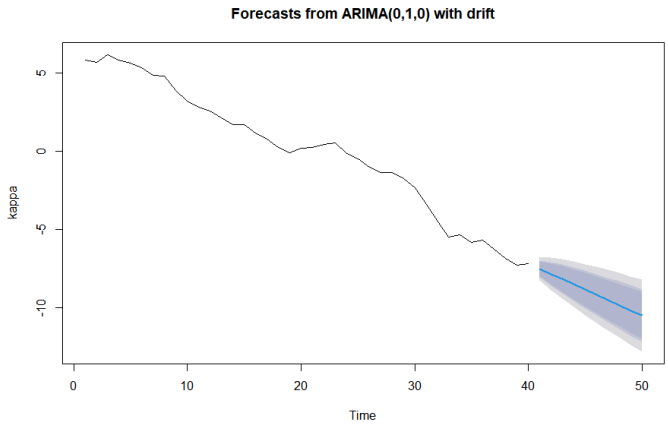

Figure 8: κ_t estimation by ARIMA, Middle Atlantic, male.

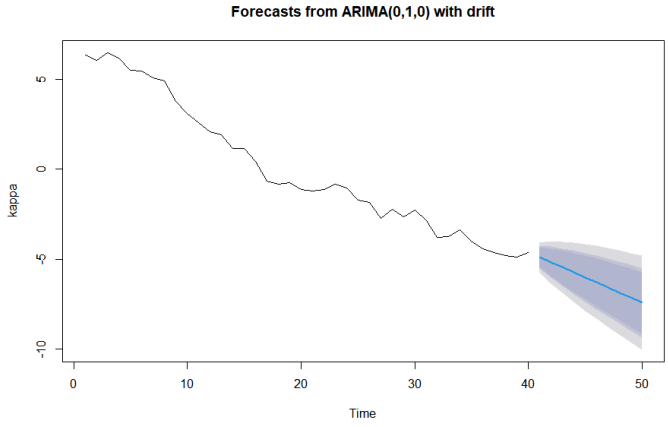

Figure 9: κ_t estimation by ARIMA, East North Central, female.

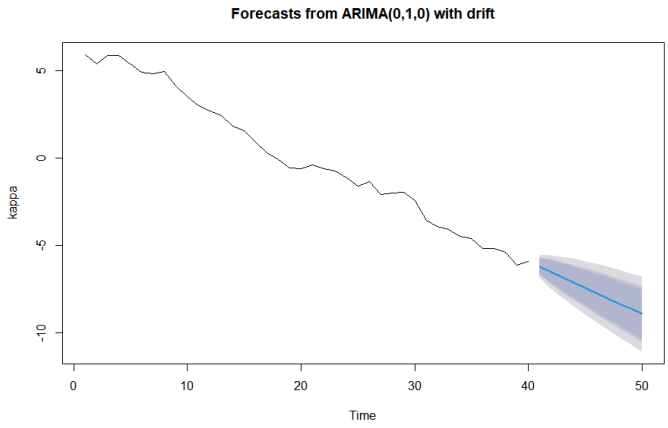

Figure 10: κ_t estimation by ARIMA, East North Central, male.

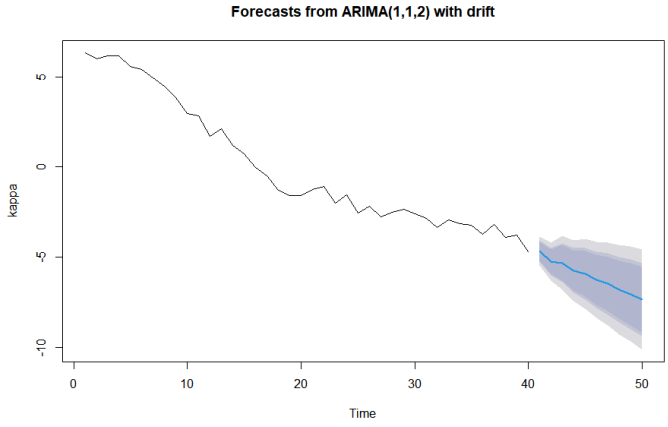

Figure 11: κ_t estimation by ARIMA, West North Central, female.

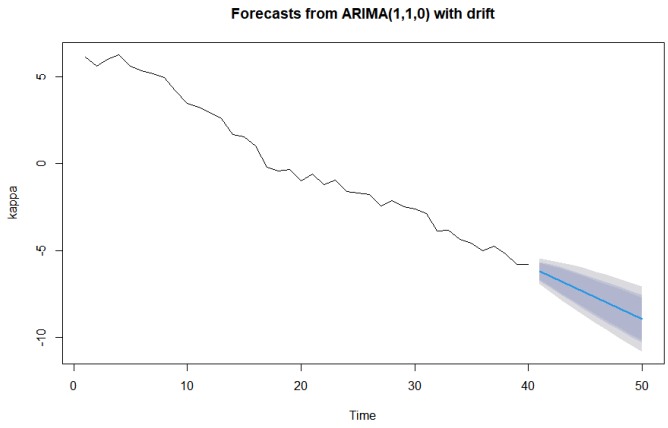

Figure 12: κ_t estimation by ARIMA, West North Central, male.

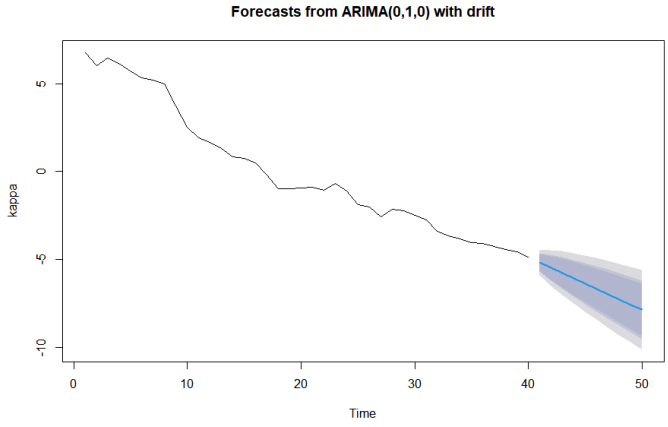

Figure 13: κ_t estimation by ARIMA, South Atlantic, female.

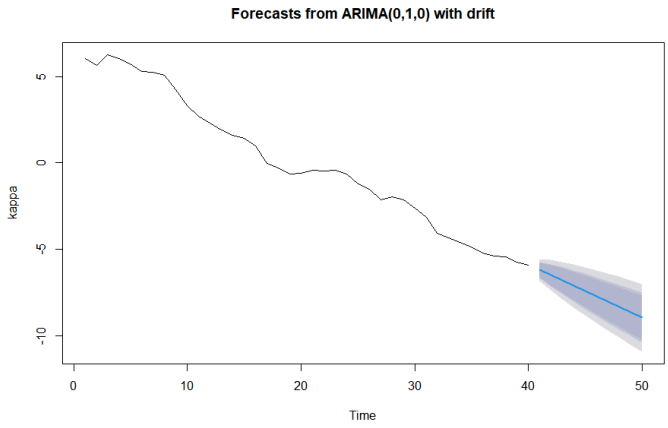

Figure 14: κ_t estimation by ARIMA, South Atlantic, male.

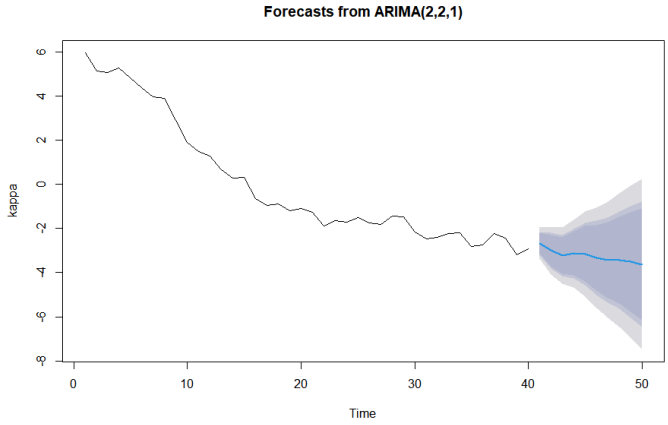

Figure 15: κ_t estimation by ARIMA, East South Central, female.

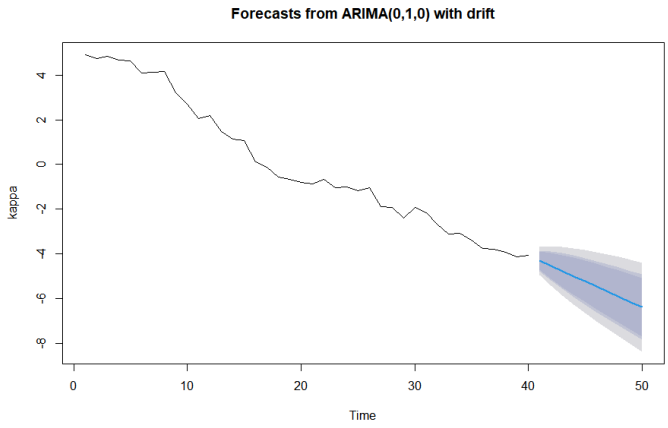

Figure 16: κ_t estimation by ARIMA, East South Central, male.

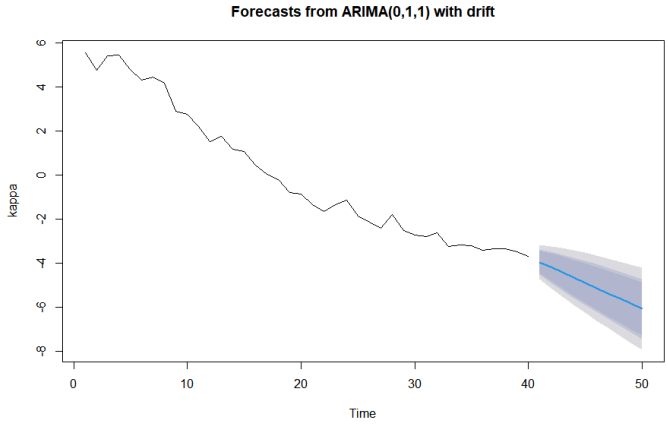

Figure 17: κ_t estimation by ARIMA, West South Central, female.

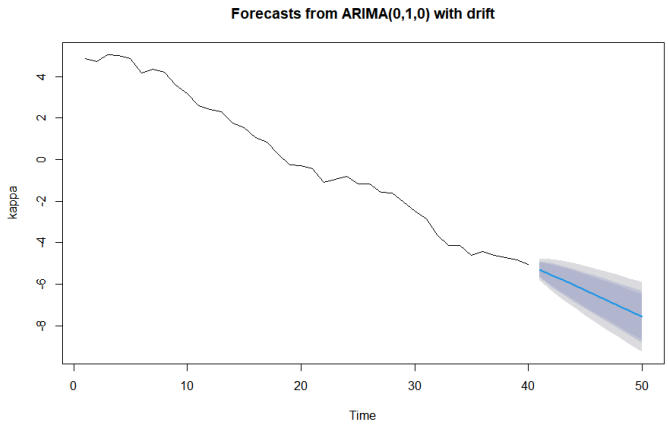

Figure 18: κ_t estimation by ARIMA, West South Central, male.

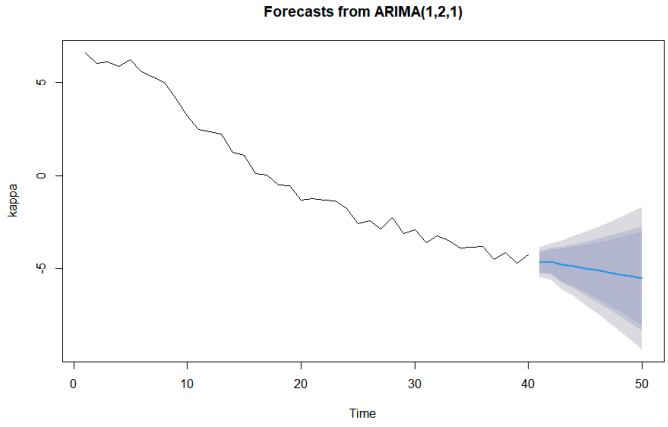

Figure 19: κ_t estimation by ARIMA, Mountain, female.

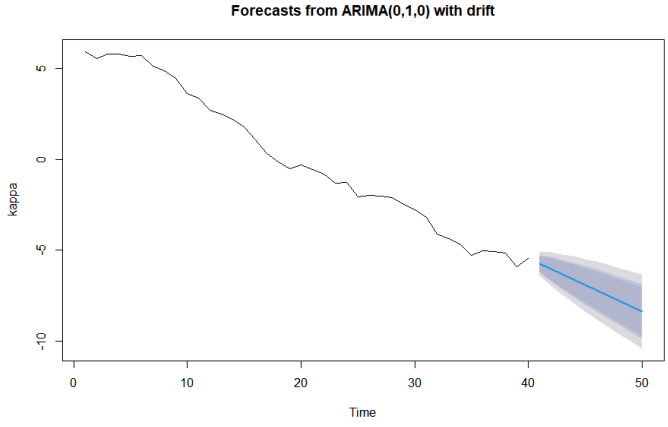

Figure 20: κ_t estimation by ARIMA, Mountain, male.

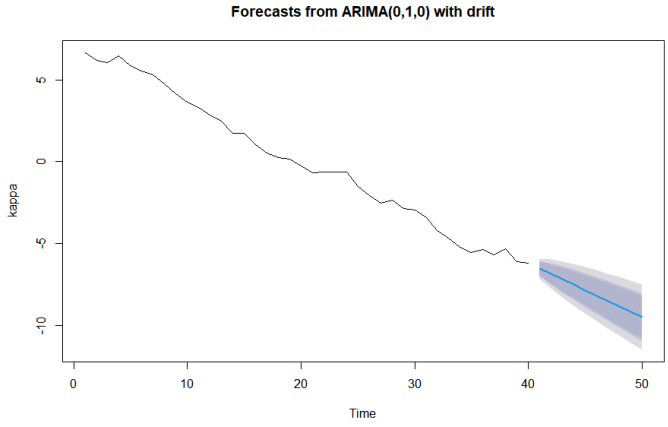

Figure 21: κ_t estimation by ARIMA, Pacific, female.

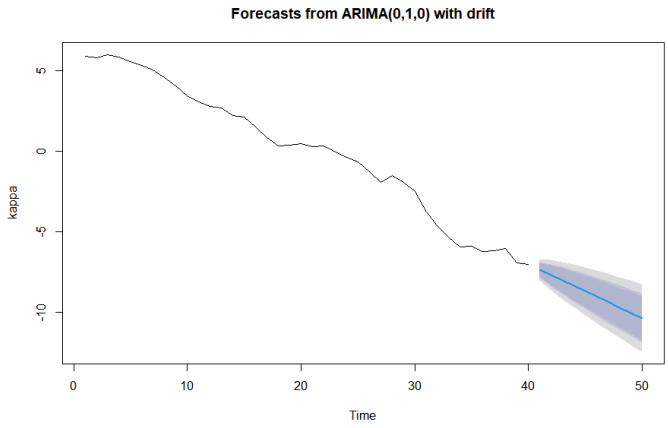

Figure 22: κ_t estimation by ARIMA, Pacific, male.

The following tables shown the mean average error (MAE) and the root mean square error (RMSE) for the estimation values by Lee-Carter ARIMA and deep learning hybrid Lee-Carter models. From the results, we can see that the deep learning hybrid Lee-Carter models offer pretty comparable results to the traditional Lee-Carter model. LC-GRU model shows better predictive performance in terms of MAE value and RMSE value on all the database, compared to the traditional ARIMA model, which the LSTM, Bi-LSTM and Bi-GRU provided 61.1%/61.1% (MAE/RMSE), 44.4%/55.6% (MAE/RMSE) and 66.7%/77.8%.

It is important to analyze the models in the context of each specific divisions and genders to gain a comprehensive understanding of the model performance. After a more detailed analysis of the tables, it can be observed from the Table 11 that the Bi-LSTM model is exhibiting competitive performance to the GRU model in terms of MAE values across different census divisions and genders. When considering RMSE values as a measure of predicting ability in Table 12, the Bi-GRU model consistently outperforms the GRU model in 10 out of the 18 data sets. Therefore, the Bi-LSTM and Bi-GRU models exhibit good predictive performance compared to the GRU model in terms of MAE and RMSE, respectively. Again, all of the models examined in this study produced very similar estimations, indicating their comparable performance. The GRU model consistently yielded the best results among the models in terms of MAE and RMSE metrics.

Table 11: Summary of MAE for ARIMA, LSTM, Bi-LSTM, GRU, and Bi-GRU in Hybrid Lee-Carter model, the best MAE value for each row in bold

Census Division	**Gender**	**ARIMA**	**LSTM**	**Bi-LSTM**	**GRU**	**Bi-GRU**
New England	female	0.003580	0.003611	0.003687	**0.003574**	0.003579
New England	male	0.003814	**0.003746**	0.003038	0.003553	0.003144
Middle Atlantic	female	0.002296	0.002266	**0.002208**	0.002245	0.002265
Middle Atlantic	male	0.003419	**0.003003**	0.003234	0.003114	0.003201
East North Central	female	0.004458	0.004404	**0.004399**	0.004419	0.004417
East North Central	male	0.004280	**0.004062**	0.004435	0.004199	0.004123
West North Central	female	0.006313	0.005383	**0.005173**	0.005537	0.005201
West North Central	male	0.005871	0.006105	0.006022	**0.005731**	0.005795
South Atlantic	female	0.004249	0.004597	0.004650	0.004248	**0.003102**
South Atlantic	male	0.004342	0.004989	0.004662	**0.003828**	0.004932
East South Central	female	0.005919	0.005809	0.006121	**0.005782**	0.006123
East South Central	male	0.006056	0.006297	0.006824	**0.005585**	0.006910
West South Central	female	0.003881	0.004094	**0.003614**	0.003811	0.004435
West South Central	male	0.004401	0.003565	0.004671	**0.003043**	0.003114
Mountain	female	0.005875	**0.005866**	0.005877	0.005868	0.005870
Mountain	male	0.005863	0.005804	**0.005611**	0.005766	0.005840
Pacific	female	0.003030	0.003034	0.003068	**0.002951**	0.002995
Pacific	male	0.003856	**0.003772**	0.003831	0.003827	0.003844

Table 12: Summary of RMSE for ARIMA, LSTM, Bi-LSTM, GRU, and Bi-GRU in Hybrid Lee-Carter model, the best RMSE value for each row in bold

Census Division	**Gender**	**ARIMA**	**LSTM**	**Bi-LSTM**	**GRU**	**Bi-GRU**
New England	female	0.008577	0.008591	0.008630	**0.008573**	0.008574
New England	male	0.007061	0.006945	**0.005719**	0.006611	0.005946
Middle Atlantic	female	0.005549	0.005497	0.005413	0.005471	**0.005250**
Middle Atlantic	male	0.006418	**0.005772**	0.006190	0.005910	0.006037
East North Central	female	0.010601	0.010432	**0.010397**	0.010484	0.010471
East North Central	male	0.008134	**0.007759**	0.008428	0.007996	0.007860
West North Central	female	0.014708	0.012972	**0.012285**	0.013355	0.012412
West North Central	male	0.012320	0.012716	0.012581	**0.012074**	0.012205
South Atlantic	female	0.010067	0.010639	0.007428	0.010059	**0.007250**
South Atlantic	male	0.007902	0.009112	0.008492	**0.007011**	0.008997
East South Central	female	0.013795	0.013558	0.014204	0.013510	**0.013085**
East South Central	male	0.012114	0.012448	0.013558	**0.011113**	0.013729
West South Central	female	0.009499	0.009820	**0.008036**	0.009486	0.010329
West South Central	male	0.008112	0.007457	0.009865	0.006260	**0.006021**
Mountain	female	0.013608	**0.013453**	0.013481	0.013473	0.013461
Mountain	male	0.011635	0.011499	**0.011114**	0.011429	0.011578
Pacific	female	0.006329	0.006345	0.006388	**0.006235**	0.006291
Pacific	male	0.007343	**0.007235**	0.007313	0.007372	0.007340

5.4 Model Comparison

First of all, we recall the experiment results from the paper: Chen and Khaliq. In this paper, we applied the LSTM, Bi-LSTM and GRU on the same dataset with different hyperparameters, the mainly difference is on the activation functions. After analysing the results, we found that the activation function tanh provided us a better performance on mortality rate prediction. The following table shows the average mean average error (MAE) and root mean square error (RMSE) for the three models we used in the previous paper.

Table 13: Averaged MAE and RMSE for the Models by Genders in the previous studies with activation function ReLu

Error Metric	**LC**	**LSTM**	**Bi-LSTM**	**GRU**
MAE female	0.0044	0.004274	0.003989	0.004253
MAE male	0.004656	0.005407	0.004992	0.003639
RMSE female	0.010304	0.009219	0.009907	0.01
RMSE male	0.009004	0.011451	0.009907	0.007741
Average MAE	0.004528	0.004841	0.00449	0.003946
Average RMSE	0.009654	0.010579	0.009563	0.008871

Below, the results obtained from the deep learning models after transitioning the activation function from ReLU to tanh are shown in table 14. We can see, that each model demonstrates a significant improvement in both MAE and RMSE. Especially, the bidirectional gated recurrent unit (Bi-GRU) gives the best prediction results among the models. The results obtained from the deep learning hybrid Lee-Carter model are shown in table 15.

The application of deep learning models directly on the data set has shown better results in terms of MAE and RMSE and all the deep learning models or deep learning hybrid Lee-Carter models gave us better estimation on mortality rates compared to the traditional Lee-Carter-ARIMA model. The bidirectional structures have

Table 14: Averaged MAE and RMSE for the Models by Genders with activation function tanh

Error Metric	**LC**	**LSTM**	**Bi-LSTM**	**GRU**	**Bi-GRU**
MAE female	0.004400	0.003742	0.002612	0.002503	0.002163
MAE male	0.0046556	0.004122	0.003593	0.003320	0.002407
RMSE female	0.010304	0.007843	0.006650	0.005744	0.004660
RMSE male	0.009004	0.008778	0.007367	0.007032	0.005107
Average MAE	0.004528	0.003932	0.003103	0.002912	0.002285
Average RMSE	0.009654	0.008310	0.007008	0.006388	0.004884

Table 15: Averaged MAE and RMSE for the Hybrid Lee-Carter Models by Genders

Error Metric	**ARIMA**	**LSTM**	**Bi-LSTM**	**GRU**	**Bi-GRU**
MAE female	0.004400	0.004340	0.004311	0.004270	0.004221
MAE male	0.004656	0.004594	0.004703	0.004294	0.004545
RMSE female	0.010304	0.010145	0.009585	0.010072	0.009680
RMSE male	0.009004	0.008994	0.009251	0.008420	0.008857
Average MAE	0.004528	0.004467	0.004507	0.004282	0.004383
Average RMSE	0.009654	0.009569	0.009418	0.009246	0.009269

shown better predictive performance than the traditional models in terms of average MAE and RMSE when we applied them to the data set directly (Bi-LSTM/ LSTM: 0.003103/0.003932 on MAE, 0.007008/ 0.008310 on RMSE, Bi-GRU/ GRU: 0.002285/ 0.002912 on MAE, 0.004884/ 0.006388 on RMSE). The bidirectional structure incorporates both forward and backward information flow, led to better accuracy and predictive performance.

However, when the bidirectional neural networks were applied within the hybrid Lee-Carter models, this advantage disappeared. Overall, the results showed that the traditional models outperformed the bidirectional models in terms of two error metric (LSTM/Bi-LSTM: 0.004467/0.004507 on MAE, 0.009569/0.009418 on RMSE, GRU/Bi-GRU: 0.004282/0.004383 on MAE,0.009246/0.009269 on RMSE).

Here, we use the Mountain division data set as an example of life expectancy pre-

diction. The first 40 years are training set and the last 10 years are test set. Here we pick the age groups 40-44 and 90-94, which are respectively to represent the middle age group and the elderly group. The results from age group 40-44 are shown in figure 23-26, the results from age group 90-94 are shown in figure 27-30.

The results showed that the application of deep learning models on data set directly is capable to catch more details from the data set with more nonlinear trend. The deep learning hybrid Lee-Carter models yielded comparable results to the traditional Lee-Carter model. Similar to Lee-Carter model, sometimes they would overrate or underrate the future values (See figure 24 and figure 30). These results are discussed in Bergeron-Boucher et al.[28] and Booth et al.[29].

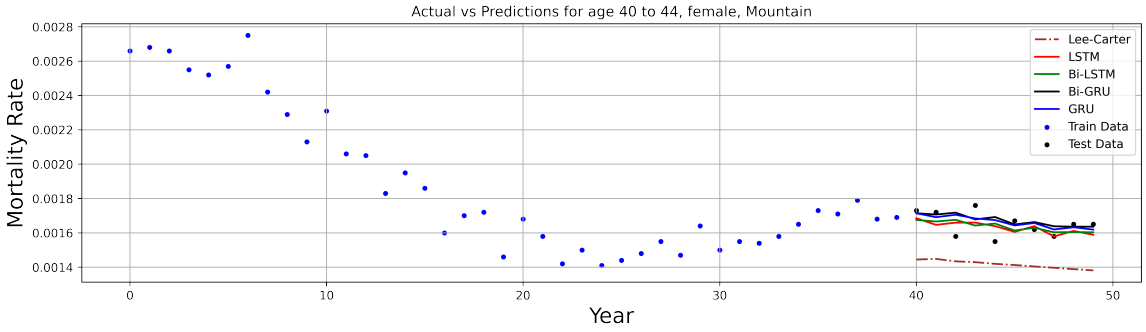

Figure 23: The life expectancy of deep learning models at age group 40-44, Mountain female population.

The predictive results for a single year are also considered in this study, we pick year 2015 as an example. We compare the log-mortality rate estimations by all the models to the real data in following figures. We show the results of all the models in one figure first, and then compare each model with Lee-Carter model one by one.

The figures showed that the deep learning models give better predictive perfor-

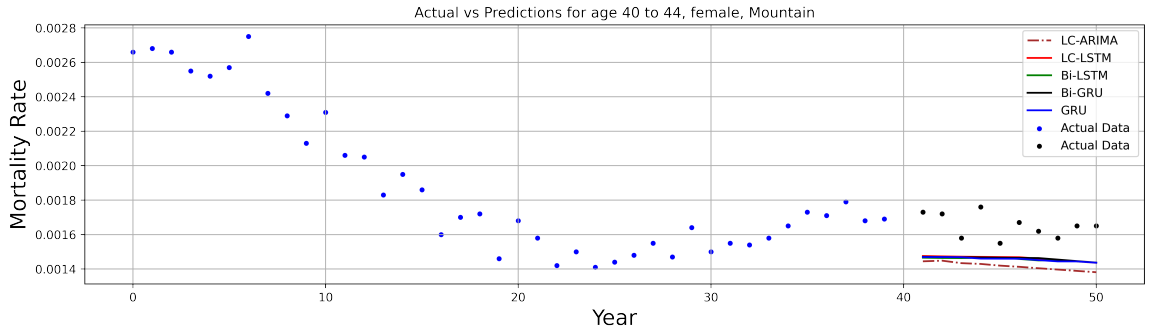

Figure 24: The life expectancy of deep learning hybrid Lee-Carter models at age group 40-44, Mountain female population.

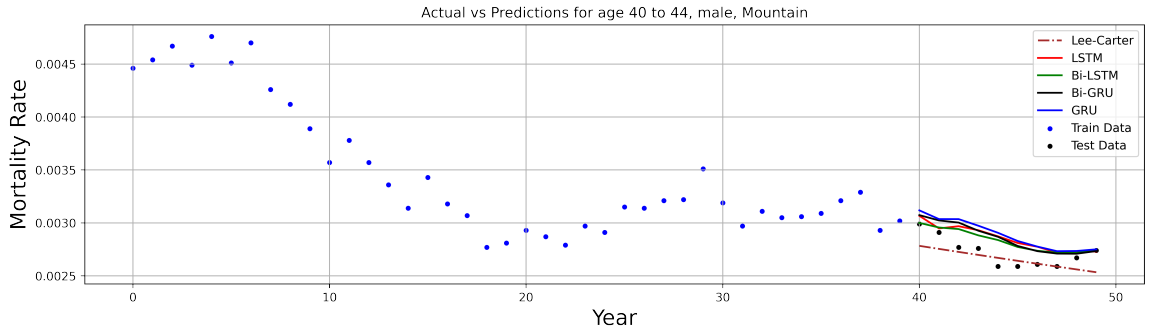

Figure 25: The life expectancy of deep learning models at age group 40-44, Mountain male population.

mance than their hybrid Lee-Carter models on female data set while hybrid models produce more reliable and accurate predictions than the deep learning models on the male data set.

However, if we consider the comparison between the deep learning models and traditional Lee-Carter model, the traditional Lee-Carter ARIMA model shows an incredible estimation on male data set, especially for the first few age groups, a perfect bath tub curve can be observed. The deep learning models outperform the traditional

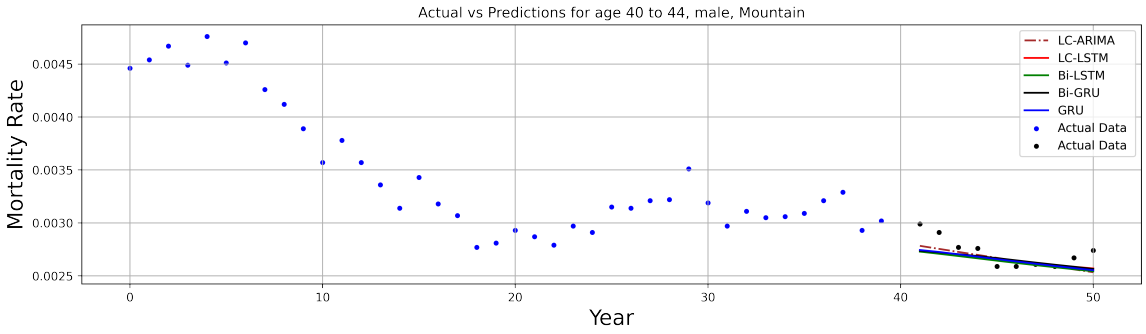

Figure 26: The life expectancy of deep learning hybrid Lee-Carter models at age group 40-44, Mountain male population.

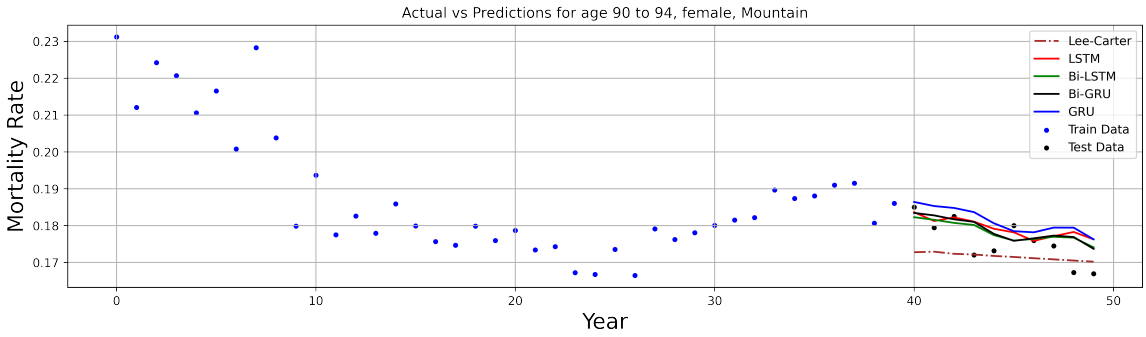

Figure 27: The life expectancy of deep learning models at age group 90-94, Mountain female population.

Lee-Carter model on female data set. For instance, in the figure 40, the Bi-GRU model shows a superior fit to the data points, especially the data points between the 8th age group and 15th age group.

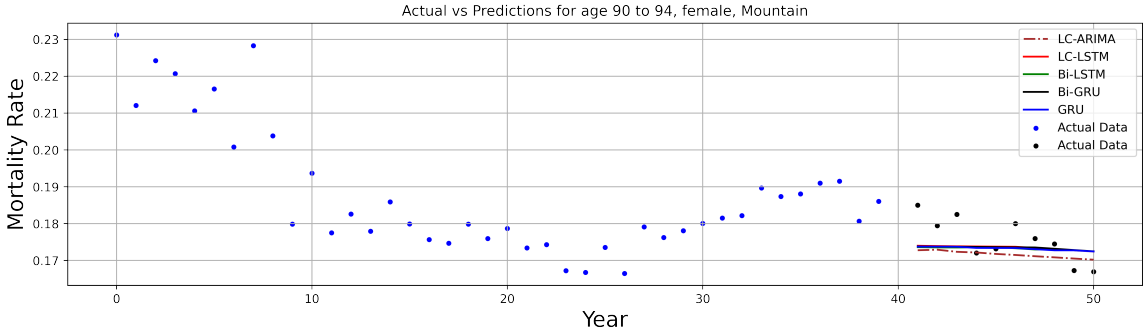

Figure 28: The life expectancy of deep learning hybrid Lee-Carter models at age group 90-94, Mountain female population.

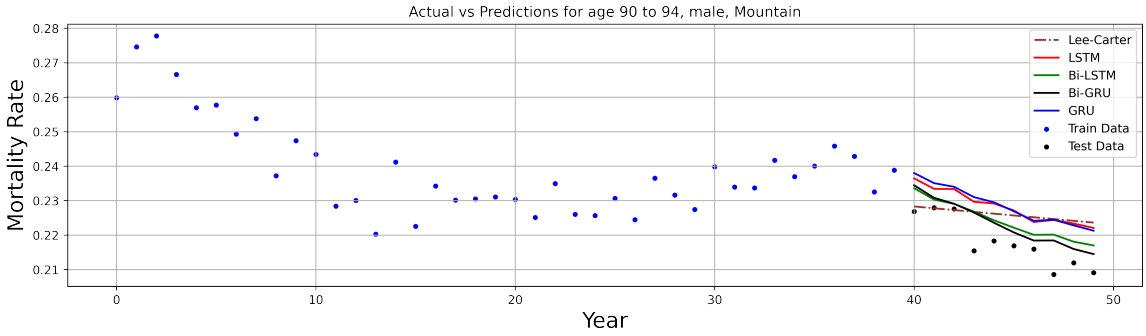

Figure 29: The life expectancy of deep learning models at age group 90-94, Mountain male population.

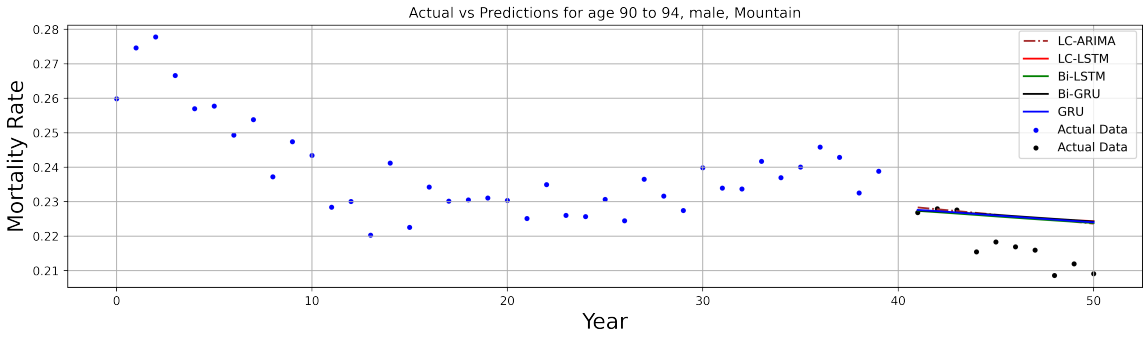

Figure 30: The life expectancy of deep learning hybrid Lee-Carter models at age group 90-94, Mountain male population.

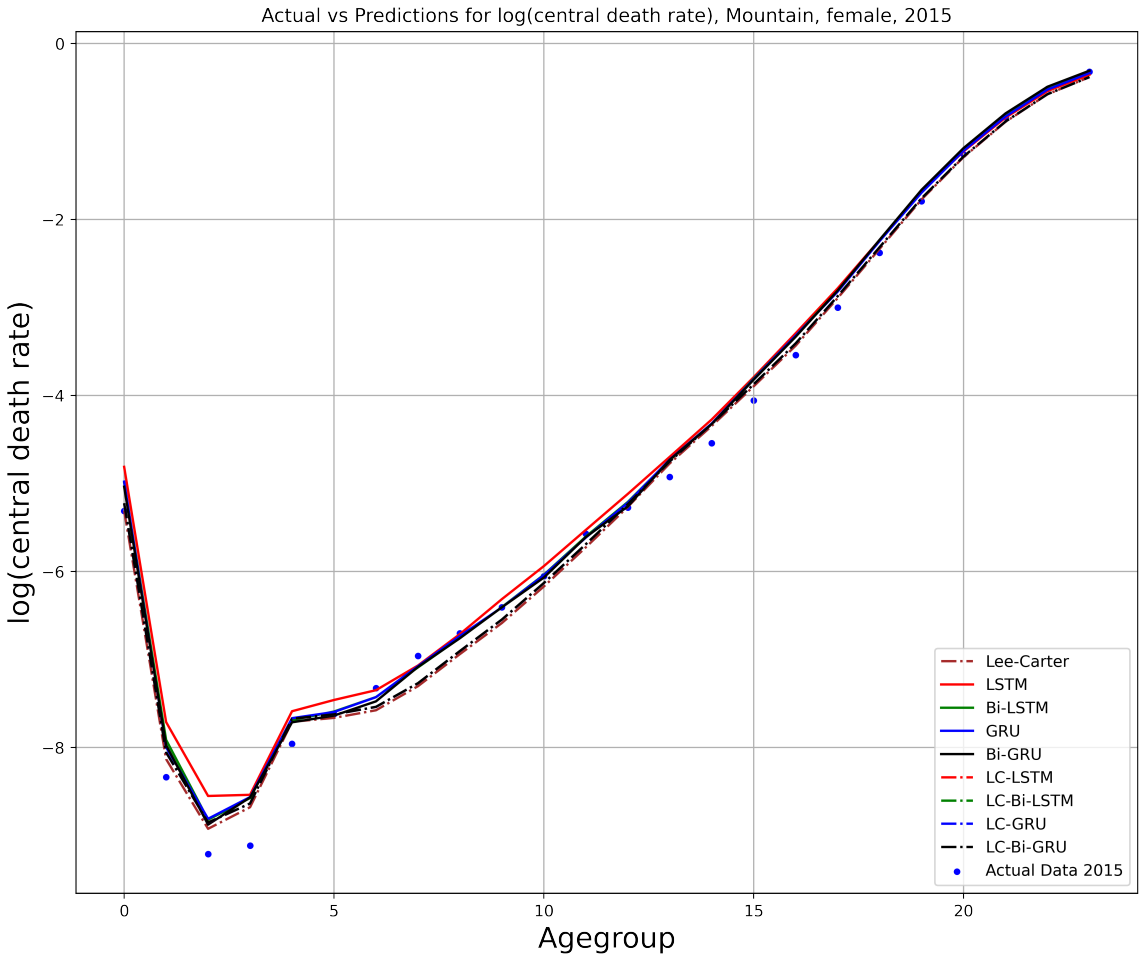

Figure 31: The predictions of log-mortality rate Mountain female population, 2015.

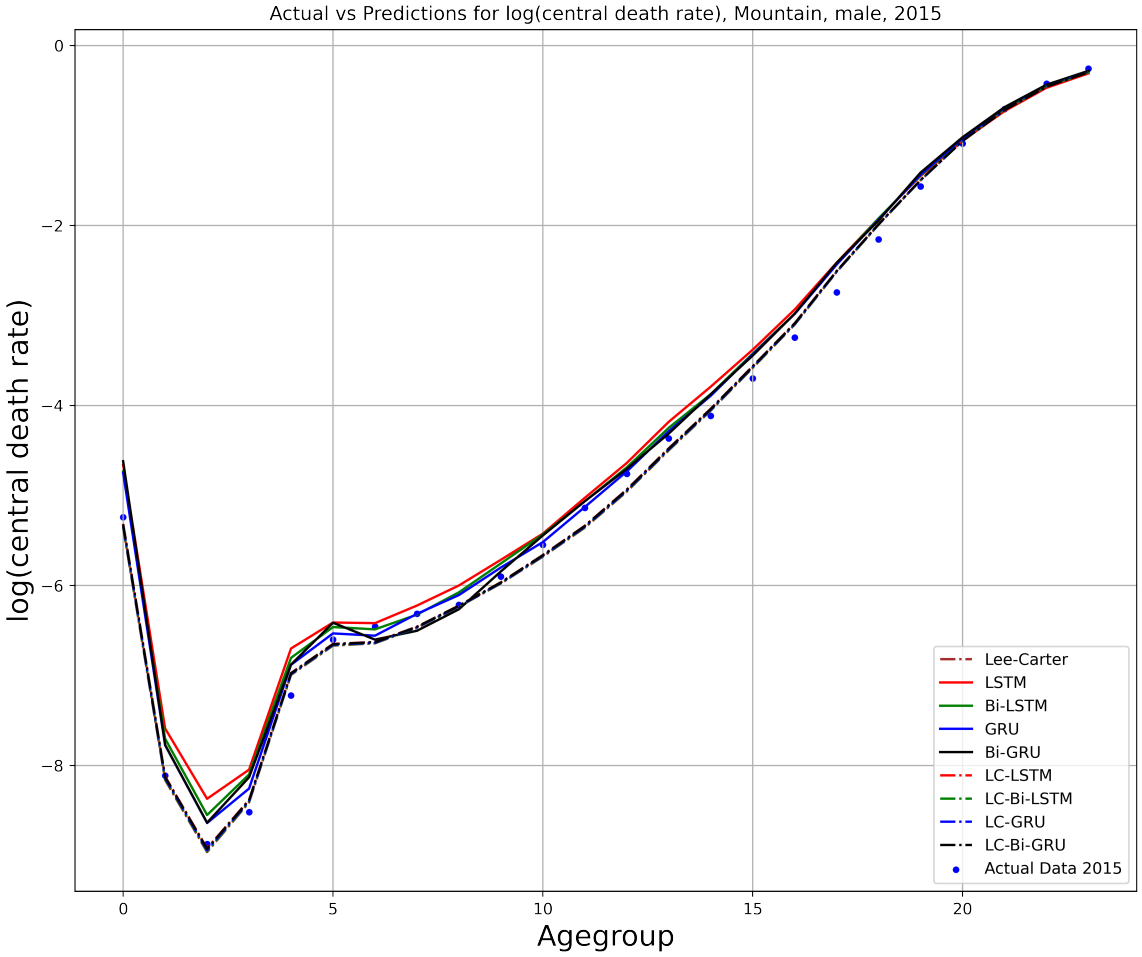

Figure 32: The predictions of log-mortality rate Mountain male population, 2015.

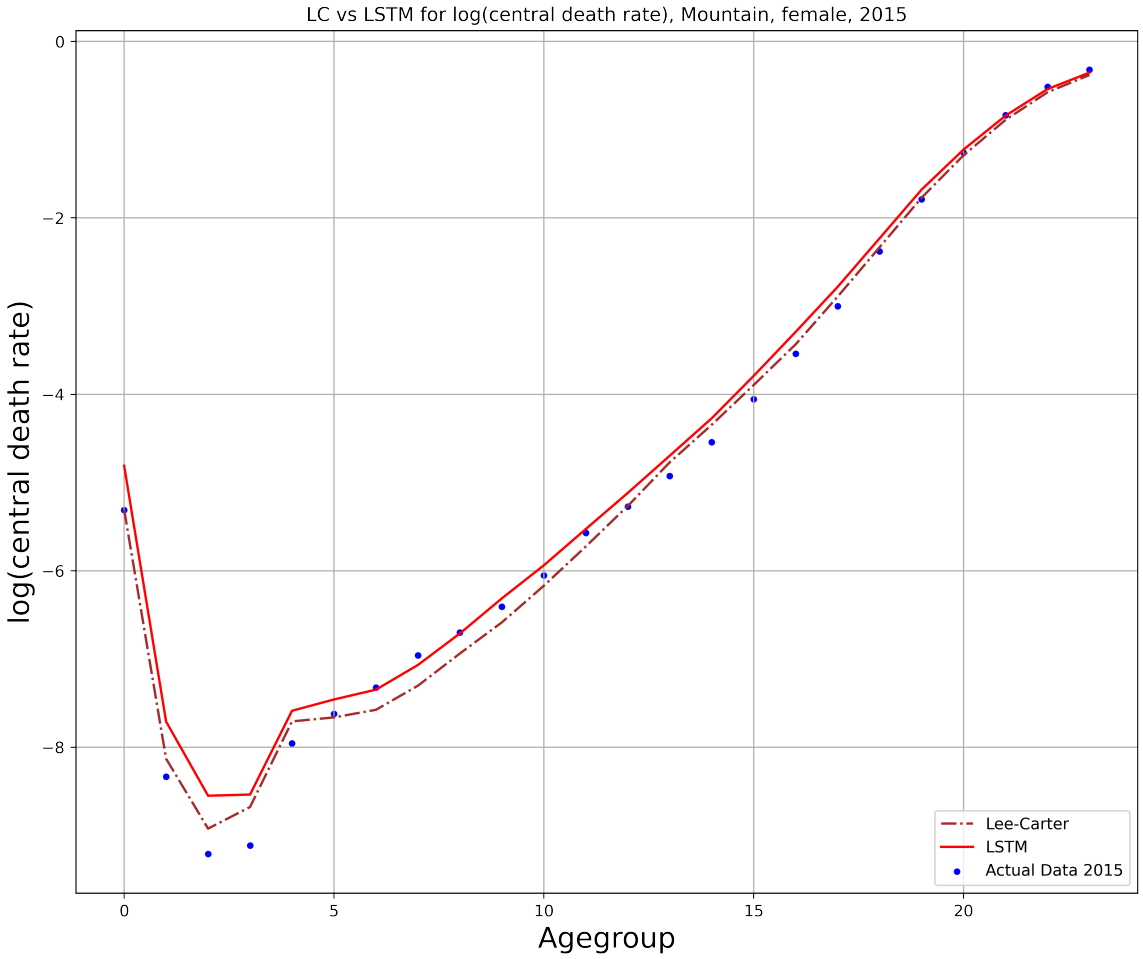

Figure 33: The predictions of log-mortality rate Mountain female population, LC vs LSTM, 2015.

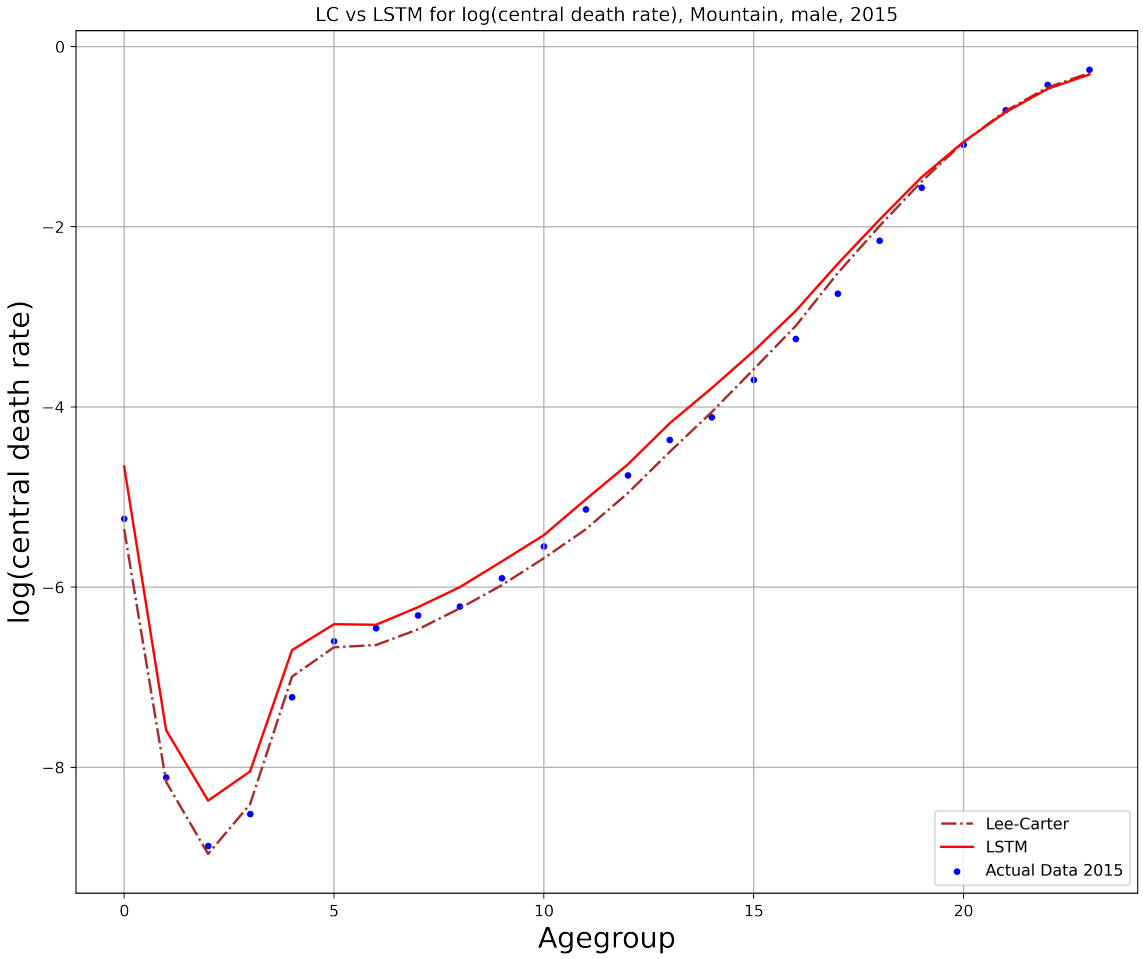

Figure 34: The predictions of log-mortality rate Mountain male population, LC vs LSTM, 2015.

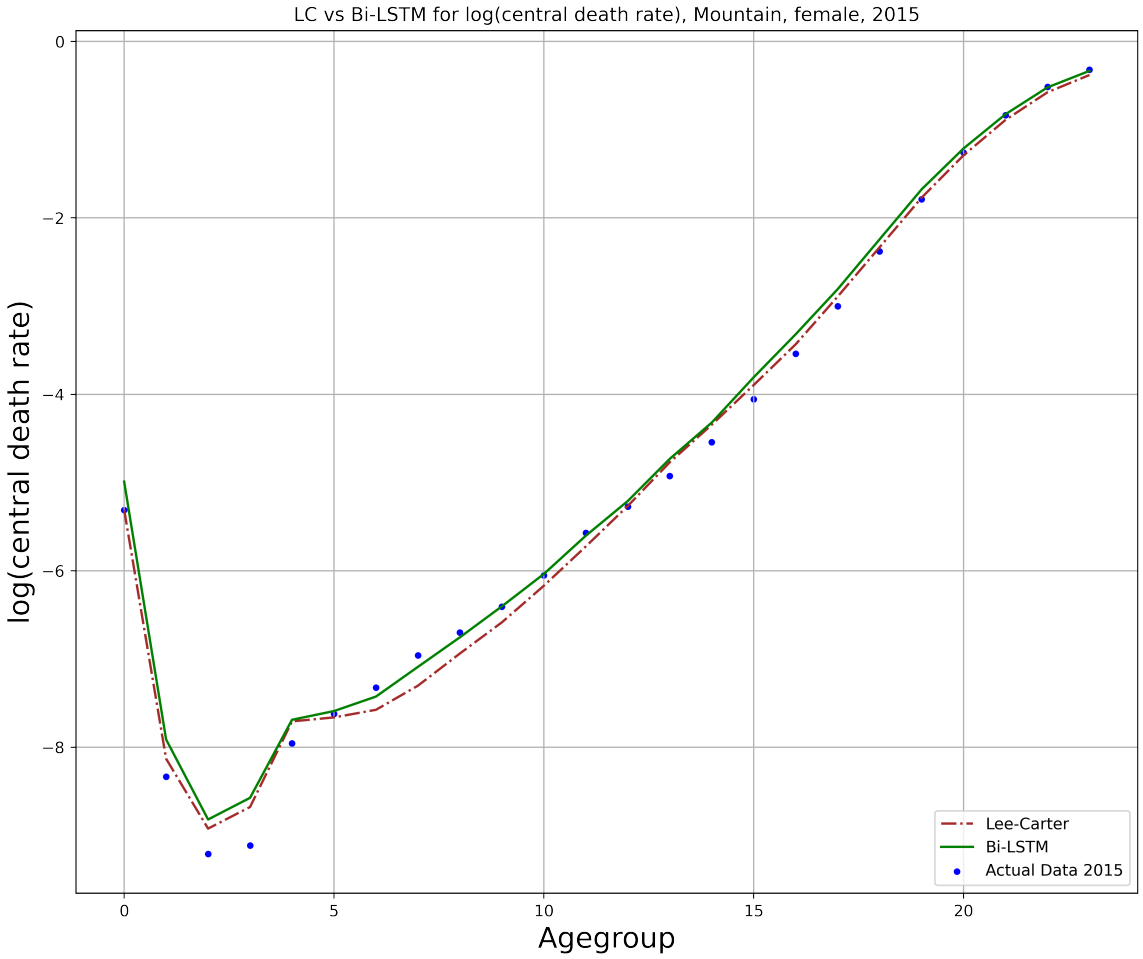

Figure 35: The predictions of log-mortality rate Mountain female population, LC vs Bi-LSTM, 2015.

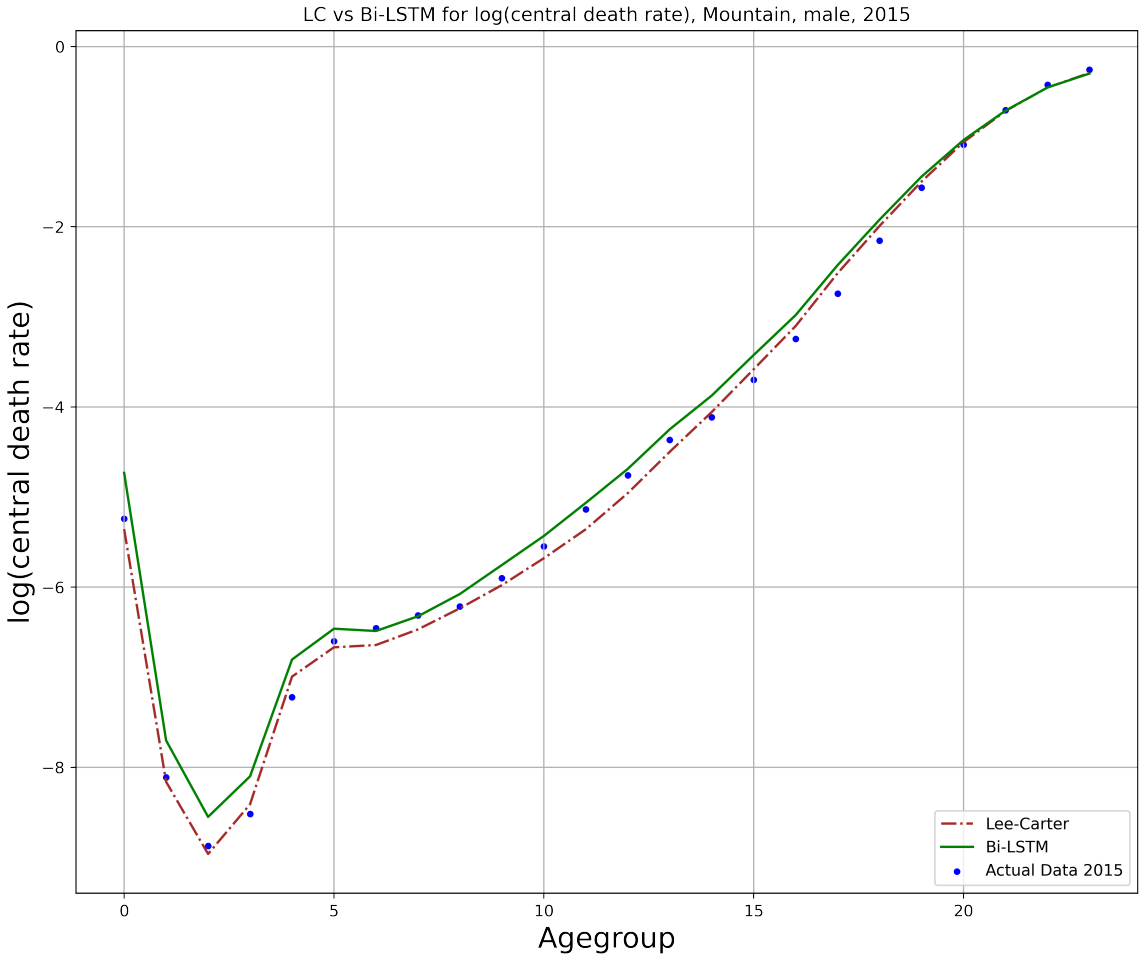

Figure 36: The predictions of log-mortality rate Mountain male population, LC vs Bi-LSTM, 2015.

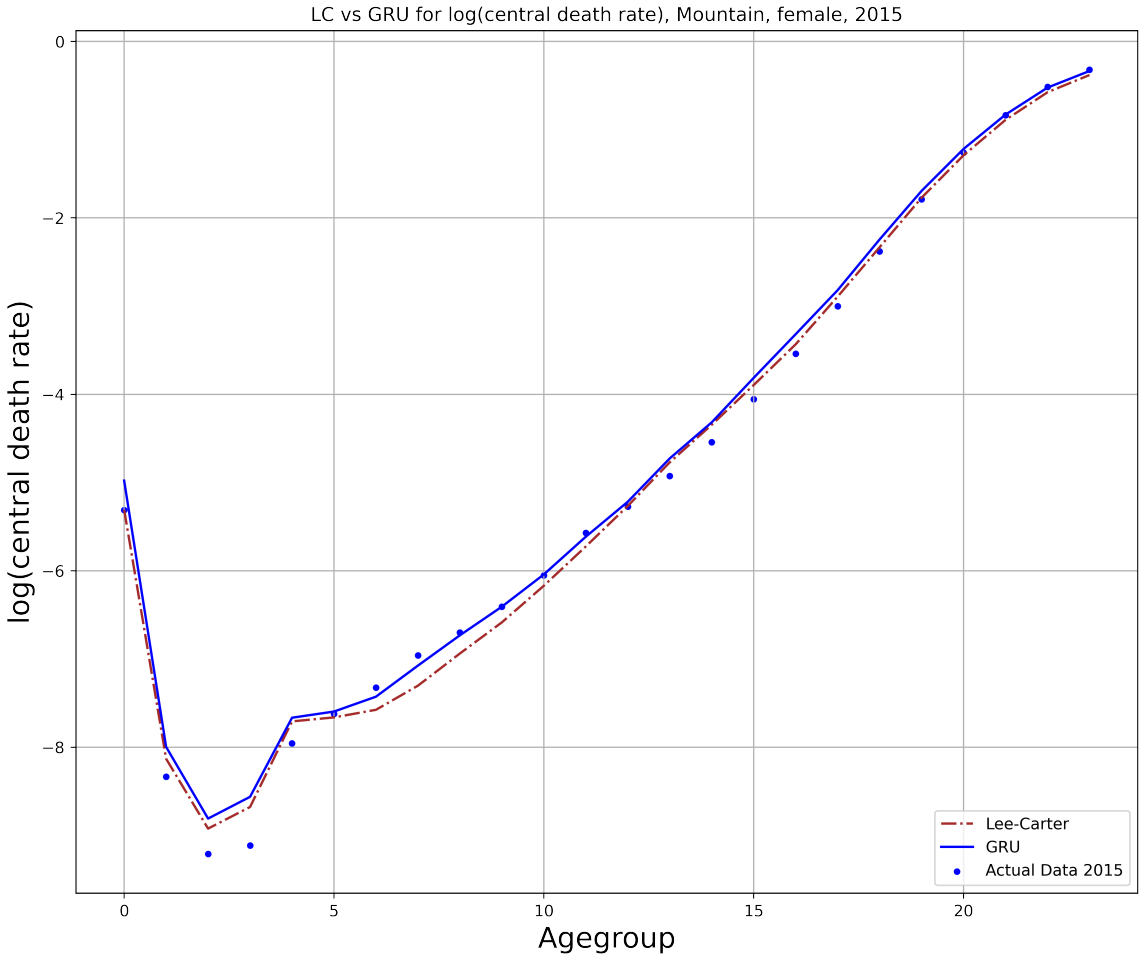

Figure 37: The predictions of log-mortality rate Mountain female population, LC vs GRU, 2015.

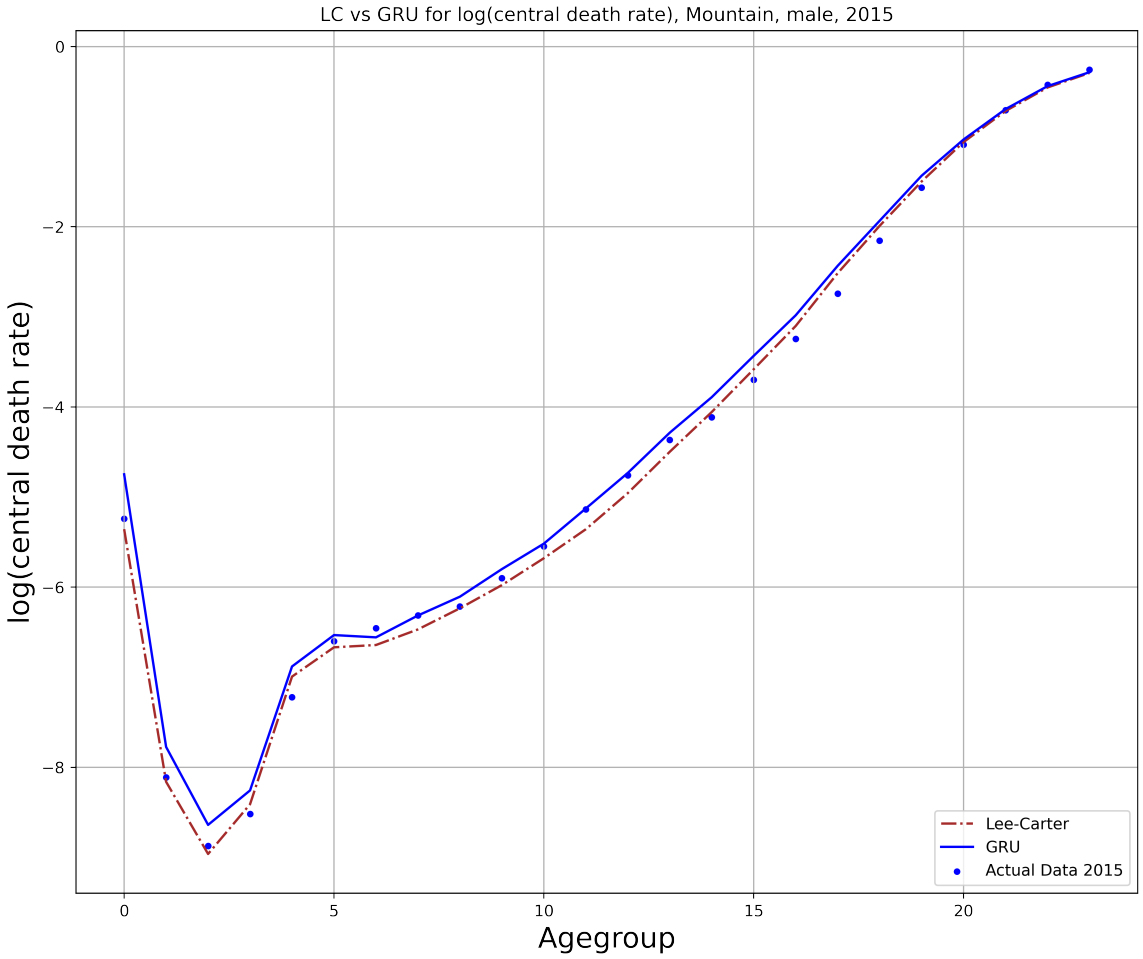

Figure 38: The predictions of log-mortality rate Mountain male population, LC vs GRU, 2015.

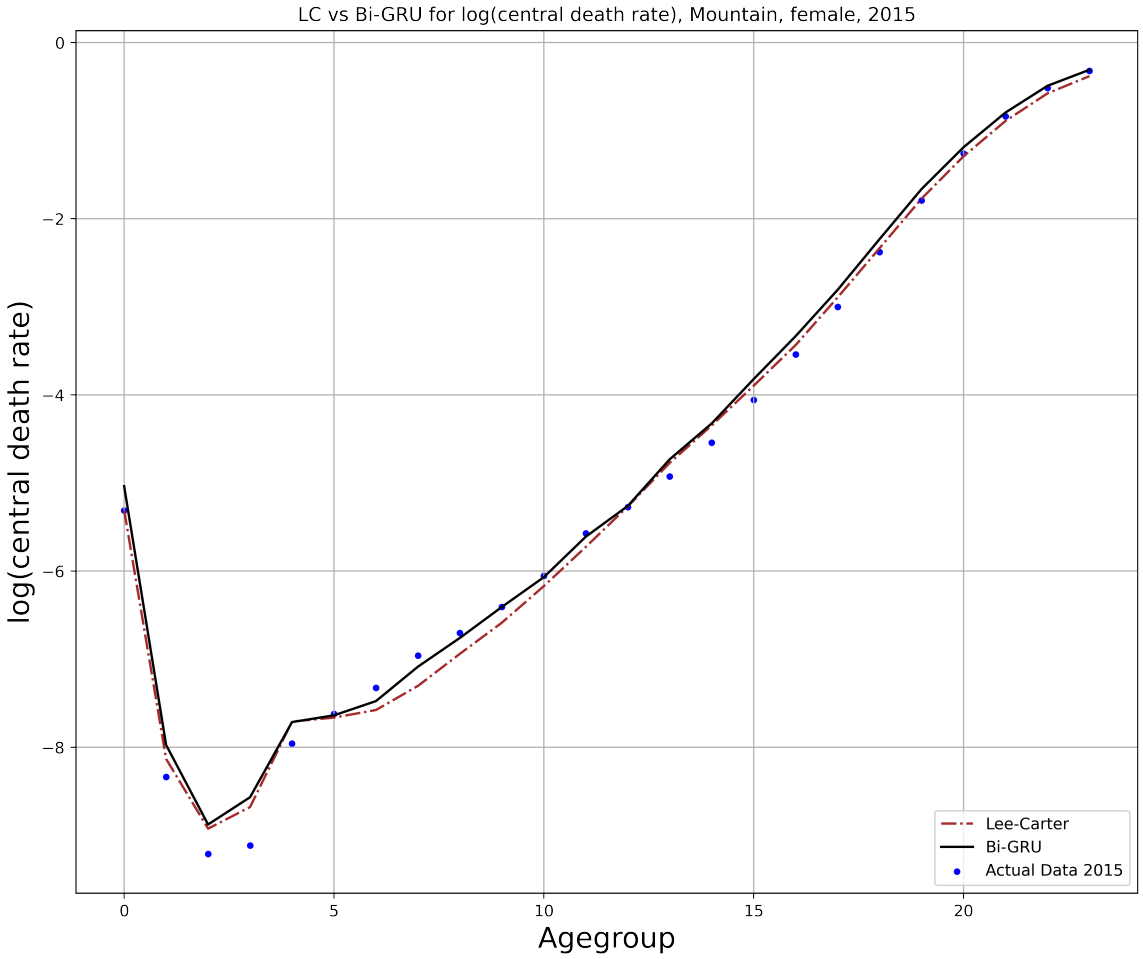

Figure 39: The predictions of log-mortality rate Mountain female population, LC vs Bi-GRU, 2015.

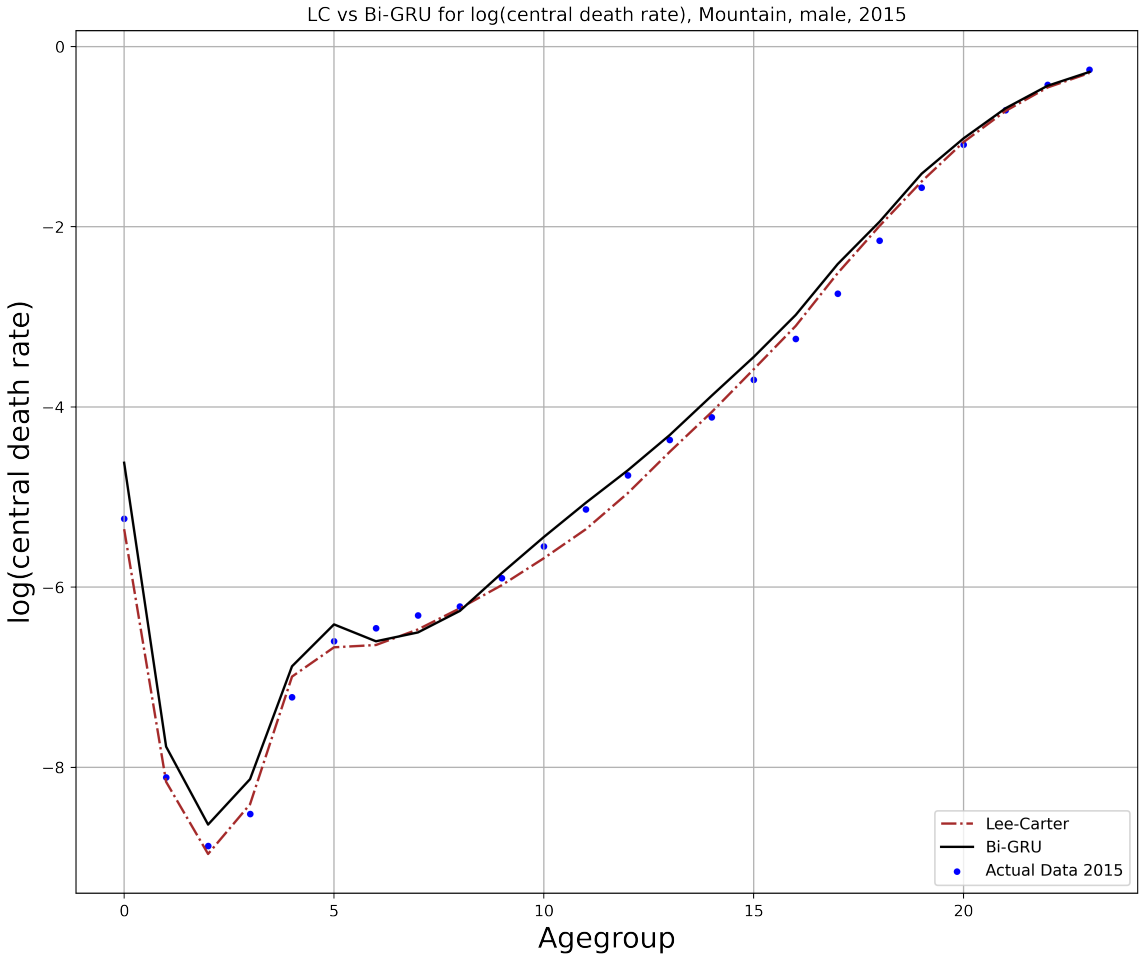

Figure 40: The predictions of log-mortality rate Mountain male population, LC vs Bi-GRU, 2015.

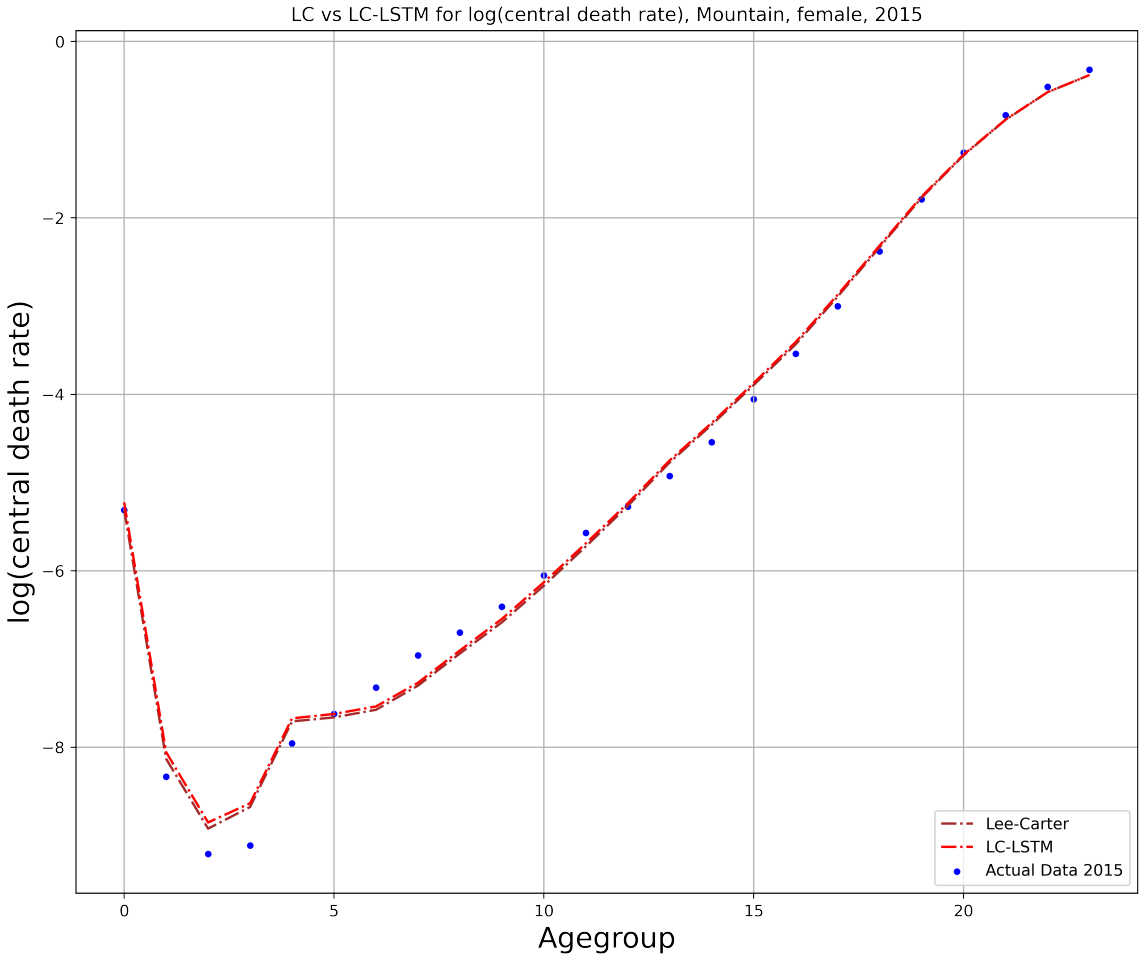

Figure 41: The predictions of log-mortality rate Mountain female population, LC vs LC-LSTM, 2015.

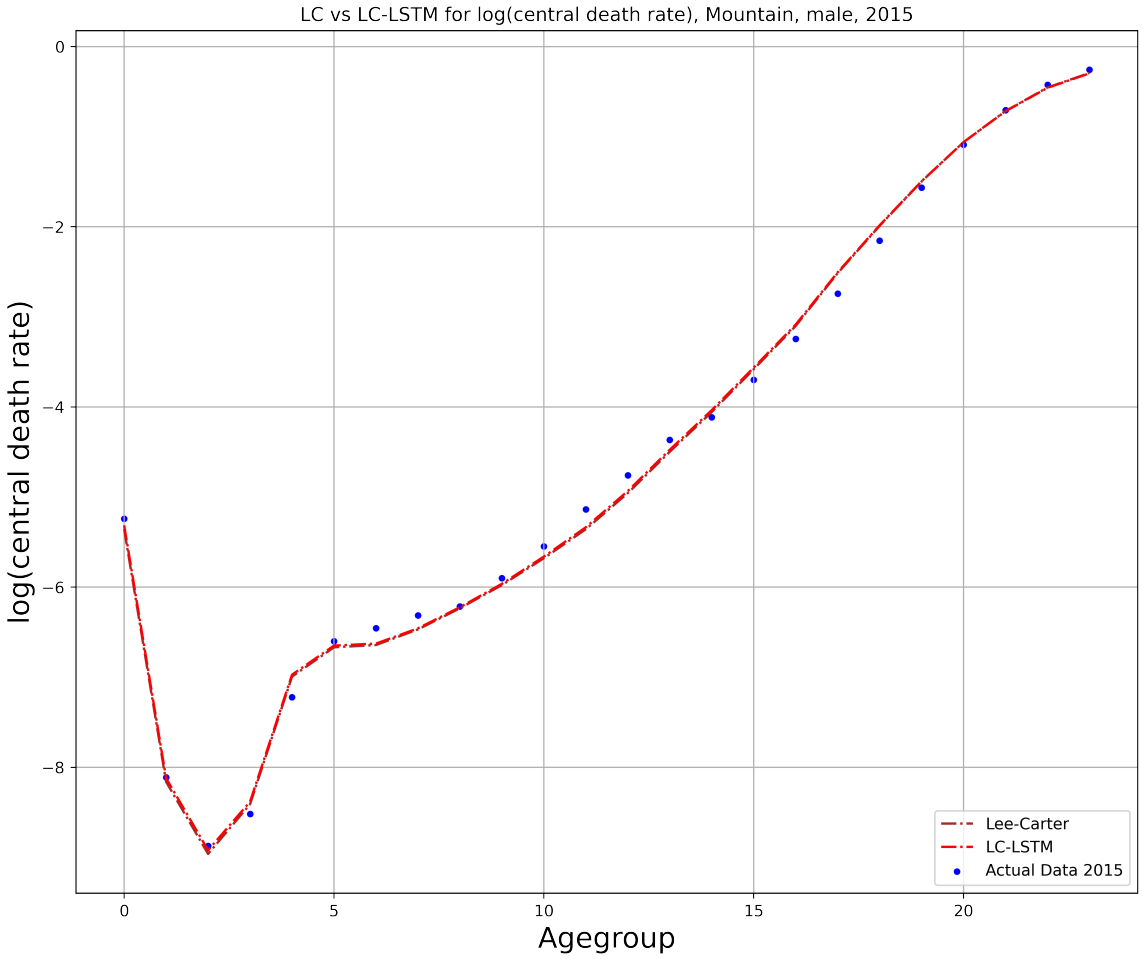

Figure 42: The predictions of log-mortality rate Mountain male population, LC vs LC-LSTM, 2015.

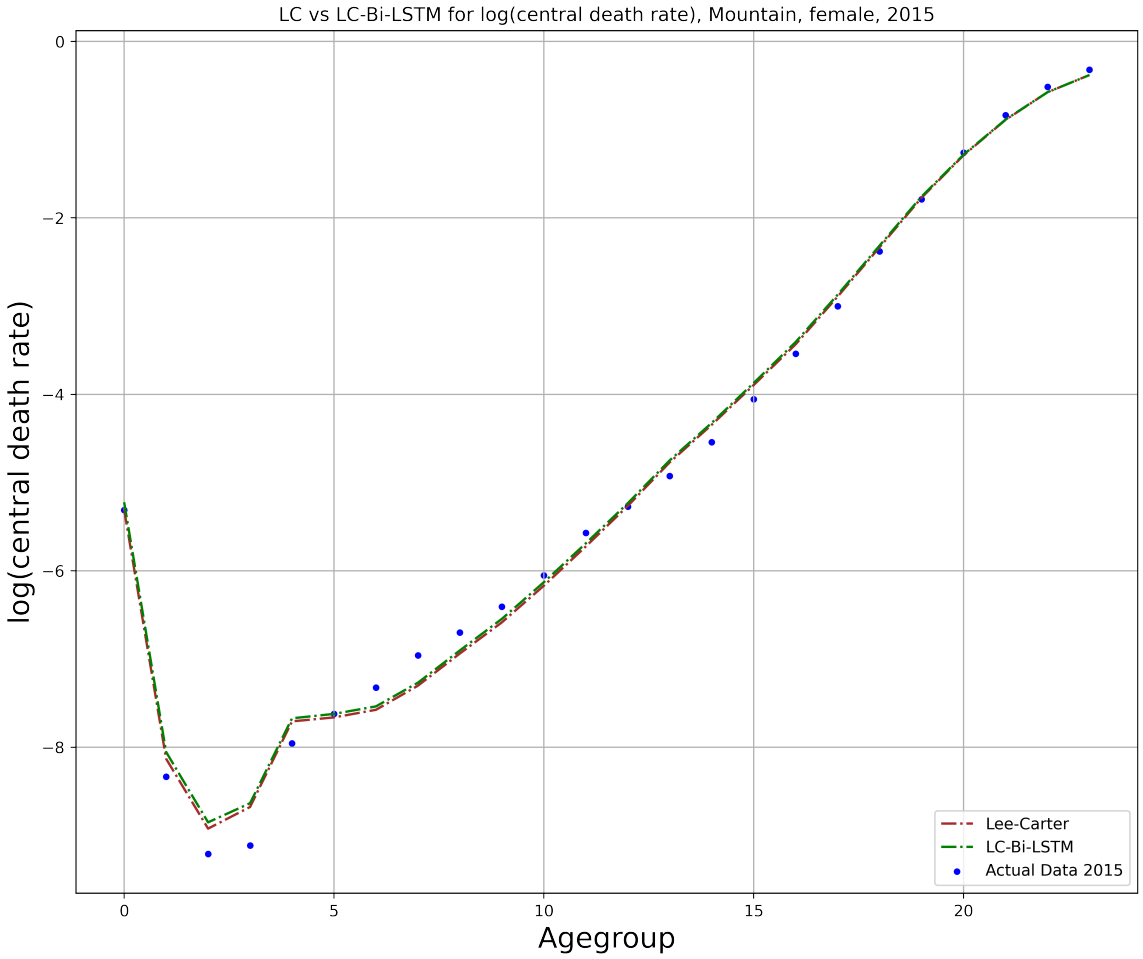

Figure 43: The predictions of log-mortality rate Mountain female population, LC vs LC-Bi-LSTM, 2015.

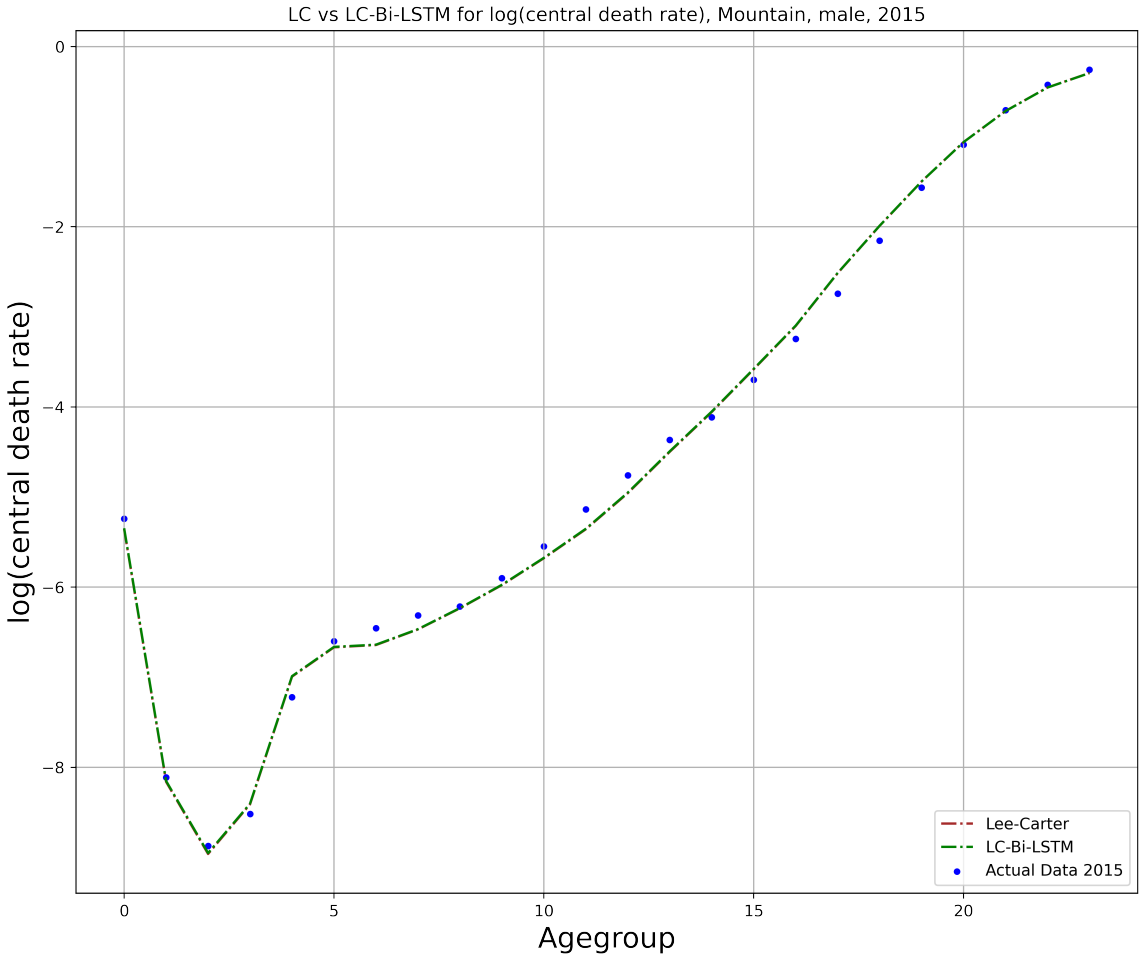

Figure 44: The predictions of log-mortality rate Mountain male population, LC vs LC-Bi-LSTM, 2015.

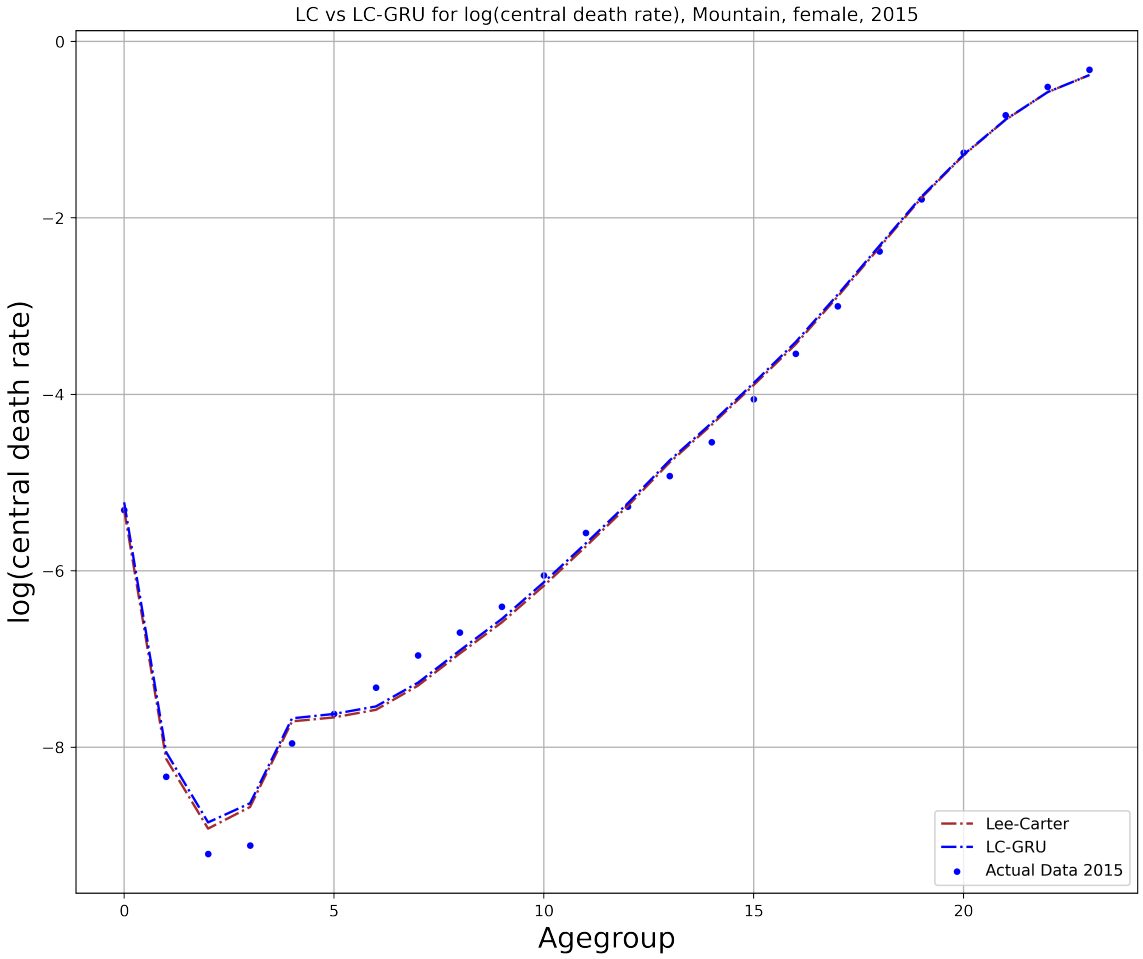

Figure 45: The predictions of log-mortality rate Mountain female population, LC vs LC-GRU, 2015.

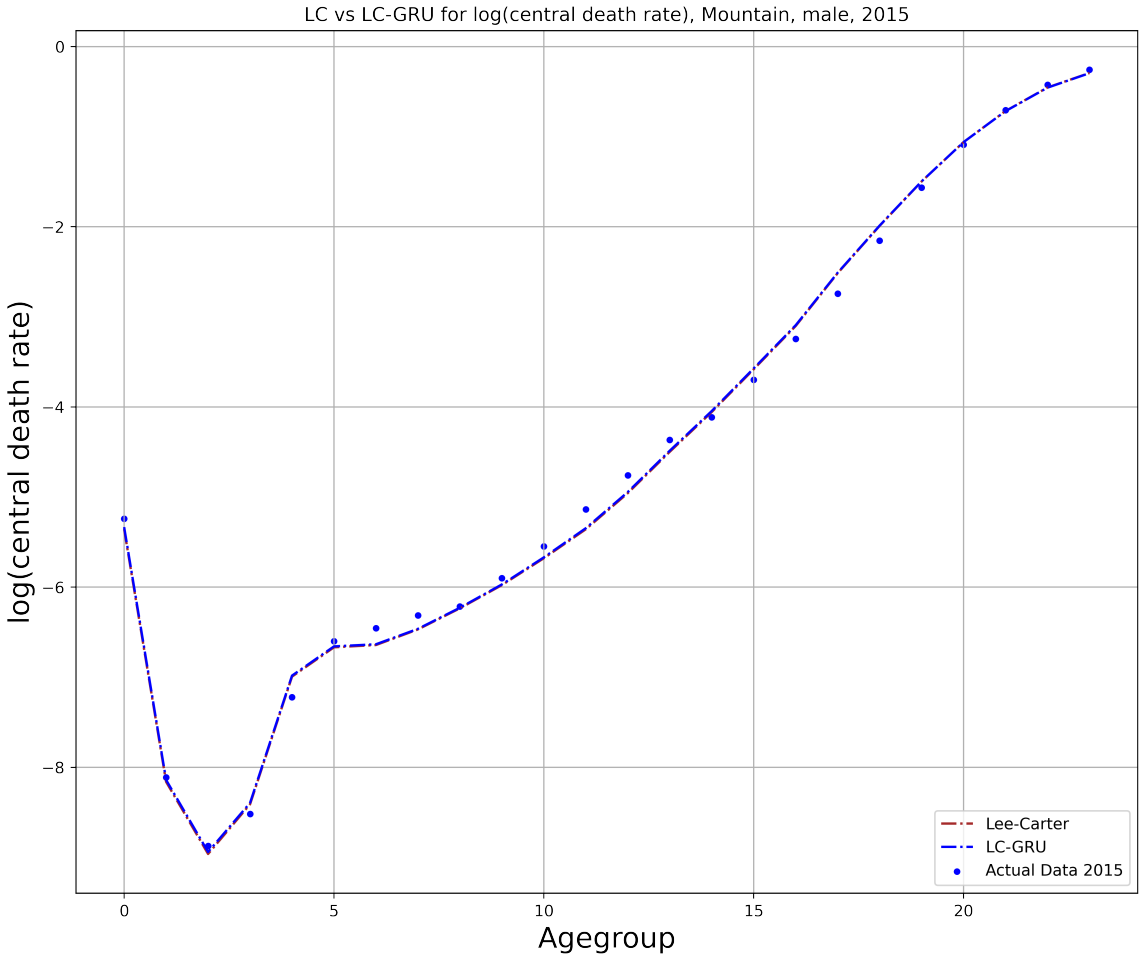

Figure 46: The predictions of log-mortality rate Mountain male population, LC vs LC-GRU, 2015.

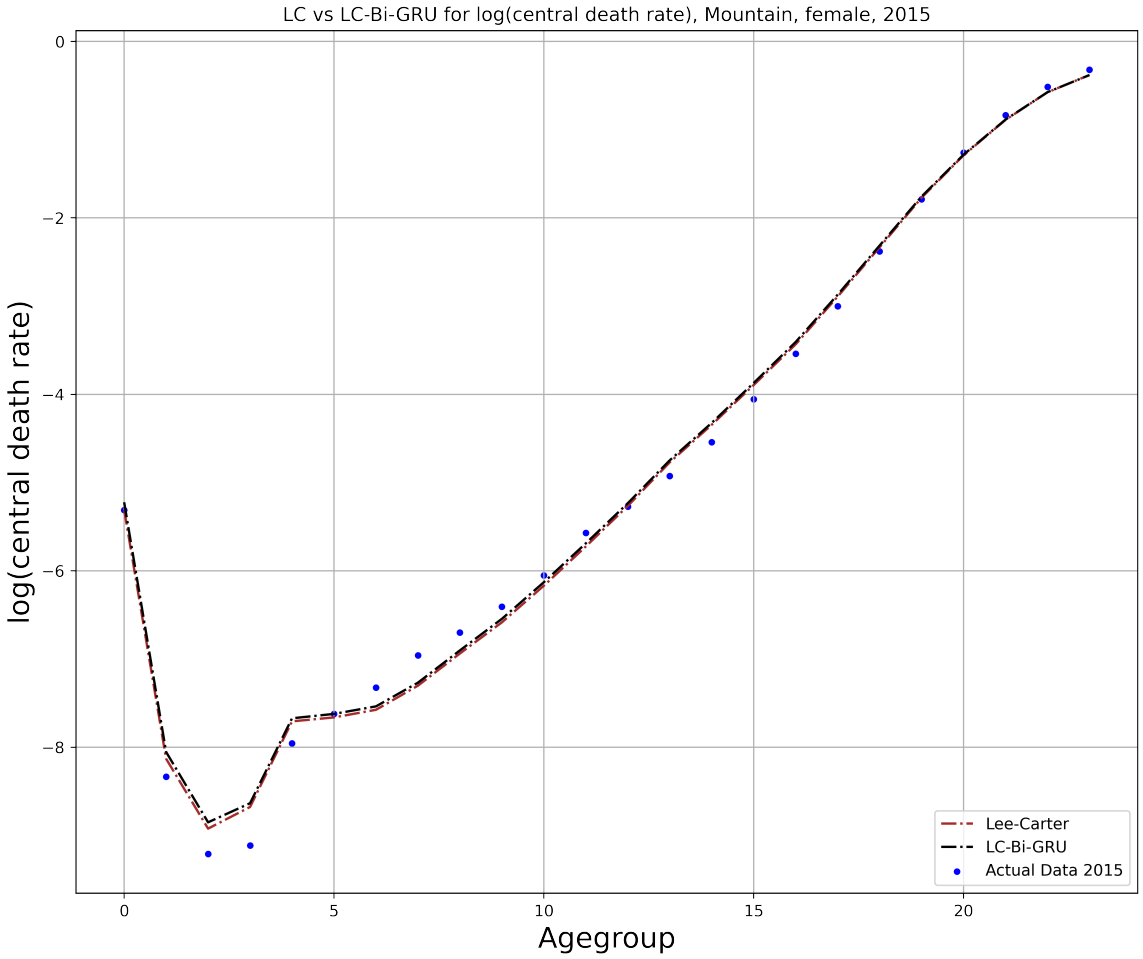

Figure 47: The predictions of log-mortality rate Mountain female population, LC vs LC-Bi-GRU, 2015.

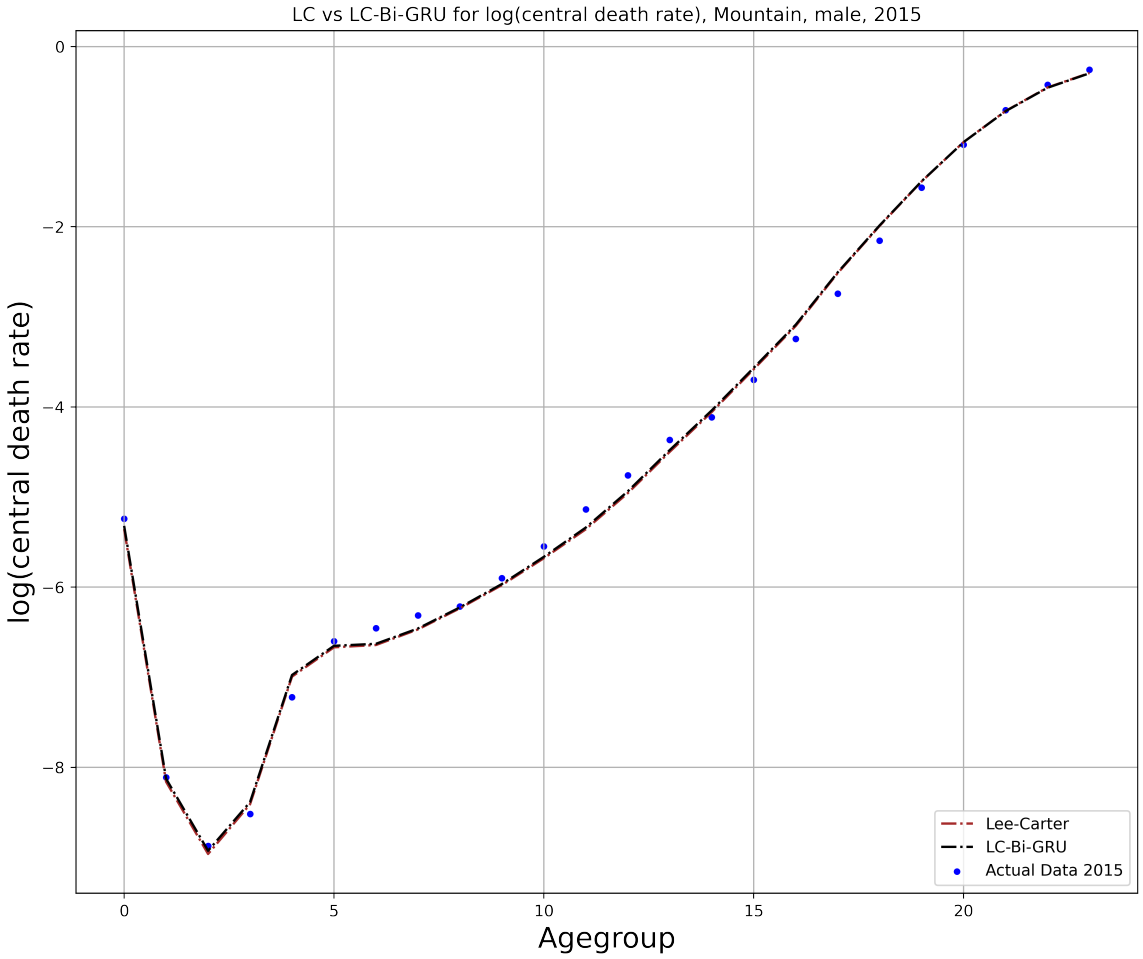

Figure 48: The predictions of log-mortality rate Mountain male population, LC vs LC-Bi-GRU, 2015.

CHAPTER 6

CONCLUSION

In this thesis, we introduced total of 8 models for mortality prediction, 4 RNNs models, including the LSTM model, Bi-LSTM model, GRU model and Bi-GRU model, 4 hybrid Lee-Carter models, including the LC-LSTM model, LC-Bi-LSTM model, LC-GRU model and LC-Bi-GRU model. These models were applied on the mortality data from 9 census divisions in the United States genders considering both genders. However, this thesis does not provide a comparison of the computational time required for the different models, as the hybrid model was implemented using both Python and R, while the deep learning models were implemented in Python. Therefore, we propose a comparison of the predictive performance of the models has been done based on the accuracy, precisely, 2 error metrics, MAE and RMSE. Among the deep learning models, the Bi-GRU model demonstrated the best prediction accuracy for this task. As for the hybrid Lee-Carter models, the LC-GRU model performed the best. Overall, the deep learning models yield significantly better predictive performance than the deep learning hybrid Lee-Carter models. This study can serve as a good reference for finding the most effective model for the mortality rate prediction in the United States.

However, it is not a straightforward conclusion that the deep learning models outperformed the hybrid Lee-Carter models on mortality prediction based on two error metrics.

There are two reasons that we still need hybrid Lee-Carter model:

(i) The hybrid Lee-Carter models offer improved model validation compared to deep learning models. The Lee-Carter model has been extensively studied, validated,

and utilized in mortality forecasting over many years. Its reliability and accuracy have been established through rigorous testing and analysis. By combining deep learning models with the well-established Lee-Carter model, the validation methodologies can ensure that the models produce not only accurate but also reliable estimations.

(ii)The hybrid Lee-Carter models offer better model explainability compared to deep learning models. Deep learning models often have complex architectures and generate results with inherent uncertainty, making it challenging to interpret the underlying factors driving the predictions. In contrast, the Lee-Carter model, with its well-defined methodology and framework, offers high level of explainability. The hybrid models' unique combination of interpretability and predictive power makes them into an excellent model for mortality forecasting, providing a comprehensive and insightful approach to analyzing and predicting mortality trends.

About the future work, there are several potential improvement in the current study. Firstly, in order to deal with the uncertainty by neural networks, which could be considered as the most significant challenge in the the application field of deep learning models. Some techniques could be applied to construct a prediction interval (such as bootstraps), instead of the point estimation in this study. Secondly, it would be valuable to investigate the application of neural networks for mortality prediction using data from other countries and regions in the human mortality database [10] or the United States mortality database [27]. This would provide a broader perspective on the effectiveness of the models. Thirdly, integrate of more stochastic models with deep learning techniques, such as Cairns-Blake-Dowd (CBD) model and its extensions. Lastly, expanding the repertoire of deep learning models for mortality rate prediction could lead to further advancements in this field.

The code for Lee-Carter model sample R file, deep learning model sample for

mortality rate forecasting and κ_t forecasting python files can be found on github at the following link: https://github.com/YuanChenmt/Thesis.git

BIBLIOGRAPHY

[1] Lee, R. and Carter, L., 1992. Modeling and Forecasting U.S. Mortality. Journal of the American Statistical Association, 87(419), pp.659-671.

[2] Cairns, A., Blake, D. and Dowd, K., 2006. A Two-Factor Model for Stochastic Mortality with Parameter Uncertainty: Theory and Calibration. Journal of Risk Insurance, 73(4), pp.687-718.

[3] Broun's, N., Denuit, M. and Vermunt, J., 2002. A Poisson log-bilinear regression approach to the construction of projected lifetables. Insurance: Mathematics and Economics, 31(3), pp.373-393.

[4] Hyndman R. J. Ullah M. S. (2007). Robust forecasting of mortality and fertility rates: a functional data approach. Computational Statistics Data Analysis 51(10), 4942–4956.

[5] Renshaw, A. and Haberman, S., 2006. A cohort-based extension to the Lee–Carter model for mortality reduction factors. Insurance: Mathematics and Economics, 38(3), pp.556-570.

[6] Deprez, P., Shevchenko, P. and Wüthrich, M., 2017. Machine learning techniques for mortality modeling. European Actuarial Journal, 7(2), pp.337-352.

[7] Levantesi, S. and Pizzorusso, V., 2019. Application of Machine Learning to Mortality Modeling and Forecasting. Risks, 7(1), p.26.

[8] Hainaut, Donatien. 2018. A neural-network analyzer for mortality forecast. Astin Bulletin 48: 481–508

[9] Francesca Perla, Ronald Richman, Salvatore Scognamiglio Mario V. Wüthrich (2021) Time-series forecasting of mortality rates using deep learning, Scandinavian Actuarial Journal, 2021:7, 572-598, DOI: 10.1080/03461238.2020.1867232

[10] Human Mortality Database. University of California, Berkeley (USA) and Max Planck Institute for Demographic Research (Germany). Available online: http://www.mortality.org

[11] Hong, W.H., Yap, J.H., Selvachandran, G. et al. Forecasting mortality rates using hybrid Lee–Carter model, artificial neural network and random forest. Complex Intell. Syst. 7, 163–189 (2021). https://doi.org/10.1007/s40747-020-00185-w

[12] Nigri, A., Levantesi, S., Marino, M., Scognamiglio, S. and Perla, F., 2019. A Deep Learning Integrated Lee–Carter Model. Risks, 7(1), p.33.

[13] Nigri, A., Levantesi, S. and Marino, M., 2020. Life expectancy and lifespan disparity forecasting: a long short-term memory approach. Scandinavian Actuarial Journal, 2021(2), pp.110-133.

[14] Marino, Mario and Levantesi, Susanna, Measuring Longevity Risk Through a Neural Network Lee-Carter Model (March 15, 2020). Available at SSRN: https://ssrn.com/abstract=3599821 or http://dx.doi.org/10.2139/ssrn.3599821

[15] Richman, Ronald and Wuthrich, Mario V., Lee and Carter go Machine Learning: Recurrent Neural Networks (August 22, 2019). Available at SSRN: https://ssrn.com/abstract=3441030 or http://dx.doi.org/10.2139/ssrn.344103 .

[16] Castellani, Gilberto, Ugo Fiore, Zelda Marino, Luca Passalacqua, Francesca Perla, Salvatore Scognamiglio, and Paolo Zanetti. 2018. An investigation of Machine Learning Approaches in the Solvency II Valuation Framework. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract$_id = 3303296$

[17] Petneházi, G. and Gáll, J., 2022. Mortality rate forecasting: can recurrent neural networks beat the Lee-Carter model?. [online] arXiv.org. Available at: ¡https://arxiv.org/abs/1909.05501¿

[18] Richman, Ronald and Wuthrich, Mario V., A Neural Network Extension of the Lee-Carter Model to Multiple Populations (October 22, 2018). Available at SSRN: https://ssrn.com/abstract=3270877 or http://dx.doi.org/10.2139/ssrn.3270877

[19] Gabrielli, Andrea, and Mario V Wüthrich. 2018. An Individual Claims History Simulation Machine. Risks 6: 29.

[20] Petneházi, G. and Gáll, J., 2022. Mortality rate forecasting: can recurrent neural networks beat the Lee-Carter model?. [online] arXiv.org. Available at: ¡https://arxiv.org/abs/1909.05501¿

[21] Chen, Y.; Khaliq, A.Q.M. Comparative Study of Mortality Rate Prediction Using Data-Driven Recurrent Neural Networks and the Lee–Carter Model. Big Data Cogn. Comput. 2022, 6, 134. https://doi.org/10.3390/bdcc6040134

[22] Hyndman, Rob John, and Yeasmin Khandakar. 2008. Automatic time series forecasting: the forecast package for R. Journal of Statistical Software 26: 1–22.

[23] 23.Hocreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8):1735-1780.

[24] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078

[25] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

[27] United States Mortality DataBase. University of California, Berkeley (USA). Available at usa.mortality.org

[28] Bergeron-Boucher, M.-P., Canudas-Romo, V., Oeppen, J. E., and Vaupel, J. (2017). Coherent forecasts of mortality with compositional data analysis. Demographic Research, 37(17):527–566.

[29] Booth, H., Hyndman, R. J., Tickle, L., and De Jong, P. (2006). Lee-Carter mortality forecasting: a multi-country comparison of variants and extensions. Demographic Research, 15:289–310.