Forecasting small area populations with Long Short-Term Memory Networks

Irina Grossman[1]

Tom Wilson [2]

Jeromey Temple[3]

1. Corresponding author. Address: Melbourne School of Population and Global Health, The University of Melbourne, 207 Bouverie St, Melbourne, Vic 3010, Australia.
Email: Irina.grossman@unimelb.edu.au

2. Melbourne School of Population and Global Health, The University of Melbourne, Australia. Email: wilson.t1@unimelb.edu.au

3. Melbourne School of Population and Global Health, The University of Melbourne, Australia. Email: jeromey.temple@unimelb.edu.au

Key words

Population forecasts; small area; Long short-term memory; Australia

Abstract:

Local and state governments depend on small area population forecasts to make important decisions concerning the development of local infrastructure and services, including schooling, transportation, healthcare, energy, telecommunications, and water supply. Despite their importance, current methods often produce highly inaccurate forecasts, especially at the small area scale. Recent years have witnessed promising developments in time series forecasting using Machine Learning across a wide range of social and economic variables. However, almost no work has been undertaken to investigate their potential application in demography, particularly for small area population forecasting. In this paper we describe the development of two Long-Short Term Memory network architectures for work with small area populations. We employ the Keras Tuner to select layer unit numbers, vary the window width of input data, and apply a double training and validation regime which supports work with short time series and prioritises later sequence values for forecasts. These methods are transferable and can be applied to other data sets. Retrospective small area population forecasts for Australia were created for the periods 2006-16 and 2011-16. Model performance was evaluated against actual data and two benchmark methods. The deep learning methods produced forecast errors comparable to the benchmark methods, markedly improving the 2006-based forecasts, but not 2011-based forecasts produced by one of the benchmark methods.

1. Introduction

1.1 Small Area Population Forecasting

Small area population forecasts are widely used by government and business for many purposes. They inform decisions about the construction and allocation of funding for roads, schools, aged care services, transport, and health services, amongst other uses. Yet small area population forecasts tend to be far more prone to error than forecasts at larger geographies. The smaller the population, the larger the error tends to be, particularly for populations of just a few thousand people (Wilson et al., 2018). In part, this is because data sets for small areas often have short time series, less detailed data, data quality issues, and noisy patterns. But it is also due to the limited amount of research dedicated to small area population forecasting compared to national-level forecasting. In recent years one of the key areas for methodological developments in time series forecasting has occurred in Machine

Learning (ML). However, literature investigating the development and evaluation of these methods on small area population datasets remains limited.

1.2 Artificial Neural Networks and Long Short-Term Memory models

Recent years have seen a surge in research relating to the use of machine learning methods, for time series forecasting[1]. Much of the research in the machine-learning forecasting literature is based on the use of artificial neural networks. These neural networks were originally inspired by the structure of the brain, where an input is mapped to an output as it travels through a series of interconnected neurons organised in layers. Each neuron (or unit) receives weighted inputs from connected units, which are transformed through the unit's architecture to produce an output. This output is then sent to the next set of neurons, or combined with outputs from other neurons, and transformed into the network's predicted value. For supervised learning problems - where there are exemplars of inputs and outputs - training refers to the assignment of weights such that the error produced when feeding the network exemplar inputs is minimised relative to the exemplar outputs. Then when a new input is presented, the network can produce an output based on what it was trained to expect from the exemplars.

Early neural networks did not have memory. They could only see and react to the most recent input, and subsequently could not consider context in their predictions. Early versions of Recurrent Neural Networks (RNNs) began to solve this problem by including "hidden states," loops that would take the output from a unit, weight it, and then feed it back into the unit at the next time step (Rumelhart et al., 1986). However, as information from past time steps are assigned a weight each time they propagate backwards in time, the impact of past inputs tends to decrease to zero or increase exponentially. Consequently, the limitations of these early RNNs include their ability to only learn from the immediate past, and subsequently their inability to capture and use long term dependencies for their forecasts. Long Short-Term Memory (LSTM) networks are a relatively newer type of RNN architecture which partially solves this problem (Gers et al, 1999; Hochreiter & Schmidhuber, 1997). Each unit in this network includes a cell which can hold information for arbitrary lengths of time. Gates within the units decide what should be forgotten and what should be added to the

[1] A Scopus search of research documents containing (forecast* and "machine-learning" or "deep learning" or "artificial neural networks") revealed just four documents in 1990, 172 in 2000, and almost 12,000 by 2020.

cell state. Information from the cell state can then be forwarded to other neurons, the network's output, or fed back into itself. This allows LSTMs to consider longer term dependencies and patterns when making predictions.

1.3 The art of developing Neural Networks and implications for population forecasting

There are an infinite number of ways in which a neural network can be built. The network is created in layers, where each layer can be considered as a set of functions which take an input, transform it, and forward it to the next layer, or produce an output. When a neural network has multiple layers, it can be referred to as Deep Learning, a subfield of Machine Learning. Practitioners creating neural networks have many different types of layers to choose from. One of the most common layer types used is the "Dense" layer, so called because each unit within it receives connections from each of the previous layer's units (Huang et al. 2017). Other layer options include LSTMs and other RNNs. There are multiple other characteristics that need to be selected when building neural network architectures, including the number of units that exist in each layer, modification of the functions which transform inputs within each of the layers, and the algorithm which searches for the best weights to model minimise error.

Automated tuning methods - such as the Keras Tuner used in this study (O'Malley et al., 2019; TensorFlow, 2019) - can be used to support the selection of some of these characteristics (called "hyperparameters"). However, the architecture and use of the methods remains a property of the skill, experience, and goals of the practitioner. It is important to emphasise the subjectivity of these types of methods as they begin to become more prominent in demography. Indeed, just as conventional forecasting techniques used by demographers are often considered to be as much of an art as a science, the same is true of machine learning methods. However, unlike conventional demographic methods, machine learning methods are often black boxes whose inner working are difficult to explain.

As ML methods begin to be introduced in demography, great care needs to be taken to ensure that any such methods which are applied for real world forecasting have been shown to produce significant, reproducible benefits over existing methods. A cautious approach is necessary. There has been some research suggesting that ML methods are often less accurate than traditional statistical methods on data from a range of domains. Makridakis et al. (2018) set out to evaluate the accuracy of forecasts produced by ML methods (including LSTM

networks, Bayesian Neural Networks, and CART regression trees, and several others) against traditional statistical methods (these included ETS, ARIMA, and an average of Simple exponential smoothing, Holt exponential smoothing, and Damped exponential smoothing). Their evaluation was carried out using monthly time series from the M3 competition, which were from multiple domains (microeconomic, macroeconomic, industry, finance, and demographic); these were the same time series as used in Ahmed et al. (2010). They found that traditional methods were more accurate and had lower computational requirements than the ML methods, for multiple tested forecasting horizons. The authors suggest that the reason why many articles claim superior ML performance is that they do not include a comparison to a suitable benchmark method (particularly a statistical one).

1.4 Recent research on using machine learning methods for small area population forecasting

Limited research has been undertaken on the development and evaluation of ML methods for population forecasting, and very little at the small area scale (Wilson et al. 2021a). Riiman et al. (2019) compared 10 year population forecasts for Alabama counties produced by small LSTM neural network models with those generated from a cohort-component model. The cohort-component model benchmark produced a Mean Absolute Percentage Error (MAPE) of 6.5% for a 10 year forecast. The LSTM networks used were relatively simple, each containing a LSTM layer with four or five units and a dense layer. They tested these models with two data types: decennial census counts and annual mid-year population estimates. Models trained on the decennial census counts were only required to produce a single 10 year interval forecast; mid-year population estimates required 10 annual outputs to produce a 10 year horizon forecast. Both data sources were used to train two types of models: single-area models, where forecasts for each area were made using a model trained only on that area's time series, and full-dataset models, where models were trained using data from all areas. Single-area model forecasts using census counts produced their best results (MAPE: 5.0%). Forecasts using a full-dataset model trained on decennial data also performed well (MAPE: 6.1%). For mid-year population estimates, forecast error after 10 years using a full-dataset model were poor (MAPE: 16.7%), whilst single-area models produced a MAPE comparable to their cohort component model benchmark (MAPE: 6.3%). These results were promising: the LSTM networks out-performed the cohort-component models despite requiring only

population totals as input, as opposed to the more detailed age-sex data on population, mortality and migration required by the cohort-component model. It is important to note that the time series lengths in the Alabama county population datasets is relatively long; decennial census data were from 1910 and mid-year population estimates from 1969. It is common for small area datasets to have shorter time series.

The better model performance in Riiman et al. (2019) with the use of decennial data is not surprising as the evaluated forecast only required one step. Error often accumulates with each step, making it more difficult to produce accurate multi-stepped forecasts. However, real-world forecasting often requires multiple-steps into the future (Makridakis et al., 2018), and forecast values for each year in a 10 year horizon are often required in practice. It is also important to note that reporting averaged error metrics, such as MAPE, for a set of small area time series is likely insufficient for the evaluation of ML forecasts; these models can often produce errors in unexpected ways. Additional metrics should also be used which give some indication of the number of forecasts with very large errors. Furthermore, the benefit of ML methods often lies in their ability to consider more data than traditional statistical methods. Therefore, it is sensible to develop ML methods which are trained on whole datasets.

The Riiman et al. (2019) study is the closest existing work to the research presented in this paper. Research investigating ML methods for small area population forecasting has been limited. Chen et al. (2020) used several ML methods (XGBoost, random forest and a neural network) to forecast built-up land expansion and populations at the grid cell level (100m^2) for China over the 2015 to 2050 period. They used high resolution maps and spatial data (such as distance to city centre) as data inputs, and constrained forecasts to the Shared Scenario Pathways, which present several demographic scenarios based on different climate change outcomes (O'Neill et al., 2017). Other demographic work using ML methods includes forecasting the components of small area population change, such as migration (Weber, 2020), and population age structures (Striessnig et al., 2019). However, the literature is very limited and not enough work has been undertaken to allow any of the methods to be recommended to practitioners.

1.4 Aims of the study

The aims of this study were to (1) investigate and develop deep learning methods suitable for small area population forecasting, and (2) characterise the performance of these methods

using an Australian small area population dataset. We consider two deep learning model architectures with varying windows of input data and automated tuning methods, and present a training and validation scheme which supports work with small area datasets. Five and 10 year forecasts were then compared against two conventional demographic methods proven to work well on such datasets.

2. Data and methods

2.1 Population data

Population data was obtained for the longest time series available on a consistent set of geographical boundaries for Statistical Area Level 2 (SA2) areas, the smallest official spatial unit for which Estimated Resident Populations (ERPs) were published (ABS, 2017). The data consists of annual ERP totals for the period 1991 to 2016 (ABS, 2017) based on 2011 geographical boundaries. The median 2016 SA2 population in Australia was 9,681 with 95% lying within the range 2,559 to 29,279. SA2 areas with populations <100 in any of the years of the fitting period were excluded from the training of models. These populations were combined into a 'remainder' area, which was excluded from training, but included in the predictions so that forecasts could be constrained to the national forecast. National population forecasts were required as data inputs or constraints to the small area forecasts. We used the main series forecasts produced by the ABS which was closest in date to the 2006 and 2011 jump-off years of the forecasts (ABS 2008, 2013a).

2.2 Data pre-processing

ML methods converge faster when data is scaled: fewer trials were required to find the parameters which produce a minimum error when the values fed into the model were on the same scale. For this study, we scaled data using the min-max method using the maximum and minimum values from the entire training set.

$$x' = \frac{x - \min(d)}{\max(d) - \min(d)} \quad (1)$$

where x is the original value, x' is the new value, and $\max(d)$ and $\min(d)$ were the maximum and minimum values in the training set, respectively.

We created forecasts from 2006 to 2016 and 2011 to 2016. ERPs past the jump off years were withheld from training and used only to evaluate the results. The sliding window technique was used to create the training data. The technique breaks a time series up into multiple exemplars which demonstrate what a forecast can look like for input sequences with specific window widths. For example, consider the time series {0, 5, 10, 15, 20, 25} which is converted into exemplars using a sliding window of width three. This time series indicates that an input sequence of{0, 5, 10} produces an output of {15}; an input of {5, 10, 15} produces an output of {20}; and an input of {10, 15, 20} outputs {25}. The sliding window approach turns the time series into a supervised learning problem. During training, different sets of model parameters were tried such that when the input sequences were entered into the model the error of the predicted output is minimised when compared to the actual output. We tested input windows widths of 5, 8, and 11 years to investigate the sensitivity of forecast accuracy to window width.

2.3 Error Metrics

We make use of several types of error metrics. Percentage error (PE) is defined as:

$$PE = \frac{(F-A)}{A}\ 100\% \qquad (2)$$

where F denotes the forecast value, A the actual value.

The distributions of percentage errors from population forecasts tend to have a long tail. These outlier values influence the Mean Absolute Percentage Error (MAPE). For this reason, we utilise the **Median Absolute Percentage Error** (MedAPE) to indicate average error. We present the MAPE in Table 3 to allow readers to compare our results with those presented in other papers which do use the MAPE metric, although we do not discuss the MAPE results.

Whilst minimising an averaged error metric across small areas is important, this is insufficient for producing a set of good forecasts. It is also important to reduce the number of bad forecasts. To evaluate the number of bad forecasts by our models we define the **Percentage of Bad Forecasts** (Wilson et al. 2021b) as the percentage of forecasts which have an Absolute Percentage Error greater than 10% after 5 years, or greater than 20% after 10 years. These values were selected because they are larger than the levels of error acceptable to population forecast users surveyed by Wilson and Shalley (2019).

The distributions of the percentage errors were visualised using Python 3.8.5 using the Seaborn (Waskom, 2021) and Matlplotlib (Hunter, 2007) packages. We used box plots to show the outliers present in both the benchmark and machine learning forecasts. Violin charts were used to visualise the distributions of percentage errors; for these plots percentage errors with an absolute value greater than 25% were removed from the visualisations.

Because the accuracy of a forecast for a small area may be influenced by its remoteness and population size, we considered these factors in our study. Remoteness can be considered a proxy indicator of the type of demographic regime experienced by an area (e.g., high immigration and young adult in-migration in the largest cities; low immigration and high out-migration of young adults in regional and remote regions). We used the ABS's remoteness area classification which classifies areas as Major Cities, Inner Regional, Outer Regional, Remote or Very Remote (ABS, 2013b). The number of SA2 areas included in each category after removal of areas with <100 people are: 1,199 SA2s in Major Cities, 476 Inner Regional SA2s, 306 Outer Regional SA2s, 47 Remote SA2s and 50 Very Remote SA2s.

2.4 Model description, tuning and training

The LSTM networks were implemented in Python (version 3.7.9) using the Keras deep learning library (version 2.3.1) with Tensorflow backend (version 2.1.0). Computations were run on GPU NVIDIA GeForce GTX 1660 Ti with Max-Q Design. We trialled two LSTM architectures, each with 4 layers:

- Simple: LSTM + 3 dense layers
- Bidirectional: Bidirectional LSTM + 3 dense layers

In addition to the LSTM layers, three dense layers were included in the architectures because deeper neural networks were often able to model more complex functions (Bengio & Delalleau, 2011). We trialled the Bidirectional LSTM additional to the "Simple" LSTM network because bidirectional neural networks are given greater exposure to the time series, allowing them to optimise weights based on future values, not just past ones (Schuster & Paliwal, 1997). Mean squared error (MSE) was used as the loss function. The Adam method (Kingma & Ba, 2014) was selected as the optimiser, which is the algorithm used during the training process to update model parameters such that the error is minimised.

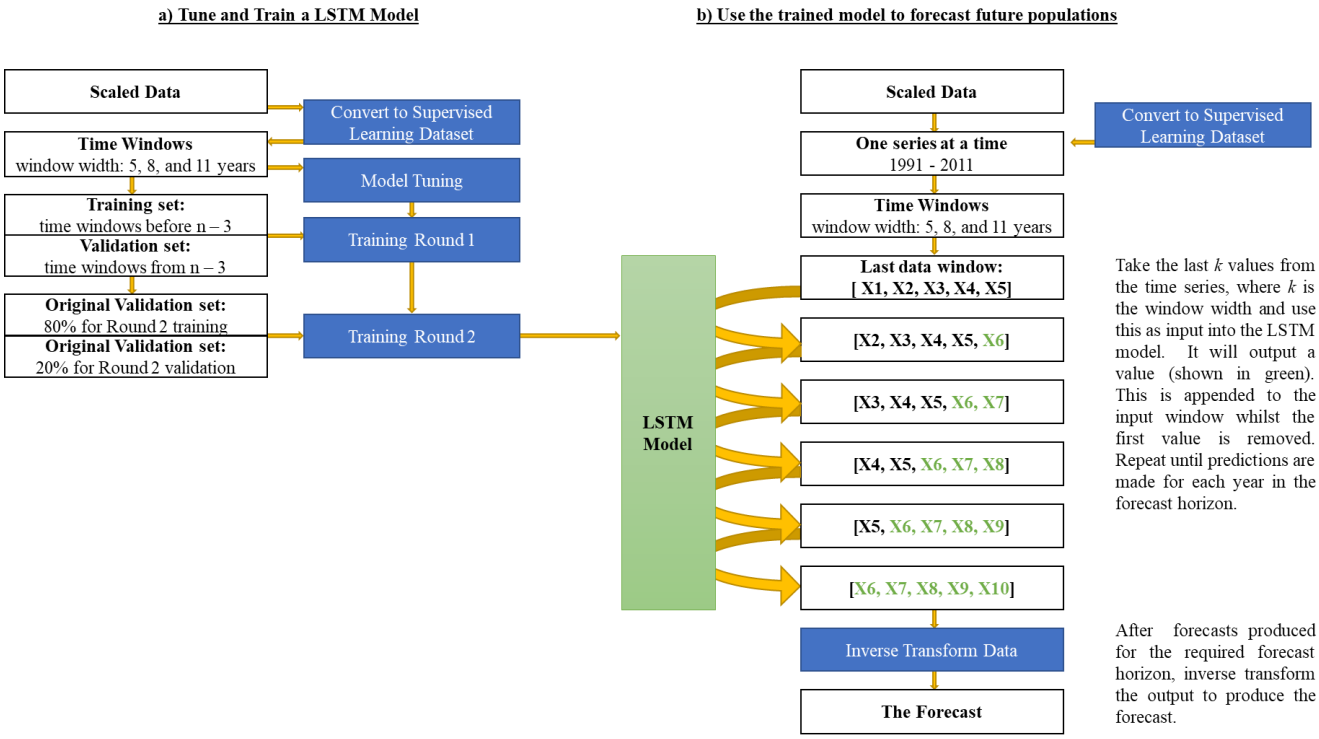


Figure 1. An overview of the process used to produce forecasts using our LSTM networks.

The process of tuning and training the LSTM networks is depicted in Figure 1a. For each of the trialled window widths, the scaled data is converted into a supervised learning dataset using the sliding window method. The Keras Tuner with Bayesian optimisation (O'Malley et al., 2019; TensorFlow, 2019) was used to select the number of units in the layers for each of the tested architectures, and each of the tested input window widths (5 steps, 8 steps, and 11 steps), where each step corresponds to one year. Thirty trials and 5 training rounds per trial were used to choose the numbers of units from a search space of 128 to 512 units; the objective was to select a number which minimised the MSE. The entire training dataset was used to tune the model.

During training, data is fed into the model and parameters were sought which minimise error between outputs for exemplar input and exemplar outputs. During each round of training (an "epoch"), model weights were updated in the direction of the error. The more epochs, the smaller the error on the training data. If epochs are too few then the optimal model parameters are unlikely to be found; if there are too many epochs the model can become overfitted, i.e., it becomes too specific for the training data and produces erroneous values when it tries to predict new values. One way to handle this is to set aside a subset of the

training data for "validation." This data is not used to train the model; it is used to estimate the model's prediction error on "new" data as the model is trained. When the error on the validation dataset begins to increase, this indicates that the model is becoming overfitted to the training dataset and suggests that forecast performance on new data will begin to decrease with further training. The two-stage training and validation scheme we describe below was created to prioritise the model for the most recent time points in the small area time series, and to support work with small datasets whilst keeping the computational load low enough to enable the models to be run on a single laptop.

In the first stage, the last three time windows from each time series in the training data (which exclude data beyond the jump-off year) were set aside as validation data. The model is trained with up to 500 epochs. We use early stopping with a patience value of 50; if there is no improvement in the error for the validation data during this time the training will stop earlier. The best model weights were restored if early stopping is utilised. Additionally, we update the learning rate when improvements in the MSE for the validation data did not improve for a period of 10 epochs, the learning rate is decreased until a minimum value of 5e-6. We can achieve such low values in the first stage of training as the Bayesian optimisation process used for tuning allows the model training process to start with quite low errors. This single round of training performed relatively well with, with the MedAPEs for the 2006-based 10 year forecasts in the range of 6.4 to 7.7 for the two architectures, however performance was improved with the second stage of training and validation.

A common challenge with small area demographic datasets is that the time series are often short. Therefore, removing the last three time windows from the training data to use for validation data could be detrimental to the accuracy of the forecasts. Our approach involved taking the data used for validation during the first stage of training and splitting it 80:20 using the train_test_split function from python's scikit-learn library (Pedregosa et al., 2011). We then undertake another round of training with the larger fraction, and validate with early stopping as before, using the smaller fraction. This produces models which have been optimised for the latter time points in the time series. During the second stage of training the minimum learning rate is lowered to 1E-6. As before, if early stopping is utilised, the best model weights were restored. Table 2 reports how many epochs were run for each of the models, stages, and jump-off years. Early stopping was not utilised in stage 2 of training and validation for the Simple 11 step model for the 2006 forecasts, suggesting that this model

may have benefitted from further epochs. The trained models were provided with the supplementary information.

After these two rounds of training with validation a tuned and trained LSTM network is produced. This model is used to make predictions, as depicted in 1b). The summary of the model architecture following tuning for the Simple 5 step LSTM, that was trained to perform the 2006- based forecasts, is provided in Table 1. Summaries for each of the other models used for the forecasts were provided with the Supplementary Information.

Table 1. Summary of the Simple 5 step LSTM network architecture (after tuning), used for the 2006-based forecasts.

Layer (type)	Output Shape	Number of Parameters
Input_1 (Input Layer)	[(None, 5, 1)]	0
LSTM	(None, 448)	806400
Dense	(None, 448)	201152
Dense_1	(None, 448)	201152
Dense_2	(None, 1)	449
Total parameters	1,209,153	
Trainable parameters	1,209,153	
Non-trainable parameters	0	

Table 2. The number of training epochs for each model, jump-off year, and stage of training.

			Model type			
		Simple			**Bidirectional**	
2006	**5**	**8**	**11**	**5**	**8**	**11**
Stage 1	77	82	230	74	82	436
Stage 2	56	320	500	56	225	202
2011	**5**	**8**	**11**	**5**	**8**	**11**
Stage 1	123	68	87	123	140	73
Stage 2	55	56	51	55	55	51

Note: The number of epochs listed is the total number of epochs used for training. If the listed number of epochs is less than 500, this means that the best set of model weights was found 50 epochs before the listed number, training stopped early, and the best model weights were restored.

2.5 LSTM forecasts

The trained and tuned models were used to forecast future values using the approach described in Figure 1b. Forecasts were produced separately for each of the time series; the last window is taken from each time series and fed into the LSTM network. For example, the 2011-based forecasts with the Simple 5 step model involved taking the last five years of data up to the jump-off year (2007-2011) and feeding it into the trained LSTM network. This would produce the prediction for the 2012 population. Then data from 2008-2012 is fed into the model to produce the forecast for 2013, and so on until predictions have been made for the entire forecast horizon. The model outputs were based on data scaled using the min-max method. To produce the forecast, the set of outputs corresponding to the years from the launch year to the target year is rescaled using the inverse of the min-max method.

2.5 The benchmark models

For our benchmark methods, we chose two simple models which have been shown to perform well in previous evaluations of small area population forecasting methods (Wilson, 2015). These models were the Linear/Exponential model (LIN/EXP) and the Constant Share of Population - Variable Share of Growth (CSP-VSG) averaged model (Wilson, 2015). The LIN/EXP model uses a linear extrapolation method if an area's growth over the past decade has been positive, and an exponential extrapolation approach otherwise. The CSP-VSG method is an average of the Constant Share of Population and the Variable Share of Growth forecast methods. The Constant Share of Population method assumes that a small area's share of the national population remains constant throughout the forecast horizon. It requires the area's proportion of the national population to be calculated for the jump off year. Small area forecasts were calculated by multiplying the proportions by the national population forecast. The Variable Share of Growth method involves adjusting LIN/EXP projections of population growth using the plus-minus method (Smith et al., 2013) such that the small area population growth matches national forecast growth.

2.6 Retrospective forecasts

Forecasts were calculated from the two jump-off years of 2006 and 2011, out to 2016. Two versions of each of the forecasts were created: constrained, and unconstrained, where the

small area forecasts in the former have been constrained to the national population forecasts. Constraining involves scaling each of the small area forecasts by the national population forecast divided by the sum of the unconstrained small area forecasts. This is carried out because both the producers and users of small area forecasts generally expect them to be consistent with the national forecast. We present forecast evaluations for all small areas, and evaluations where the SA2s have been grouped by size, and by their remoteness categorization (ABS, 2013b)

2.7 Data and Code Availability

Code, data, and population forecasts were provided as Supplementary Information to this paper.

3. Results

3.3 An Overview of Error

A summary of the constrained and unconstrained forecast errors for the SA2 areas is presented in Table 3. For ease of reading, the smallest error in each row has been bolded and LSTM forecast errors were highlighted where they beat both benchmarks.

In the unconstrained forecasts, the LSTMs models all performed well for the 2006-based 5 year forecasts, each producing a MedAPE of 3.1 - 3.2%, below the 3.4% MedAPE produced by the CSP-VSG model, and the 4.2% of the Lin/EXP model. For the 2006-based 10 year forecasts, the LSTM methods had MedAPEs of 5.9 - 6.2%, except for the Simple 11-step LSTM (MedAPE: 6.9%). The benchmark methods performed notably poorer with MedAPEs of 7.3% and 7.1% for the LIN/EXP and CSP-VSG methods, respectively. The benchmark LIN/EXP model was the best performing method for the 2011-based 5 year forecasts, with a MedAPE of 3.3%. The LSTM networks all performed similarly with MedAPEs of 3.6 - 3.8%, outperforming the CSP-VSG model (4.3%).

When evaluating model forecasts by the Percentage of Bad Forecasts, the LSTM networks tended to outperform the benchmark models, except for the 2011-based forecast where the LIN/EXP model produced 11.9% Bad Forecasts, compared to 12.0% and 12.1% Bad Forecasts for the Bidirectional 5 step and Simple 5 step LSTM methods, respectively.

Constraining the forecasts had variable effects but tended to increase errors. This is particularly true for the Percentage of Bad Forecasts by LSTM methods. Constraining increased the number of bad forecasts for the 10 year, 2006-based forecasts by 0.4% to 1.0% for all of the LSTM networks except the Simple 11 Step LSTM, which 0.4% fewer bad forecasts after constraining.

Table 3: Total population forecast errors

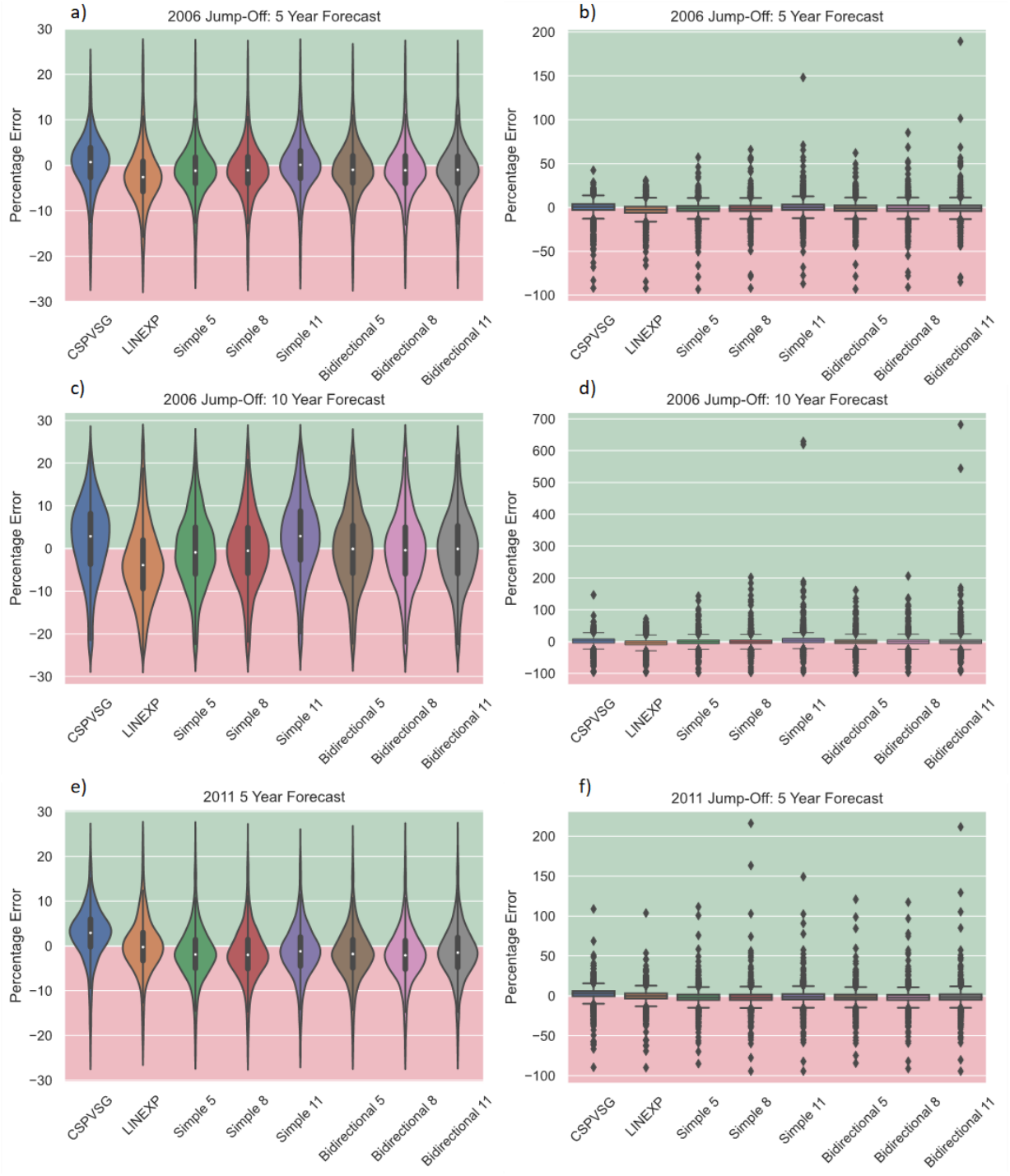
	Benchmark		Simple LSTM			Bidirectional LSTM		
	LIN/EXP	CSP-VSG	5 steps	8 steps	11 steps	5 steps	8 steps	11 steps
					MedAPE (%)			
After 5 years					**Unconstrained**			
2006-based	4.2	3.4	3.2	3.2	**3.1**	3.2	3.2	3.2
2011-based	**3.3**	4.3	3.7	3.8	3.6	3.6	3.8	3.7
After 10 years								
2006-based	7.3	7.1	**5.9**	**5.9**	6.9	6.0	6.0	6.2
					Constrained			
After 5 years								
2006-based	3.6	3.4	**3.0**	3.1	3.1	3.1	3.1	3.1
2011-based	3.6	4.3	3.7	**3.5**	3.7	3.7	**3.5**	3.9
After 10 years								
2006-based	6.5	7.1	6.5	**6.1**	6.8	6.7	6.4	6.5
					MAPE (%)			
After 5 years					**Unconstrained**			
2006-based	6.1	5.4	**5.0**	5.1	5.3	**5.0**	5.2	5.2
2011-based	**5.2**	6.0	5.5	5.8	5.6	5.4	5.6	5.7
After 10 years								
2006-based	10.5	9.9	**9.5**	9.6	11.8	9.6	9.8	10.6
					Constrained			
After 5 years								
2006-based	5.7	5.4	**4.9**	5.1	5.3	5.0	5.2	5.2
2011-based	5.6	6.0	5.6	5.7	5.8	5.7	**5.5**	6.0
After 10 years								
2006-based	10.1	**9.9**	10.0	10.0	11.5	10.3	10.3	11.2
					Percentage Bad Forecasts			
After 5 years					**Unconstrained**			
2006-based	15.9	11.9	**11.2**	11.4	12.0	11.3	11.9	11.7
2011-based	**11.9**	14.4	12.1	12.3	12.3	12.0	12.5	12.4
After 10 years								
2006-based	12.6	11.1	**10.1**	**10.1**	13.5	10.8	10.8	11.1
After 5 years					**Constrained**			
2006-based	14.8	11.9	11.2	11.3	12.1	**11.1**	11.9	11.9
2011-based	14.4	14.4	13.4	13.1	13.6	13.4	**12.8**	14.7
After 10 years								
2006-based	12.2	11.1	**10.7**	10.9	13.1	11.8	11.8	11.7

Note: The lowest error is bolded. Shading indicates LSTM errors which beat both benchmarks. LIN/EXP - The Linear/Exponential Model; CSP-VSG- The Constant Share of Population - Variable Share of Growth model. LSTM - Long short-term memory. Simple - deep learning networks which include a standard LSTM layer. Bidirectional - deep learning networks which include a bidirectional LSTM layer. Steps - the width (in years) of the input windows used during training.

3.2 Distributions of Percentage Errors

Figure 3 presents the distributions of percentage errors for each of our forecasts. Population forecasts tend to have long tails of large percentage errors, which are depicted in the box plots on the right-hand side of Figure 3. Percentage errors with a magnitude greater than 25% were removed from the Violin plots on the left, which illustrate the underlying probability density function, using the kernel density estimation method (Waskom, 2020). The box plots show that LSTM methods produce a few particularly erroneous forecasts; this can be seen by the larger magnitude of several of the outliers as compared to the benchmark methods. The violin plots reveal that the CSP-VSG models tended to over-project Australian small area populations whilst the LIN/EXP model was more likely to underestimate them. The LSTM network forecasts tended to exhibit less bias than the benchmark models, except for the 2011 5 year forecasts where the medians of their error distributions tended to be slightly negative.

Figure 2. Distribution of Percentage Errors for each model, 2006 and 2011 Jump-off, by Forecast Horizon



Note: Box plots showing distributions with outliers are shown on the right, whilst violin plots with |error values| >25 excluded are shown to the left. The violin plots include mini box plots within them with white dots representing median percentage errors. LIN/EXP - The Linear/Exponential Model; CSP-VSG- The Constant Share of Population - Variable Share of Growth model. LSTM - Long short-term memory. Simple - deep learning networks which include a standard LSTM layer. Bidirectional - deep learning networks which include a bidirectional LSTM layer. The number following 'Simple' or 'Bidirectional' indicates the width (in years) of the input windows used during training.

3.3 The impact of SA2 population size on forecast accuracy

Table 4 presents MedAPEs by SA2 population size at the jump-off year. For both the 5- and 10- year 2006-based forecasts, LSTM methods outperformed the benchmark methods for population size >5,000. However, the benchmark methods did better for the smallest areas with <5000 populations. The LIN/EXP method was clearly the better performer for the 2011-based forecasts, regardless of population size category. The LSTM methods tended to perform better than the CSP-VSG forecasts for the 2011-based forecasts.

Constraining the forecasts decreased the MedAPE of the 2011-based LIN/EXP forecasts, whilst increasing the MedAPE of the 2006- based forecasts; this was true for all population? size categories. The impact of constraining on the accuracy of the LSTM forecasts varied with population size, generally decreasing error for larger SA2s (>15,000 people) and increasing error for smaller SA2s (<5000 people).

Table 4: Median Absolute Percentage Errors by population size category of SA2 area total population forecasts

	Benchmark		**Simple LSTM**			**Bidirectional LSTM**		
	LIN/EXP	CSP-VSG	5 steps	8 steps	11 steps	5 steps	8 steps	11 steps
After 5 years				**Unconstrained**				
2006-based								
0-4,999	4.2	**4.1**	4.5	4.4	5.5	4.4	4.8	4.6
5,000-9,999	4.0	3.6	**3.0**	3.2	**3.0**	3.1	3.2	3.1
10,000-14,999	4.2	3.2	2.8	2.8	**2.4**	2.7	2.8	2.8
15,000+	4.4	2.9	2.9	2.7	**2.4**	3.0	2.8	2.7
2011-based								
0-4,999	**4.1**	5.5	4.4	4.8	4.5	4.4	4.7	4.9
5,000-9,999	**3.4**	4.5	3.8	4.0	3.8	3.7	3.9	3.9
10,000-14,999	**2.9**	4.2	3.3	3.4	3.2	3.3	3.3	3.2
15,000+	**3.0**	3.3	3.3	3.6	3.3	3.3	3.4	3.5
After 10 years								
2006-based								
0-4,999	**7.0**	8.2	7.7	8.0	13.0	8.8	8.4	8.6
5,000-9,999	7.1	7.5	5.8	5.8	6.9	5.8	5.6	6.1
10,000-14,999	7.4	6.3	4.8	4.5	4.9	4.8	4.7	4.8
15,000+	8.0	6.1	5.4	5.0	4.6	5.6	5.3	5.4
After 5 years				**Constrained**				
2006-based								
0-4,999	**3.8**	4.1	4.8	5.1	5.7	5.1	5.3	5.1
5,000-9,999	3.7	3.6	3.1	3.1	**3.0**	3.1	3.1	3.1
10,000-14,999	3.2	3.2	**2.3**	2.4	2.4	**2.3**	2.4	2.6
15,000+	3.5	2.9	2.4	2.5	**2.3**	2.4	2.4	**2.3**
2011-based								
0-4,999	**4.5**	5.5	5.0	4.8	4.7	5.0	4.7	4.8
5,000-9,999	3.9	4.5	3.9	**3.8**	**3.8**	3.9	**3.8**	4.1
10,000-14,999	3.8	4.2	3.4	**3.2**	3.4	3.5	**3.2**	3.7
15,000+	**2.9**	3.3	3.1	2.8	3.1	3.1	**2.9**	3.2
After 10 years								
2006-based								
0-4,999	**6.5**	8.2	9.6	9.7	12.5	10.8	10.5	10.7
5,000-9,999	6.7	7.5	6.7	**6.1**	6.8	6.6	6.6	6.8
10,000-14,999	6.5	6.3	5.2	4.9	4.8	5.3	4.9	**4.5**
15,000+	7.0	6.1	4.8	**4.6**	4.7	4.7	4.8	4.7

Note: The lowest error is bolded. Shading indicates LSTM errors which beat both benchmarks. LIN/EXP - The Linear/Exponential Model; CSP-VSG- The Constant Share of Population - Variable Share of Growth model. LSTM - Long short-term memory. Simple - deep learning networks which include a standard LSTM layer. Bidirectional - deep learning networks which include a bidirectional LSTM layer. Steps - the width (in years) of the input windows used during training.

3.4 The impact of remoteness on forecast accuracy

Finally, we consider how the remoteness of an SA2 impacts forecast accuracy for each of the methods (Table 5). For both the 5 and 10 year horizon 2006-based forecasts, the LSTM networks outperformed the benchmark models for larger areas, with the difference being most evident for the SA2 areas in Major Cities, where the Simple 8 step LSTM method outperformed the CSP-VSG method, the better performing benchmark, by 0.5% after 5 years and by 1.2% after 10 years. However, this pattern of results was different for the 2011-based forecast where the LIN/EXP method was the top performer for all categories, except Outer Regional where the 5 step LSTM methods did marginally better. Constraining the forecasts tended to decrease LSTM network error size for Major cities and increase the error for more remote areas - particularly for LSTM networks. Constraining generally improved LIN/EXP model forecasts for 2006-based forecasts and worsened the 2011-based forecasts, across most of the remoteness categories.

Table 5: Median Absolute Percentage Errors by remoteness of SA2 area total population forecasts

	Benchmark		Simple LSTM			Bidirectional LSTM		
	LIN/EXP	CSP-VSG	5 steps	8 steps	11 steps	5 steps	8 steps	11 steps
After 5 years				**Unconstrained**				
2006-based								
Major Cities	4.8	3.4	3.1	**2.9**	3.0	3.1	3.1	3.1
Inner Regional	3.2	3.1	**2.9**	3.0	3.0	3.0	**2.9**	**2.9**
Outer Regional	**3.5**	4.0	**3.5**	3.8	3.6	3.6	3.6	3.5
Remote	**4.5**	4.9	5.6	5.3	5.1	5.7	5.4	5.2
Very Remote	8.4	**6.0**	8.5	8.9	8.8	8.3	9.3	8.0
2011-based								
Major Cities	**3.2**	4.0	3.6	3.8	3.6	3.5	3.7	3.7
Inner Regional	**3.1**	3.9	3.5	3.6	3.4	3.5	3.6	3.5
Outer Regional	3.4	5.4	**3.3**	3.5	3.5	**3.3**	3.5	3.4
Remote	**5.4**	9.8	7.3	7.4	8.3	7.9	7.5	7.4
Very Remote	9.4	11.9	9.8	10.1	9.6	9.7	**9.3**	10.0
After 10 years								
2006-based								
Major Cities	8.4	6.7	5.7	**5.5**	5.8	5.8	**5.5**	5.9
Inner Regional	5.9	6.4	**5.5**	5.8	7.4	5.7	5.8	5.8
Outer Regional	**5.0**	7.8	7.0	6.8	10.2	6.9	6.7	6.7
Remote	**7.0**	11.2	9.2	7.1	12.3	10.5	8.4	9.8
Very Remote	**10.0**	11.4	12.2	12.7	17.4	12.6	13.2	14.6
After 5 years				**Constrained**				
2006-based								
Major Cities	3.8	3.4	**2.7**	2.8	2.9	2.8	2.8	2.9
Inner Regional	3.0	3.1	3.0	3.0	3.1	3.1	3.0	**2.9**
Outer Regional	**3.2**	4.0	3.9	3.8	4.0	4.0	4.0	3.8
Remote	**4.2**	4.9	5.1	5.5	4.9	5.5	5.7	4.7
Very Remote	7.8	**6.0**	8.3	9.1	8.6	8.3	9.4	8.1
2011-based								
Major Cities	3.5	4.0	3.4	**3.2**	3.5	3.5	3.3	3.6
Inner Regional	3.4	3.9	3.5	3.5	3.5	3.6	**3.3**	3.8
Outer Regional	**3.7**	5.4	4.1	4.0	4.0	4.2	3.8	4.6
Remote	**6.0**	9.8	11.1	10.8	10.5	12.1	10.6	10.1
Very Remote	**11.4**	11.9	13.8	12.5	12.7	13.8	12.4	12.5
After 10 years								
2006-based								
Major Cities	7.3	6.7	5.7	**5.5**	5.6	5.7	5.6	5.7
Inner Regional	**5.6**	6.4	6.4	6.1	7.1	6.9	6.4	6.1
Outer Regional	**4.8**	7.8	9.0	8.6	9.5	9.1	8.8	8.3
Remote	**8.4**	11.2	11.1	9.8	12.4	12.7	10.7	12.1
Very Remote	**10.1**	11.4	13.4	15.6	16.8	14.2	15.6	17.3

Note: The lowest error is bolded. Shading indicates LSTM errors which beat both benchmarks. LIN/EXP - Linear/Exponential Model; CSP-VSG - Constant Share of Population - Variable Share of Growth model. LSTM - Long short-term memory. Simple - deep learning networks which include a standard LSTM layer. Bidirectional - deep learning networks which include a bidirectional LSTM layer. Steps - the width (in years) of the input windows used during training.

4. Discussion

4.1 Key findings

This paper has presented a deep learning approach for forecasting small area population totals. We demonstrated that LSTM methods were capable of matching, or improving, the accuracy of top performing benchmark models for Australian small area forecasts at the SA2 level. This is notable given that the top performing benchmark method was different for the 2006- and 2011- based forecasts, whilst the top performing LSTM methods consistently produced comparable forecasts and markedly improved the 10- year forecast accuracy. We first focus our discussion on the unconstrained forecasts, before considering the impacts of constraining and then discussing the limitations, and opportunities for further research.

Models performed differently depending on jump-off years. An examination of overall model performance for the 2006- based forecasts found that the LSTM methods produced more accurate forecasts than our two benchmark models. This improvement was particularly marked after 10 years with the Simple 5 step LSTM obtaining a MedAPE of 5.9% and 10.1% Percentage Bad Forecasts, compared to a 7.1% MedAPE and 11.1% Percentage Bad Forecasts for the CSP-VSG method and a 7.3% MedAPE and 12.6% Percentage Bad Forecasts for the LIN/EXP method. When we consider performance by population size, we find that the benchmark methods outperform the LSTM methods for populations with fewer than 5,000 people, whilst LSTM methods did better for larger areas. Similarly, the benchmark methods were the best performers for Very Remote areas whilst the LSTM methods did better for Major Cities and Inner Regional SA2s.

The LIN/EXP method was the best performer for the 2011-based forecasts overall, for each of the SA2 population size categories, and for most of the remoteness categories. The LSTM methods tended to perform better than the CSP-VSG method; this was true for all remoteness categories and for all the size categories except the largest where the CSP-VSG method did as well as the top LSTM methods. The CSP-VSG method is a combination of an adjusted LIN/EXP model (The VSG method) and the CSP model. The LIN/EXP model performs well for the 2011 based forecasts. The relatively poorer performance of the CSP-VSG method for the 2011-based forecasts was due to the lower accuracy of the CSP method. This suggests that there was greater variability in growth rates across SA2 areas during the 2011-2016 period compared to 2006-2011.

4.2 The impacts of varying window width

Next, we consider the performance of our Simple and Bidirectional LSTM architectures and the impact of changing the input window width. Shynkevich et al. (2017) investigated the impacts of varying input window length for stock price forecasting. They used 3 types of ML methods: support vector machines, artificial neural networks, and k-nearest neighbours. Forecast accuracy was found to improve when the input window width matched the forecast horizon. Riiman et al. (2019) focused on forecasts of small area population totals. They noted that the "prediction window of five gives good results (though for some specific counties the best results varied)" for their 10 year horizon forecasts (Riiman et al., 2009; p. 107). In the results presented in this paper, we show that there was no consistency in improvement of forecast precision by increasing window width from five to 10 years. However, consideration of model performance by area size and remoteness suggests that a longer input window may be less advantageous for Remote areas, and more helpful for areas with larger populations and for those that are in Major Cities. Perhaps this finding could help account for some of the variation found by Riiman et al., (2019) for different counties.

4.3 Constraining forecasts

In practice, forecasts for small areas often need to be consistent with those for higher geographies; practicing demographers often constrain their small area projections to ensure this is the case. There were several studies which achieve consistent forecasts by disaggregating population forecasts produced for national or larger regional areas (e.g., McKee et al., 2015; Boke-Olén et al. 2017). Constraining has been used in published small area forecast methods to produce consistent forecasts. In Chen et al. (2020) grid level population forecasts were constrained to the Shared Socioeconomic Pathways scenarios. However, there is no comprehensive evaluation of different constraining methods. We chose a simple method of constraining that involved simply scaling the forecasts to be consistent with the Australian national population projection. The overall impact of constraining the LIN/EXP projections was variable. Constraining the LSTM projections tended to increase error, except for forecasts for SA2s in Major Cities and for areas with populations >15,000, which were generally improved. Our results show that constraining can alter the results of our evaluation. This calls into question the value of recommending a method based on a small improvement in a particular error metric. Such small improvements may not translate to real

world practice where further refinements to the forecasts (such as constraining) may negate the benefits. Further research is required to develop alternatives to simple constraining methods which involve a universal scaling factor, understand how these methods interact with population forecasting methods, and understand properties of datasets that would support the use of one constraining method over another. Furthermore, it may be better to constrain an LSTM forecast after each of the forecasting steps to prevent the forecasts from veering too far away from a national projection and to prevent some of the runaway growth that can occur. This should be considered in further research work.

4.4 Limitations and future research

This paper developed and evaluated two LSTM architectures for the purposes of forecasting small area population totals. Our models performed favourably relative to proven benchmark models and produced markedly improved 10 year forecasts. We considered issues in the constraining of small area forecasts. There were several limitations to this work that must be noted:

- The methods were only evaluated using an Australian small area population dataset and over a limited time period. These methods should be tested for longer forecast horizons, as real-world small area forecasts often require 10-20 year horizons. ML methods would need to be tested on datasets from many countries covering a range of historical periods to better understand which methods were most suitable for real world small area forecasts. An example of such an evaluation is found in Olson et al. (2018), where 13 machine learning algorithms were trialled on 165 datasets to prepare guidance for bioinformaticians.
- Most of the SA2s in our dataset were from Major Cities. Therefore, the trained models would be better suited for urban areas. A model trained with only Remote areas, or only the SA2s with smaller populations, may have produced more accurate forecasts for rural areas. Further work should consider the inclusion of an area's remoteness as an additional feature in a multivariate dataset or to prepare separate models for remote areas.
- In this paper we developed and evaluated two deep learning based methods to support work with small area datasets. However, further work is required to evaluate other architectures, and other training and validation regimes with more datasets.

- We only trialled one simple method for constraining populations to the national projection. This method could decrease the accuracy of the forecasts if the national projection turns out to be inaccurate. Further work is required.

The work presented in this paper was undertaken on a single laptop computer. Each LSTM forecast took between 24 and 50 minutes to produce. This was additional to the time taken to prepare the data and write the code. The benchmark models were faster and easier to run and can be produced in Excel; therefore, they were often more appropriate for use in practice. The forecasts we presented were based on univariate datasets containing only population totals. Univariate datasets are easier to prepare and the additional variables available may differ between countries; methods developed with multivariate datasets may not be useful for regions which lack other data. However, multivariate datasets may help improve forecast accuracy and should be investigated in future research. These more complex approaches may be what is required to build models which can consistently beat the benchmark models. However, the additional costs which accompany greater complexity may make them unusable in many practical settings.

4.5 Conclusions

In this paper we show that there is potential for small area population forecasts produced by deep learning methods to outperform conventional methods. The strong performance of many of the machine learning forecasts is encouraging. However, further work is required before they can be justifiably used in practice. Machine learning methods are usually black boxes which produce forecasts that are not easily interpretable by end-users. Further work is required to develop and evaluate machine learning methods that are proven to work across multiple small area demographic datasets. Research is also required to develop a system of constraints which consider underlying demographic processes to mitigate the large errors that occur for some small areas and to produce forecasts which are consistent with forecasts for higher geographies.

Acknowledgements

This research was supported by the Australian Research Council (Discovery Project DP200101480).

Data Availability Statement

Code, model summaries, SA2 area population forecasts, Estimated Resident Populations, and Statistical Area Level 2 2011 to Remoteness Area 2011 are available at: https://github.com/irigrossman/LSTM-for-Small-Area-Populations

5. References

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6), 594-621. https://doi.org/10.1080/07474938.2010.481556

Australian Bureau of Statistics, (2017). *ERP by SA2 and above (ASGS 2011), 1991 to 2016* [Dataset]. Australian Bureau of Statistics ABS.Stat Beta. http://stat.data.abs.gov.au/Index.aspx?DataSetCode=ABS_ANNUAL_ERP_ASGS

Australian Bureau of Statistics, (2013a). *TABLE B9. Population projections, By age and sex, Australia - Series B* [Dataset]. Australian Bureau of Statistics repository. https://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/3222.02012%20(base)%20to%202101?OpenDocument

Australian Bureau of Statistics, (2013b). *1270.0.55.005 - Australian Statistical Geography Standard (ASGS): Volume 5 - Remoteness Structure, July 2011* [Dataset]. Australian Bureau of Statistics repository. https://www.abs.gov.au/AUSSTATS/abs@.nsf/allprimarymainfeatures/17A7A350F48DE42ACA258251000C8CA0?opendocument

Australian Bureau of Statistics, (2011). *ERP by SA2 and above (ASGS 2011), 1991 to 2016* [Dataset]. Australian Bureau of Statistics ABS.Stat Beta. http://stat.data.abs.gov.au/Index.aspx?DataSetCode=ABS_ANNUAL_ERP_ASGS

Australian Bureau of Statistics, (2008). *TABLE B9. Population projections, By age and sex, Australia - Series B* [Dataset]. Australian Bureau of Statistics repository. https://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/3222.02006%20to%202101?OpenDocument

Bengio Y., & Delalleau O. (2011). On the Expressive Power of Deep Architectures. In J. Kivinen, C. Szepesvári, E. Ukkonen & T. Zeugmann (Eds.), *Algorithmic Learning Theory. ALT 2011. Lecture Notes in Computer Science* (pp. 18 - 36). Springer. https://doi.org/10.1007/978-3-642-24412-4_3

Boke-Olén, N., Abdi, A. M., Hall, O., & Lehsten, V. (2017). High-resolution African population projections from radiative forcing and socio-economic models, 2000 to 2100. Scientific data, 4(1), 1-9. https://doi.org/10.1038/sdata.2016.130

Chen, H., Matsuhashi, K., Takahashi, K., Fujimori, S., Honjo, K., & Gomi, K. (2020). Adapting global shared socio-economic pathways for national scenarios in Japan. *Sustainability Science, 15*(3), 985-1000. https://doi.org/10.1007/s11625-019-00780-y

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). *Learning to forget: Continual prediction with LSTM.* Paper presented at the 9th International Conference on Artificial Neural Networks: ICANN '99, Edinburgh, UK.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). *Densely connected convolutional networks.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *IEEE Annals of the History of Computing, 9*(03), 90-95.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint. https://arxiv.org/abs/1412.6980.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE, 13*(3), e0194889.

McKee, J. J., Rose, A. N., Bright, E. A., Huynh, T., & Bhaduri, B. L. (2015). Locally adaptive, spatially explicit projection of US population for 2030 and 2050. Proceedings of the National Academy of Sciences, 112(5), 1344-1349. https://doi.org/10.1073/pnas.1405713112

Olson, R. S., Cava, W. L., Mustahsan, Z., Varik, A., & Moore, J. H. (2018). Data-driven advice for applying machine learning to bioinformatics problems. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium* (pp. 192-203).

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., & et al. (2019). Keras Tuner. Retrieved from https://github.com/keras-team/keras-tuner

O'Neill, B. C., Kriegler, E., Ebi, K. L., Kemp-Benedict, E., Riahi, K., Rothman, D. S., . . . Solecki, W. (2017). The roads ahead: Narratives for shared socioeconomic pathways describing world futures in the 21st century. Global Environmental Change, 42, 169-180. doi:https://doi.org/10.1016/j.gloenvcha.2015.01.004

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Riiman, V., Wilson, A., Milewicz, R., & Pirkelbauer, P. (2019). Comparing Artificial Neural Network and Cohort-Component Models for Population Forecasts. *Population Review*, 58(2). https://doi.org/10.1353/prv.2019.0008

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536. https://doi.org/10.1038/323533a0

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing, 45*(11), 2673-2681.

Smith, S. K., Tayman, J., & Swanson, D. A. (2013). *A practitioner's guide to state and local population projections.* Springer.

Striessnig, E., Gao, J., O'Neill, B. C., & Jiang, L. (2019). Empirically based spatial projections of US population age structure consistent with the shared socioeconomic pathways. *Environmental Research Letters, 14*(11), 114038. https://doi.org/10.22022/pop/10-2019.54

TensorFlow. (2019). Introduction to the Keras Tuner. Retrieved from https://www.tensorflow.org/tutorials/keras/keras_tuner

Waskom, M. L. (2020). seaborn.violinplot. Retrieved from https://seaborn.pydata.org/generated/seaborn.violinplot.htmlWaskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021.

Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021.

Weber, H. (2020). How Well Can the Migration Component of Regional Population Change be Predicted? A Machine Learning Approach Applied to German Municipalities. *Comparative Population Studies, 45*.

Wilson, T. (2015). New evaluations of simple models for small area population forecasts. *Population, Space and Place*. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/psp.1847

Wilson, T., Brokensha, H., Rowe, F., & Simpson, L. (2018). Insights from the evaluation of past local area population forecasts. *Population Research and …*. Retrieved from https://link.springer.com/article/10.1007/s11113-017-9450-4

Wilson, T., & Shalley, F. (2019). Subnational population forecasts: Do users want to know about uncertainty? *Demographic Research*, 41, 367-392. https://doi.org/10.4054/DemRes.2019.41.13

Wilson, T., Grossman, I., Alexander, M., Rees, P., & Temple, J. (2021a, April 29). Methods for small area population forecasts: state-of-the-art and research needs. https://doi.org/10.31235/osf.io/sp6me

Wilson, T., Grossman, I., & Temple, J. (2021b). *Evaluation of the best M4 competition methods for small area population forecasting.* Manuscript submitted for publication.