Final Report

Alexandar Nikolic

Abstract

Accurate forecasting of carbon dioxide is essential in assisting the development of policies to effectively mitigate climate change. This study develops and evaluates 6 separate machine learning models being Random Forest, XGBoost, Long-Short Term Memory, Ridge Regression, Lasso Regression and ElasticNet. They were developed using annual data ranging from 1982 to 2022, integrating data from both 'Our World in Data' and the Australian Energy Update 2024. Environmental, economic and energy factors from these datasets were implemented to create these models. A comparison between Random Forest's and XGBoost's feature importance evaluation has also been made, finding that population and fossil fuel and energy consumption are significant factors in CO_2 emissions in Australia. Of the models developed, Lasso Regression was found to have the highest accuracy (R^2 = 0.9852) and the lowest error, indicating its superior ability to identify complex relationships between features. This performance is then followed by Ridge Regression, ElasticNet, XGBoost, LSTM and finally random forest. This research highlights the strength and adaptability of regularisation methods combined with feature importance analysis, providing valuable insights for policy makers, allowing them to make informed decisions on reducing carbon emissions in creating sustainable regulations.

Introduction

Carbon dioxide (CO_2) is the most prevalent greenhouse gas, accounting for approximately 63% of emissions within Australia (Australian Government, 2024). These emissions are the result of human activities, such as burning fossil fuels, which greatly contribute to the significant issue of climate change. In order to combat this issue, strategies to reduce CO_2 emissions are being implemented, such as transitioning to renewable energies and improving energy efficiency. To determine how successful these efforts are, precise forecasting of these emissions is necessary. Utilising machine and deep learning models is an effective methodology to predict CO_2 emissions within Australia.

This report targets the problem statement of "What is the predicted CO_2 emissions for Australia using machine and deep learning models?". The research questions to answer this problem statement are:

- Which machine or deep learning algorithm creates the most accurate model?
- Which dataset/s is most suitable for Australian CO_2 emission prediction models?
- What features are most effective in determining CO_2 emissions in Australia?
- How to verify the accuracy of models?
- What feature engineering techniques produce the best Australian CO_2 emission models?

Within this study, which methodologies are most effective for creating CO_2 prediction models are explored. Which algorithm is best, which features are most suitable, as well as other variables have been determined through this research. This research would be beneficial for Australian government policy makers, as they can use the information on what factors are most indicative of CO_2 emissions to take measures against them. Additionally, this research benefits other scholars aiming to complete similar research, as findings on what

methodologies are most effective can be used to create their own models in different contexts.

Literature Review

There are numerous research reports dedicated to determining the most effective machine learning or deep learning algorithm to predict CO_2. The two main approaches commonly used are; prediction based on influencing factors and prediction based on historical data. The first method selects factors that have a high correlation with CO_2 emissions as inputs for a model, allowing it to predict future emissions. The second approach exclusively utilises historical CO_2 emission values in the form of a univariate time series as inputs to determine the emission trends (Li & Zhang, 2023). Additionally, many articles implement statistical models to predict CO_2 emissions, however, papers which compare these different methods often discover that machine learning models are more effective in performing these functions. For example, Li and Zhang (2023) compared the statistical models of Grey Model, Autoregressive integrated moving average and Seasonal autoregressive-integrated moving average with exogenous factors (SARIMAX), to the machine learning models and found all the machine learning models were significantly more effective in predicting CO_2 emissions. These different machine learning algorithms are explored further in this review.

Common algorithms used to predict CO_2 in scholarly articles are Artificial Neural Networks (ANN), Support Vector Machine (SVM), Linear Regression Model (LR), Random Forest (RF) and Long-Short Term Memory (LSTM) (Akkaya & Akkaya, 2023; Chukwunonso et al, 2024; Kumari & Singh, 2022; Le et al, 2024; Li & Zhang, 2023; Yoa et al, 2024).

Methodology

In terms of methodology in creating CO_2 prediction models, many reports utilise similar methods, this being made up of 3 sections. These sections are data preparation, model building and model selection.

Data Preparation

Data preparation involves collecting CO_2 emission data, pre-processing it through various methods and splitting the dataset to either develop the model or evaluate it. In selecting appropriate datasets, there are many relevant resources which detail CO_2 emissions and other relevant statistics in the majority of countries. Datasets such as CAIT (Kumari & Singh, 2022), 70th Statistical Review of World Energy (Akkaya & Akkaya, 2023) and Carbon Monitor (Li & Zhang, 2023) are utilised depending on the article. Other reports used more local data, such as Chukwunonso et al using US Energy Information Administration data source for USA data (2024).

Pre-processing of data most often involves filling in missing values, smoothing data, resolving inconsistencies and eliminating outliers (Al-Nefaie & Aldhyani, 2023). However, other feature engineering steps can be taken to validate the correlation of the model's inputs to CO_2 emissions. These feature selection methods are rarely implemented in the context of CO_2 emissions. Chukwunonso et al. (2024) implemented min-max normalisation to apply a

linear transformation to their data, converting every input to a value between 0 and 1. This method preserves the distribution of the data while bringing them into a desired range. Furthermore, variation mode decomposition (VMD) was utilised for the purpose of denoising the data. Through both these processes, it was found that models became more accurate. Additionally, Ran et al. (2023) utilised RF to confirm feature relevance, using it to determine that the inputs of energy emission intensity, industrial employees, urbanisation rate, total population and the number of industrial technical papers, to be the most significant factors in CO_2 emissions in China. Other inputs were considered, however, to avoid redundancy of input variables only the most significant were used to build the models. Kong et al. (2022) also implemented feature selection, using a combination of partial autocorrelation function (PACF) and ReliefF to find feature importance. In terms of data splitting, it is extremely common within this context for CO_2 emission data to be split 80% for training and 20% for testing, with it being used in a multitude of studies (Chukwunonso et al, 2024; Li & Zhang, 2023; Kong el al, 2022).

The limited implementation of feature engineering processes leads to the possibility for studies to determine how significant their impact is. Algorithms to determine feature importance such as RF, ReliefF and Gradient Boosting could be compared to assist in finding the most optimal model for CO_2 predictions.

Model Building

Model building involves the development of the various models to test their effectiveness in predicting CO_2 emissions. Below is a description of 8 articles which focused on using machine learning to predict CO_2 emissions. It details what their aim was, what algorithms they tested, what inputs or features were used and which algorithm created the best result and why.

Chukwunonso et al. (2024) explored different machine learning algorithms in predicting CO_2 emissions in the United States of America (USA). They tested various ANNs being Feedforward Neural Network (FFNN), Layered Recurrent Neural Network (L-RNN) and two Convolutional Neural Network (CNN). The two CNNs differed, both having two layers, one with 10 and 5 filters and the other with 50 and 25 filters. Inputs for the models were various energy consumptions, such as coal and natural gas. The result of the report determined that L-RNN performed the best. The most effective models were consistent regardless of pre-processing techniques used, however, models with pre-processed data, including min-max normalisation and VMD, achieved more accurate results than their non-pre-processed counterparts. This result was due to L-RNN's ability to incorporate a time delay during the training process through a connection between the output layer and hidden layers.

Li and Zhang (2023) utilised the machine learning models of ANN, RF and LSTM to predict CO_2 emissions in China. Only historical CO_2 emission data was used to test the models. They discovered that LSTM was most successful in predicting CO_2 emissions as it can use historical data to predict information at specific times.

Ran et al. (2023) studied using machine learning to estimate China's industrial carbon emissions peak, using back propagation neural network (BP) and SVM. Inputs used were

energy emission intensity, industrial employees, urbanisation rate, total population and the number of industrial technical papers. They found that SVM was more suitable to predict China's peak CO_2 emissions as it is more suitable for small sample, non-linear data sets when compared to BP.

A recent study by Le et al. (2024) forecasted energy consumption and CO_2 emissions in Vietnam using LR, RF, Extreme Gradient Boosting (XGBoost) and AdaBoost. Inputs used were related to population, Gross Domestic Product (GDP) and energy consumption. In evaluating their models, XGBoost was determined as the most effective for forecasting CO_2 emissions because of its proven effectiveness in handling difficult regression problems.

Kumari and Singh. (2022) utilised machine learning time series models for CO_2 emission prediction in India, using LR, RF and LSTM. Again, LSTM was discovered to be the most effective in predicting CO_2 emissions due to the same reasons as Li and Zhang (2023).

Akkaya and Akkaya. (2023) developed and compared machine learning models to predict energy-related CO_2 emissions, using data from 68 countries. They tested SVM, Ensemble of Trees (ET) and Gaussian Process Regression (GPR), as well as variants of each, giving them 9 models in total. Inputs used to create models were primary energy consumption, electricity generation, non-fossil based electricity generation, and GDP. They found that the GPR exponential variant was superior in predicting CO_2 emissions as it has the ability to adapt to different data patterns and model non-linear relationships.

Kong et al. (2022) predicted daily CO_2 emissions in China using both statistical and machine learning models.The machine learning models used were Extreme Learning Machine (ELM) and BP with many variants created, especially for ELM. Through their studies ELM combined with an Improved Sparrow Search Algorithm (ISSA) was most suited for daily CO_2 emission prediction. The result was caused by ELM-ISSA effectively finding useful information in historical time-series with its optimised input weights and hidden layer bias thanks to ISSA and its strong generalisation ability of ELM.

Yoa et al. (2024) created a model which predicted CO_2 emissions within the Jiangsu province in China, using machine learning and econometric models. Machine learning models used were BP, SVM, LSTM and Heuristic Neural Network (HNN). This HNN was a custom RNN algorithm, implementing more complex memory units to improve the models performance. Inputs used were population size, urbanisation rate, industrial structures, GDP per capita and carbon emission intensity. After in-depth testing, it was determined that the HNN model performed most effectively, outperforming the commonly used LSTM. This performance difference can be attributed to the smaller dataset used which HNN was more suited to with its effective non-linear processing ability, self organisation and adaptability.

Overall, LSTM was the algorithm that was most frequently proven to generate successful results in CO_2 emission predictions. This result is due to LSTM being proven to be extremely effective in time series forecasting with its memory cells being able to store and retrieve information over long periods of time. This feature grants LSTM the ability to determine accurate and complex patterns based on a number of features. Despite this, articles not utilising LSTM found other algorithms which were effective, such as L-RNN, GPR and ELM. Additionally, Yoa et al. (2024) created a model superior to LSTM with HNN. These findings

lead to indecisiveness as to which process is most effective, requiring further testing and research to determine a more conclusive result.

Model Selection

Finally, model selection evaluates how effectively each model performed, utilising various evaluation criteria. Many reports utilise similar evaluation criteria to test their models. Some of the most common ones being mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and coefficient of determination (R^2) (Akkaya & Akkaya, 2023; Chukwunonso et al, 2024; Kong et al, 2022; Kumari & Singh, 2022; Le et al, 2024; Li & Zhang, 2023; Ran et al, 2023; Yoa et al, 2024). Through analysis of the evaluation criteria, the most effective model can then be determined.

Research Methodology

Predicting CO_2 emissions is essential for reducing the drastic impacts of climate change within Australia. Australia was chosen for this research project as there has been limited study of CO_2 emissions in Australia, with no other articles at time of writing attempting to predict Australian CO_2 emissions using machine learning.

Data Sources and Development Environment

To predict CO_2 emissions in Australia, data used to train the machine learning models was sourced from a combination of Australian datasets. Our World in Data's (OWID) CO_2 and Greenhouse Gas Emissions dataset provides a comprehensive database of different countries' CO_2 emissions, including Australia (Ritchie et al, 2023). This dataset was combined with the Australian Energy Update 2024, as it includes many important details on Australia's energy consumption, which greatly influence Australia's CO_2 emissions (Department of Climate Change, Energy, the Environment and Water, 2024). This dataset spans from 1982-2022, utilising the most up to date data available. There are 116 inputs that can be extrapolated from this dataset. Some of these include factors such as population, GDP, primary energy consumption, coal, oil, gas, renewables among others factors. Which of these features that have been used was determined by which features were found to be most relevant for each model to achieve the best performance. All pre-processing, model building and model evaluation has been completed on Google Colab, utilising Python 3.10 and the Scikit-learn 1.5.2 library. The flowchart below (Figure 1) describes the creation process for the models development. Each colour represents each step in development, with brown being data pre-processing, green being model building and red being model evaluation.

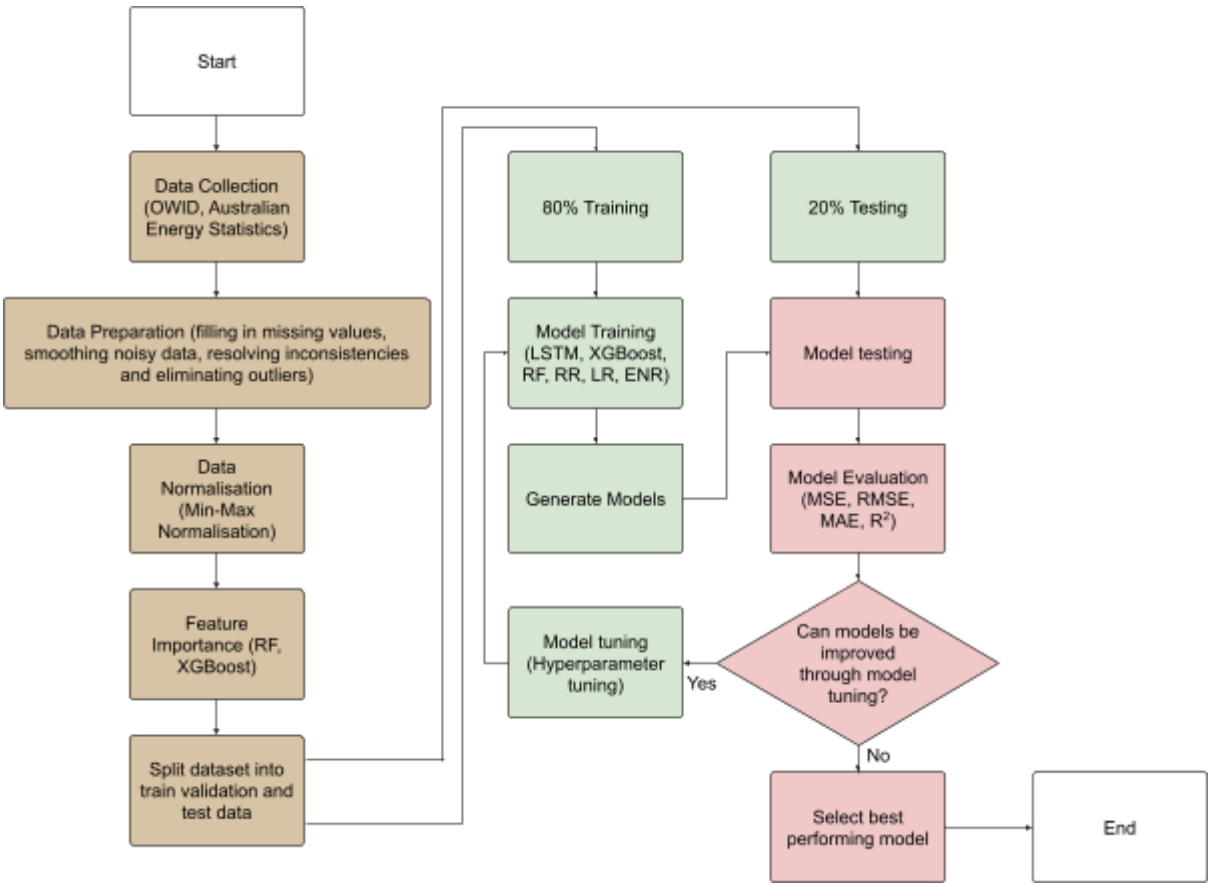


Figure 1. FlowChart for the development of machine and deep learning models

Pre-processing

With the numerous potential features to be used, feature engineering techniques have been implemented to ensure the dataset is suitable for machine learning model training. Initially data preparation techniques were utilised against the dataset. These data preparation techniques include filling in missing values, smoothing noisy data, resolving consistencies and eliminating outliers. Backward filling was utilised to fill in missing data as some features lacked entries in the first few values of the data.

Min-Max Normalisation

Once this stage was completed, Min-Max normalisation was implemented for select models, similar to Chukwunonso et al. (2024). Min-Max Normalisation applies a linear transformation to the dataset (Akanbi et al, 2015). It maps all values between a specified range, using the minimum and maximum values from the dataset. The specified range for the purposes of this project was between 0 and 1 as this is the most typically used range for the purposes of normalising machine learning datasets. This process preserves the order and distance of data points and allows features that have smaller values to be equally considered compared to features which consist of larger values. Min-Max normalisations' ineffectiveness in handling outliers has been mitigated, as outliers were removed from the dataset during data preparation. The effect of Min-Max Normalisation can be seen in the Figures 2 and 3 below

with each feature's data being transformed between ranges 0-1.

year	population	gdp	cement_co2	co2	ghg_excluding_lucf_per_capita	land_use_change_co2	methane	methane_per_capita	nitrous_oxide	nitrous_oxide_per_capita
1982	15192485	100.0	2.823	218.001	17.572	76.011	154.946	10.199	73.563	4.842
1983	15414891	104.6	2.377	207.645	16.842	249.682	139.737	9.065	61.029	3.959
1984	15620650	110.1	2.683	217.516	17.426	118.902	143.298	9.174	61.882	3.962
1985	15835742	114.5	2.891	224.556	17.413	63.932	164.234	10.371	81.961	5.176
1986	16069630	117.4	2.908	224.040	17.303	22.861	153.657	9.562	70.487	4.386
1987	16322272	124.2	2.878	234.600	17.787	-14.839	142.504	8.731	58.610	3.591
1988	16594102	129.0	3.136	240.737	17.680	-30.655	140.107	8.443	59.409	3.580
1989	16869361	133.6	3.380	256.837	18.485	-39.529	147.204	8.726	62.941	3.731
1990	17126301	133.1	3.463	278.160	19.425	-18.620	158.576	9.259	73.174	4.273
1991	17353191	133.7	3.183	279.534	19.343	-56.967	159.446	9.188	72.551	4.181
1992	17549282	139.1	2.923	284.529	19.503	22.293	157.682	8.985	71.629	4.082
1993	17722905	144.6	3.005	288.874	19.560	42.207	155.667	8.783	71.312	4.024
1994	17897430	150.2	3.484	293.701	19.500	25.788	154.743	8.646	72.340	4.042
1995	18095338	156.0	3.358	305.056	20.066	189.387	154.361	8.530	70.983	3.923

Figure 2. Dataset before Min-Max normalisation

year	population	gdp	cement_co2	co2	ghg_excluding_lucf_per_capita	land_use_change_co2	methane	methane_per_capita	nitrous_oxide	nitrous_oxide_per_capita
1982	0.000000	0.000000	0.279624	0.049759	0.133970	0.228788	0.501015	0.970196	0.408980	0.839868
1983	0.020203	0.018639	0.000000	0.000000	0.000000	0.527588	0.292381	0.773696	0.166609	0.575576
1984	0.038894	0.040924	0.191850	0.047428	0.107176	0.302582	0.341230	0.792584	0.183103	0.576474
1985	0.058433	0.058752	0.322257	0.081254	0.104790	0.208006	0.628426	1.000000	0.571373	0.939838
1986	0.079679	0.070502	0.332915	0.078775	0.084603	0.137344	0.483333	0.859816	0.349499	0.703382
1987	0.102629	0.098055	0.314107	0.129514	0.173426	0.072481	0.330338	0.715820	0.119832	0.465430
1988	0.127321	0.117504	0.475862	0.159001	0.153790	0.045270	0.297457	0.665916	0.135283	0.462137
1989	0.152326	0.136143	0.628840	0.236358	0.301523	0.030002	0.394812	0.714954	0.203581	0.507333
1990	0.175666	0.134117	0.680878	0.338811	0.474032	0.065976	0.550811	0.807312	0.401458	0.669560
1991	0.196276	0.136548	0.505329	0.345413	0.458983	0.000000	0.562745	0.795010	0.389411	0.642023
1992	0.214089	0.158428	0.342320	0.369413	0.488346	0.136366	0.538547	0.759834	0.371582	0.612391
1993	0.229861	0.180713	0.393730	0.390289	0.498807	0.170628	0.510906	0.724831	0.365452	0.595031
1994	0.245714	0.203404	0.694044	0.413482	0.487796	0.142380	0.498230	0.701092	0.385331	0.600419
1995	0.263692	0.226904	0.615047	0.468041	0.591668	0.423851	0.492990	0.680991	0.359090	0.564801

Figure 3. Dataset after Min-Max normalisation

When developing models it was found that models aside from LSTM received reductions in performance when normalisation was performed to the data, however, it was especially critical for LSTM, therefore only the LSTM model will receive normalisation of its dataset. This finding is due to large feature values distorting how these features are weighted within LSTM. This behavior is unlike RF and XGBoost as they utilise decision trees which split data based on feature thresholds and not on distance metrics, making normalisation irrelevant. For the regularisation models, performance generated was more than sufficient without normalisation and therefore was not used.

Machine Learning based Feature Importance

After the dataset is normalised, which features being used and what their importance is can be evaluated. For this purpose, ML algorithms can be effectively used to determine what key factors influence CO_2 emissions. For this study, the ML techniques of RF and XGBoost were used with RF being used similarly to Ran et al (2023). The 116 potential inputs for the models were evaluated to determine which factors are most influential in Australian CO_2

emissions. Both RF and XGBoost have been used for feature importance and indicate which features are most influential in determining the final output of predicted CO_2 emissions for their respective models. Additionally, which features that were determined to be most important by each algorithm are compared to provide a deeper analysis of which features are most influential on CO_2 emissions in Australia.

After the feature importance was completed, an RF and XGBoost dataset was created, being the culmination of the most important features of their respective models. Models aside from these have utilised either the RF or the XGBoost dataset depending on which dataset it performs the best with.

Model Building

To begin the creation of the machine and deep learning models, the dataset needs to be split between training and testing. In similar models, the most common split of the dataset was 80% for training and 20% each for testing, however, due to the limited amount of data with annual time steps between 1982 and 2022 more data has been assigned to training(Chukwunonso et al, 2024; Li & Zhang, 2023; Kong el al, 2022). For these models cross validation has been used to split the data, for the final fold the years 1982 - 2016 have been used as training data while the years 2017-2022 have been used for testing. Further information on how other folds have split the data can be found in the section Cross Validation. This split gives $83.\overline{3}\%$ of data to training and $16.\overline{6}\%$ for testing. The training set is used to train the model and the test is utilised to evaluate the models performance against data it has not seen (Xu & Goodacre, 2018). With the limited amount of data available cross validation was implemented, instead of allocating data towards a dedicated validation set. Through the process of cross validation the models performance during the training is assessed, allowing for the adjustment of suitable hyperparameters to improve performance.

Once the data has been split, the model training can be performed. The chosen algorithms which have been implemented are LSTM, XGBoost and Random Forest, Ridge, Lasso and ElasticNet as LSTM had promising results from other studies and XGBoost and Random Forest have been found to be more suitable for smaller data sizes due to their ensemble nature (Ghazaryan, 2023).

The regularisation models were chosen due to their effectiveness in solving multicollinearity issues originating from the dataset's many similar features. These 6 models are compared to analyse the differences in traditional machine learning compared to deep learning.

Cross Validation

Cross validation is a technique used to evaluate how well a machine learning model performs on unseen data. To perform cross validation the dataset is split into multiple segments which the models are individually trained and tested on. Once all segments have been trained and validated the average results are determined to find the final performance. For time series problems cross validation of how different data splits are generated must be done carefully. Cross validation methodologies such as K-Fold Cross Validation randomly splits data among each fold. This process does not consider the temporal order of the data which is essential for time series problems.

To maintain the temporal order Expanding Window cross validation was implemented, starting each fold with a small training window of the data and gradually expanding it until all training data has been used. This process grants a better understanding on how the developed model would perform against general data. In implementing the cross validation the TimeSeriesSplit function from the sklearn library was utilised, this function is visualised in figure 4. This function creates the Expanding window folds allowing each set to be looped through to train and validate models. Within this study the data was split within 5 folds, the years of data that these folds contain can be seen in Table 1 .

Fold	Training Years	Testing Years
1	1982-1992	1993-1998
2	1982-1998	1999-2004
3	1982-2004	2005-2010
4	1982-2010	2011-2016
5	1982-2016	2017-2022

Table 1. Training and testing splits for each fold

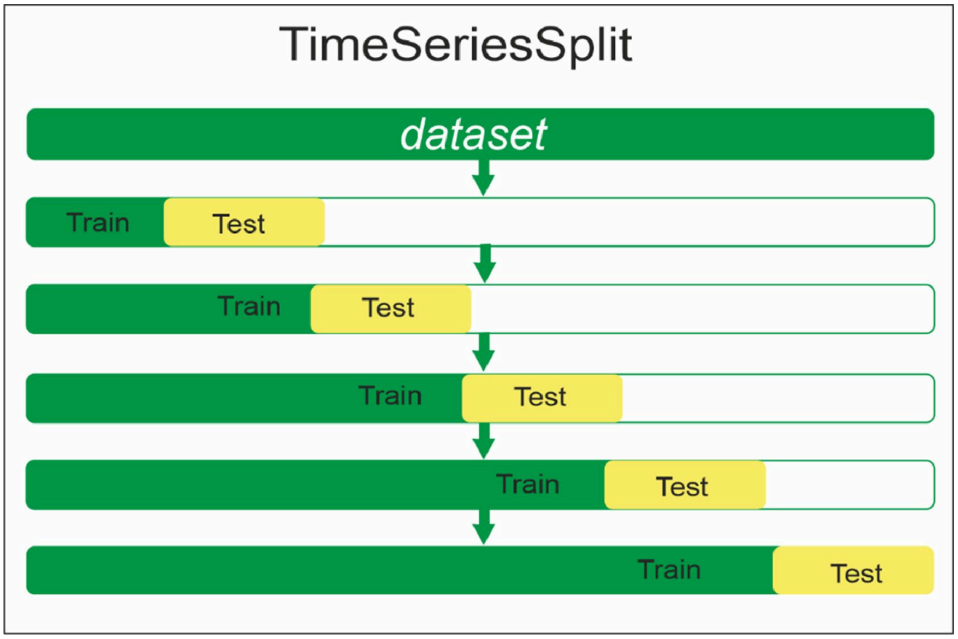


Figure 4. Expanding Window Cross Validation (da Silva et al, 2022)

XGBoost

XGBoost is a machine learning algorithm that is often used for regression and classification tasks (Darmawan, 2023). The model combines multiple predictions from weaker models to create a more accurate final model, this process is known as gradient boosting. Decision trees are used as a base and the model trains them through an iterative process. For each iteration, the error residuals are used to fit in the next model. Finally, the result is created from the weighted summation of all the individual tree predictions.

Hyperparameters for XGBoost include the number of sub trees, maximum tree depth, learning rate, regularisation rates, complexity control and minimum child weight (XGBoost, n.d.). The values of these hyperparameters have been defined during development, creating an optimised XGBoost model. Hyperparameters were chosen to prioritise the performance of the last fold. This is due to this fold containing the most training data and being most likely to obtain the best results.

To perform feature selection for XGBoost initially all 116 features were included within the model. This choice was made to ensure there was no bias in picking certain features based on what was thought to be intuitive rather than allowing the algorithms to determine the patterns to predict CO_2 emissions. Once an initial model was made using all of these features the feature importance was examined and features which had little to no feature importance were removed from the dataset. This methodology was then repeated until model performance decreased from removing further features. At the end of this process there were 23 features remaining to create the best performing XGBoost model.

In selecting hyperparameters the documentation for the XGB regressor was extensively investigated (XGBoost, n.d.). All possible parameters were tested, determining whether changing them from their default value improved performance. In these tests parameters such as booster, nthread and gamma among others were tested. Parameters that improved performance from its default state were included while parameters that reduced performance were not used. Within development, random search was also used to find optimal parameters, however it failed to produce better results than the manual configuration. The final configuration of this model can be found below, this specific configuration was found to produce the best results (Figure 5).

```
xgbModel = XGBRegressor(learning_rate=0.179, max_depth=6, subsample=1, colsample_bytree=1, n_estimators=125, base_score=400)
```

Figure 5. XGBoost configuration

RF

RF is an ensemble model which applies the results of many decision trees to create more accurate predictions. RF begins with creating a decision tree to find a variable and its associated value from the data (Bulagang et al, 2020). This process is continuously repeated until the maximum depth of the tree is reached or when a subset can no longer be further divided. For feature filtering with RF, the training set of each decision tree is around two thirds of the whole training set, the remaining data is known as out-of-bag data, which is extremely important in filtering feature importance.

For feature importance Gini index is then used to determine the optimal subset of features, as it measures how to split the data into smaller subsets (Zhang et al, 2023). It completes this goal by measuring the purity of each feature, meaning how similar the values in the dataset are, and the feature with the lowest index is split.

Key hyperparameters that have been optimised within the random forest are the number of estimator and minimum samples split among other variables, creating a superior model.

Feature selection was performed similarly to XGBoost with the initial RF model being developed from all available features with least performing features being gradually removed. In this case RF benefited from a larger number of features, gaining its best performance with 31 features.

The selection of other hyperparameters also followed a similar process to the XGBoost model. The documentation of the RandomForestRegressor was extensively examined and all possible parameters were tested to validate whether changing them to other values from their default would improve their performance (scikit-learn, n.d.). Parameters such as oob_score, n_jobs and ccp_alpha were tested to discover their effect on the model's performance. The most beneficial parameters found were n_estimators, min_samples_split and max_leaf_nodes, while random_state was used to create reproducible results. The best found RF configuration can be found below (Figure 6).

```
rf_regressor = RandomForestRegressor(n_estimators=100, random_state=42, min_samples_split=4, max_leaf_nodes=12)
```

Figure 6. RF Configuration

LSTM

LSTM deep learning model, being an improved version of the RNN, is able to handle long term dependencies (Li & Zhang, 2023). LSTM directly addresses RNN's issues with gradient explosion or vanishing gradients, by incorporating an input gate, forest gate and output gate. These gates give LSTM the ability to retain and discard information as needed, making it extremely effective in training against time series data. Figure 7 below details the structure of LSTM in which the input gate, forest gate and output gate are labelled as i, f and o respectively. C is the memory unit, X is the input data and H is the unit's state. The input gate updates a cell's state and finds what information should be remembered. The forget gate determines what information should be removed or retained. Finally, the output gate sets the value of the next state.

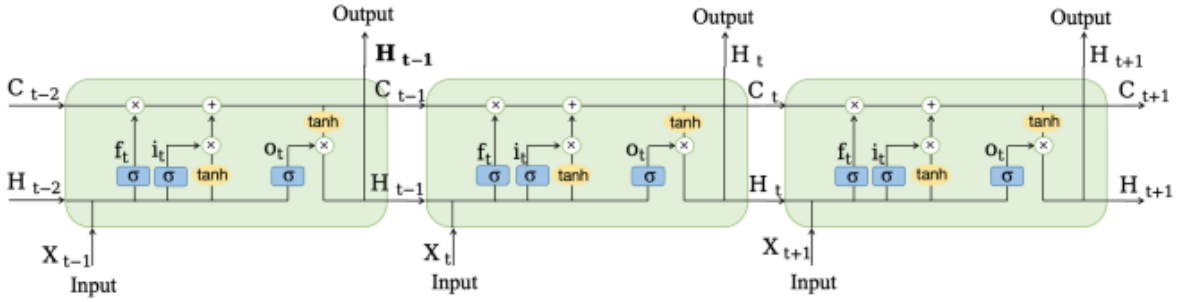


Figure 7. Architectural structure of LSTM (Li & Zhang, 2023)

Hyperparameters that have been continually adjusted during development for LSTM are the number of layers, loss function, optimiser, epochs and batch size. The number of layers for the purposes of predicting CO_2 emissions is commonly 3, with each layer having 50 nodes except for the output. Mean squared error has been used as the loss function, as it is commonly used for regression problems. The Adam optimiser has additionally been used as it has shown to be effective in similar studies (Li & Zhang, 2023). Epochs, batch size and other hyperparameters have been defined and tuned during development to ensure optimal performance.

$$K_E(x_i, x') = \sigma_f^2 \exp\left(-\frac{(x_i - x')}{\sigma_l}\right)$$

Figure 8. Exponential kernel function formula (Akkaya & Akkaya, 2023)

In developing the LSTM model, due to the inability to produce reproducible results as well as a limited amount of time it was difficult to find the best performing parameters. Additionally, LSTM's structure can be configured in a multitude of ways, especially with the number of and types of layers, meaning not all possible configurations could be tested. Due to this inconsistency, feature selection was done by utilising the same dataset that was used for RF, this choice was made as the RF dataset had a greater number of features compared to XGBoost meaning LSTM like other deep learning models would benefit for more data to predict patterns in CO_2 emissions and achieve better results. From testing the most consistent model configuration can be found below (Figure 9). It contains only 1 LSTM layer as more layers were found to have worse performance and was trained over 100 epochs with a batch_size of 16. These smaller parameters were likely most effective due to the small dataset as too many epochs or layers would cause overfitting.

```
# Build the LSTM model
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(x_train_lstm.shape[1], x_train_lstm.shape[2])))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse', metrics=[metrics.mean_squared_error])

model.summary()

# Train the model
history = model.fit(x_train_lstm, y_train, epochs=100, batch_size=16, validation_data=(x_test_lstm, y_test))
```

Figure 9. LSTM Configuration

Ridge Regression

Ridge regression (RR) is a regularisation technique which prevents the standard linear regression from overfitting. The model uses L2 regularisation to solve the multicollinearity problem faced with linear regression. This problem is when independent variables within a model are correlated it is difficult to determine the unique effect of each independent variable on the dependent variable, being CO_2 (Nur at el, 2024). The model produces more stable coefficient estimates by reducing variability of coefficients. L2 regularisation adds a penalty term to the loss function, finding the coefficients that minimise the sum of squared errors (Aydın et al, 2025).

When developing all of the three regularised regression models feature selection was performed the same way. Due to these models lacking their own feature importance metrics like RF and XGBoost instead the datasets for these models were compared. It was found that the XGBoost dataset performed better with these models. This result is due to the smaller dataset for XGBoost reducing the complexity of the dataset for these models to handle. The main hyperparameter of this model is the alpha which is the constant that multiplies the L2 term, controlling the regularisation strength. Due to time constraints other parameters such as solver could not be optimised, this limitation applied to all other regularisation models (scikit-learn, n.d.). Though testing the final models configuration can

be found below having an alpha of 20 (Figure 10).

```
ridgeReg = Ridge(alpha=20)
```

Figure 10. RR configuration

Lasso Regression

Lasso Regression (LR) is a similar regularisation technique to RR as it also solves the multicollinearity problem to prevent overfitting. Instead, LR uses L1 regularisation which has a penalty applied to regression coefficients, moving them closer to zero. If a feature reaches zero it is removed, helping LR to serve as variable selection. This penalty adds the absolute value of magnitude to the loss function (Aydın et al, 2025).

In selecting hyperparameters for the LR it offers very similar options to the RR model with the main parameter being the alpha. Instead, this parameter is the constant that multiplies the L1 term, controlling regulisation strength (scikit-learn, n.d.). The final configuration for this model can be seen in Figure 11 with an alpha of 0.5.

```
lasso = Lasso(alpha=0.5)
```

Figure 11. LR Configuration

ElasticNet

ElastiNet regression (ENR) is a combination of both RR and LR, adding both L1 and L2 regularisation to the model. This hybrid serves to be able to perform feature selection similar to LR and stabilise coefficients similar to RR. Overall, ENR serves as a balance between the two previous techniques (Nur et al, 2024).

For the development of the ENR model two key hyperparameters were optimised. These being alpha and l1_ratio. The alpha is the constant that multiplies the penalty terms and the l1_ratio which is the ENR mixing parameter, determining in what ratio is the L1 or L2 ratio applied. The closer this value is to 0 the more L2 penalty is used while the closer the value is to 1 the more L1 penalty is used (scikit-learn, n.d.). The best configuration of the model found in this study can be seen below (Figure 12).

```
elastic_net = ElasticNet(alpha=0.5, l1_ratio=0.3)
```

Figure 12. ENR Configuration

Model Evaluation

Once the models were fully trained, they were tested against evaluation criteria to determine which model is most effective for predicting Australian CO_2 emissions. The criteria used in this project was MSE, RMSE, MAE and R^2. These metrics are defined as follows (Figure 4):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} | \hat{y}_i - y_i |}{n}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \frac{\sum_{j=1}^{n} y_j}{n}\right)^2}$$

Figure 13. Evaluation Criteria Formulas (Li & Zhang, 2023)

MSE determines how large the error of the model is and its variance, the smaller the value the more accurate the model is (Setiadi et al, 2023).

RMSE is the root of MSE and similarly, the smaller the value the more accurate the model, however, RMSE determines the standard deviation of the error instead, scaling the value down for easier comparison with the data.

MAE calculates how close a model's prediction is from the true value (Kotu & Deshpande, 2019). For example, if a model's MAE equals 1000, this means that the model's prediction is likely to be 1000 away from the true value, making a lower value the better result.

R^2 measures how well the regression line fits in the data, with 1 being a perfect fit and 0 meaning there is no relationship between the dependent and independent variables (Cheng et al, 2014). Unlike the other evaluation criteria being used, a higher value indicates a better model.

A custom accuracy metric was added to evaluate the models. It compares the models predictions with the actual results for each time step, calculating what percentage of the models predictions are correct. If a prediction is within 5 of the actual value then the prediction is considered to be correct. This accuracy metric was only applied onto the final fold for the testing results. The implementation of this metric can be seen below (Figure 14).

```
results = []
print(test['xgb_co2_pred'])
print(test['co2'])
for year in test['co2'].keys():
    if abs(test['xgb_co2_pred'][year] - test['co2'][year]) < 5:
        results.append(1)
    else:
        results.append(0)
```

```
print('Accuracy: ', sum(results)/len(results) * 100)
```
Figure 14. Accuracy metric implementation

Research Methodology Summary

Overall, through this in-depth process in creating these machine and deep learning models, meaningful progress has been made in determining the most optimal methodology to create CO_2 emissions prediction models. In using Min-max normalisation, and comparing RF XGBoost for feature importance, LSTM, RF, XGBoost, RR, RL and ENR for model building and MSE, RMSE, MAE and R^2 for model evaluation, valuable research has been made in defining the most suitable process to predict Australian CO_2 emissions.

Results and Discussion

The results obtained from the 6 generated models are presented in this section. First each method is analysed individually, followed by a comparison between each model's effectiveness in predicting Australian CO_2 emissions.

XGBoost

The XGBoost model achieved an R^2 score of 0.908 when forecasting against unseen data with an RMSE 3.899. The performance metrics for both training and testing within all folds can be seen within Tables 2 and 3. Additionally, the feature importance scores have also been detailed in Figure 15 highlighting which variables were most influential towards the performance of the model. The most influential of these features being population, cement_co2 and coal_co2. This result aligns with expectations as population would be expected to correlate with CO^2 emissions along with fossil fuel consumption. These performance metrics indicate good predictive ability in evaluating CO_2 emissions as visualised in Figure 16. As expected training metrics are significantly better than testing scores, however it can be seen that despite differences across each fold are small the training metrics improve when the model is trained on more data. In terms of the testing metrics, as the last folds performance was prioritised, all of the other folds had lackluster performance. Similar to the training metrics, the XGBoost does achieve better and better scores as it has access to more data, allowing it to make more accurate predictions.

	MSE	RMSE	MAE	R^2
Fold 1	0.0000002825	0.0005315	0.0003802	0.9999999996
Fold 2	2.5495e-07	0.0005049	0.0003797	0.9999999998
Fold 3	2.2989e-07	0.0004795	0.0003560	0.99999999992
Fold 4	2.8922e-07	0.0005378	0.0004112	0.99999999993
Fold 5	2.4865e-07	0.0004986	0.0003848	0.99999999995

Table 2: Training XGBoost metrics

	MSE	RMSE	MAE	R^2	Accuracy(%)
Fold 1	836.4721	28.9218	24.4788	-2.5255	
Fold 2	906.0657	30.1009	27.1925	-4.4956	
Fold 3	505.1333	22.4752	19.2155	-7.7650	
Fold 4	36.8733	6.0723	4.9530	-0.1912	
Fold 5	15.2044	3.8993	3.7140	0.9084	66.6%

Table 3: Testing XGBoost metrics

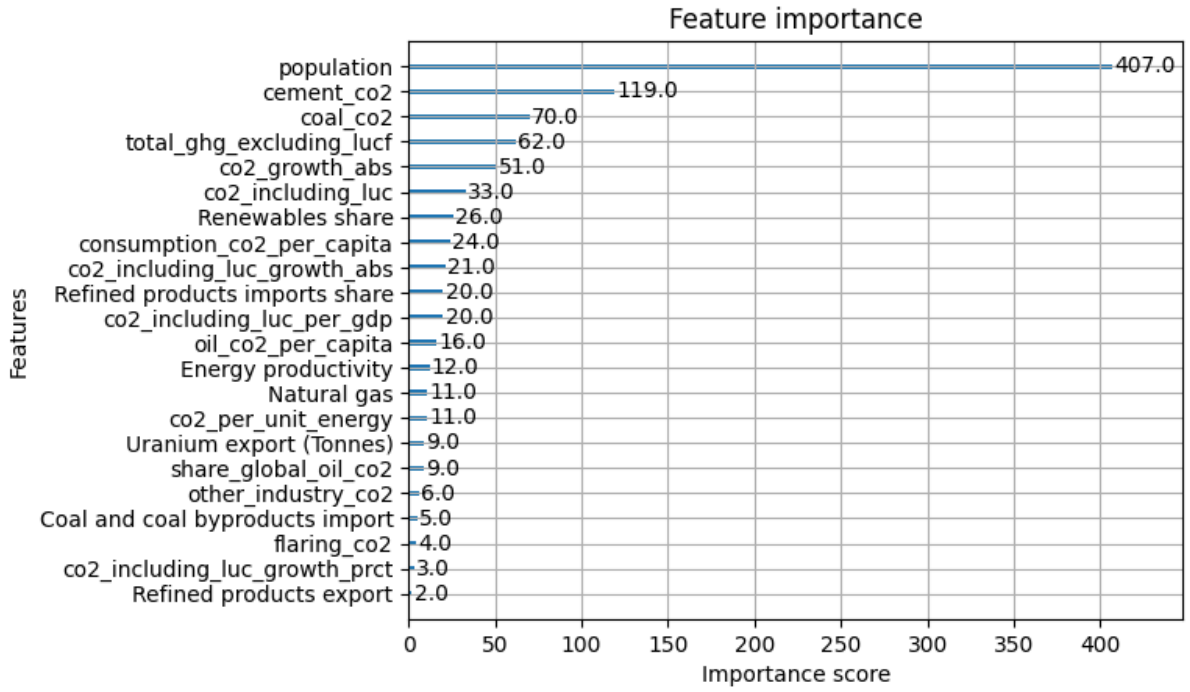


Figure 15: Feature Importance for XGBoost model

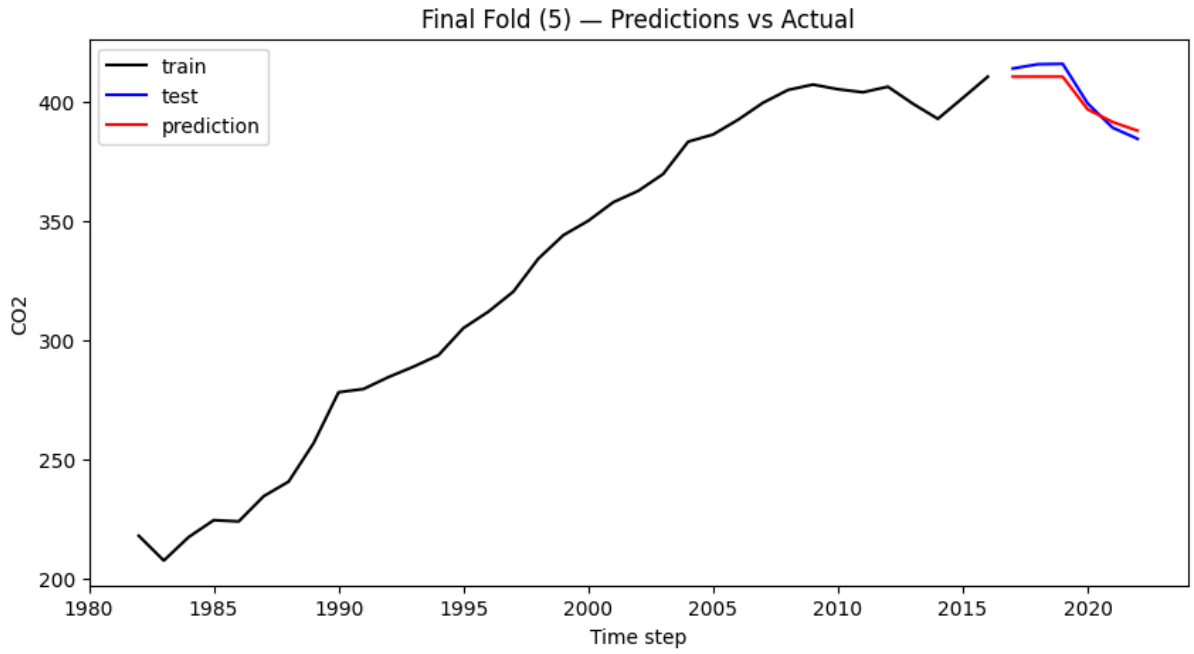


Figure 16: Graphed predictions of 5th fold for XGBoost model

RF

The RF model achieved an R^2 score of 0.730 when forecasting against unseen data with an RMSE 6.692. The performance metrics for both training and testing within all folds can be seen within Tables 4 and 5. Additionally, the feature importance scores have also been detailed in Figure 17 highlighting which variables were most influential towards the performance of the model. Interestingly, the feature importance graph differs quite greatly from the XGBoost model. Instead the most important features are Energy intensity, share_global_culmilative_cement_co2 and energy_per_gdp. This result indicates that the RF benefits more from economic factors rather than fossil fuel consumption when compared to XGBoost. These performance metrics indicate decent predictive ability in evaluating CO_2 emissions however ultimately worse predictions when compared to the XGBoost model, visualised in Figure 18. Similar to XGBoost, the model's performance improves as it is given more data in both train and test set. Interestingly, when comparing RF and XGBoost testing results it can be noticed that they follow a similar pattern on how the model improves. Both models obtain worse metrics when moving from the 1st to 2nd fold and afterwards they continuously improve until their best performance is achieved in the final fold. This similarity can be attributed to them both utilising decision trees to obtain their predictions, granting comparable behaviour.

	MSE	RMSE	MAE	R^2
Fold 1	25.5622	5.0559	4.0944	0.9639
Fold 2	29.8338	5.4620	3.2844	0.9808
Fold 3	17.2948	4.1587	2.6795	0.9941
Fold 4	10.1037	3.1786	2.1816	0.9976

Fold 5	6.2259	2.4952	1.7089	0.9987

Table 4: Random Forest training metrics

	MSE	RMSE	MAE	R^2	Accuracy(%)
Fold 1	1495.9688	38.6778	35.1772	-5.3051	
Fold 2	3000.6008	54.7777	52.7298	-17.1996	
Fold 3	912.3432	30.2050	29.2424	-14.8308	
Fold 4	27.9279	5.2847	4.2703	0.09776	
Fold 5	44.7795	6.6917	6.2599	0.7302	33.3%

Table 5: Random Forest testing metrics

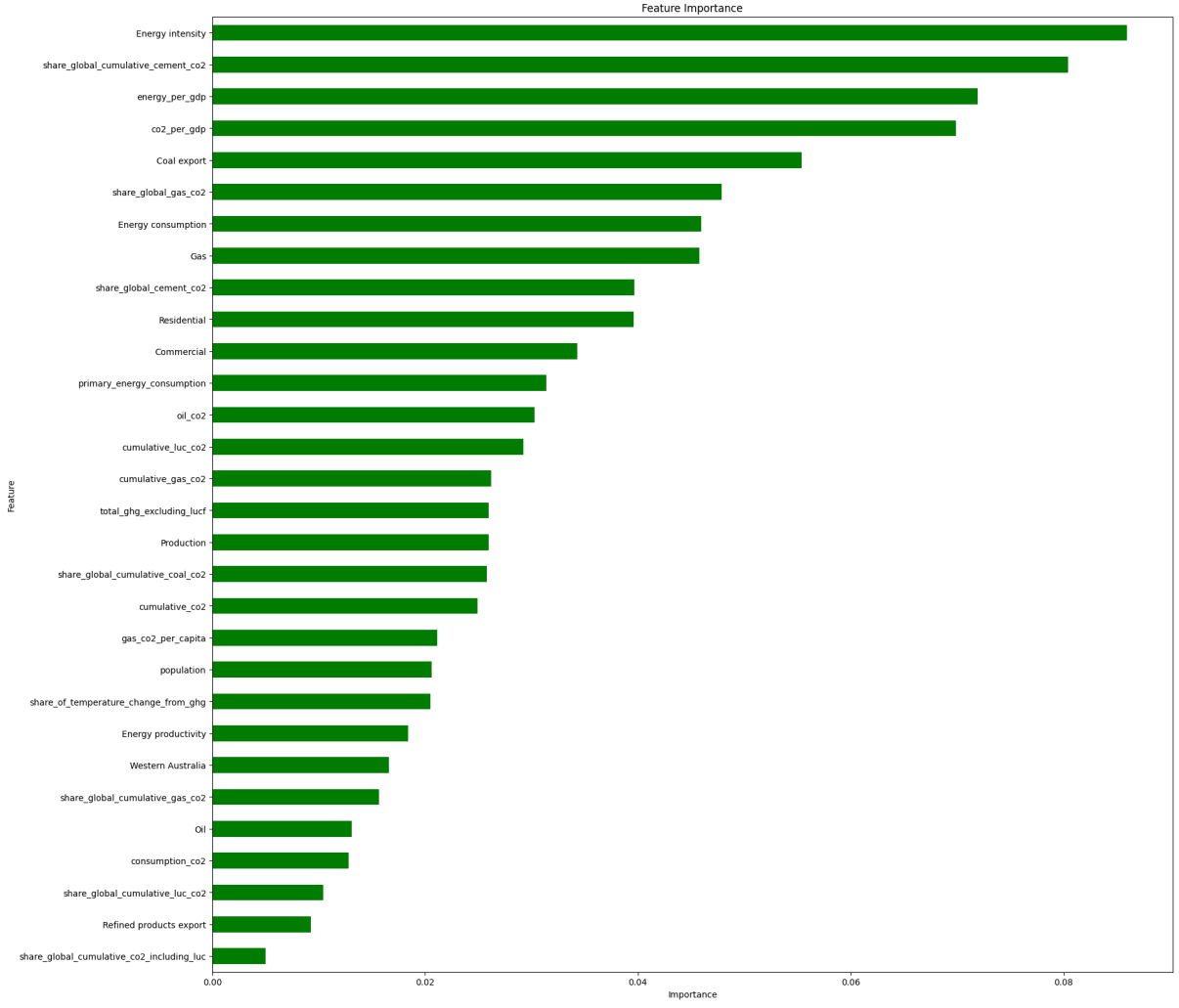
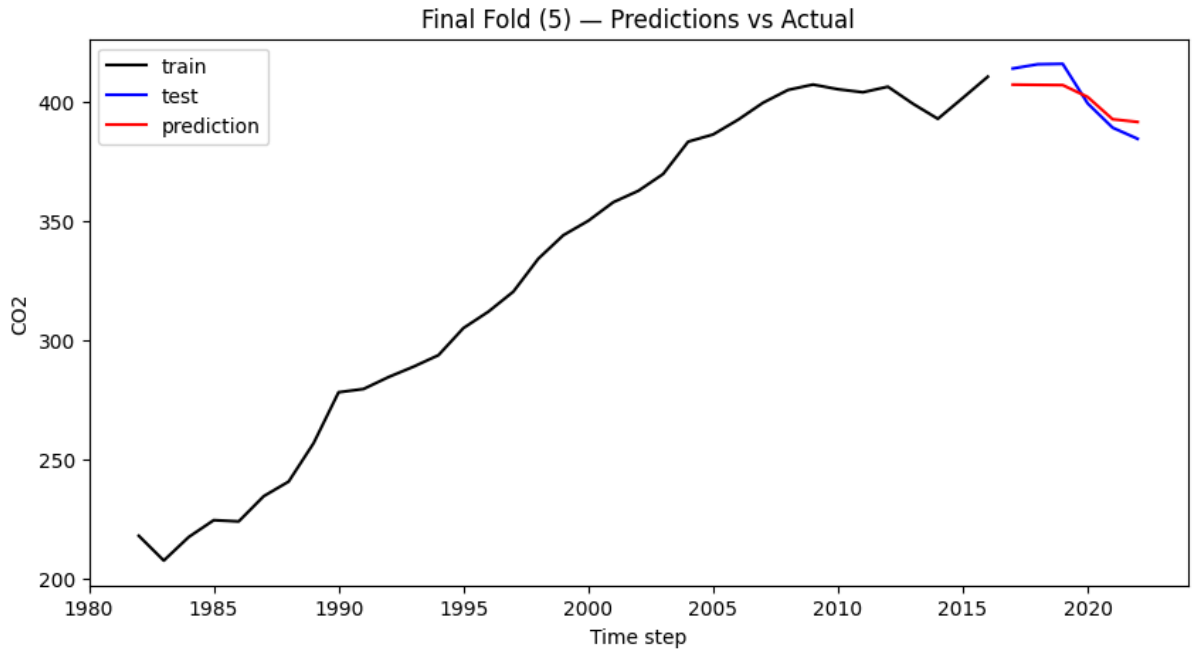


Figure 17: Feature Importance for Random Forest model



Feature 18: Graphed predictions of 5th fold for Random Forest model

LSTM

When recording the results for the LSTM model a different methodology had to be implemented. This is due to the inherent random nature of LSTM and there being no way to obtain reproducible results within the environment used in this study. To obtain valid results for this model the model was trained and tested 30 times and the average result was calculated. Additionally, due to the LSTM models input being normalised, this normalisation needed to be inverted to allow for metrics to be comparable to the other models. The results from this process can be found below. The LSTM model obtained an R^2 of 0.7744 and an RMSE of 5.7575 indicating better results than the RF but failing to improve upon the XGBoost model. Interestingly, LSTM was able to gain decent results within the first fold of the dataset, potentially due to simple patterns appearing in the initial fold compared to later folds introducing more diverse patterns. The training and testing metrics of this model can be found in Tables 6 and 7 and a visualisation of this model can be found in Figure 19.

	MSE	RMSE	MAE	R^2
Fold 1	13.8708	3.6575	3.0774	0.9804
Fold 2	19.1095	4.3461	3.6570	0.9877
Fold 3	17.9461	4.2077	3.4432	0.9939
Fold 4	30.0261	5.4246	4.3809	0.9930
Fold 5	19.5152	4.3793	3.5916	0.9958

Table 6: LSTM training metrics

	MSE	RMSE	MAE	R^2	Accuracy(%)

Fold 1	56.9334	7.2953	6.0855	0.7600	
Fold 2	197.0633	12.9979	11.9295	-0.1953	
Fold 3	250.5402	15.3764	12.6802	-3.3473	
Fold 4	1228.6844	34.5453	33.5461	-38.6939	
Fold 5	37.4511	5.7575	4.8436	0.7744	50

Table 7: LSTM testing metrics

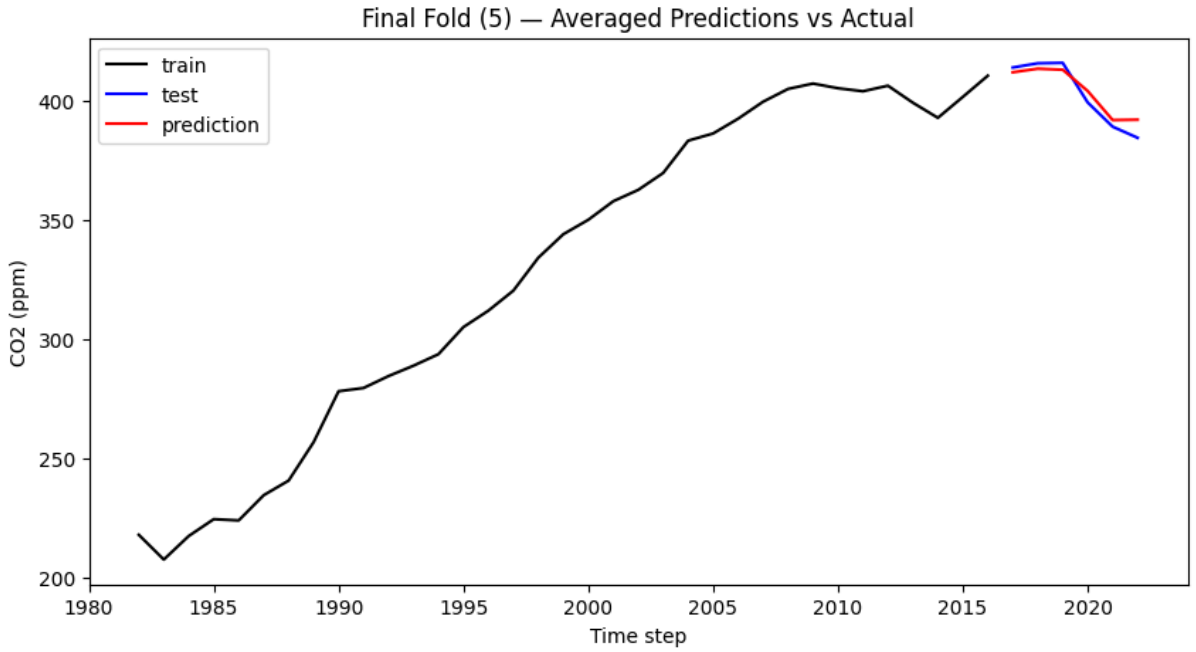


Figure 19: Graphed predictions of 5th fold for LSTM model

RR

The RR model achieved an R^2 score of 0.9768 and RMSE of 1.5286 when forecasting against unseen data. The exact results of the model against all folds for training and testing can be seen with Tables 8 and 9 and visualisation can be seen in figure 20. The performance metrics indicate very strong predictive ability in evaluating CO_2 emissions with obtaining good performance across most folds. Unlike previous models which could only achieve serviceable performance across 1 or 2 folds the Ridge model only performed poorly within the 4th fold. This great performance offers a significant improvement compared to previous models, granting extremely accurate forecasting.

	MSE	RMSE	MAE	R^2
Fold 1	0.8910	0.9439	0.8500	0.9987
Fold 2	0.7777	0.8819	0.7994	0.9995

Fold 3	0.7899	0.8888	0.7682	0.9997
Fold 4	0.6739	0.8209	0.6308	0.9998
Fold 5	2.3366	1.5286	1.1982	0.9995

Table 8: Training Ridge Regression metrics

	MSE	RMSE	MAE	R^2	Accuracy(%)
Fold 1	26.7294	5.1700	3.7356	0.8873	
Fold 2	17.0463	4.1287	3.3102	0.8966	
Fold 3	2.1565	1.4685	1.0012	0.9626	
Fold 4	67.3827	8.2087	6.6919	-1.1769	
Fold 5	3.8430	1.9603	1.6756	0.9768	100

Table 9: Testing Ridge regression metrics

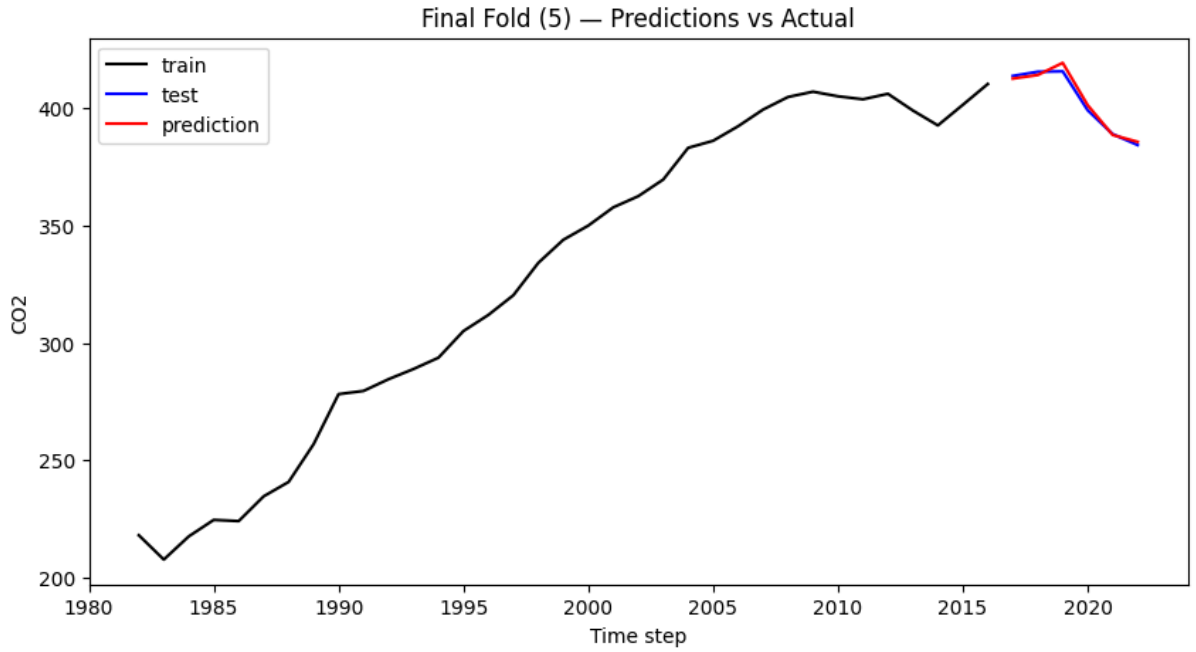


Figure 20: Graphed predictions of 5th fold for RR model

LR

The LR model achieved an R^2 score of 0.9852 and a RMSE 1.5667 when forecasting within the final fold. The results of this model are seen below in tables 10 and 11 and a visualisation of this result can be seen in Figure 21. The performance metrics surpass the RR model giving even more accurate predictions. This result indicates that the L1 regularisation performs better than the L2 regression due to LR having the capability to remove features and create the most optimal series of coefficients rather than only creating more stable coefficients like the RR model.

	MSE	RMSE	MAE	R^2
Fold 1	0.7971	0.8928	0.7222	0.9989
Fold 2	0.9990	0.9995	0.8715	0.9994
Fold 3	1.0327	1.0162	0.8438	0.9996
Fold 4	1.0044	1.0022	0.7922	0.9998
Fold 5	2.7220	1.6498	1.2695	0.9994

Table 10: Training LR metrics

	MSE	RMSE	MAE	R^2	Accuracy(%)
Fold 1	9.3539	3.0584	2.2969	0.9606	
Fold 2	2.7578	1.6607	1.1391	0.9833	
Fold 3	2.2798	1.5099	1.0783	0.9604	
Fold 4	54.2107	7.3628	5.5185	-0.7513	
Fold 5	2.4546	1.5667	1.4765	0.9852	100

Table 11: Testing LR metrics

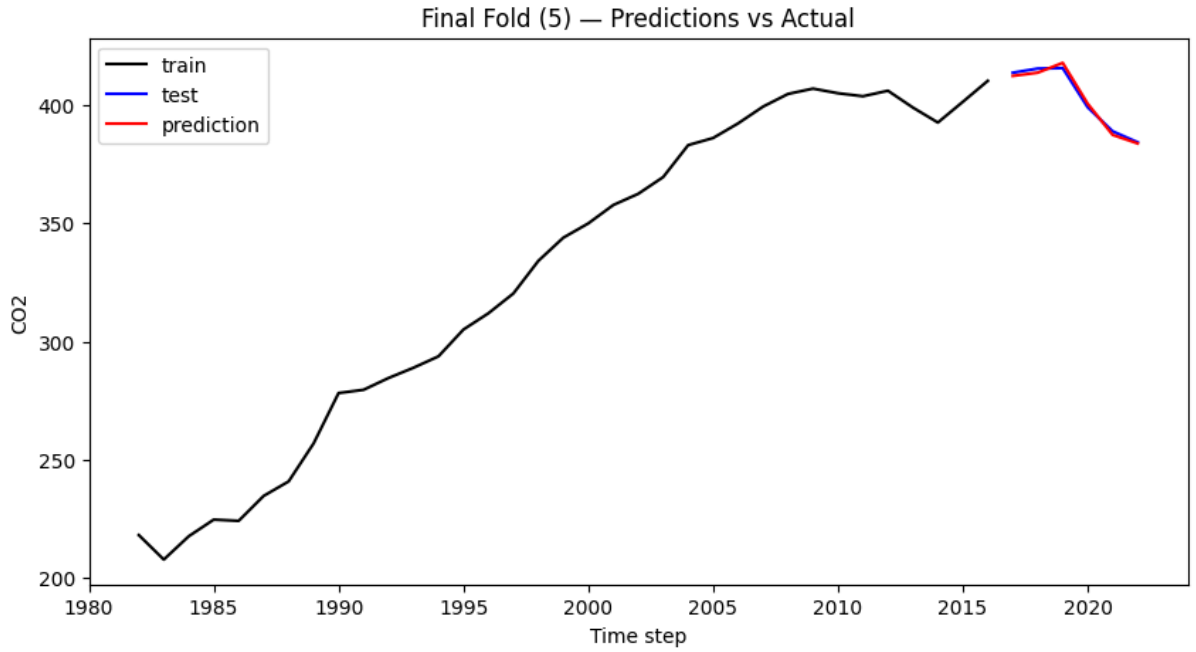


Figure 21: Graphed predictions of 5th fold for LR model

ENR

The ENR model achieved a R^2 score of 0.9659 and an RMSE of 2.3775 when forecasting against unseen data. More indepth results of this model's performance can be found in Tables 12 and 13 for both training and testing and visualisation can be seen in Figure 21.

The performance of this model is very strong, however, it performs slightly worse than the previous RR and LR models. This result could be due to the added configuration complexity of the ElasticNet compared to the other models, making it more difficult to determine the optimal configuration of the model. Interestingly, all the three regularisation models performed poorly against Fold 4. The strong performance across all regularisation techniques indicate that they are particularly suited to predicting the CO_2 emissions in Australia. This performance is due to the shape of the available data. With the small number of data points being 41 compared to a similar number of features, the other model types struggle to prevent overfitting, however, these models excel at preventing overfitting through their penalty terms making them achieve the best performance.

	MSE	RMSE	MAE	R^2
Fold 1	0.7060	0.8403	0.6836	0.9990
Fold 2	0.9953	0.9976	0.8345	0.9994
Fold 3	0.9063	0.9520	0.7343	0.9997
Fold 4	0.8391	0.9160	0.6614	0.9998
Fold 5	2.4983	1.5806	1.2464	0.9995

Table 12: Training ElasticNet metrics

	MSE	RMSE	MAE	R^2	Accuracy(%)
Fold 1	66.8465	8.1760	7.6784	0.7183	
Fold 2	10.2058	3.1947	2.7990	0.9381	
Fold 3	4.4777	2.1160	1.6369	0.9223	
Fold 4	54.4771	7.3809	5.2044	-0.7599	
Fold 5	5.6526	2.3775	1.7441	0.9659	100

Table 13: Testing ElasticNet metrics

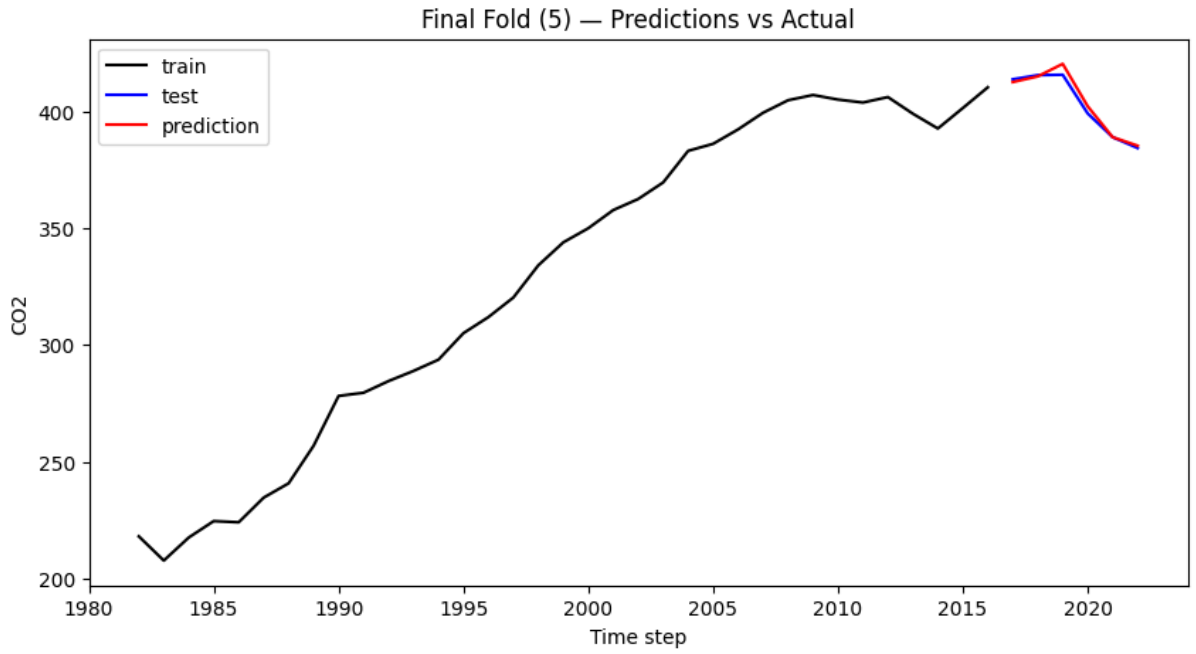


Figure 22: Graphed predictions of 5th fold for ENR model

Conclusion and Recommendations

This study explored the application of ML algorithms to predict CO_2 emissions within Australia utilising data found from OWID as well as the Australian Energy Update. The ML algorithms of RF, XGBoost, LSTM, RR, LR and ENR were compared to determine which performed the best predictions. Additionally, RF's and XGBoost feature importance evaluation allowed for an understanding of which economic and environmental factors most impacted CO_2 emissions within Australia. RF found energy consumption and economic activity to be the key drivers of CO_2 emissions with Energy intensity, share_global_culmilative_cement_co2 and energy_per_gdp being the most important features. XGBoost instead found population, cement_co2 and coal_co2 to be the most important factors, highlighting differences in how these models calculate their importances yet still relying on economic and energy consumption factors. The results of this study can provide an outlook on the most significant factors influencing CO_2 emissions with these findings having the ability to assist policy makers to develop strategies to mitigate CO_2 emissions within Australia. Furthermore, the methodologies and findings can be implemented within other countries allowing other researchers insight on how to develop models for their own region and discover their own most influential factors.

Despite these results, there were some limitations encountered in this study. The lack of available data was a significant issue in developing models. With only 41 time stamps separated at yearly intervals there is not enough data to provide the necessary complexity to models, especially for LSTM and other deep learning models. This yearly interval also fails to find seasonal patterns, reducing the models' abilities to account for these complexities. Improvements in the available data by collecting over a longer period of time or granting quarterly or even monthly data would greatly improve the models' predictive precision. Other limitations were due to limited time available, meaning not all hyperparameters or feature

combinations could be explored, especially when concerning the LSTM model. This limit reduces to what extent model performance could be optimised as well as the number of algorithms which could be tested within this study. Due to time constraints, extensive optimisation of the regularisation techniques was not possible, despite this limitation they produced excellent results. In the future, studies could focus their efforts upon these models, fully exploring their potential.

Additionally, Future studies could further explore different deep learning structures to more comprehensively find their impact on model performance. Other models could also be explored to compare their predictive ability to the algorithms discussed within this study giving further understanding on which models are suitable.

References

Akanbi, O, A., Amiri, I, S., & Fazeldehkordi, E. (2015). *A Machine-Learning Approach to Phishing Detection and Defence.* Elsevier. https://doi.org/10.1016/B978-0-12-802927-5.00004-6.

Akkaya, E.K., & Akkaya, A.V. (2023). Development and performance comparison of optimized machine learning-based regression models for predicting energy-related carbon dioxide emissions. *Environmental Science and Pollution Research,* 30(2023), 122381–122392. https://doi.org/10.1007/s11356-023-30955-1

Al-Nefaie, A. H., & Aldhyani. T. H. H. (2023). Predicting CO_2 Emissions from Traffic Vehicles for Sustainable and Smart Environment Using a Deep Learning Model. *Sustainability,* 15(9), 7615. https://doi.org/10.3390/su15097615

Australian Government. (2024). *Quarterly Update of Australia's National Greenhouse Gas Inventory: March 2024*. https://www.dcceew.gov.au/sites/default/files/documents/nggi-quarterly-update-march-2024.pdf

Aydın, Y., Bekdaş, G., & Nigdeli, S. M. (2025). *Dimensioning of the Retaining Wall Using Linear Regression, Ridge Regression and Lasso Regression*. New Technologies, Development and Application VIII, Cham.

Bulagang, A, F., Weng, N, G., Mountstephens, J., & Teo, J. (2020). A review of recent approaches for emotion classification using electrocardiography and electrodermography signals*. Informatics in Medicine Unlocked*, *20*(2020). https://doi.org/10.1016/j.imu.2020.100363

Cheng, C, L., Shalabh., & Garg, G. (2014). Coefficient of determination for multiple measurement error models. *Journal of Multivariate Analysis, 126*(2014), 137-152. https://doi.org/10.1016/j.jmva.2014.01.006

Chukwunonso, B. P., Al-Wesabi, I., Shixiang, L., AlSharabi, K., Al-Shamma'a, A. A., Hussein Farh, H. M., Saeed, F., Kandil, T., & Al-Shaalan, A. M. (2024). Predicting carbon

dioxide emissions in the United States of America using machine learning algorithms. *Environmental Science and Pollution Research, 31*(2024), 33685-33707. https://doi.org/10.1007/s11356-024-33460-1

Darmawan, H., Yuliana, M., & Zen Samsono Hadi, M. (2023). GRU and XGBoost Performance with Hyperparameter Tuning Using GridSearchCV and Bayesian Optimization on an IoT-Based Weather Prediction System. *International Journal on Advanced Science, Engineering and Information Technology, 13*(3), 851-862. https://doi.org/10.18517/ijaseit.13.3.18377

da Silva, D. G., Geller, M. T. B., Moura, M. S. d. S., & Meneses, A. A. d. M. (2022). Performance evaluation of LSTM neural networks for consumption prediction. *e-Prime - Advances in Electrical Engineering, Electronics and Energy, 2,* 100030. https://doi.org/https://doi.org/10.1016/j.prime.2022.100030

Department of Climate Change, Energy, the Environment and Water. (2024). *Australian Energy Update 2024*. https://www.energy.gov.au/publications/australian-energy-update-2024

Kong, F., Song, J. & Yang, Z. (2022). A daily carbon emission prediction model combining two-stage feature selection and optimized extreme learning machine. *Environmental Science and Pollution Research,* 29(2022), 87983–87997. https://doi.org/10.1007/s11356-022-21277-9

Kotu, V., & Deshpande, B. (2019). *Data Science (Second Edition)*. Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-814761-0.00012-5

Kumari, S., & Singh, S.K. (2022). Machine learning-based time series models for effective CO_2 emission prediction in India. *Environmental Science and Pollution Research*, 30(2023), 116601–116616. https://doi.org/10.1007/s11356-022-21723-8

Le, T.T., Sharma, P., Osman, S.M., Dzida, M., Nguyen, P.Q.P., Tran, M.H., Cao, D.N., & Tran, V.D. (2024). Forecasting energy consumption and carbon dioxide emission of Vietnam by prognostic models based on explainable machine learning and time series. *Clean Technologies Environmental Policy,* 1(2024). https://doi.org/10.1007/s10098-024-02852-9

Li, X., & Zhang, X. (2023). A comparative study of statistical and machine learning models on carbon dioxide emissions prediction of China. *Environmental Science and Pollution Research, 30*(2023), 117485-117502. https://doi.org/10.1007/s11356-023-30428-5

Nur, A., Jaya, A., & Siswanto, S. (2024). Comparative Analysis of Ridge, LASSO, and Elastic Net Regularization Approaches in Handling Multicollinearity for Infant Mortality Data in South Sulawesi. *Jurnal Matematika Statistika dan Komputasi, 20,* 311-319. https://doi.org/10.20956/j.v20i2.31632

Ran, Q., Bu, F., Razzaq, A., Ge, W., Peng, J., Yang, X., & Xu, Y. (2023). When will China's industrial carbon emissions peak? Evidence from machine learning. *Environmental Science and Pollution Research,* 30(2023), 57960–57974.

https://doi.org/10.1007/s11356-023-26333-6

Ritchie, H., Rosado, P., & Roser, M. (2023). *CO_2 and Greenhouse Gas Emissions.* https://ourworldindata.org/co2-and-greenhouse-gas-emissions#citation

Safonova, A., Ghazaryan, G., Stiller, S., Main-Knorn, M., Nendel, C., & Ryo, M. (2023). Ten deep learning techniques to address small data problems with remote sensing. *International Journal of Applied Earth Observation and Geoinformation, 125,* 103569. https://doi.org/https://doi.org/10.1016/j.jag.2023.103569

Schulz, E., Speekenbrink, M., & Krause, A. (2018). A tutorial on Gaussian process regression: Modelling, exploring and exploiting functions. *Journal of Mathematical Psychology, 85*(2018), 1-16. https://doi.org/10.1016/j.jmp.2018.03.001

scikit-learn. (n.d.). *ElasticNet*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

scikit-learn. (n.d.). *Lasso.* https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

scikit-learn. (n.d.). *RandomForestRegressor*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

scikit-learn. (n.d.). *Ridge*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

Setiadi, D, R, I, M., Rustad, S., Andono, P, N., & Shidik, G, F. (2023). Digital image steganography survey and investigation (goal, assessment, method, development, and dataset). *Signal Processing, 206*(2023). https://doi.org/10.1016/j.sigpro.2022.108908

Tabassum, S., & Altaf, S. (2025). Addressing multicollinearity in log Birnbaum–Saunders regression model: a ridge regression estimation approach. *International Journal of Data Science and Analytics*. https://doi.org/10.1007/s41060-025-00861-5

XGBoost. (n.d.). *XGBoost Parameters*. XGBoost. https://xgboost.readthedocs.io/en/stable/parameter.html

Xu, Y., & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing, 2*(2018), 249-262. https://doi.org/10.1007/s41664-018-0068-2

Yao, X., Zhang, H., Wang, X., Jiang, Y., Zhang, Y., & Na, X. (2024). Which model is more efficient in carbon emission prediction research? A comparative study of deep learning models, machine learning models, and econometric models. *Environmental Science and Pollution Research,* 31(2024), 19500–19515. https://doi.org/10.1007/s11356-024-32083-w

Zhang, Y., Nie, B., Du, J., Chen, J., Du, Y., Jin, H., Zheng, X., Chen, X., & Miao, Z. (2023). Feature selection based on neighborhood rough sets and Gini index. *PeerJ Computer Science, 9*(2023). https://doi.org/10.7717/peerj-cs.1711

Appendix

Appendix 1: Project Communication Log

Project Title:	What is the predicted future CO2 emissions for Australia using machine and deep learning models?		
Student Name:	Alexandar Nikolic	Supervisor Name:	Dr Safa Ghannam
Date	Event	Topic of Communication	Outcome
1/08/2024	Online Meeting	Discuss AI Project	Brainstorm possible AI project ideas.
9/08/2024	Email	Project Name	Determined name and goals of the project.
13/08/2024	Online Meeting	Research questions discussion	Further develop research questions into developing predictive models.
27/08/2024	Online Meeting	Prediction model topic	Determine creating prediction models to predict Australian CO_2 emissions.
13/09/2024	In-person Meeting	Task 2 Discussion	Determine where to get articles for the Literature Review
25/09/2024	Email	Task 2 question	Determined that brief definitions of models is sufficient in literature review.
27/09/2024	In-person Meeting	Task 2 Review	Received feedback on Task 2 draft
11/10/2024	In-person Meeting	Model selection	Determined which models to use for study, being LSTM, XGBoost among other options.
21/10/2024	Email	Database question	Determined that the Australian energy

			update contained enough data to develop models.
25/10/2024	In-person Meeting	Research Proposal	Discussed the completion on Task 3 and what details need to be included.
7/08/2025	In-person Meeting	Catch up	Caught up on any progress made on models and what the next steps are for development.
8/08/2025	Microsoft Teams message	Feature selection	Given example feature list.
12/08/2025	Microsoft Teams message	Clarification of LSTM model	Given example of LSTM model.
14/08/2025	Microsoft Teams message	Further clarification on LSTM model	Given an example paper of LSTM with good results, also suggested to try RF and XGBoost models.
18/08/2025	Microsoft Teams message	XGBoost hyperparameter tuning	Suggested to use RandomisedSearch to find optimal parameters for XGBoost.
21/08/2025	In-person Meeting	Dataset feature selection	Suggested to update features of the dataset and given example papers.
29/08/2025	Microsoft Teams message	Project update	Informed supervisor that I achieved good performance on the XGBoost model.
11/09/2025	In-person Meeting	Cross Validation implementation	Discussed Cross validation methods for models.
18/09/2025	Microsoft Teams message	Cross Validation	Suggested to implement Expanding Window CV.
3/10/2025	Online Meeting	Cross	Discussed how to

		Validation/Model Configuration	record consistent LSTM results, elaborated on the purpose of CV.
15/10/2025	Microsoft Teams message	Time Series Split	Given example to implement time series split.
22/10/2025	Microsoft Teams message	Regularisation Models	Suggested to try ElasticNet, Ridge regression and Lasso regression.
24/10/2025	Online Meeting	Finalising Project	Discussing final tweaks of project.