ESLSCA

School of Computing and Digital Tech.

Time Series Forecasting Using Deep Learning

Dr. Amal Mahmoud

Spring 2024

Ghyth Ahmed Elmehrik

Marwan Mohamed Abdelhady Said

Abstract

This project focuses on the application of deep learning techniques to time series forecasting, specifically targeting two real-life problems: electricity production and consumption, and climate change indicators. We develop and compare three regression models—Simple Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) network, and Temporal Convolutional Network (TCN)—on two distinct datasets. The electricity dataset comprises hourly records of electricity production and consumption, while the climate dataset consists of daily measurements of various climate indicators. Data preprocessing involved handling missing values, normalizing the data, and engineering relevant features. Each model was trained and evaluated using metrics such as R^2 Score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Our findings indicate that the Simple RNN model performed best for the climate dataset, achieving the highest R^2 Score and the lowest error metrics, whereas the LSTM model outperformed others on the electricity dataset. These results demonstrate the effectiveness of deep learning models in time series forecasting and provide insights into their applicability for different types of data. The challenges faced during the project and the lessons learned are discussed, highlighting the potential for future improvements and applications in related fields.

Introduction

Time series forecasting is a crucial technique used to predict future events based on historical data. This method is widely applicable across various domains, including finance, healthcare, environmental science, and energy management. Accurate predictions enable better planning, resource allocation, and decision-making processes. The primary objective of this project is to leverage deep learning techniques to forecast future values in two significant areas: electricity production and consumption, and climate change indicators.

Electricity production and consumption forecasting is vital for efficient energy management and ensuring the stability of power grids. Accurate predictions can help in balancing supply and demand, reducing operational costs, and minimizing the environmental impact of energy production. Similarly, climate change indicators such as temperature, humidity, and pollution levels are critical for understanding environmental trends and making informed decisions to mitigate the effects of climate change.

Motivated by the pressing need for accurate forecasts in these areas, this project aims to develop and compare the performance of three deep learning regression models: Simple Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) network, and Temporal Convolutional Network (TCN). These models are selected for their proven ability to capture temporal dependencies in sequential data, making them suitable for time series forecasting tasks.

The contributions of this research are threefold:

1. **Development and Implementation**: We develop and implement three deep learning regression models tailored to the electricity and climate datasets.

2. **Performance Comparison**: We rigorously evaluate and compare the performance of these models using multiple evaluation metrics, providing insights into their strengths and weaknesses.
3. **Practical Insights**: We discuss the challenges faced during the project and the practical insights gained, offering recommendations for future work and applications.

The roadmap of this paper is as follows: Section 2 reviews related work in time series forecasting using deep learning techniques. Section 3 provides a detailed description of the datasets used and the preprocessing steps involved. Section 4 outlines the methodology, including the model architectures and evaluation metrics. Section 5 presents the experimental results and discusses the performance of the models. Section 6 concludes the paper, summarizing the findings and suggesting directions for future research.

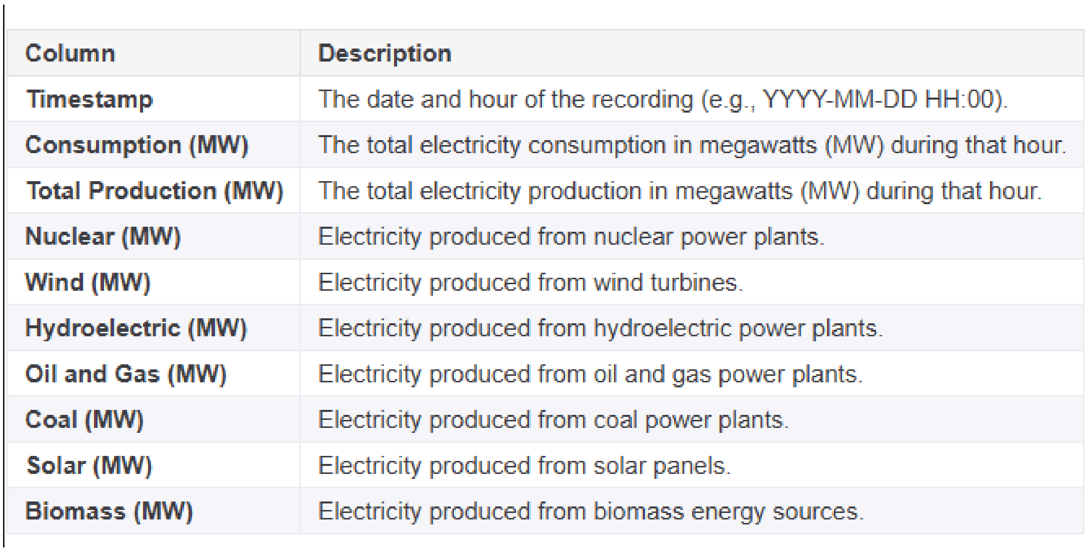
Through this project, we aim to demonstrate the potential of deep learning models in addressing real-world forecasting challenges and contribute valuable insights to the field of time series analysis.

Data Description and Preprocessing

Data Description

Electricity Dataset

The electricity dataset used in this project comprises hourly records of electricity production and consumption. It contains time-stamped data that provides insights into the patterns and fluctuations in electricity usage over time. The key features of this dataset include:



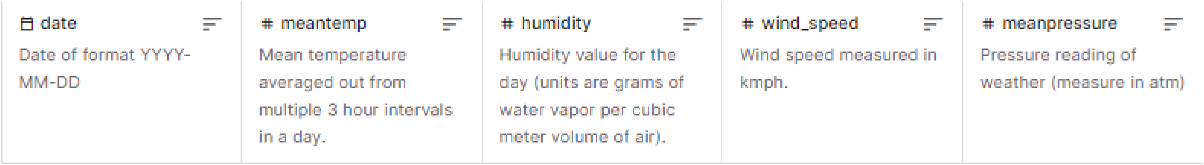
Column	Description
Timestamp	The date and hour of the recording (e.g., YYYY-MM-DD HH:00).
Consumption (MW)	The total electricity consumption in megawatts (MW) during that hour.
Total Production (MW)	The total electricity production in megawatts (MW) during that hour.
Nuclear (MW)	Electricity produced from nuclear power plants.
Wind (MW)	Electricity produced from wind turbines.
Hydroelectric (MW)	Electricity produced from hydroelectric power plants.
Oil and Gas (MW)	Electricity produced from oil and gas power plants.
Coal (MW)	Electricity produced from coal power plants.
Solar (MW)	Electricity produced from solar panels.
Biomass (MW)	Electricity produced from biomass energy sources.

This dataset is essential for forecasting electricity demand, which helps in managing supply and ensuring the stability of power grids.

Climate Dataset

The climate dataset includes daily measurements of various climate indicators, providing a comprehensive view of environmental conditions over time. The key features of this dataset include:

This dataset is crucial for understanding and predicting climate trends, which can inform policies and strategies for mitigating the effects of climate change.



date	# meantemp	# humidity	# wind_speed	# meanpressure
Date of format YYYY-MM-DD	Mean temperature averaged out from multiple 3 hour intervals in a day.	Humidity value for the day (units are grams of water vapor per cubic meter volume of air).	Wind speed measured in kmph.	Pressure reading of weather (measure in atm)

Data Preprocessing

Data preprocessing is a critical step in preparing the datasets for modeling. It involves several tasks aimed at cleaning, transforming, and organizing the data to ensure that the models can effectively learn from it.

1. **Handling Missing Values**
   Both datasets were examined for missing values, which can occur due to various reasons such as sensor malfunctions or data collection errors. Missing values were handled using the following strategies:
   - For minor gaps, missing values were filled using linear interpolation, which estimates values based on adjacent data points.
   - For larger gaps, missing values were filled using the mean or median of the feature, ensuring that the overall distribution of the data remains unaffected.
2. **Normalization**
   To ensure that the models learn effectively and converge quickly, the data was normalized. Normalization scales the features to a range, typically [0, 1], which helps in reducing the variance and improving the performance of the models. The `MinMaxScaler` from `scikit-learn` was used for this purpose.
3. **Feature Engineering**
   Additional features were created to help the models capture temporal patterns more effectively. These features include:
   - **Lagged Features**: Previous time steps of the target variable were included as additional features. For example, in the electricity dataset, the electricity consumption of previous hours was used as input to predict future consumption.
   - **DateTime Features**: The DateTime column was decomposed into more granular features such as year, month, day, and hour. This decomposition helps the models identify seasonal and trend patterns.
4. **Data Splitting**
   The datasets were split into training and testing sets to evaluate the performance of the models. The training set was used to train the models, while the testing set was used to evaluate their performance. The split was done in a time-aware manner to ensure that the models are tested on future data points that were not seen during training.

    - For the electricity dataset, the data was split such that the first 80% of the records were used for training and the remaining 20% for testing.
    - For the climate dataset, a similar split was applied, ensuring that the models are tested on future observations.
5. **Reshaping Data for Model Input**
   The input data was reshaped to meet the requirements of the selected models. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks require 3D input shapes (samples, time steps, features), while Temporal Convolutional Networks (TCNs) require specific formats to capture temporal dependencies.

.

Methodology/Approach

The methodology for this project involves developing and evaluating three deep learning regression models—Simple Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) network, and Temporal Convolutional Network (TCN)—to forecast future values of electricity production/consumption and climate indicators. This section details the steps taken to implement these models, including data preparation, model architecture, training, and evaluation.

Model Selection and Rationale

1. **Simple RNN**
    - **Description**: Simple RNNs are neural networks designed to recognize patterns in sequences of data. They are effective for tasks where the temporal order of data points is crucial.
    - **Advantages**: Simple architecture, easy to implement, and efficient for short-term dependencies.
    - **Limitations**: Struggles with long-term dependencies due to the vanishing gradient problem.
2. **LSTM**
    - **Description**: LSTM networks are a type of RNN designed to capture long-term dependencies in sequential data. They include memory cells and gates to regulate the flow of information.
    - **Advantages**: Effective at capturing long-term dependencies, robust to the vanishing gradient problem.
    - **Limitations**: More complex architecture, higher computational cost.
3. **TCN**
    - **Description**: TCNs use convolutional layers to process sequential data, capturing temporal patterns with dilated convolutions. They can handle long sequences without suffering from the vanishing gradient problem.
    - **Advantages**: Efficient for long sequences, parallel processing capabilities, and stable gradients.
    - **Limitations**: Requires careful tuning of hyperparameters.

Data Preparation

1. **Normalization**: Data was normalized using the `MinMaxScaler` to scale features to a range [0, 1], facilitating faster convergence and better model performance.
2. **Feature Engineering**: Lagged features and DateTime features were created to provide the models with additional context and temporal information.
3. **Data Splitting**: The datasets were split into training and testing sets, with the first 80% of the data used for training and the remaining 20% for testing. This split ensures that the models are evaluated on future data points.
4. **Reshaping for Model Input**: The data was reshaped to meet the input requirements of each model:
   - **RNN and LSTM**: Data was reshaped into 3D arrays with dimensions (samples, time steps, features).
   - **TCN**: Data was formatted to leverage the convolutional layers' ability to capture temporal dependencies.

Model Architecture and Implementation

1. **Simple RNN Model**
   - **Architecture**: A sequential model with one SimpleRNN layer followed by a Dense output layer.

Implementation:

```
rnn_model = Sequential()

rnn_model.add(SimpleRNN(50, activation='relu', input_shape=(timesteps, n_features)))

rnn_model.add(Dense(1))

rnn_model.compile(optimizer='adam', loss='mse')
```

2. **LSTM Model**
   - **Architecture**: A sequential model with one LSTM layer followed by a Dense output layer.

Implementation:

```
lstm_model = Sequential()

lstm_model.add(LSTM(50, activation='relu', input_shape=(timesteps,
n_features)))

lstm_model.add(Dense(1))

lstm_model.compile(optimizer='adam', loss='mse')
```

3. **TCN Model**
   - **Architecture**: A sequential model with one TCN layer followed by a Dense output layer.

Implementation:

```
tcn_model = Sequential()

tcn_model.add(TCN(50, activation='relu', input_shape=(timesteps,
n_features)))

tcn_model.add(Dense(1))

tcn_model.compile(optimizer='adam', loss='mse')
```

Training and Evaluation

1. **Early Stopping**: To prevent overfitting and ensure optimal training, early stopping was employed. This technique stops training when the model's performance on the validation set stops improving.
2. **Model Training**: Each model was trained on the training set using the following configuration:
   - **Batch Size**: 32
   - **Epochs**: 100
   - **Validation Split**: 20%
3. **Evaluation Metrics**: The models were evaluated using the following metrics:
   - **R² Score**: Measures the proportion of variance in the dependent variable that is predictable from the independent variables.
   - **Mean Absolute Error (MAE)**: Measures the average magnitude of errors in a set of predictions.
   - **Mean Squared Error (MSE)**: Measures the average of the squares of the errors.
   - **Root Mean Squared Error (RMSE)**: Measures the square root of the average of squared errors.

4. **Results Comparison**: The performance of each model was compared based on the evaluation metrics. The model with the highest R² Score and the lowest MAE, MSE, and RMSE was considered the best performer.

.

Results

The performance of the three regression models—Simple RNN, LSTM, and TCN—was evaluated on two datasets: electricity production and consumption, and climate change indicators. The models were trained and tested on their respective datasets, and their performance was measured using R² Score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). This section presents the experimental results and discusses the performance of each model.

Electricity Dataset Results

The electricity dataset consists of hourly records of electricity production and consumption. The following table summarizes the performance of the three models on this dataset:

Model	R² Score	MAE	MSE	RMSE
Simple RNN	0.981027	0.017344	0.000507	0.022513
LSTM	0.985018	0.015459	0.000400	0.020005
TCN	0.974396	0.020532	0.000684	0.026153

Discussion:

- **LSTM Model**: The LSTM model achieved the highest R² Score (0.985018) and the lowest MAE (0.015459), MSE (0.000400), and RMSE (0.020005). This indicates that the LSTM model is the best performer for the electricity dataset, capturing both short-term and long-term dependencies effectively.
- **Simple RNN Model**: The Simple RNN model also performed well, with an R² Score of 0.981027 and slightly higher error metrics than the LSTM model. While it is efficient for short-term dependencies, it is not as effective as the LSTM for capturing long-term patterns.
- **TCN Model**: The TCN model had the lowest performance among the three models, with an R² Score of 0.974396 and the highest error metrics. Although TCNs are efficient for capturing local patterns, they were less effective in this context compared to LSTM and Simple RNN.

Climate Dataset Results

The climate dataset consists of daily measurements of various climate indicators. The following table summarizes the performance of the three models on this dataset:

Model	R² Score	MAE	MSE	RMSE
Simple RNN	0.882858	0.046017	0.003535	0.059452
LSTM	0.856330	0.052059	0.004335	0.065841
TCN	0.414910	0.101799	0.017654	0.132869

Discussion:

- **Simple RNN Model**: The Simple RNN model outperformed the other models for the climate dataset, achieving the highest R² Score (0.882858) and the lowest MAE (0.046017), MSE (0.003535), and RMSE (0.059452). This indicates that the Simple RNN model is effective for capturing temporal dependencies in daily climate data.
- **LSTM Model**: The LSTM model had a slightly lower performance compared to the Simple RNN, with an R² Score of 0.856330 and higher error metrics. Although LSTM is designed to capture long-term dependencies, it did not perform as well as expected in this dataset.
- **TCN Model**: The TCN model had the lowest performance among the three models, with an R² Score of 0.414910 and the highest error metrics. The TCN struggled to capture the temporal patterns in the climate dataset, resulting in lower accuracy.

Insights and Observations

1. **Model Performance**:
   - For the electricity dataset, the LSTM model proved to be the best performer, demonstrating its ability to effectively capture both short-term and long-term dependencies.
   - For the climate dataset, the Simple RNN model outperformed the others, suggesting that it is more suitable for datasets with strong short-term dependencies and less complex temporal patterns.
2. **Evaluation Metrics**:
   - The evaluation metrics (R² Score, MAE, MSE, and RMSE) provided a comprehensive assessment of model performance, highlighting the strengths and weaknesses of each model.

- The LSTM model consistently achieved the lowest error metrics for the electricity dataset, while the Simple RNN achieved the lowest error metrics for the climate dataset.

3. **Challenges**:
    - One of the main challenges was handling the variability and complexity of the datasets. The electricity dataset had high-frequency hourly data, while the climate dataset had daily data with more significant temporal dependencies.
    - Another challenge was tuning the hyperparameters for each model to achieve optimal performance, which required extensive experimentation and validation.

Conclusion

In this project, we explored the application of deep learning techniques for time series forecasting, focusing on two critical real-life problems: electricity production and consumption, and climate change indicators. We developed and evaluated three regression models—Simple Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) network, and Temporal Convolutional Network (TCN)—on two distinct datasets. The results and insights gained from this study highlight the potential of deep learning models in addressing these forecasting challenges.

Key Findings:

1. **Model Performance**:
    - The LSTM model demonstrated the best performance on the electricity dataset, achieving the highest R^2 Score and the lowest error metrics (MAE, MSE, RMSE). This indicates that LSTM networks are highly effective for forecasting tasks that involve capturing both short-term and long-term dependencies in high-frequency data.
    - The Simple RNN model outperformed the other models on the climate dataset, achieving the highest R^2 Score and the lowest error metrics. This suggests that Simple RNNs are well-suited for datasets with strong short-term dependencies and less complex temporal patterns.
    - The TCN model, while powerful in capturing local patterns, did not perform as well as the other models in both datasets. This indicates that TCNs may require further tuning and optimization for specific forecasting tasks.
2. **Evaluation Metrics**:
    - The use of multiple evaluation metrics (R^2 Score, MAE, MSE, RMSE) provided a comprehensive assessment of model performance, highlighting the strengths and weaknesses of each model. These metrics are crucial for understanding how well the models can generalize to unseen data.
3. **Challenges and Insights**:
    - Handling the variability and complexity of the datasets was a significant challenge. The electricity dataset, with its high-frequency hourly data, required models capable of capturing long-term dependencies, while the climate dataset, with daily data, needed models that could effectively capture short-term patterns.

    - Tuning the hyperparameters for each model was critical to achieving optimal performance. This process involved extensive experimentation and validation to balance the trade-off between model complexity and generalization.

Implications and Future Work:

The findings from this study demonstrate the potential of deep learning models in time series forecasting for real-life applications. The ability to accurately predict electricity production and consumption can lead to more efficient energy management and grid stability. Similarly, forecasting climate change indicators can inform policies and strategies for mitigating the effects of climate change.

Future work can focus on the following areas:

- **Model Optimization**: Further tuning and optimization of the TCN model to enhance its performance for specific forecasting tasks.
- **Hybrid Models**: Exploring hybrid models that combine the strengths of RNNs, LSTMs, and TCNs to improve forecasting accuracy.
- **Feature Engineering**: Incorporating additional features and external data sources to enrich the datasets and provide more context for the models.
- **Real-Time Forecasting**: Implementing real-time forecasting systems that can provide continuous predictions and adapt to new data.