**ADVANCED MACHINE LEARNING**

**FINAL PROJECT REPORT**

**ANALYZING TRAFFIC FLOW DYNAMIC WITH RNNs & LSTMs**

A blue and yellow logo

Description automatically generated

|  |
| --- |
|  |
|  |
|  |
| **Nandini Raveendran Nair Subhadra** |
|  |

**Weather-Driven Traffic Flow Prediction: A Comparative Analysis of RNNs and LSTMs**

**Introduction**:

This project investigates the complex relationship that exists between traffic flow dynamics and weather. Utilizing Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs), the study aims to predict traffic patterns during holidays, and different weather conditions. According to the hypothesis, accurate forecasting will be possible if one can comprehend how weather variables interact with traffic flow through RNN and LSTM models. Choosing certain months in the year can give us accurate results to predict how traffic flow is affected. Therefore, we decided to use August, September, and October months to conduct our analysis.

**Objective:**

The objective of this analysis is to leverage advanced machine learning models, specifically Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs), to comprehensively understand the intricate relationship between weather conditions and traffic flow dynamics during holiday periods. The goal is to develop a robust predictive model using RNNs and LSTMs, with a focus on model selection and configuration, to enhance the accuracy of traffic flow predictions. The study aims to contribute valuable insights to traffic management and planning strategies by identifying the most effective model for forecasting under weather-influenced conditions.

**Methodology:**

Data preparation involves reading a .CSV file normalized with Pandas Data Frame, and one-hot encoding is performed for continuous and categorical variables. The code primarily utilizes Pandas and Keras libraries for importing, manipulating, and analyzing time series data.

RNN & LSTM models are chosen due to their inherent ability to capture sequential dependencies and patterns within time series data. Traffic flow dynamics are influenced by temporal factors, such as hourly variations and consecutive patterns, making RNNs and LSTMs well-suited for this task. These models are constructed, fine-tuned, and trained on the dataset, and later their validation loss and test loss are compared to see which model performs better.

**Data Analysis and Preprocessing:**

* Data loaded from a CSV file into a Pandas Data Frame.
* Data Frame columns include Timestamp, Traffic Flow, Weather, and Holiday.
* Timestamp converted to a datetime object to extract temporal features.
* Data normalization and one-hot encoding are applied to continuous and categorical variables.
* Average traffic flow by hour analyzed.
* Visual representation of peak traffic hours provided.
* Data reshaped to create sequences suitable for RNN/LSTM models.

**Model Architecture & Training and Validation:**

* Several RNN and LSTM models were initiated with different configurations.
* Models trained and evaluated using mean squared error loss.
* The best model was selected based on validation loss.
* Training and validation loss visualizations for model convergence assessment.
* Experimentation with stacked LSTMs.
* Exploration of batch sizes and learning rate affects model performance.

The below graphs show the validation losses for RNN and LSTM

RNN

A graph of a number of different batch sizes

Description automatically generated

LSTM

A graph of a line graph

Description automatically generated

**Model Evaluation:**

* Models evaluated on a validation and test set to estimate generalization performance.
* LSTM model with a batch size of 128 outperformed RNN model based on the test loss.
* After doing Hyperparameter fine-tuning, adjusting units, and learning rate, we got the following results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RNN | | LSTM | |
| Validation Loss | Test Loss | Validation Loss | Test Loss |
| Batch Size 32 | 0.04486 | 0.03606 | 0.03665 | 0.03218 |
| Batch Size 64 | 0.04354, | 0.03515 | 0.03926 | 0.03264 |
| Batch Size 128 | 0.04638 | 0.03737 | 0.03752 | 0.03197 |

**Results and Conclusions:**

The results reinforce the observation that LSTMs perform better than RNNs for this task across all batch sizes. It also highlights that the choice of batch size impacts model performance, with smaller batch sizes slightly favoring model accuracy.

As we saw, LSTMs are more effective than RNNs. This could be due to their architecture, which is better at handling long-term dependencies and avoiding issues like vanishing gradients. We noticed also, batch sizes seem to yield better results, though the differences are not extremely large. This could be a point of consideration for further optimization.

Future studies may explore scalability and integration of additional variables for enhanced predictive accuracy.

* The below graphs show the Training and validation losses for RNN, Stacked LSTM

A group of graphs with blue lines

Description automatically generated

**Summary**:

The results demonstrate that Long Short-Term Memory Networks (LSTMs) outperform Recurrent Neural Networks (RNNs) in a given task across all batch sizes. Smaller batch sizes slightly improve model accuracy, suggesting batch size choice is crucial for performance optimization. The superiority of LSTMs is attributed to their architecture's ability to manage long-term dependencies and avoid issues like vanishing gradients. While batch size impacts results, the differences are not markedly large, indicating room for further optimization. Future research could focus on the scalability of these models and the incorporation of additional variables to enhance predictive accuracy.

Another important point to consider is the nature of the data. As we had small data, therefore differences between the models and batch sizes seem not too large. But when we have large data sets using these methods models could give varying results and their performance could be more visible. However, as the objective is the project is to learn and show how we can apply the time series model to sequential data, the project shows that RNN and LSTM could give different results and even batch size could differ. So, in real-world projects through following these steps and model building we can compare and pick the fitting model with its fitting batch size.

As the “No Free Lunch Theory” argues there is no one theory that explains everything and is applicable to all data types. Thus, depending on the real-world data and problem different methods, models, and strategies could be used.

It is important to highlight one last note regarding model performance. The model performances could change as we run our models several times and the presentation file model result could be different from the project model results. But, we prepared this report according to the latest version of our models as colab pdf and ipynb files show.