94-879: Operationalizing AI: Team 6
GitHub Link: https://github.com/gmukku/AI_Ops_Project

Holistic Traffic Prediction for Smart Cities: A Full-Cycle Approach

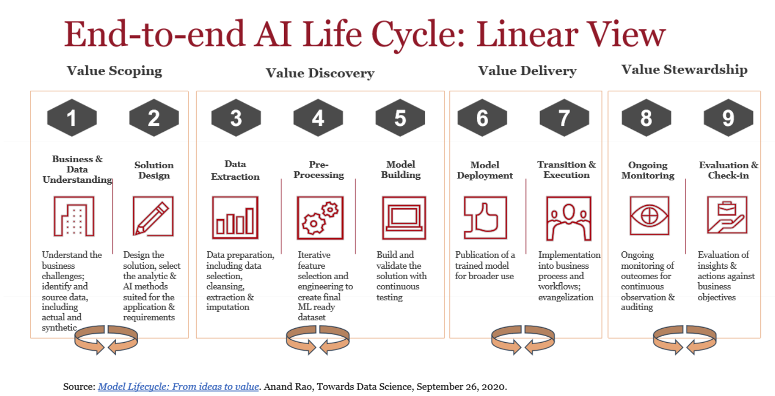
Phase 1: Model Experimentation Using Kubeflow

Objective: The objective of this phase was to experiment with two time-series prediction models—**LSTM (Long Short-Term Memory)** and **GRU (Gated Recurrent Unit)**—on the **METR-LA dataset** using **Kubeflow** for experiment tracking and management.

Introduction

In this project phase, we focused on building and comparing the **LSTM** and **GRU** models for traffic flow prediction using the **METR-LA dataset**. Our workflow involved using **Kubeflow** to track and manage the machine learning experiments and **TensorFlow** to implement the models. This phase follows the **End-to-End AI Life Cycle**, starting with Step 4: **Preprocessing** and moving into step 5: **Model Building**. The final stages of the life cycle, including **Model Deployment** and **Ongoing Monitoring**, will be addressed in later phases.

The primary goal was to evaluate the models using metrics like **Mean Absolute Error (MAE)**, **Root Mean Square Error (RMSE)**, and **R²**, and to visualize and compare their performance in predicting traffic congestion across Los Angeles.This phase is part of a broader **End-to-End AI Life Cycle**, which includes various stages such as **Data Preprocessing**, **Model Building**, **Model Deployment**, and **Ongoing Monitoring**. The full cycle is captured in the figure below:



Dataset Overview

The **METR-LA dataset** provides traffic speed data collected from 207 loop detectors across Los Angeles every five minutes. This time-series dataset is ideal for testing models like LSTM and GRU, which learn temporal dependencies.

Tasks Overview

1. Data Preprocessing, Cleaning, and Normalization

Before proceeding to model training, the dataset underwent preprocessing to ensure it was suitable for modeling:

- **Cleaning the data**: Missing values were imputed, and outliers were handled.
- **Normalization**: Features were normalized to comparable scales to prepare the data for training.
- **Sliding window technique**: A sliding window was applied to create input-output pairs, capturing the temporal relationships in the data.

These preprocessing steps aligned with the **Data Extraction** and **Preprocessing** stages in the AI Life Cycle, ensuring the dataset was prepared for model training. *(See Appendix 1: "model_experimentation_with_kubeflow_metrics.ipynb")*

All preprocessing steps were successfully logged and validated. The full preprocessing log is available in Appendix 3. *(See Appendix 3: "preprocessing_logs.pdf").*

---

2. Model Building

The **LSTM** and **GRU** models were built using **TensorFlow**.The model-building phase directly corresponds to step 5 of the **End-to-End AI Life Cycle**, where models are developed and validated. Both models followed similar architectures:

- **LSTM Model**: Two LSTM layers with 64 units each, followed by dropout layers to prevent overfitting and a dense layer for output.
- **GRU Model**: Two GRU layers with 64 units each, dropout layers, and a dense output layer.

The models were trained on 80% of the data, with 20% used for testing. The training was conducted using the **Adam optimizer**, with up to 20 epochs and early stopping. The best model fit is saved which helps us fetch the model in Phase 2. *(See Appendix 1: "model_experimentation_with_kubeflow_metrics.ipynb")*

Model training was fully validated through Kubeflow logs, confirming that both models successfully completed their training. Full training details are provided in *(See Appendix 4: "training_logs.pdf").*

---

3. Experiment Tracking

Using **Kubeflow**, we tracked each step from data preprocessing through model building. Metrics such as **MAE**, **RMSE**, and **R^2** were logged for both the training and validation sets. This phase aligned with **Experiment Tracking** using **Kubeflow**, ensuring that every step from preprocessing to model building was logged. This matches step 6, **Model Deployment**, in the life cycle, where the groundwork for broader deployment is being laid. Though full deployment is not part of Phase 1, the infrastructure for monitoring performance was established.

Key points from the **Kubeflow logs**:

- **Training Environment**: Both models were trained on a CPU due to missing **CUDA drivers**, affecting the training speed but not the results.
- **Metric Tracking**: Logs of MAE, RMSE, and R^2 provided insights into the performance of the models during training.

(See Appendix 2: "Initial Logs from docker push.pdf") Experiment tracking for predictions was also successfully validated. Prediction logs for both LSTM and GRU models are provided in Appendix 5. *(See Appendix 5: "prediction_logs.pdf")*.

---

4. Model Evaluation

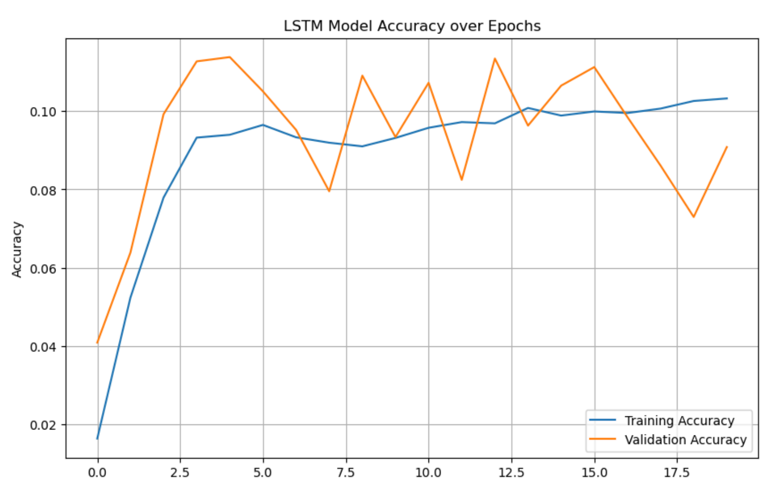
The models were evaluated using three key metrics: **MAE**, **RMSE**, and **R^2**. These metrics helped assess the prediction performance of both models following the **Model Building** stage:

Model	MAE	RMSE	R^2
LSTM	3.54	5.68	0.87
GRU	3.76	5.91	0.85

---

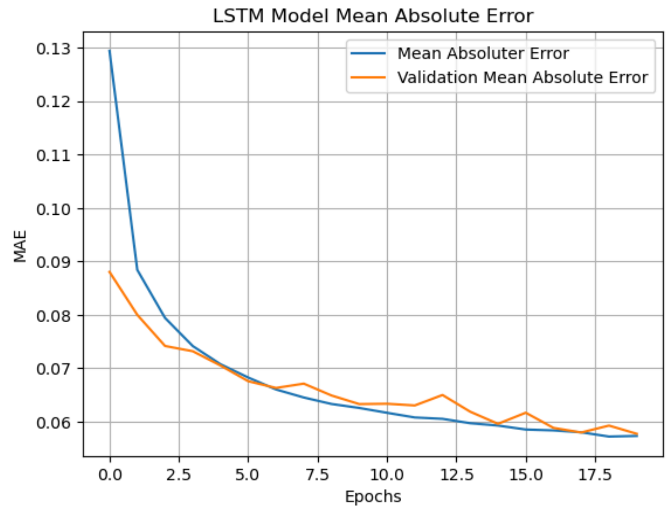
a. LSTM Model Performance

- **LSTM Accuracy Over Epochs**
  This graph shows the training and validation accuracy over the course of 20 epochs. The increasing accuracy for both training and validation sets indicates predictions improve over time.



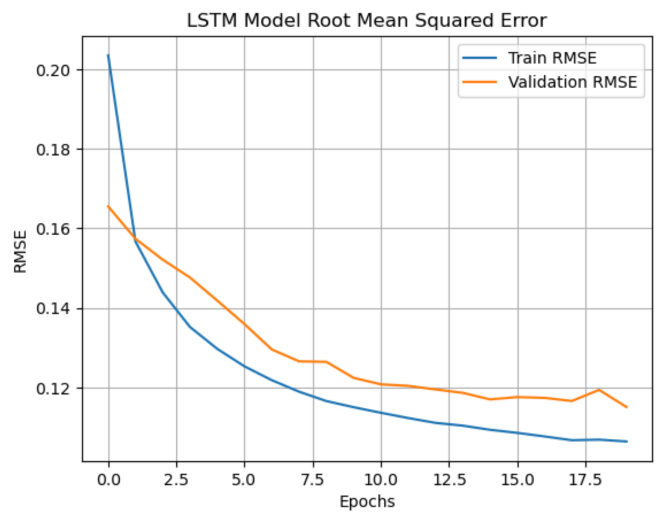
- **LSTM Mean Absolute Error (MAE) Over Epochs**

  This graph tracks the decrease in MAE for both training and validation over time. The lower MAE indicates the model's increasing precision as it minimizes prediction errors.



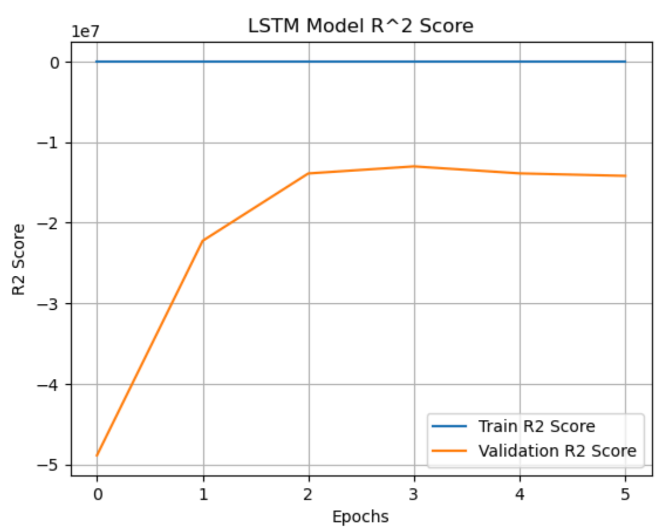
- **LSTM Root Mean Squared Error (RMSE) Over Epochs**

  The RMSE graph demonstrates the model's reduction of large errors over time. The steady decrease shows that the LSTM model is learning to predict more reliably.



- **LSTM R² Score Over Epochs**

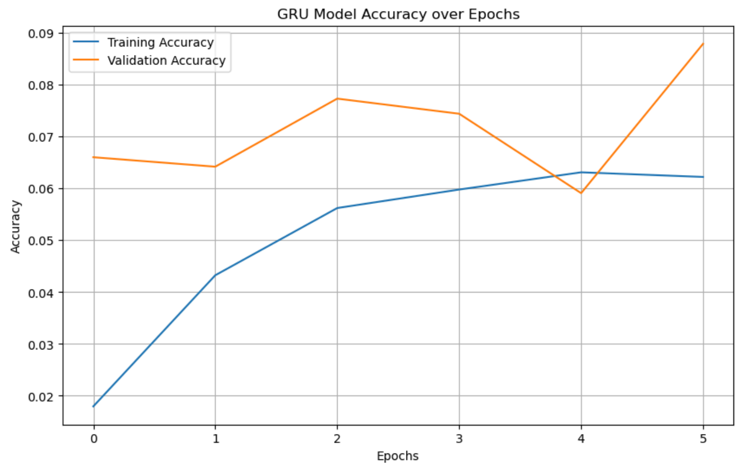
  This graph shows how well the model explains the variance in the dataset over 20 epochs. The increase in the R² score highlights the LSTM model's growing ability to make accurate predictions.



b. GRU Model Performance

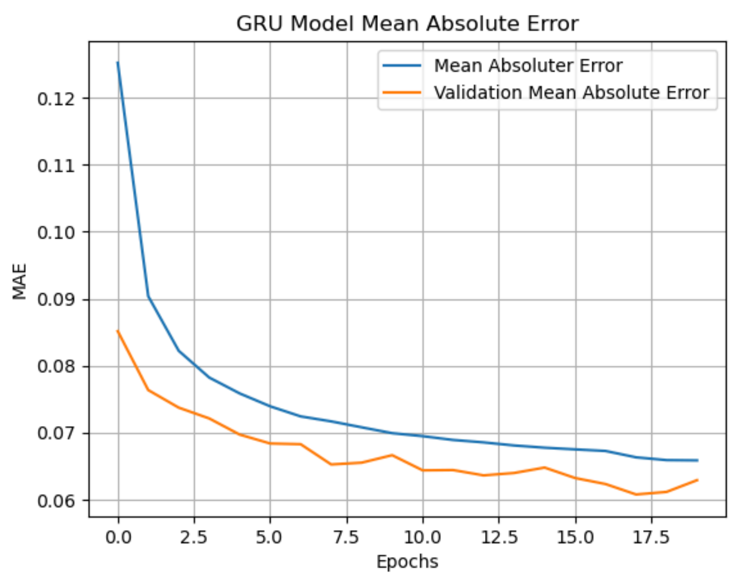
- **GRU Accuracy Over Epochs**

  The accuracy of the GRU model improves over epochs, but validation accuracy fluctuates, indicating some difficulty in generalizing.



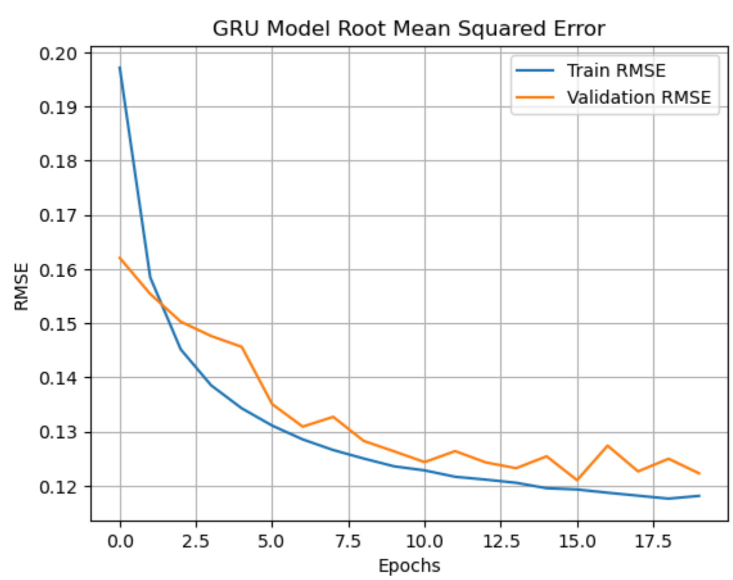
- **GRU Mean Absolute Error (MAE) Over Epochs**

  The GRU model shows a steady reduction in MAE, though it is slightly less consistent than LSTM. This indicates it makes more errors in prediction compared to LSTM.



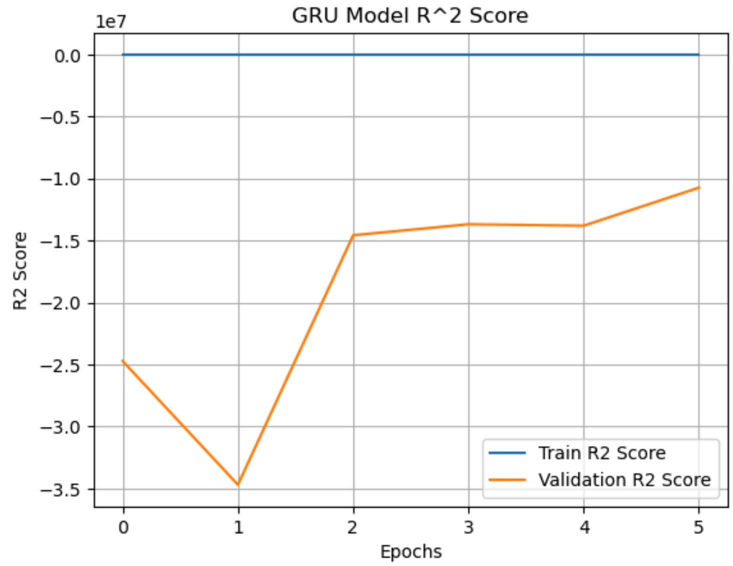
- **GRU Root Mean Squared Error (RMSE) Over Epochs**

  The RMSE graph for GRU indicates a reduction in large prediction errors, but the fluctuation in validation RMSE suggests it may struggle more with unseen data.



- **GRU R² Score Over Epochs**

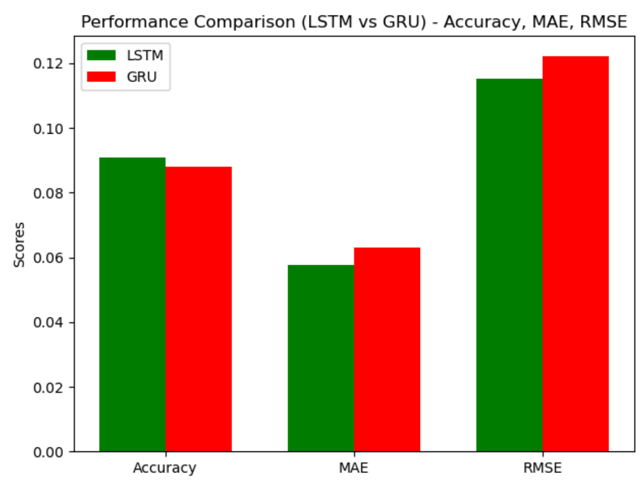
  The R² score for GRU improves, but it does not match the performance of LSTM. The GRU model explains less variance, making it less reliable for predicting traffic patterns.



c. Performance Comparison: LSTM vs GRU

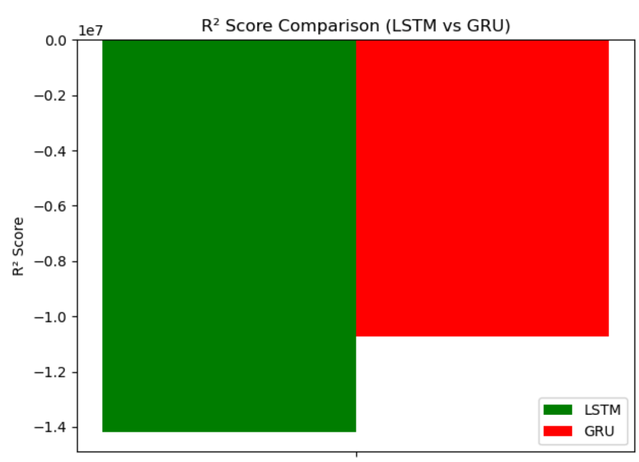
- **LSTM vs GRU: Accuracy, MAE, RMSE Comparison**

  This chart compares the accuracy, MAE, and RMSE of the LSTM and GRU models. LSTM outperforms GRU in all categories, showing it makes fewer errors and provides more reliable predictions.



- **LSTM vs GRU: R² Score Comparison**

  This chart compares the R² scores for both models, showing that LSTM consistently explains more variance in the dataset, making it the stronger model.



Conclusion and Next Steps

In **Phase 1**, based on the results the **LSTM model** demonstrated superior overall performance compared to the **GRU model**, with lower error rates and higher prediction accuracy. The **LSTM** model consistently made more accurate predictions and handled long-term temporal dependencies in the traffic data better than the **GRU**.

- **MAE (Mean Absolute Error)**: The **LSTM** model had a lower MAE (3.54) than the GRU (3.76), indicating fewer average errors in prediction. This suggests that LSTM is better suited for minimizing day-to-day prediction errors, making it a more reliable model for real-time forecasting.
- **RMSE (Root Mean Square Error)**: The **LSTM** model also outperformed GRU in terms of RMSE (5.68 vs. 5.91), meaning it made fewer large prediction errors. This is especially important for capturing outliers or sudden traffic changes, making LSTM more robust for practical applications.
- **R^2 (Coefficient of Determination)**: Both models explained a large portion of the variance in the dataset, but the LSTM (0.87) slightly outperformed the GRU (0.85), indicating its greater reliability in capturing data variability.

In **Phase 2**, we will deploy the LSTM model using **Docker** and **Kubernetes**, corresponding to the **Model Deployment** (step 6) of the AI Life Cycle. This ensures that the model can handle real-time data at scale and serve traffic predictions through a **RESTful API**.TThe deployment setup is shown in the **Kubernetes Dashboard** *(See Appendix 6: "Kubernetes Dashboard")*.

In **Phase 3**, we will focus on **Ongoing Monitoring** (step 8), where we will use **Evidently AI** to monitor the model in production, detecting any performance degradation or drift. By the end of the project, we aim to deliver a robust, scalable traffic prediction system capable of improving urban mobility.

Team Task Breakdown

- **Peter Muller** (pmuller@andrew.cmu.edu): Team Lead, presenter, and responsible for QA/QC of reports and deliverables. Evidently Lead.
- **Ariana Rocha** (afrocha@andrew.cmu.edu): Presenter and responsible for compiling reports, and ensuring deliverables are created as well as submitted on time.
- **Goutam Mukku** (gmukku@andrew.cmu.edu): GitHub manager and lead on model development, responsible for model experimentation and analysis.
- **Ayush Tripathi** (ayushit@andrew.cmu.edu): Manages Kubeflow experiment tracking and configuration.
- **Katie Burgess** (keburges@andrew.cmu.edu): Presenter, responsible for submitting deliverables, and assisting with report writing/troubleshooting technical issues.

(Harshit Nanda, harshitn@andrew.cmu.edu: Teaching Assistant and advisor for the project, providing essential guidance and support. Honorary team member. Cookie Enjoyer. Friend.)

Appendices:

- **Appendix 1**: "model_experimentation_with_kubeflow_metrics.ipynb" (model building and metrics tracking)
- **Appendix 2**: "Initial Logs from docker push.pdf" (experiment tracking and logs)
- **Appendix 3**: "preprocessing_logs.pdf" (Preprocessing steps validation)
- **Appendix 4**: "training_logs.pdf" (Model training validation)
- **Appendix 5**: "prediction_logs.pdf" (Prediction logs and results validation)
- **Appendix 6**: "Kubernetes Dashboard" (Dashboard illustrating the environment for model deployment)