Forecasting Daily Car Wash Traffic Through LSTM and XGBoost Models

Albert Shilling
College of Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida
albertshilling1225@gmail.com

***Abstract*— Accurately forecasting car wash traffic is essential for optimizing daily operations, as customer volume is closely linked to weather conditions. In this study, two predictive models will be compared: XGBoost, a tree-based ensemble algorithm enhanced by extreme gradient boosting for performance and scalability, and Long-Short Term Memory (LSTM), a recurrent neural network (RNN) that handles time series forecasting well by taking in a sequence of data to make predictions. XGBoost leverages extreme gradient boosting to handle structured data efficiently, while LSTM utilizes gating systems, like the forget gate, to handle which data from the current cell state to pass to the next cell state. This can help the LSTM find any patterns while also mitigating overfitting issues.**

In the findings, the LSTM model achieved a lower Mean Squared Error (MSE) whereas the XGBoost model yielded a higher coefficient of determination (R^2). This comparison shows the trade-offs between a boosted regressor and sequential modelling.

To support real-time forecasting, a user interface was developed using Streamlit, allowing users to input current weather and temporal data and receive immediate car wash count predictions from the trained model.

I. Introduction

At Rising Tide Car Wash where 80% of the staff is on the Autism Spectrum, it is important to have methods in place to help manage efficiency and profitability as a for-profit business. One key strategy involves maintaining a 10% conversion of retail customers to monthly membership customers.

For adults on the Autism Spectrum, this can be difficult to estimate or visualize what 10% of their customer interactions will be due to changing factors like weather which can strongly impact the daily volume. Car wash traffic volume is directly impacted by weather conditions seasonality, and holiday travel. While precipitation has a negative effect, as air quality index (AQI) increases so will customer traffic. The AQI measures air pollution levels which includes but not limited to dust and pollen in the air [1]. During months with less precipitation, AQI will naturally be higher as precipitation can help manage the pollutants in the air. Early months in the year here in Florida, combine these factors: low precipitation and high AQI readings which leads to higher customer traffic due to cars being dirtier more often.

Trained on three years of historical car wash counts and corresponding weather patterns, this prediction model will assist the management in making necessary labor adjustments and help the staff accurately know how many memberships to sell per day. This study will use two models to compare which has the greatest performance, a decision tree-based structure XGBoost model or a LSTM RNN architecture.

XGBoost is less tunable but highly efficient on structured data and builds upon supervised machine learning and ensemble learning [2]. XGBoost achieves high efficiency with its gradient boosting techniques, which sets a targeted outcome for the following model to minimize any error [2]. XGBoost uses these decision trees as base learners, then combines these trees sequentially to try and improve performance. [3]

Comparing to a LSTM framework, which has highly tunable parameters, can dive deeper and can learn different patterns based on the length of the sequence. Using a sequence of seven, this LSTM model will aim to predict weekday vs. weekend patterns to better understand customer habits as well as how weather effects these patterns. LSTMs have memory cells which have an internal state that has a self-connected recurrent edge weight of one, this ensures gradients can pass without vanishing or exploding which can lead to overfitting errors [4]. LSTM RNNs are explicitly used for vanishing gradient problems [5]. In traditional RNNs, they can struggle with learning long term dependencies due to these vanishing gradients [5], while the LSTM design using the current cell state and forget gate can help manage this problem.

II. Methodology and System Framework

Forecasting car wash traffic will help in managing labor, ensuring daily profitability, and giving the sales staff a clear and concise goal to aim for when trying to convert 10% of retail customers to monthly memberships. Instead of the sales staff being told to have a goal of 10%, this model will have a user interface that shows a numerical representation of how many customers to convert per day and per hour.

This model is trained on three years of historical car wash count and corresponding weather data from January 1st, 2021, to December 31st, 2024. To predict car count accurately, reports were downloaded from the SQL server and parsed for correct counts as shown in Figure 1. This total count represents total cars washed including both retail washes, the target for this prediction, and existing member washes. As more memberships are sold, the daily count of member washes will naturally increase so this way, the retail washes can be calculated as this rate changes.

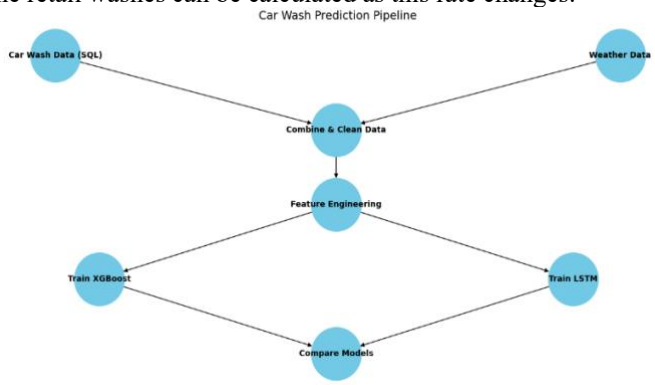

Figure 1. System architecture for car wash prediction pipeline

The dataset initially consists of 9 features: 'Date', 'Count' (daily car wash count), 'temp' (temperature), 'humidity', 'precip' (precipitation), 'precipicover' (cloud cover), 'uvindex' (UV index), 'conditions', and 'AQI' (air quality index). Through feature engineering additional features can be derived to increase model performance and learning. Using pandas datetime library, features including 'month', 'dayofweek' and binary indicators 'is_weekend' and 'is_holiday' can be found. Lag-based features were introduced to capture effects precipitation had on customer traffic trends. These features include 'rolling_rain_3' and 'rolling_rain_7' which allow the model to learn not only short patterns but also weekly patterns as well. To learn from recent trends, a 'prev_day_count' feature is added to learn the relationship between previous day and current day to help provide insight to customer behavior. The categorical 'conditons' feature requires encoding as it described current conditions as Clear, Partly Cloudy, Overcast, and Rain. Since the 'Date' column is in date time formatting, this column is excluded from use in the training. In total, 16 features make up the data that both models will be trained on.

XGBoost is selected as the benchmark model due to its ease of implementation while still having strong performance on structured data. The data is partitioned into training and testing using a Pareto split, 80% will be used for training while the remaining 20% will be for the test set. For XGBoost hyperparameter tuning, the number of estimators (n_estimators) was set to 500, meaning the model consists of 500 decision trees. In each tree, there is a max depth (max_depth) of 15, which allows the model to capture complex patterns otherwise missed on shallower depth models. In addition, with a learning rate of 0.05, a subsample ratio (subsample) of 0.6 which indicates 60% of the data is randomly sampled for each boosting round, and a colsample_bytree value of 0.8, which controlled the fraction of features sampled for each tree. A fixed random seed (random_state=42) ensures the results can be replicated. Figure 2 shows a section of the first tree in the XGBoost model.

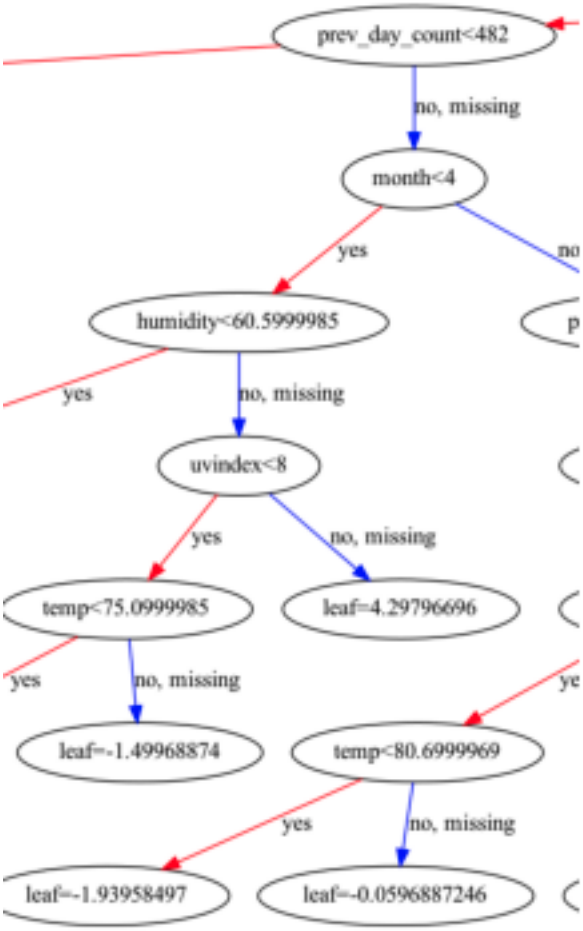

Figure 2. Section of tree 1 in XGBoost

The LSTM framework is used as the deep learning model under evaluation, with a goal of matching or surpassing the performance of the XGBoost benchmark. Features and target feature ('Count') are separated into data frames and both sets are normalized to improve convergence. A sequence length of 7 days is chosen to capture both weekday and weekend temporal patterns while avoiding any overfitting. Unlike the random splitting used in tree-based models like XGBoost, the LSTM requires data to be split in chronological order to preserve time dependencies. The dataset is divided into 70% training, 15% validation, and 15% testing. The LSTM model consists of five LSTM layers with a dropout layer following each LSTM layer. LSTM layers start at 150 neurons in the first layer and decrease to 50 neurons in the last LSTM layer as shown in Figures 3 and 4. Each dropout layer randomly

deactivates 15% of the neurons during training. The model is complied with the Adam optimizer with a learning rate of 0.0005. Early stopping is implemented with a patience of 20 epochs to prevent overtraining, and the batch size is set to 32 as LSTMs often perform better with smaller batch sizes that introduce small amounts of stochasticity. The model was trained for just over 225 epochs before early stopping was triggered.

```
Weights for layer: lstm_5
(17, 600)
(150, 600)
(600,)

Weights for layer: lstm_6
(150, 400)
(100, 400)
(400,)

Weights for layer: lstm_7
(100, 320)
(80, 320)
(320,)

Weights for layer: lstm_8
(80, 240)
(60, 240)
(240,)

Weights for layer: lstm_9
(60, 200)
(50, 200)
(200,)
```

Figure 3. LSTM architecture with weights

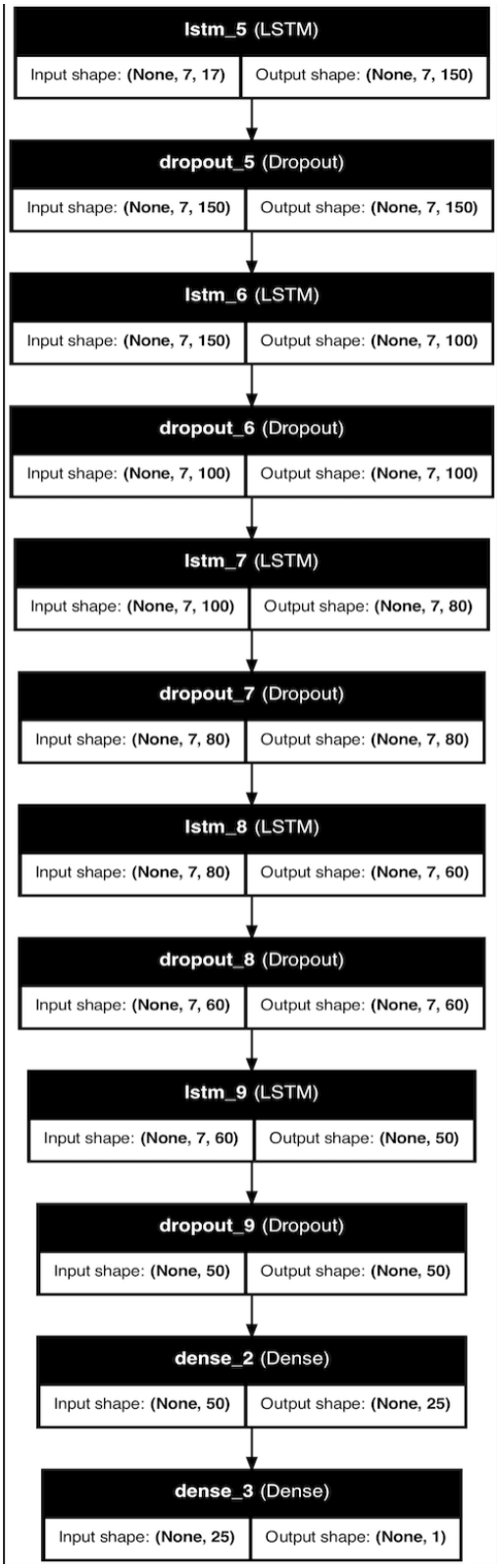

Figure 4. LSTM architecture with input and output shape sizes

III. Experiments

The primary objective of this study is to evaluate the performance of two supervised learning algorithms-

XGBoost, and LSTM RNN-in forecasting daily car wash counts based on historical weather and temporal features. The study aims to identify which model is more suitable in terms of accuracy and operational forecasting for real-world deployment in car wash environments.

The models were implemented in Python 3.11, the LSTM architecture was developed using TensorFlow and Keras, while the XGBoost model was constructed using the XGBoost library. Data preprocessing and analysis were performed using Python libraries Pandas, NumPy, and SciKit-learn.

For deployment at Rising Tide Car Wash, a user interface was created using Streamlit, a Python framework for developing web-based applications. The interface allows the user to enter in weather data and previous day car count to generate a prediction. The trained models are serialized and loaded using the joblib, pickle (XGBoost), or Hierarchical Data Format (h5) (LSTM).

To translate the model's prediction into actionable business targets, the predicted total car count for the day is first generated based on weather and temporal inputs from the user. From this value, the expected number of retail (non-member) washes is estimated by applying the current membership ratio, which currently averages approximately 55%. This is calculated as:

$$Retail\ Washes = Predicted\ Car\ Count\ \times\ 0.55$$

From this retail estimate, we can derive the number of potential membership conversions using the target conversion rate of 10%:

$$Conversion\ Goal = Retail\ Washes\ \times\ 0.10$$

To support the sales associates with hourly performance goals, this conversion goal is divided by the operating hours in the business day, resulting in:

$$Hourly\ Goal = \frac{Conversion\ Goal}{11}$$

If the conversion goal does not divide evenly into hourly targets, the remaining conversion goal is distributed to managers or additional associates to ensure all conversions are accounted for.

The evaluation metrics used in this study include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 (Coefficient of Determination). MAE and R^2 serve as the primary evaluation criteria: MAE provides interpretable error in units of car count, while R^2 quantifies the proportion of variance in the target variable explained by the model.

The LSTM model achieved lower MAE and RMSE values, indicating better predictive accuracy in terms of absolute and squared error. With a MAE of 39.51, the LSTM model's prediction error averages about 40 cars, compared to the XGBoost MAE of 43.52, suggesting that the XGBoost predictions are slightly less accurate on average. However, the LSTM's R^2 score of 0.8461 shows the model is explaining 84.61% of variance while the XGBoost achieved 87.06%. This suggests that while the LSTM performs better on average error metrics, XGBoost may generalize slightly better by capturing overall variance in the data. Overall, both models are comparable in performance from figures 5 and 6, with a slight edge to the LSTM model in terms of error reduction.

Model	MAE	RMSE	R^2
XGBoost	43.52	57.46	0.8706
LSTM	39.51	53.99	0.8461

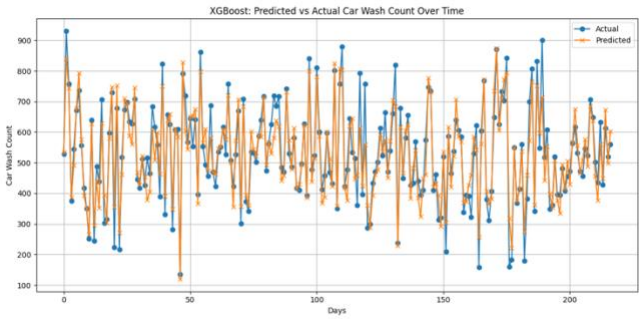

Figure 5. XGBoost predicted vs actual car count

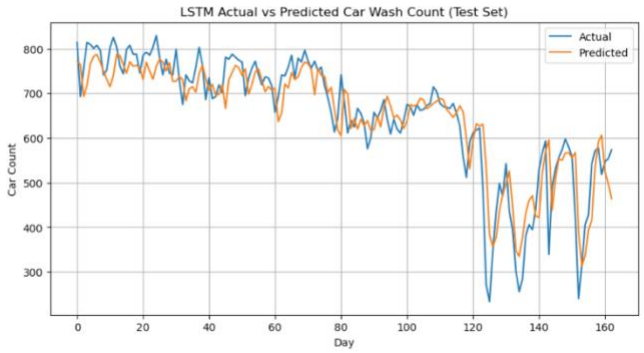

Figure 6. LSTM predicted vs actual car count

IV. Conclusion

In this study, we evaluated accuracy and performance metrics of two supervised learning algorithms with the goal of deploying the most effective model for real-time car wash count predictions. Both models demonstrated comparable performance in identifying general trends influenced by weather and temporal features. The LSTM model
outperformed XGBoost in terms of average error, achieving lower MAE and RMSE values. However, the XGBoost model explained a slightly higher proportion of variance in the data, as indicated by its superior R^2 score.
Given its performance on key error metrics, the LSTM model was selected for integration into the user interface. It will serve as the predictive engine for forecasting car wash traffic with high accuracy in practical deployment.

REFERENCES

[1]
AirNow, “Air Quality Index (AQI) Basics,” *AirNow*. https://www.airnow.gov/aqi/aqi-basics/

[2]
Nvidia, “What is XGBoost?,” *NVIDIA Data Science Glossary*, 2024. https://www.nvidia.com/en-us/glossary/xgboost/

[3]
GeeksforGeeks, “XGBoost,” *GeeksforGeeks*, Sep. 18, 2021. https://www.geeksforgeeks.org/xgboost/

[4]
DIVE INTO DEEP LEARNING, “9.2. Long Short Term Memory (LSTM) — Dive into Deep Learning 0.14.4 documentation,” *d2l.ai*. https://d2l.ai/chapter_recurrent-modern/lstm.html

[5]
S. saxena, “LSTM | Introduction to LSTM | Long Short Term Memor,” *Analytics Vidhya*, Mar. 16, 2021. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/