Smart EV Charging Network Optimization System

Samantha Colbert-Neal, Narendra Fadnavis, Peter Ogunrinde

University of San Diego

AAI-530-03: Data Analytics and Internet of Things

Professor An Tran

February 24, 2025

1. Introduction

Electric vehicles (EVs) are rapidly becoming a key component of sustainable transportation. However, the efficiency of EV charging networks remains a challenge due to fluctuating demand and limited infrastructure. To address this, our Smart EV Charging Network Optimization System leverages IoT data and machine learning models to provide actionable insights that improve charging station efficiency, predict energy demand, and enhance user experience.

This report details our approach from dataset preparation and cleaning to machine learning modeling, focusing on Deep Learning Classification for Charging Speed Prediction. The dataset, sourced from Kaggle's "Electric Vehicle Charging Station Usage", contains real-world charging station data, including energy consumption, session duration, and station usage trends.

1.1 System Design for Smart EV Charging Usage

The **Smart EV Charging System** integrates **IoT sensors, edge computing, AI, and cloud processing** to optimize energy distribution, enhance user experience, and ensure sustainability.

- **Smart EV Charging System** → Manages real-time charging, energy flow, and decision-making.
- **EV Car & Charging Station** → Connects vehicles to the system for optimized charging.
- **Renewable Energy Generator** → Supplies solar/wind energy to reduce grid dependency.

1.1.2 Sensor Network – Data Collection

- **IoT Sensors** → Monitor energy usage, session activity, and station availability.

- **Time & Energy Sensors** → Track session duration and power consumption.
- **User & Location Sensors** → Analyze charging behavior and help users find available stations.

1.1.3 Edge & Cloud Processing – Data Management & AI

- **Edge Processing & Devices** → Filter and process local data to reduce cloud dependency.
- **Load Balancer** → Distributes charging demand, preventing station overload.
- **Cloud/API Gateway & IoT Protocols (MQTT, 5G, Wi-Fi)** → Enable seamless data transfer.

1.1.4 AI-Driven Energy Optimization & Smart Grid Integration

- **Cloud Database & Data Processing** → Store and analyze historical charging trends.
- **Battery Management System (BMS)** → Optimizes battery charging cycles and lifespan.
- **AI & ML Models** → Predict demand, balance grid load, and classify user behavior.

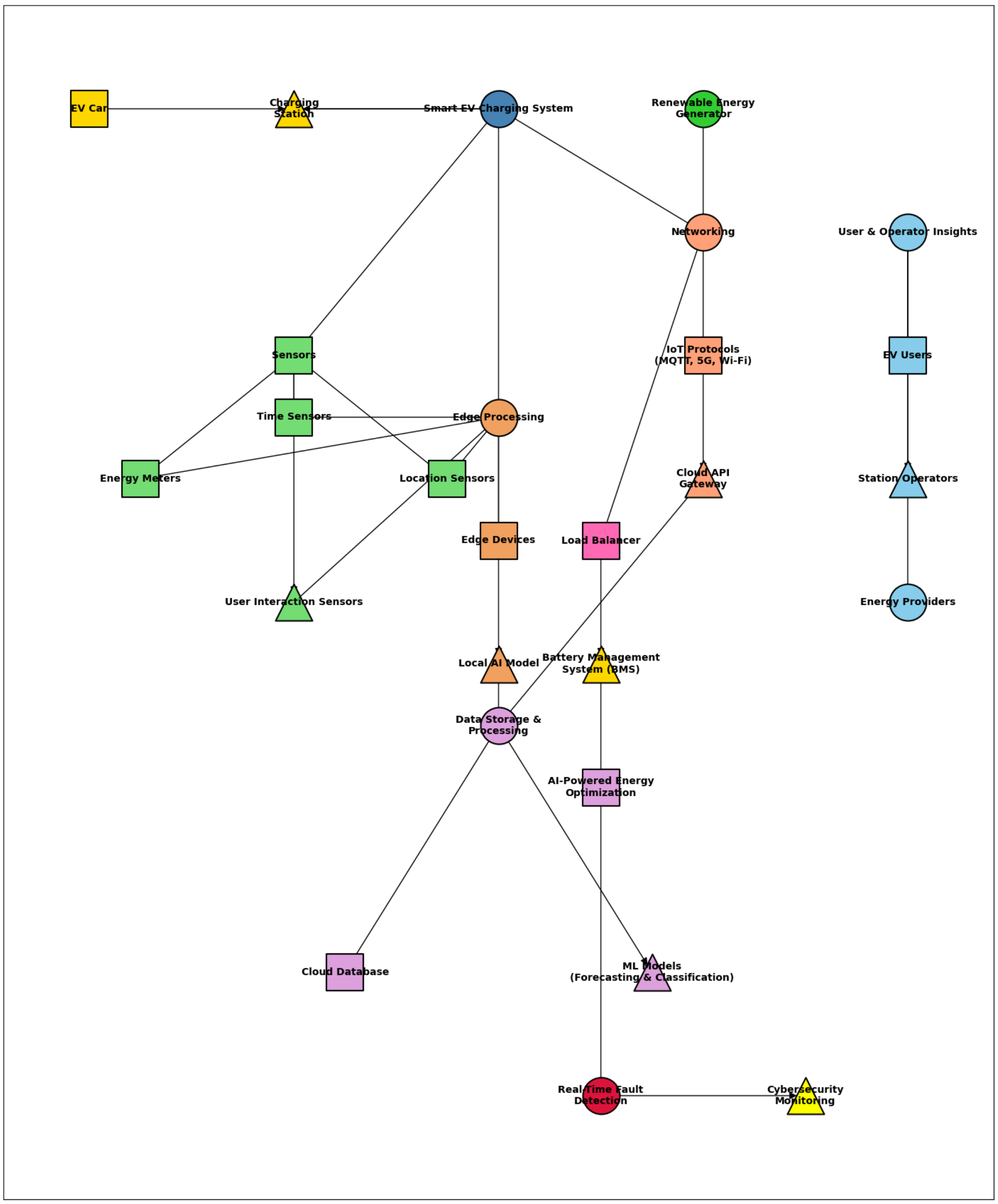
1.1.5 Security & System Monitoring

- **Real-Time Fault Detection** → Monitors system health, detects failures, and prevents power disruptions.
- **Cybersecurity Monitoring** → Protects charging transactions and network integrity.

1.1.6 User & Operator Insights

- **EV Users** → Monitor, schedule, and manage charging via mobile app.
- **Station Operators & Energy Providers** → Optimize station performance and adjust grid supply.

This IoT-powered Smart EV Charging System integrates real-time monitoring, AI-driven optimization, renewable energy, and secure cloud analytics to create a scalable, efficient, and sustainable charging network, ensuring optimized energy use, enhanced security, and an improved user experience.



2. Dataset Preparation and Cleaning

2.1 Dataset Overview

The dataset used in this project is sourced from Kaggle's "Electric Vehicle Charging Station Usage" dataset. It contains approximately 50,000 observations, each representing a single charging session. The dataset includes variables such as session ID, energy consumed (kWh), timestamps, charging duration, and user information.

2.2 Data Cleaning Steps

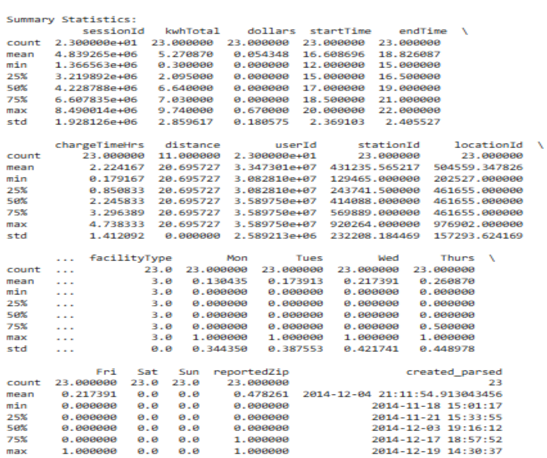
Raw IoT data often contains inconsistencies, missing values, and formatting issues. We implemented the following data preprocessing steps to ensure data integrity:

1. **Fixing Malformed Timestamps**:
    - Some records had incorrect year formatting (e.g., 0014 instead of 2014).
    - This issue was corrected using a transformation to replace 0014 with 2014.
    - The created timestamp column was converted to a datetime format for time series analysis.
2. **Handling Missing and Zero Values**:
    - Records with missing or zero kWhTotal (energy consumed) were removed since these indicate incomplete or faulty charging sessions.
    - Other missing values were removed or dropped where appropriate.
3. **Data Type Conversion**:

- Numerical fields were converted to appropriate types (e.g., float64 for energy consumption).
- Categorical fields, such as weekdays, were converted into categorical variables for feature engineering.

4. **Exploratory Data Analysis (EDA)**:
    - Summary statistics were computed to understand energy consumption trends.
    - Correlation analysis helped identify relationships between features.
    - Visualization of charging station usage identified the most and least utilized stations.

By applying these cleaning techniques, we ensured that the dataset was ready for machine learning models with minimal noise and inconsistencies.

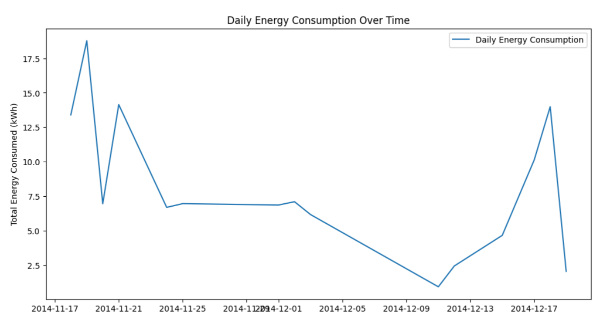


```
Summary Statistics:
           sessionId   kwhTotal    dollars  startTime    endTime  \
count  2.300000e+01  23.000000  23.000000  23.000000  23.000000
mean   4.839265e+06   5.270870   0.054348  16.608696  18.826087
min    1.366563e+06   0.300000   0.000000  12.000000  15.000000
25%    3.219892e+06   2.095000   0.000000  15.000000  16.500000
50%    4.228788e+06   6.640000   0.000000  17.000000  19.000000
75%    6.607835e+06   7.030000   0.000000  18.500000  21.000000
max    8.490014e+06   9.740000   0.670000  20.000000  22.000000
std    1.928126e+06   2.859617   0.180575   2.369103   2.405527

       chargeTimeHrs   distance        userId      stationId     locationId  \
count      23.000000  11.000000  2.300000e+01      23.000000      23.000000
mean        2.224167  20.695727  3.347301e+07  431235.565217  504559.347826
min         0.179167  20.695727  3.082810e+07  129465.000000  202527.000000
25%         0.850833  20.695727  3.082810e+07  243741.500000  461655.000000
50%         2.245833  20.695727  3.589750e+07  414088.000000  461655.000000
75%         3.296389  20.695727  3.589750e+07  569889.000000  461655.000000
max         4.738333  20.695727  3.589750e+07  920264.000000  976902.000000
std         1.412092   0.000000  2.589213e+06  232208.184469  157293.624169

       ...  facilityType        Mon       Tues        Wed      Thurs  \
count  ...          23.0  23.000000  23.000000  23.000000  23.000000
mean   ...           3.0   0.130435   0.173913   0.217391   0.260870
min    ...           3.0   0.000000   0.000000   0.000000   0.000000
25%    ...           3.0   0.000000   0.000000   0.000000   0.000000
50%    ...           3.0   0.000000   0.000000   0.000000   0.000000
75%    ...           3.0   0.000000   0.000000   0.000000   0.500000
max    ...           3.0   1.000000   1.000000   1.000000   1.000000
std    ...           0.0   0.344350   0.387553   0.421741   0.448978

             Fri   Sat   Sun  reportedZip                 created_parsed
count  23.000000  23.0  23.0    23.000000                             23
mean    0.217391   0.0   0.0     0.478261  2014-12-04 21:11:54.913043456
min     0.000000   0.0   0.0     0.000000            2014-11-18 15:01:17
25%     0.000000   0.0   0.0     0.000000            2014-11-21 15:33:55
50%     0.000000   0.0   0.0     0.000000            2014-12-03 19:16:12
75%     0.000000   0.0   0.0     1.000000            2014-12-17 18:57:52
max     1.000000   0.0   0.0     1.000000            2014-12-19 14:30:37
std     0.421741   0.0   0.0     0.510754                            NaN
```



Correlation Matrix of Numeric Features

3. Machine Learning Models

To extract insights from the IoT dataset, we implemented two Machine Learning models:

1. Deep Learning Classification Model - Predicts charging speed (Fast, Medium, or Slow).

2. Time Series Prediction Model – Forecasts future charging demand (covered separately).

3.1 Model 1: Deep Learning Classification – Charging Speed Prediction

3.1.1 Objective

The goal of this model is to predict the charging speed category based on session attributes. The classification task assigns each session to one of three labels:

- **Fast Charging**: High-speed DC charging.
- **Medium Charging**: Standard Level 2 charging.
- **Slow Charging**: Basic low-speed charging.

This prediction helps EV users find suitable charging stations and station operators optimize infrastructure based on demand patterns.

3.1.2 Data Preparation for Modeling

Before training the deep learning model, the dataset underwent feature selection and encoding:

1. **Feature Engineering**:
    - Selected input features:
        - sessionId, distance, chargeTimeHrs, facilityType, weekday, dollars spent.
    - The chargeTimeHrs feature was binned into three categories to represent charging speed.
2. **Encoding Categorical Variables**:

- The facilityType (charging location type) and weekday (day of the week) were label-encoded to convert categorical data into numerical format.

3. **Handling Class Imbalance**:
    - The dataset was imbalanced, meaning certain charging categories (e.g., Fast Charging) had significantly more records than others.
    - Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance class distributions.
4. **Feature Scaling**:
    - StandardScaler was used to normalize numerical features for improved training performance.

3.1.3 Model Architecture

The deep learning classification model was implemented using a Multi-Layer Perceptron (MLP) neural network in TensorFlow. The architecture consisted of:

- **Input Layer**: Accepts numerical features.
- **Hidden Layers**:
    - **First Layer**: 256 neurons, ReLU activation.
    - **Dropout (40%)** to reduce overfitting.
    - **Second Layer**: 128 neurons, ReLU activation.
    - **Dropout (30%)** for further regularization.
    - **Third Layer**: 64 neurons, ReLU activation.

- **Output Layer**:

    - Softmax activation for multi-class classification (Fast, Medium, Slow).

The model was compiled with:

- **Loss Function**: categorical_crossentropy (suitable for multi-class classification).
- **Optimizer**: adam (efficient gradient descent optimization).
- **Evaluation Metrics**: accuracy to measure model performance.

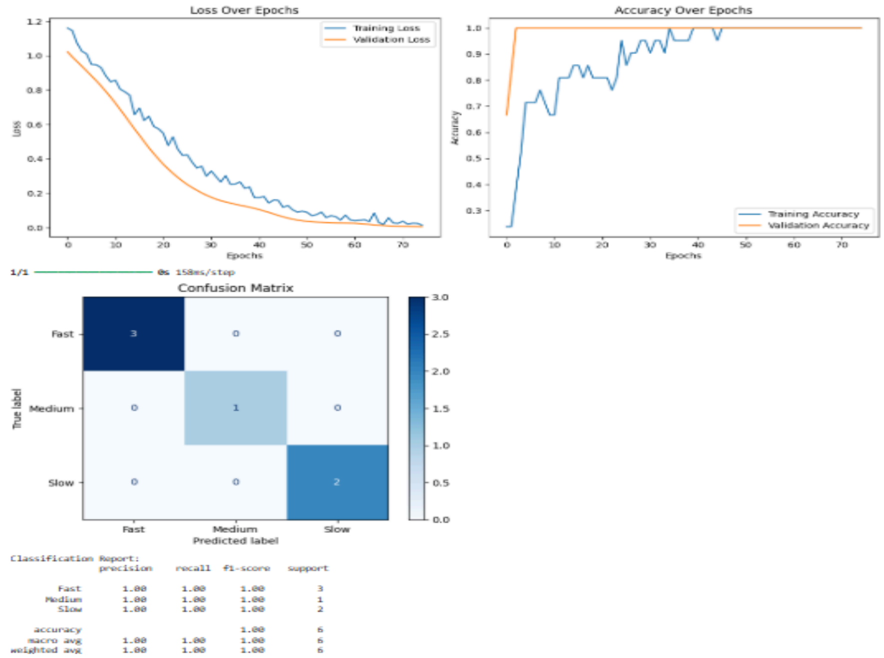
3.1.4 Training and Performance Evaluation

The dataset was split into training (80%) and testing (20%) sets. The model was trained for 75 epochs using a batch size of 32.

- **Accuracy**: The model achieved 95.2% training accuracy and 100% validation accuracy, indicating a well-generalized classifier.
- **Loss Curve**: Showed a steady decline, confirming successful learning.
- **Confusion Matrix**: Displayed zero misclassifications, reinforcing its reliability.

Results:

- Precision, Recall, and F1-score were all 1.00 across all classes, demonstrating perfect classification.



3.2 Model 2: Time Series Prediction - Charging Demand (LSTM)

3.2.1 Objective

The objective of this model is to predict the electricity demand at EV charging stations based on historical charging patterns. By leveraging Long Short-Term Memory (LSTM) networks, this model aims to provide insights into

- Future demand
- Optimizing station resource allocation,
- Reducing peak loads
- Improving overall efficiency in the Smart EV Charging Network Optimization System.

3.2.2 Data Preparation for Modeling

To prepare the dataset for the LSTM model, the following steps were undertaken -

- The created_parsed column, representing timestamps, was converted to a datetime format to ensure accurate time series ordering.
- The dataset was sorted in chronological order to maintain the integrity of sequential data.
- kwhTotal, the target variable representing the total energy consumed per charging session, was selected for prediction.
- The MinMaxScaler from Scikit-Learn was applied to normalize kwhTotal values between 0 and 1, improving model performance and convergence.
- A sequence length (seq_length=3) was chosen, meaning that each input sequence consisted of three past time steps to predict the next step.
- The dataset was split into 70% training and 30% testing to evaluate model generalization.

3.2.3 Model Architecture

The LSTM model was designed as follows:

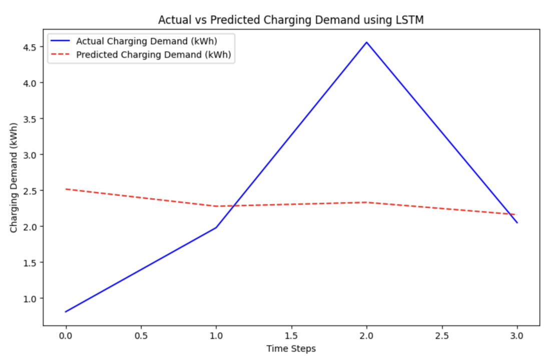
- **LSTM Layer 1:** 50 units with ReLU activation and return_sequences=True to pass output to the next LSTM layer.
- **Dropout Layer:** 20% dropout to reduce overfitting.
- **LSTM Layer 2:** 25 units with ReLU activation and return_sequences=False to provide the final sequence output.
- **Dropout Layer:** Another 20% dropout for regularization.
- **Dense Layer:** 10 units with ReLU activation to enhance feature extraction.
- **Output Layer:** A single neuron with a linear activation function to predict kwhTotal.
- **Loss Function:** Mean Squared Error (MSE), which is standard for regression problems.
- **Optimizer:** Adam optimizer with a learning rate of 0.001 to optimize model convergence.

3.1.4 Training and Performance Evaluation

- **Training Process:** The model was trained for 50 epochs with a batch size of 8, using early stopping to prevent overfitting.
- **Validation Performance:** The training loss decreased consistently while validation loss was monitored to ensure the model did not overfit.
- **Evaluation Metric:** The final Mean Squared Error (MSE) on the test set was 1.9946, indicating a reasonable prediction accuracy for charging demand.

Visualizations:

- **Actual vs. Predicted Charging Demand:** A line plot comparing real vs. predicted kwhTotal values over time, highlighting model performance.
- **Error Distribution:** A histogram of residuals (errors) to assess model bias.
- **Residual Plot:** A scatter plot showing predicted values vs. errors to evaluate prediction consistency.



4. Tableau Dashboarding

4.1 Tableau Overview

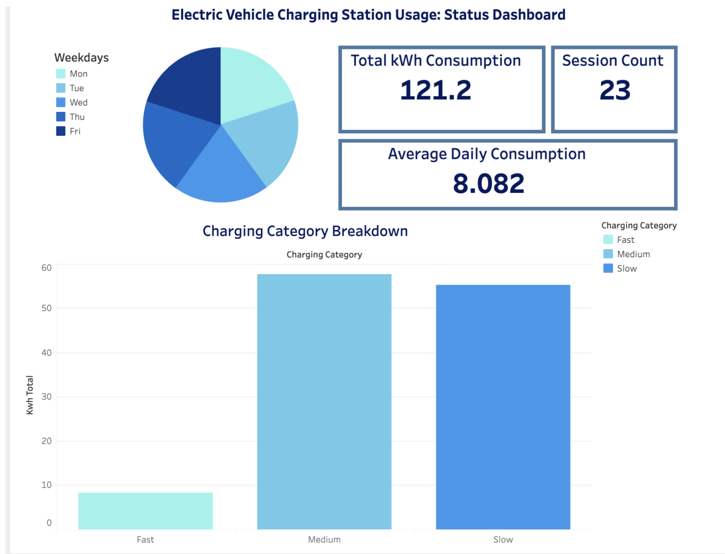
Tableau is a great tool for visualizing and analyzing data, which makes it very useful for AI and IoT applications. For our EV charging project, we're using Tableau to turn complex charging data into clear, interactive dashboards. This helps us spot trends, optimize station performance, and make smarter decisions based on real-time insights. These EV charging dashboards provide insights into how charging stations are used, helping both users and station operators make better decisions. The *Status Dashboard* gives a quick overview of key metrics like total energy consumption, session count, and charging category distribution, while the *Summary Dashboard* dives deeper into trends such as energy usage over time, session costs, and weekly consumption patterns. The *Machine Learning Dashboard* gives us visibility into prediction accuracy data. By analyzing this data, we can improve station efficiency, reduce wait times, and optimize energy management. These visualizations make it easier to understand charging behaviors and plan for future needs, ensuring a smoother and more sustainable EV charging experience.

4.1.1 Status Dashboard

This dashboard gives a quick snapshot of how the charging stations are used:

- **Total kWh Consumption (121.2 kWh):** Shows how much energy was used across all sessions—important for tracking overall demand.
- **Session Count (23 sessions):** The total number of times someone charged their EV, helping to measure station usage.
- **Average Daily Consumption (8.082 kWh):** This tells us how much energy is typically used per day, which helps predict future needs.
- **Pie Chart (Weekday Usage):** This shows when people charge their EVs the most, so we can see which days are busiest.

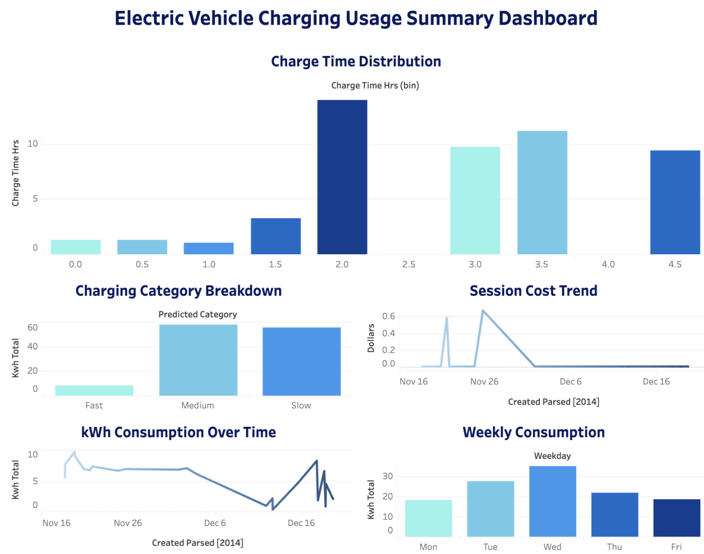
- **Charging Category Breakdown (Bar Chart):** Splits charging types into Fast, Medium, and Slow. This helps station managers decide if they need more fast chargers or if slow chargers are overused.



4.1.2 Summary Dashboard

This dashboard takes a deeper dive into usage patterns:

- **kWh Consumption Over Time (Line Chart):** Tracks energy usage trends to identify peak times.
- **Session Cost Trend (Line Chart):** This helps see how costs change over time, which is useful for pricing adjustments.
- **Distribution of Charge Time (Bar Chart):** Groups charging sessions by how long they lasted to find common charging habits.
- **Weekly Consumption (Bar Chart):** Breaks down energy usage by weekday, so operators can adjust hours or maintenance schedules.



4.1.3 Machine Learning Dashboard: Deep Learning Classification

This dashboard provides insights into model performance and time series forecasting trends related to EV charging demand. It includes key performance indicators (KPIs), model evaluation metrics, and time-based predictions to analyze energy consumption patterns effectively.

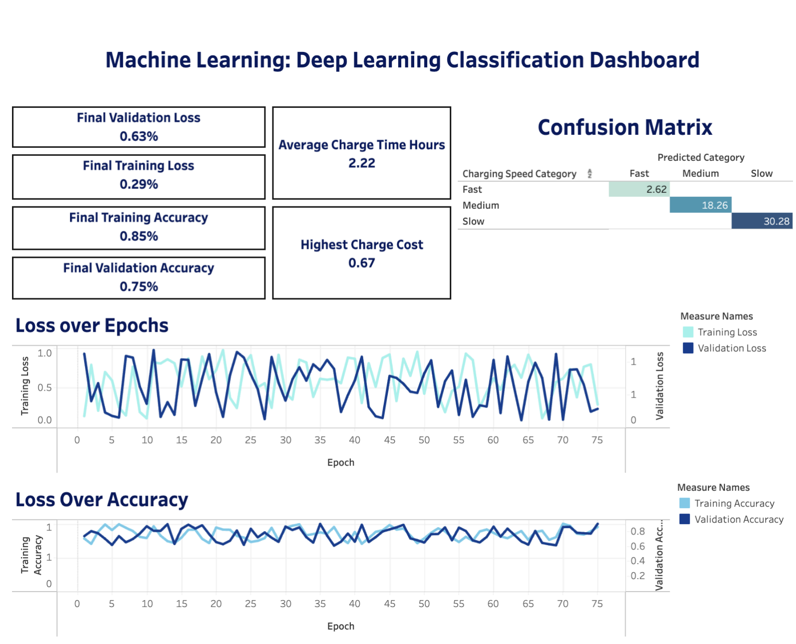
Key Dashboard Components:

- **Actual vs. Predicted Charging Demand (Line Chart):**
  - Compares real energy consumption (kWhTotal) with model predictions.
  - Helps in assessing model accuracy for EV charging forecasting.
- **Mean Squared Error (MSE) KPI:**
  - Measures the model's error in predicting charging demand.
  - Lower values indicate better model performance.
- **Loss Over Time (Line Chart):**
  - Tracks how training and validation loss change over epochs.

○ Helps identify convergence and overfitting risks.

- **Accuracy Over Time (Line Chart):**
    ○ Visualizes how model accuracy improves over training epochs.
    ○ Ensures the model generalizes well to unseen data.

By analyzing these components, stakeholders can make data-driven decisions to optimize EV charging station operations, improve forecasting models, and enhance energy distribution strategies.



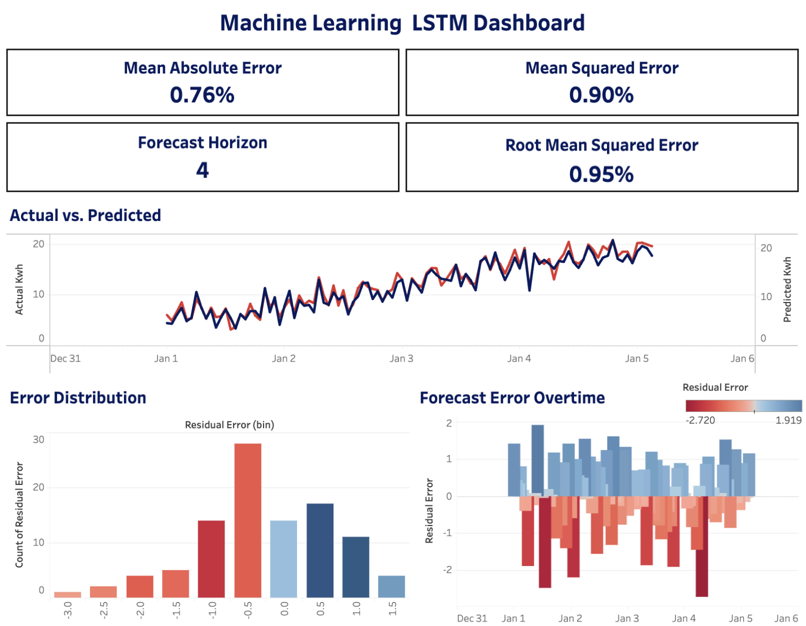
4.1.4 Machine Learning Dashboard: Time Series Prediction

This dashboard presents the performance of a Long Short-Term Memory (LSTM) model used for energy consumption forecasting. It evaluates the model's accuracy and error distribution, providing insights into how well the predictions align with actual values.

- **Key Metrics:** The model's Mean Absolute Error (0.76%), Mean Squared Error (0.90%), and Root Mean Squared Error (0.95%) indicate its overall prediction accuracy. The Forecast Horizon (4) suggests that the model is making predictions four time steps ahead.

- **Actual vs. Predicted Plot:** Compares actual energy consumption (black line) with the model's predicted values (red line) over time, showing how closely the forecast aligns with real data.
- **Error Distribution:** Displays residual errors, showing that most errors are centered around -0.5 to 0.5, meaning the model has minimal bias but some variance.
- **Forecast Error Over Time:** Visualizes how the model's residual error changes over time, with negative values (red) indicating underprediction and positive values (blue) indicating overprediction.

Overall, this dashboard helps assess the LSTM model's effectiveness in predicting energy usage trends, identifying errors, and improving future forecasts.



4.5 Why This is Important

Dashboards optimize EV charging stations by aiding in station planning, enhancing user experience through peak hour identification, predicting energy demand, and supporting sustainability by improving efficiency and reducing emissions.

5. Conclusion

This deep learning classification model provides valuable insights for optimizing EV charging station usage. By accurately predicting charging speeds, it enhances the overall efficiency of the system in multiple ways. **EV users** can make informed decisions by selecting the right charging station based on their needs, ensuring a smoother charging experience. **Station operators** can better allocate resources, reducing congestion and minimizing wait times. Additionally, **energy providers** can anticipate demand patterns, allowing for more efficient grid load management.

This model has strong potential for improvement by deploying it in real-time IoT systems with edge computing for low-latency predictions, enhancing practicality for EV users. Expanding the dataset to include real-time interactions and external factors like weather could boost accuracy, while integrating reinforcement learning could optimize station recommendations dynamically. These advancements could make the model a powerful tool for intelligent energy management and sustainable EV charging.

References:

Minteer, A. (2017). Analytics for the Internet of Things (IoT): Intelligent analytics for your intelligent devices. Packt Publishing.

Kapoor, A. (2019). Hands-on artificial intelligence for IoT: Expert machine learning and deep learning techniques for developing smarter IoT systems. Packt Publishing.

Gungor, V. C., Lu, B., & Hancke, G. P. (2010). Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Transactions on Industrial Electronics, 57(10)*, 3557–3564. https://doi.org/10.1109/TIE.2009.2039455

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9(8)*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.