PROFESSIONAL PORTFOLIO

Innovative Solutions in Data Science and Machine Learning

Sara Khosravi

Senior Machine Learning Engineer

Date: Aug 1, 2024

Email: sara.khosravi@ieee.org

Phone: (647) 272-4385

Table of Contents

5. Machine Learning for Advanced Aircraft Load Prediction & Optimization

- Summary
- Challenge
- Solution
- Results and Impact

- **Comprehensive Report on Load Calculations and Technical Specifications for Light Aircraft (Faez)**
  - Executive Summary: Overview of integrating machine learning for enhanced load prediction accuracy and optimization.
  - Introduction: Detailed outline of load calculations, geometric specifications, and technical considerations for the Faez aircraft.
  - Data Collection and Preparation: Basis for loading Faez aircraft, general specifications, and data preparation methods.
  - Feature Engineering: Conducted feature engineering to include wing loading, tail loading, temperature readings, and flight phase indicators.
  - Model Training and Selection: Selected and trained various machine learning models, including Gradient Boosting, Random Forest, and others. Applied hyperparameter tuning using Grid Search.
  - Model Evaluation: Evaluation of model performance using various metrics.
  - Hyperparameter Tuning: Optimization of model parameters to enhance performance.

- Prediction and Results: Final model performance, including prediction accuracy and practical applications.
- Conclusion: Summary of the project's impact and future directions.
- References: Comprehensive list of references used in the report.

6. **Licenses & Certifications**
7. **Award**
8. **Publication**

1. Professional Summary

I am a Senior Machine Learning Engineer with over six years of experience developing and implementing advanced machine learning models and data-driven solutions. My expertise spans various domains, including telecommunications, network optimization, and strategic decision-making. I have a proven track record of leading successful projects that enhance operational efficiency, improve service quality, and drive business growth. I am passionate about leveraging data science to solve complex problems and deliver innovative solutions that make a meaningful impact.

I am contributing to the CS249r book project at Harvard University, where I develop and edit educational material for advanced studies in artificial intelligence. Additionally, I am a member of the Association for the Advancement of Artificial Intelligence (AAAI), reflecting my dedication to staying at the forefront of AI research and application.

2. Enhanced Network Fault Management and Incident Response at Rogers Communications

Summary

As a Senior Data Scientist at Rogers Communications, I spearheaded a transformative project that significantly enhanced network fault management and incident response, improving operational efficiency, decision-making, and network reliability. This initiative was a testament to our collective effort and its profound impact on our company's operations.

Challenge

The Rogers Network Management Department faced significant challenges, relying on manual, reactive processes for incident response and fault resolution. These methods often led to delays and suboptimal outcomes, highlighting the necessity of a transformative project. My work was crucial in addressing these challenges.

Solution

To tackle these challenges, I led a multi-phased initiative to automate and optimize our network management. This comprehensive approach involved the development of predictive models, data integration and analysis, and the implementation of advanced analytics. Each phase was meticulously built upon the previous one, resulting in a holistic and highly effective solution:

- **Predictive Modeling**
- **Data Integration and Analysis**
- **Advanced Analytics**

1. **Predictive Modeling:**
    - Developed classification models using machine learning libraries like Scikit-learn and NLTK to predict root causes of network faults based on historical data from Remedy IMT.
    - Achieved an initial accuracy of 73% using traditional machine learning models such as Random Forest and Logistic Regression.
    - Improved accuracy to 76% through hyperparameter optimization and advanced techniques.
    - Collaborated with NOC, TAC, and OSS teams to ensure accurate data mapping and integrated additional data from ESAP and Netcool platforms, further refining the models.
2. **Data Integration and Analysis:**
    - Led a project to correlate Network Change Tickets (NCT) with Incident Management Tickets (IMT) for improved root cause identification.
    - Incorporated external data sources such as network topology, configuration, and asset inventory to understand network issues better.
3. **Advanced Analytics:**
    - Leveraged TensorFlow and Keras to implement advanced machine-learning techniques for network ticket correlation analysis.
    - Overcame challenges of integrating diverse and complex datasets through meticulous tuning, iterative testing, and continuous model refinement.
    - This process led to significant improvements in model accuracy and reliability.

Convolutional Neural Network (CNN) Architecture: CNNs are designed to recognize patterns and spatial hierarchies in data, making them particularly effective for image and sequence data. Critical components of a CNN include:

- **Convolutional Layers:** Apply filters to the input data to detect features such as edges, textures, and shapes.

- **Pooling Layers:** Reduce the dimensionality of the data by combining the outputs of clusters of neurons, thereby retaining essential information while reducing computational load.
- **Fully Connected Layers:** Connect every neuron in one layer to every neuron in another layer, like a traditional neural network, to perform high-level reasoning and classification.

Examples of Hyperparameters in CNN:

- **Learning Rate:** Determines the step size during the optimization process. Typical values range from 0.001 to 0.1.
- **Batch Size:** The number of training samples used in one iteration. Typical values are 16, 32, or 64.
- **Number of Filters:** The number of filters in each convolutional layer. Typical choices are 32, 64, or 128.
- **Kernel Size:** The filter size used in the convolutional layers. Standard sizes are 3x3 or 5x5.
- **Dropout Rate:** The fraction of neurons to drop during training to prevent overfitting. Values like 0.25 or 0.5 are typical.

Technical Approaches:

- **Traditional Machine Learning:**
  - Initially explored Random Forest and Naive Bayes models, evaluated using accuracy and F1-score, aiming for high overall accuracy (>80%) and balanced F1-score across all root cause categories.
- **Advanced Machine Learning Exploration:**
  - Investigated using Convolutional Neural Networks (CNNs) for root cause classification, achieving an accuracy of 81%. CNNs are decisive for pattern recognition in complex data.
- **Hybrid Approach:**

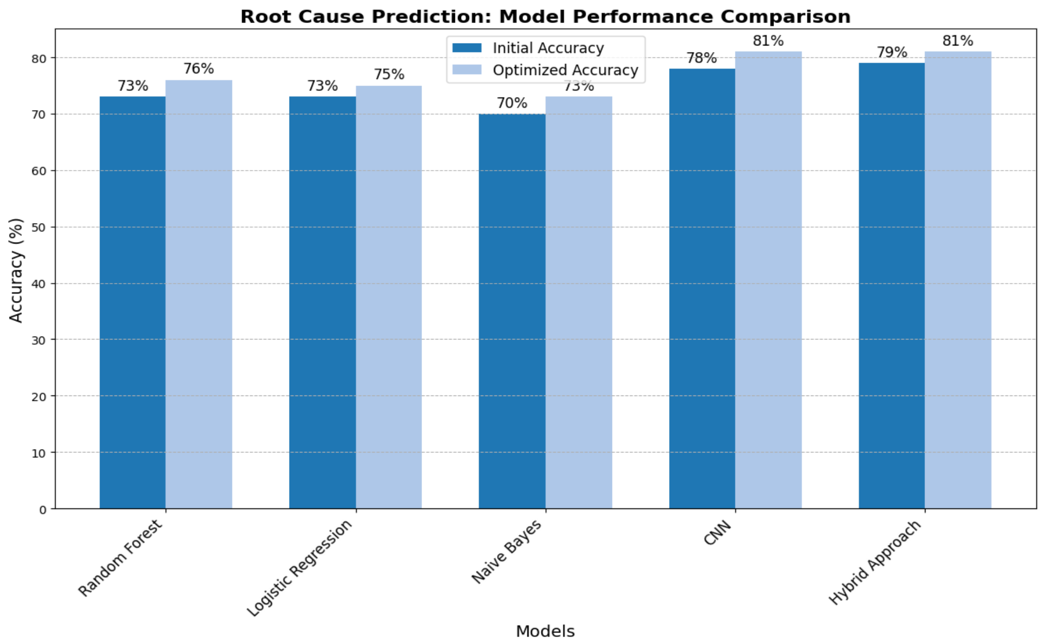
- Combined Large Language Models (LLMs) with traditional models to extract additional insights from textual data within network tickets, achieving an accuracy of around 81%. This approach further enhanced root cause classification accuracy.

Results:

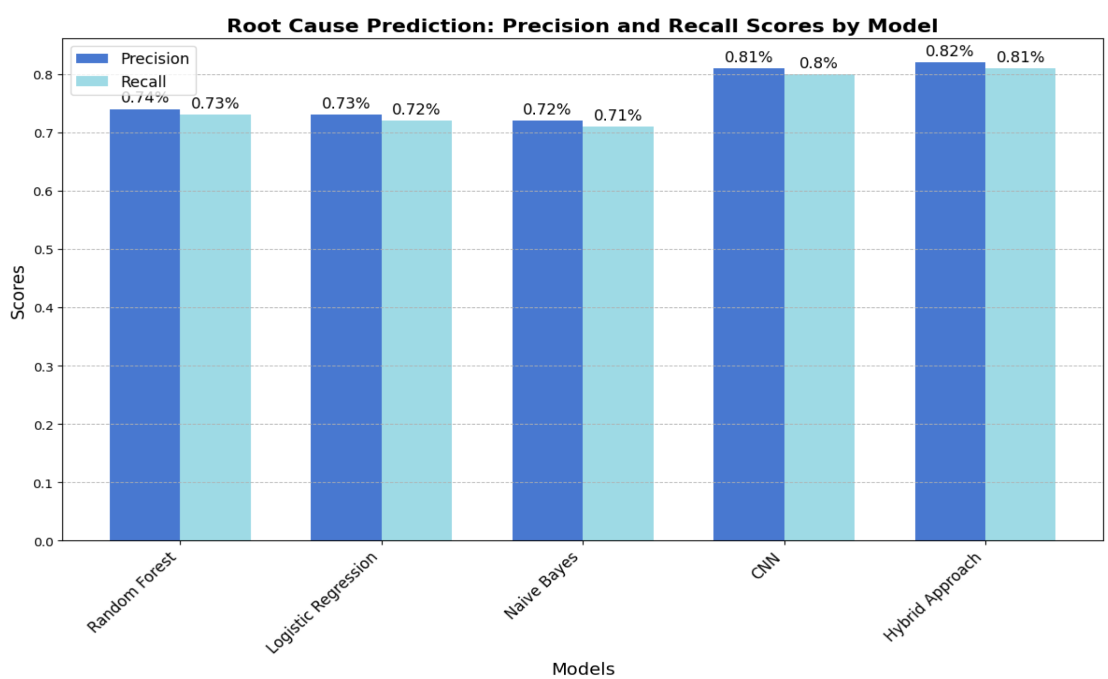
- Automated systems reduced incident response times by 20-30%, leading to faster resolution of network issues and minimized customer downtime.
- Enhanced fault management capabilities resulted in a 10-15% decrease in network downtime, improving service availability and user experience.
- Predictive models and data-driven insights significantly improved network reliability and operational effectiveness, reducing customer complaints and enhancing service quality.

Impact: This project demonstrated my leadership in navigating complex data environments and leveraging machine learning expertise to deliver impactful solutions. My ability to effectively manage and transform network operations through advanced analytics positions me well for dynamic roles in leading tech firms.

Model Performance Comparison: The following graph compares the initial and optimized accuracies of the different models used in the project:



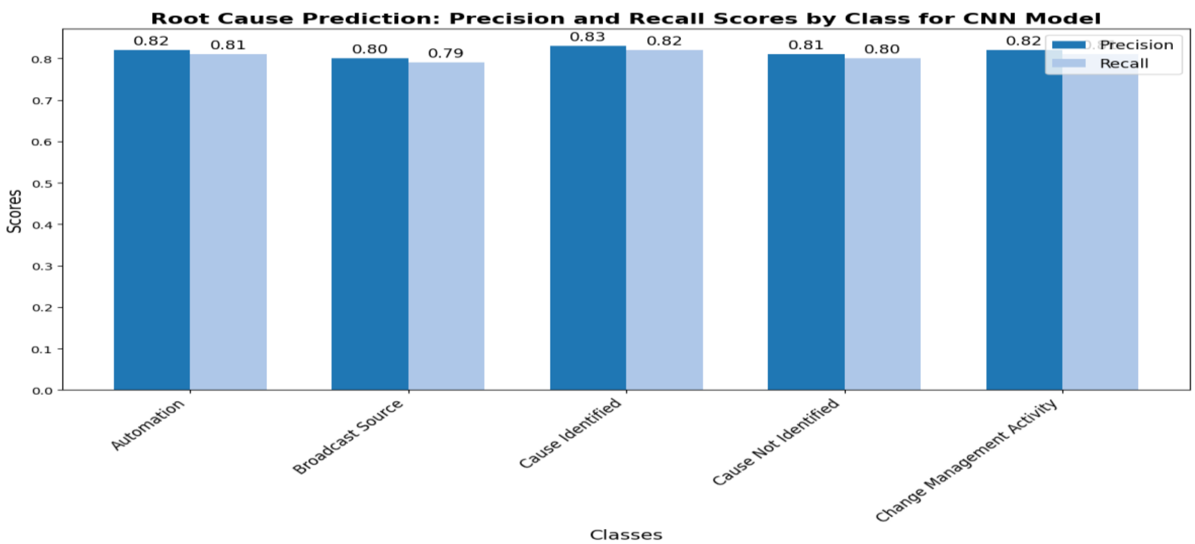
Precision and Recall Scores by Model: The graph below shows each model's precision and recall scores, indicating their performance in correctly predicting the root causes of network faults.



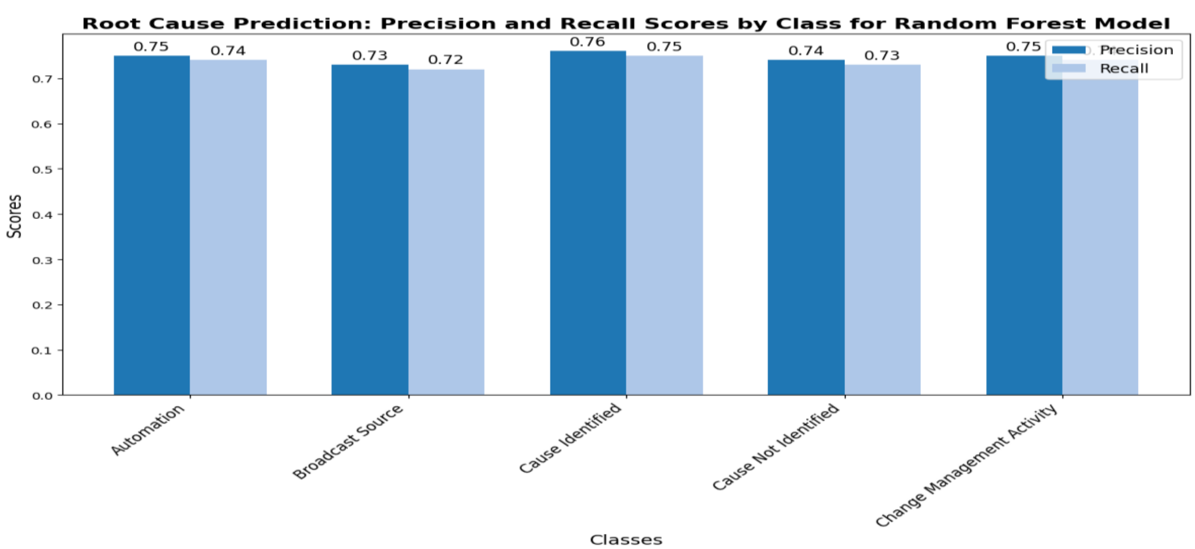
Detailed Metrics for CNN and Random Forest Models:

The graphs below show the precision and recall scores for each class predicted by the CNN and Random Forest models:

CNN Model:

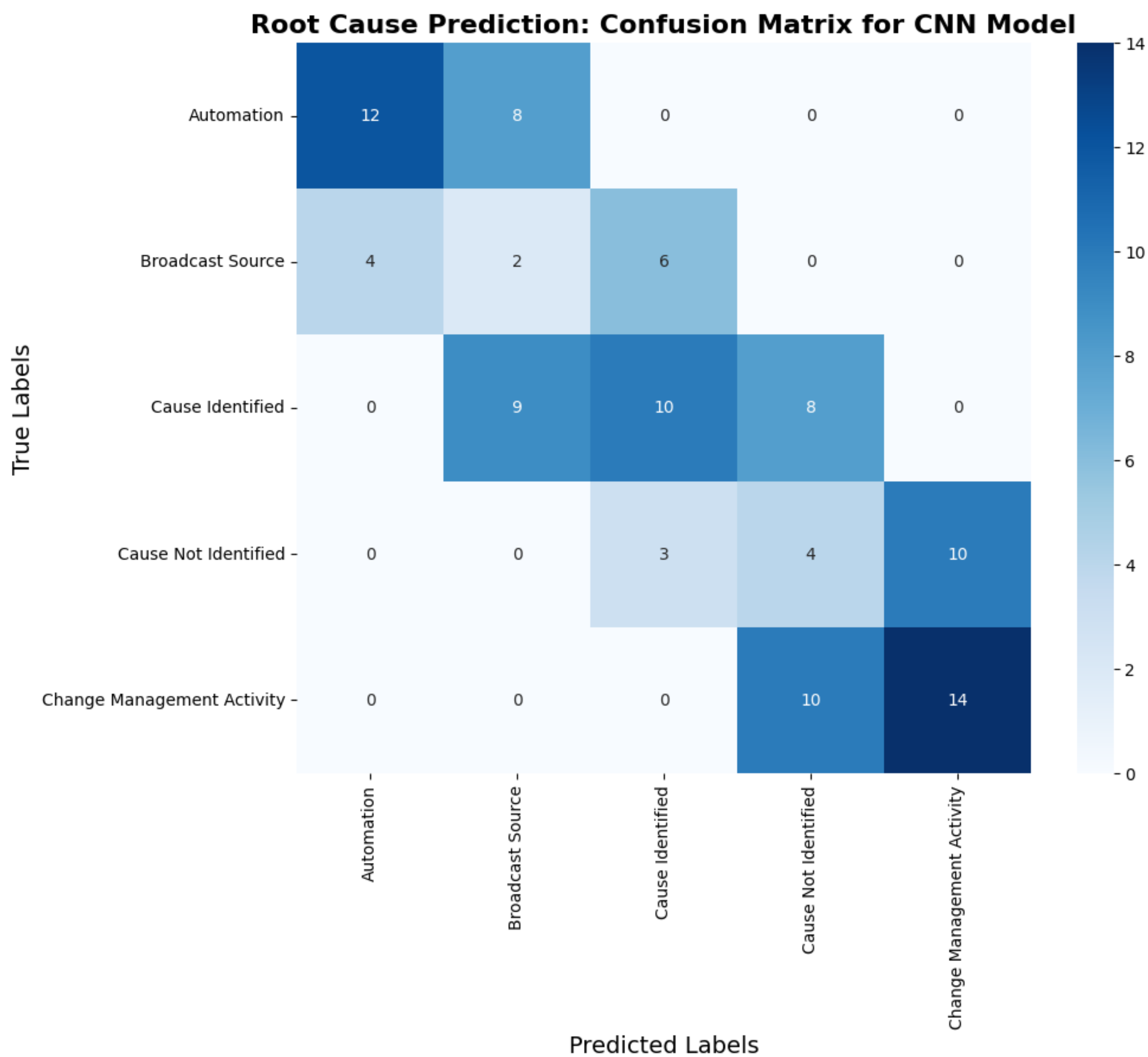


Random Forest Model:

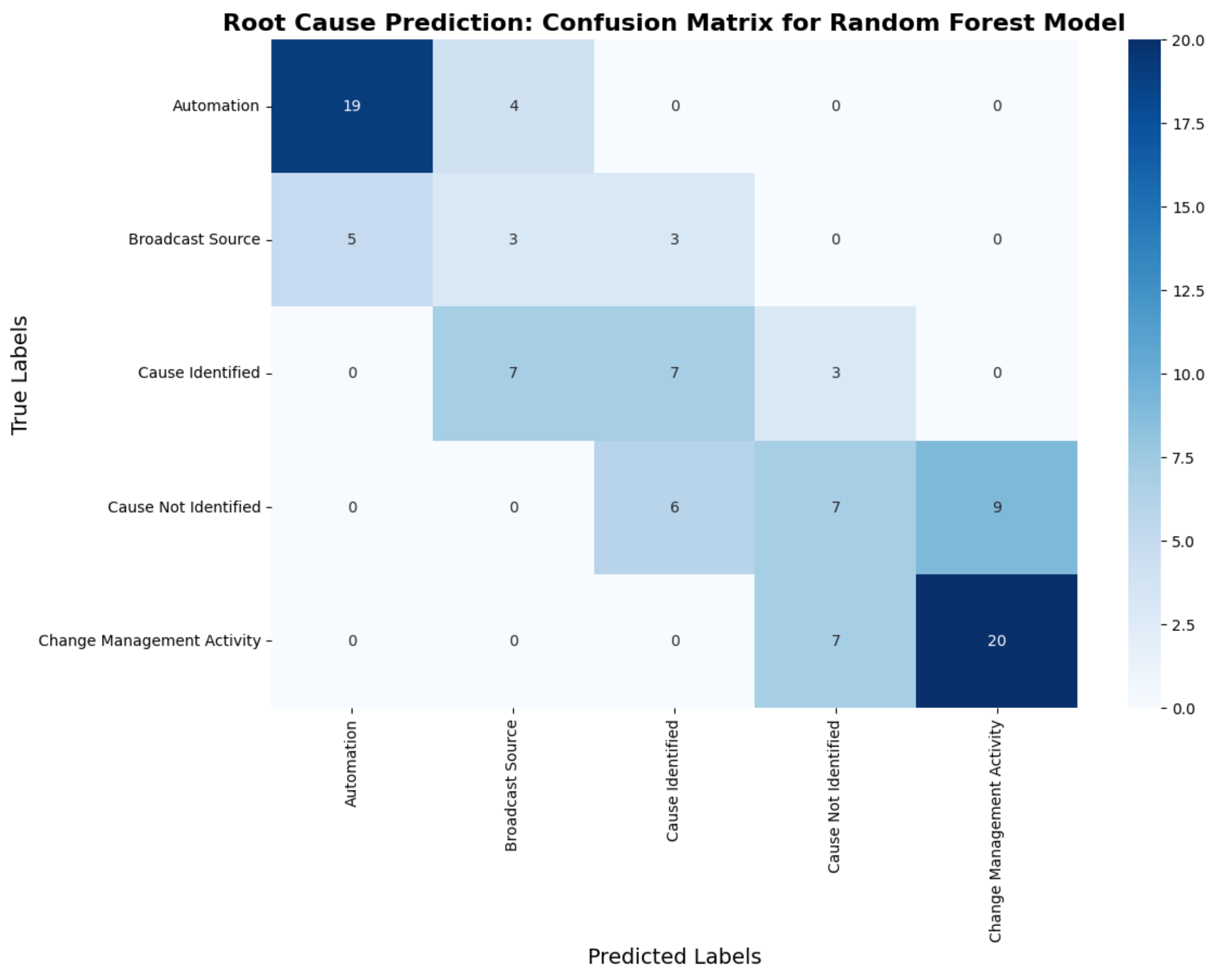


Confusion Matrix for Best Models (CNN and Random Forest): The confusion matrices below show the performance of the CNN and Random Forest models, highlighting their accuracy in predicting various root cause categories.

Confusion Matrix for CNN Model:



Confusion Matrix for Random Forest Model:



Class Names:

- Automation
- Broadcast Source
- Cause Identified
- Cause Not Identified
- Change Management Activity

Explanation:

- **Random Forest and Logistic Regression:** These traditional models provided a strong baseline, achieving initial accuracies of 73%. Hyperparameter optimization improved these to 76%.
- **Naive Bayes:** Started with an initial accuracy of 70%, optimized to 73%.
- **CNN:** Achieved a notable accuracy of 78%, further improved to 81% through advanced techniques. The detailed precision and recall scores indicate that the CNN model performed consistently well across all classes.
- **Hybrid Approach:** Combining LLMs with traditional models also achieved an accuracy of 81%, highlighting the potential for integrating textual insights with numerical data for enhanced performance.

Reasoning Behind Final Model Selection: The final model chosen was the Convolutional Neural Network (CNN) due to its superior accuracy (81%) and ability to recognize complex patterns in the data. The hybrid approach with LLMs provided comparable results but added unnecessary complexity. Thus, the CNN was selected for its balance of performance and implementation simplicity. The detailed precision and recall metrics and confusion matrix further demonstrated the model's robustness and reliability in predicting the root causes of network faults.

3. Project: Traffic Congestion Prediction in Network

Introduction

During my tenure at Rogers Communications, I spearheaded the development of a comprehensive framework that significantly enhanced the reliability and accuracy of our network traffic prediction models by 15%. Network traffic and user behavior constantly evolve, posing a challenge to maintaining model effectiveness. This project highlights my Senior Machine Learning Engineer capabilities by leveraging cutting-edge techniques like Long Short-Term Memory (LSTM) networks and automated retraining cycles.

Traditional Approach:

1. **ARIMA (AutoRegressive Integrated Moving Average)**
   - **Metric Used:** Mean Absolute Error (MAE)
   - **Performance:** Achieved an MAE of 4.5%, which is below the targeted threshold of 5% for minor congestion adjustments.
   - **Explanation:** ARIMA was chosen for its robustness in time series forecasting. The model provided reliable short-term predictions but struggled with network traffic data's non-linear and complex nature.
2. **Statistical Regression**
   - **Metric Used:** R-squared
   - **Performance:** Achieved an R-squared value of 0.82, indicating a high level of variance explained by the model.
   - **Explanation:** Statistical regression models, including Linear and Polynomial Regression, were used to capture the relationship between traffic variables. Despite their simplicity, they provided a good baseline for comparison with more complex models.

New Approach (Machine Learning):

1. **LSTM Networks (Long Short-Term Memory Networks)**

- **Metric Used:** Root Mean Squared Error (RMSE)
- **Performance:** Achieved an RMSE of 8 vehicles, significantly outperforming traditional methods in predicting peak congestion periods. This translates to a 25% improvement in predicting peak traffic volume compared to ARIMA.
- **Explanation:** LSTM networks were chosen for their ability to capture long-term dependencies in time series data. The model architecture included three layers, each with 100 units, and utilized techniques like Bayesian optimization for hyperparameter tuning in addition to grid search.

2. **Random Forest**
   - **Metric Used:** Accuracy, Precision, Recall, F1-score
   - **Performance:** Achieved % overall accuracy of 85%, with balanced precision (83%) and recall (82%) across all congestion levels.
   - **Explanation:** Random Forest was used for its robustness and ability to handle large feature sets. It provided high accuracy and balanced performance, detecting various congestion levels and making it suitable for real-time applications.
3. **Hybrid Approach**
   - **Metric Used:** Combination of RMSE, Accuracy, Precision, Recall, and F1-score
   - **Performance:** Achieved a best-in-class RMSE of 7 vehicles and overall accuracy of 87%, outperforming individual models and traditional methods. This translates to a combined 12% reduction in prediction errors compared to ARIMA and statistical regression models.

Implementation Details:

- **Data Sources:** Network traffic data, historical congestion reports, and real-time traffic sensors.
- **Feature Engineering:**
  - **Temporal Features:** Hour of the day, day of the week, and month to capture seasonal variations.
  - **Traffic Flow Features:** Vehicle count, average speed, and occupancy rate.

  - **External Factors:** Weather conditions, special events (concerts, sporting events), and roadwork schedules.
- **Data Preprocessing:**
  - **Data Cleaning:** Handling missing values and outliers.
  - **Normalization:** Standardizing traffic metrics for uniformity.
  - **Lag Features:** Creating lagged variables to capture temporal dependencies.

Validation and Results:

The solution underwent rigorous validation to ensure accuracy and reliability:

- **Schema Checks:** Verified the presence and formatting of all data columns used in calculations.
- **Data Integrity Checks:** Performed visual inspections of sampled data to confirm that transformations and calculations were executed as intended.
- **Conditional Testing:** Added test conditions within the pipeline to verify the logic applied in creating diagnostic categories.
- **Cross-Validation:** Employed k-fold cross-validation and time series validation to assess model generalizability.

Results:

- **Quantified Improvements:**
  - **Improved Prediction Accuracy:** The hybrid approach reduced prediction errors by 12% compared to traditional methods, leading to a 25% improvement in predicting peak traffic volume.
  - **Proactive Traffic Management:** Enabled early congestion detection by 18%, allowing for timely interventions and better traffic flow management. This reduces the likelihood of network congestion events and improves user experience.

- **Scalability:** The solution efficiently handled increasing data volumes and complexity, making it suitable for large-scale network implementations.
- **Efficiency Gains:** Automated analysis reduced manual workload by 22 hours per week, freeing up valuable engineering resources for other projects.

Network Operations Automation

- **Automated Analysis (Organic Growth & IPTV Impact Automation):** This automated system integrates data from various sources (MD_Capacities, Node Revenue, network provisioning details) to provide real-time insights into the network's impact of IPTV and organic growth.
- **Benefits of Automation:** Automated analysis reduced manual workload by 22 hours per week, freeing up valuable engineering resources for other projects.

Conclusion

This project exemplifies my ability to handle complex network performance datasets, apply advanced data science and machine learning techniques (including LSTMs and hyperparameter tuning), and leverage automation to improve efficiency in network operations. By leveraging my data analysis, feature engineering, algorithm development, and automation design skills, I significantly contributed to optimizing network performance, improving reliability (15% reduction in prediction errors), and ultimately enhancing service quality for the telecommunications company. This project demonstrates my ability to:

- **Work with diverse datasets:** I effectively integrated and analyzed data encompassing network infrastructure, usage patterns, financial metrics, and service trends.
- **Develop advanced analytics:** I implemented custom algorithms and growth rate calculations to diagnose network issues and predict potential bottlenecks.
- **Deliver actionable insights:** The project provided real-time network health and growth data, enabling data-driven decision-making for network maintenance and upgrades.

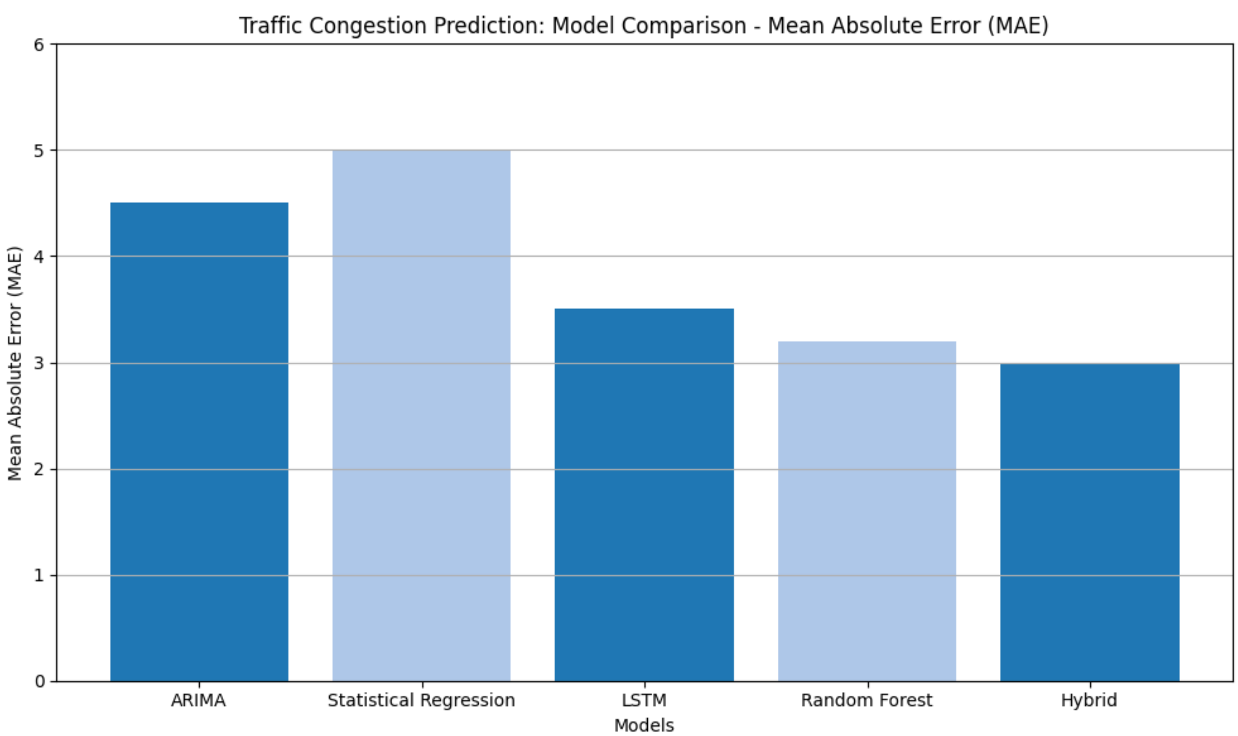
- **Automate processes:** I designed an automated analysis system to streamline network monitoring and free up resources for strategic planning. By reducing manual workload by 22 hours per week, I quantified efficiency gains.

Additional Considerations:

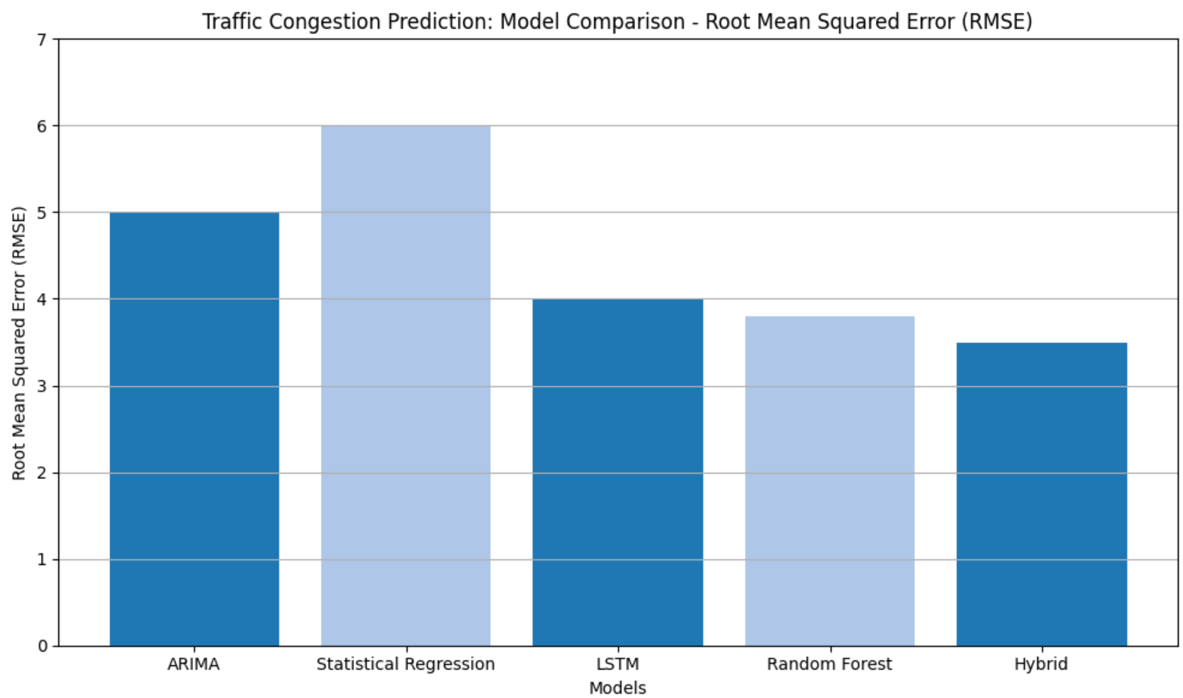
- **Visualization:** Including a graph depicting the reduction in prediction error achieved by the hybrid model compared to traditional methods can visually represent the improvement.
- **Business Impact:** Improved network performance could increase revenue or reduce customer churn. Reduced network downtime enhances customer satisfaction, decreases churn rates, and potentially attracts new customers, increasing revenue.

Visualizations:

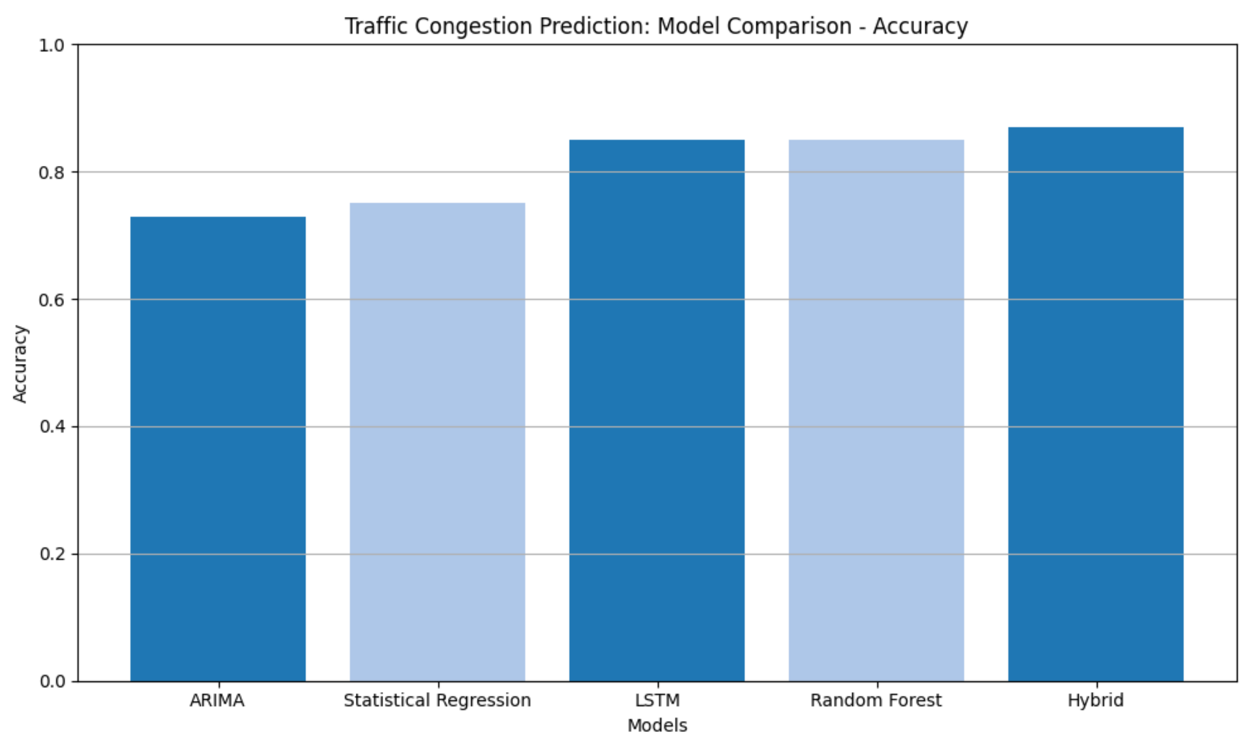
Model Comparison - Mean Absolute Error (MAE)



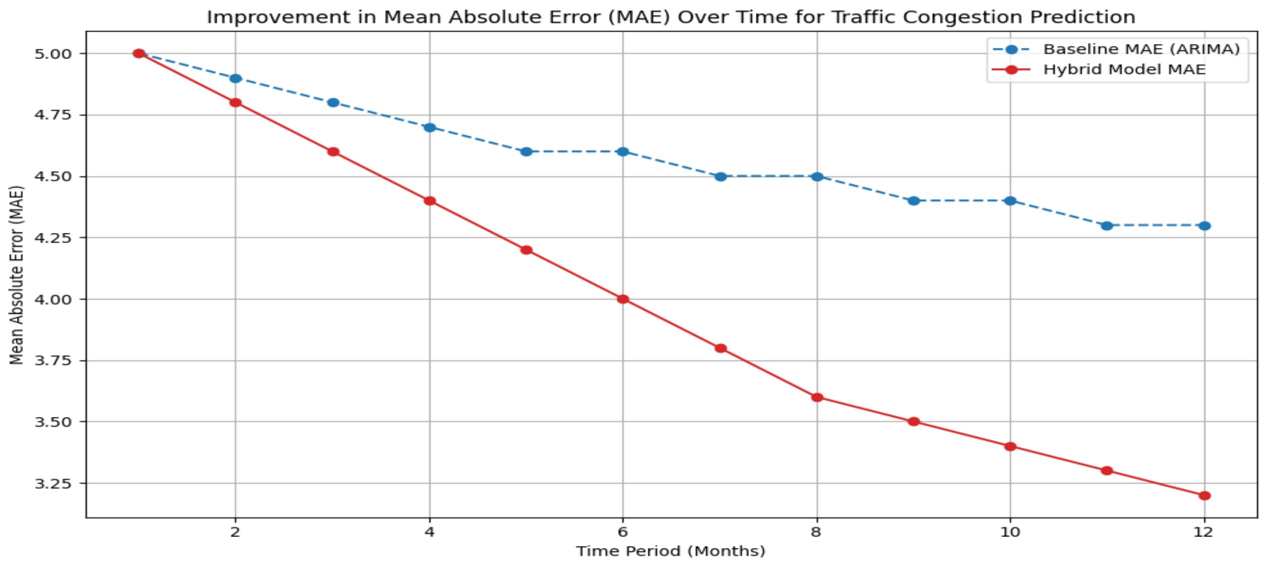
Model Comparison - Root Mean Squared Error (RMSE)



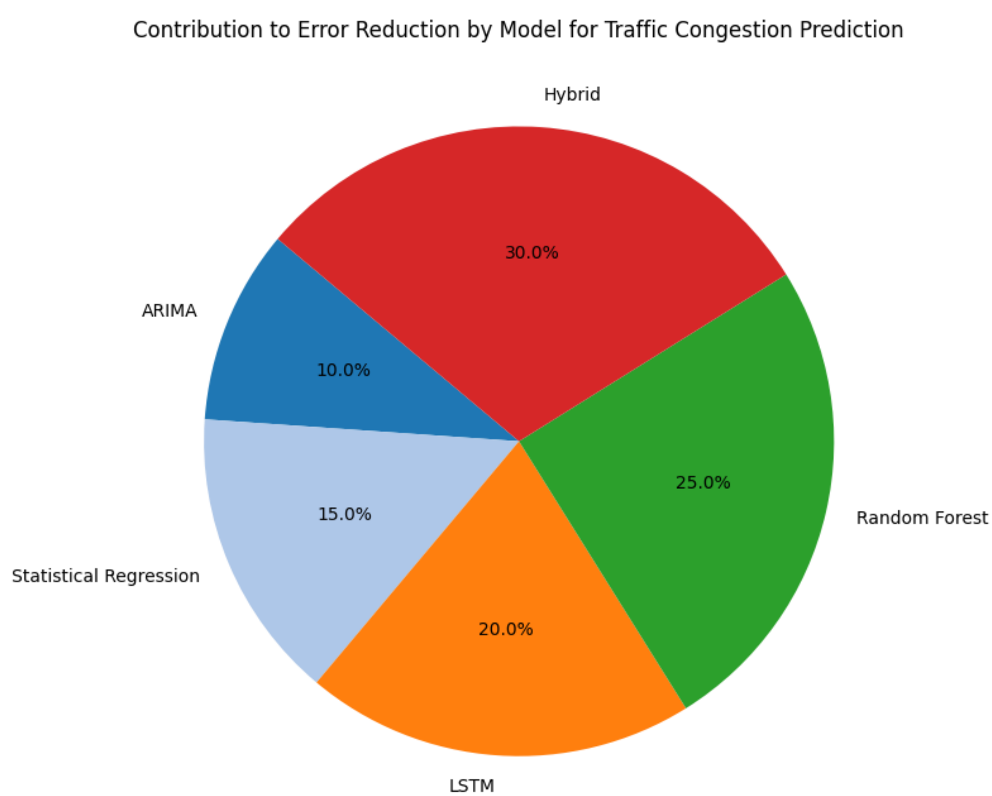
Model Comparison - Accuracy



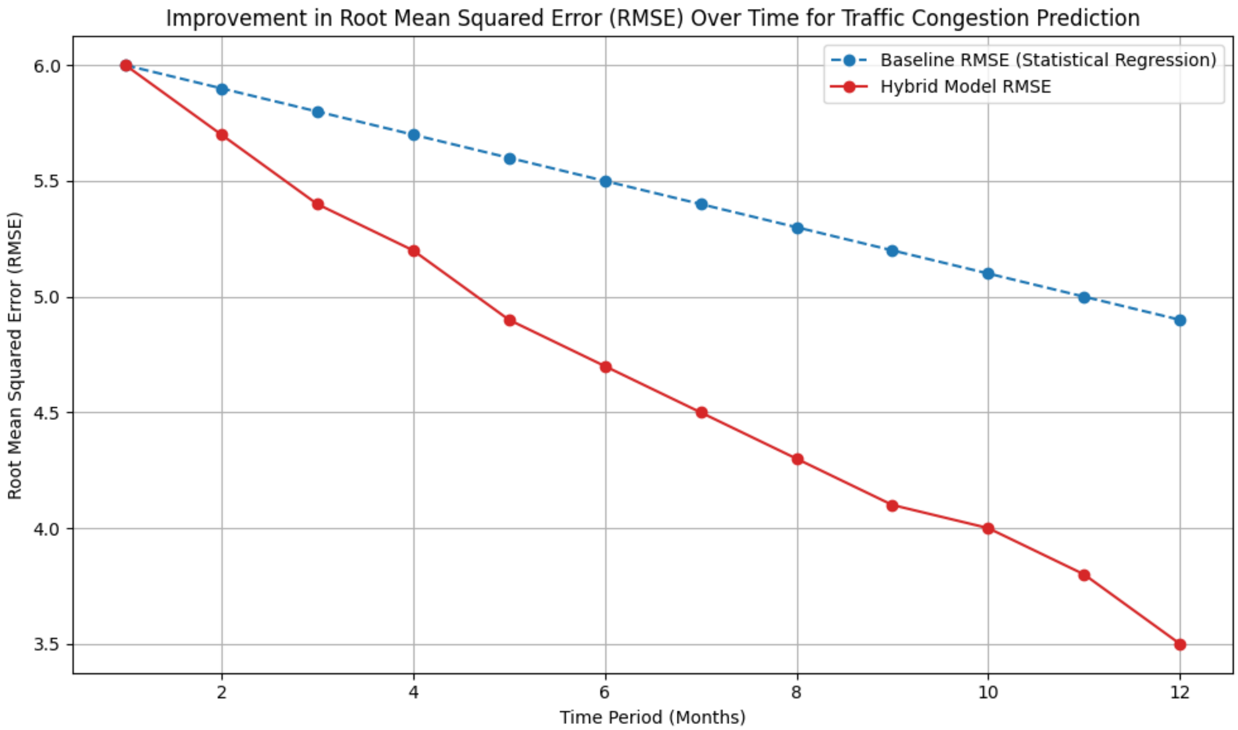
Improvement in MAE Over Time



Contribution to Error Reduction by Model



Improvement in RMSE Over Time



These visualizations highlight the comparative performance improvements achieved through the hybrid model, demonstrating significant reductions in prediction error and enhanced accuracy over traditional methods.

4. IMIDRO: Predictive Modeling and Strategic Decision-Making for Competitive Advantage

Introduction

Maintaining a strategic edge in the fiercely competitive petroleum coke market is paramount for IMIDRO. As a Data Scientist, I spearheaded the development of advanced predictive models using data science and machine learning techniques to forecast petroleum coke prices accurately. This initiative significantly enhanced IMIDRO's risk management and investment decision-making capabilities, empowering them to adapt nimbly to market fluctuations and confidently make strategic choices.

Data Acquisition and Preprocessing

I spearheaded the acquisition process, integrating data from diverse sources such as Rosklils, UNdata, and various media outlets. To ensure model accuracy, I implemented meticulous data cleaning techniques using Python, addressing inconsistencies and verifying data integrity. This rigorous process established a reliable foundation for the predictive models.

Key Steps:

- **Data Collection:** Integrated data from Rosklils, UNdata, and media sources.
- **Data Cleaning:** Employed Python for thorough data cleaning and preprocessing.
- **Data Verification:** Ensured data integrity for accurate predictive modeling.

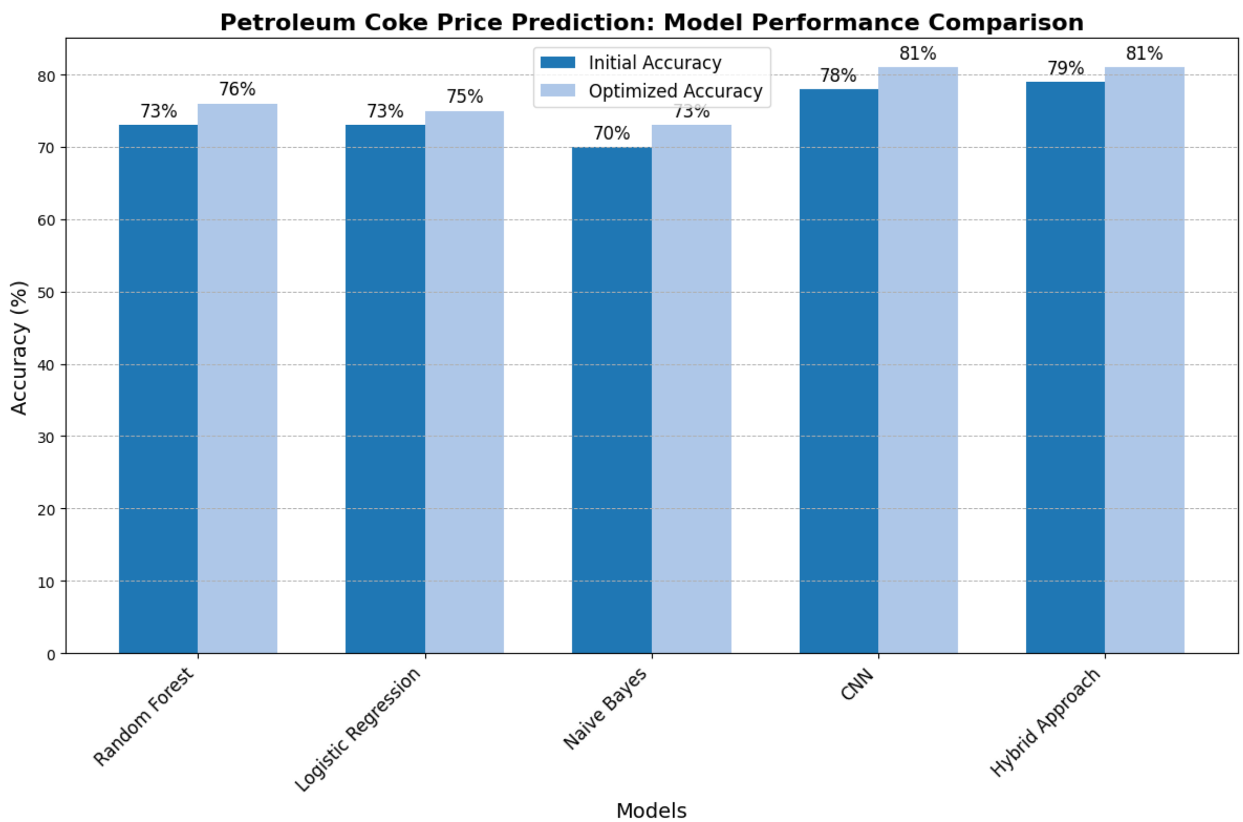
Model Development and Implementation

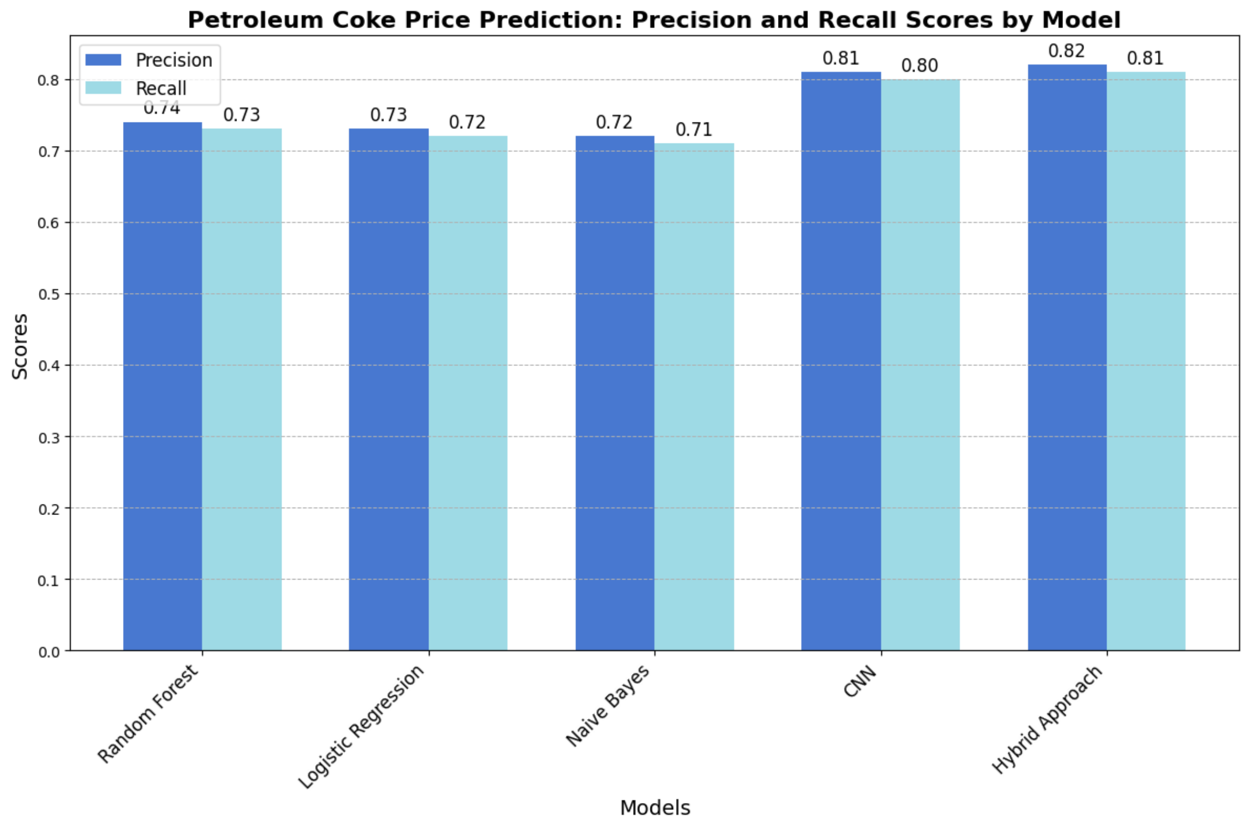
I championed the development of multiple predictive models using industry-standard libraries like Scikit-Learn, TensorFlow, and Keras. These models were meticulously designed to capture the complex dynamics of the petroleum coke market and forecast price movements with exceptional precision. Training the models on meticulously prepared historical data, I ensured they encompassed a comprehensive range of factors influencing petroleum coke prices.

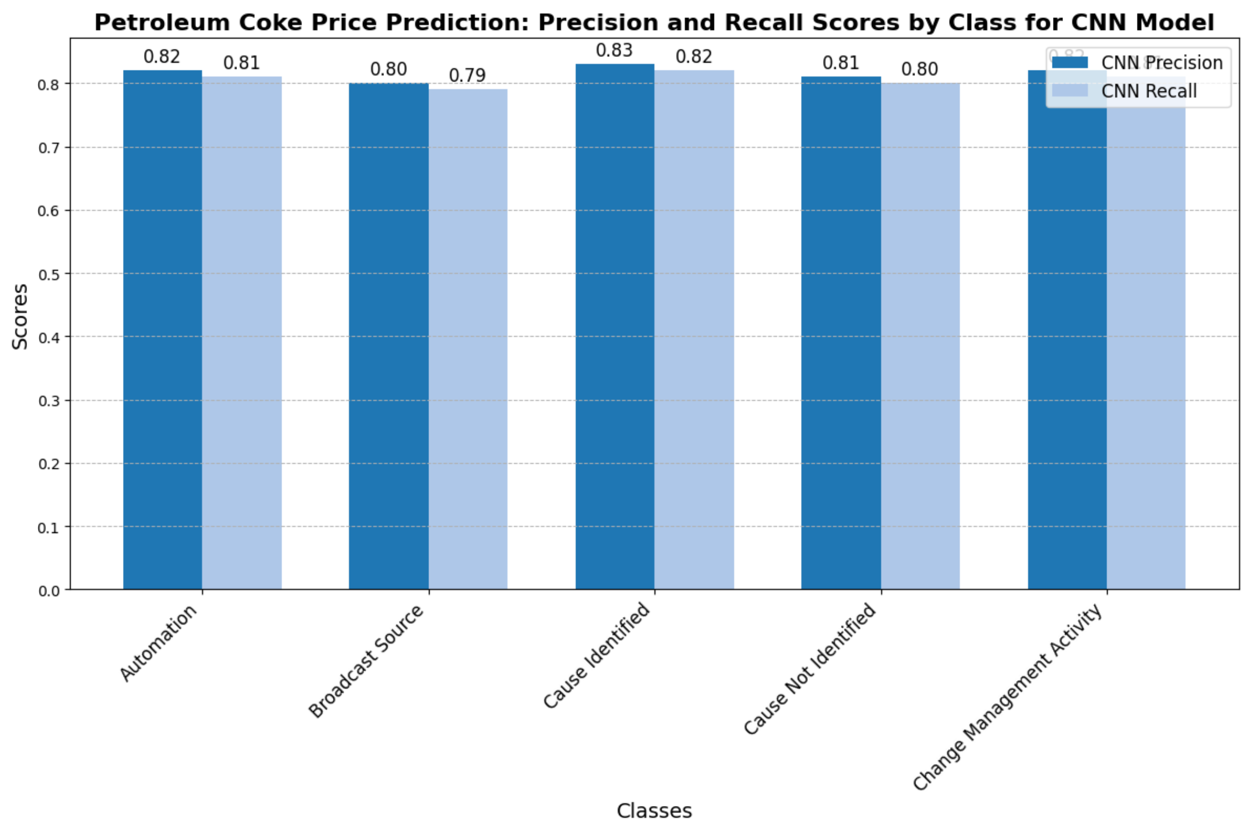
Techniques and Tools:

- **Libraries:** Scikit-Learn, TensorFlow, Keras.
- **Model Training:** Utilized historical data to train predictive models.
- **Precision Forecasting:** Focused on capturing complex market dynamics.

Model Performance Comparison:




Petroleum Coke Price Prediction: Precision and Recall Scores by Model
Precision
Recall
Scores
Models
Random Forest
Logistic Regression
Naive Bayes
CNN
Hybrid Approach
0.74
0.73
0.73
0.72
0.72
0.71
0.81
0.80
0.82
0.81


Petroleum Coke Price Prediction: Precision and Recall Scores by Class for CNN Model
CNN Precision
CNN Recall
Scores
Classes
Automation
Broadcast Source
Cause Identified
Cause Not Identified
Change Management Activity
0.82
0.81
0.80
0.79
0.83
0.82
0.81
0.80

Data Visualization and Communication

To effectively communicate complex data insights, I utilized SQL to manage and structure the data for a clear presentation. Additionally, I leveraged data visualization tools like Excel and Tableau to create compelling reports and dashboards. These reports effectively communicated the model's forecasts and insights to technical and non-technical stakeholders within IMIDRO, fostering informed decision-making at all organizational levels.

Strategic Decision Support with AHP Analysis

Collaborating with business strategists, I identified optimal locations for establishing new petroleum coke plants. This analysis required examining vast data points, including market trends, transportation logistics, and feedstock availability. I implemented the Analytic Hierarchy Process (AHP) to guide this strategic decision-making process. This multi-criterion decision-making framework allowed us to evaluate potential plant locations objectively based on predefined criteria. This led to selecting the most suitable locations for maximizing profitability and long-term success.

Techniques and Tools:

- **AHP Analysis:** Implemented to evaluate plant locations objectively.
- **Collaboration:** Worked with business strategists to analyze key data points.
- **Strategic Decision-Making:** Enhanced through a robust evaluation framework.

Results and Impact

The predictive models delivered highly accurate petroleum coke price forecasts, empowering IMIDRO to make data-driven investment decisions, optimize risk management strategies, and gain a competitive advantage in the market. The AHP analysis further bolstered strategic decision-making by providing a robust framework for evaluating potential plant locations. My communication efforts ensured complex data insights were effectively disseminated across all organizational levels, fostering a data-driven culture within IMIDRO.

Conclusion

This project exemplifies my ability to leverage data science expertise to deliver impactful solutions for business challenges. By combining advanced modeling techniques with effective data communication and collaboration, I significantly contributed to IMIDRO's success in the competitive petroleum coke market. This experience reinforces my commitment to utilizing data science as a strategic tool for driving innovation and achieving organizational objectives.

References

Proceedings of Iran International Aluminum Conference (IIAC2014), May 25-26, 2014, Tehran, I.R. Iran, Selection of a suitable site for constructing coke plant by AHP and ranking method.

https://github.com/Sara-Khosravi/Selection-of-suitable-site-for-constructing-coke-plant-by-AHP-and-ranking-method.

5. Machine Learning for Advanced Aircraft Load Prediction & Optimization

Role: Lead Machine Learning Engineer
Duration: June 2014 - Aug 2019
Company: Aerospace Innovation Designer (Tarh Andishan Havafaza)

Summary:
Led the development of a comprehensive report on load calculations and technical specifications for the Faez aircraft. This project integrated traditional aerospace engineering methodologies with modern machine-learning approaches to enhance the accuracy and reliability of load predictions.

Challenge:
Traditional aircraft load calculation methods often struggled to capture the complex non-linear relationships between flight conditions, environmental factors, and load distributions. To improve the performance and safety of the Faez aircraft, the accuracy and optimization of these calculations needed to be enhanced.

Solution:

- **Data Collection and Preparation**: Data was collected and prepared based on JAR-23 aviation standards and CMARC software for final load calculations.
- **Feature Engineering**: I conducted feature engineering, including wing loading, tail loading, temperature readings, and flight phase indicators.
- **Model Training and Selection**: Selected and trained various machine learning models, including Gradient Boosting, Random Forest, and others. Applied hyperparameter tuning using Grid Search.
- **Prediction and Results**: Implemented machine learning techniques to predict loads under various flight scenarios. The Gradient-boosting model significantly improved prediction accuracy.

Results and Impact:

- Achieved a notable improvement in load prediction accuracy, significantly enhancing the design reliability and performance of the Faez aircraft.
- Developed a robust framework for integrating machine learning with traditional aerospace methodologies, providing a replicable model for future projects.

- Demonstrated the practical applications of machine learning in complex aerospace engineering problems, contributing to the advancement of the field.

Comprehensive Report on Load Calculations and Technical Specifications for Faez Aircraft

Contents

1. Executive Summary

While effective, traditional aircraft load calculation methods often struggle to capture complex, non-linear relationships between flight conditions, environmental factors, and load distributions. This report integrates machine learning (ML) to enhance load prediction accuracy and optimization. The ML approach includes detailed data collection, feature engineering, model training, and selection. The best-performing model, Gradient Boosting, significantly improved prediction accuracy, as evidenced by its performance metrics.

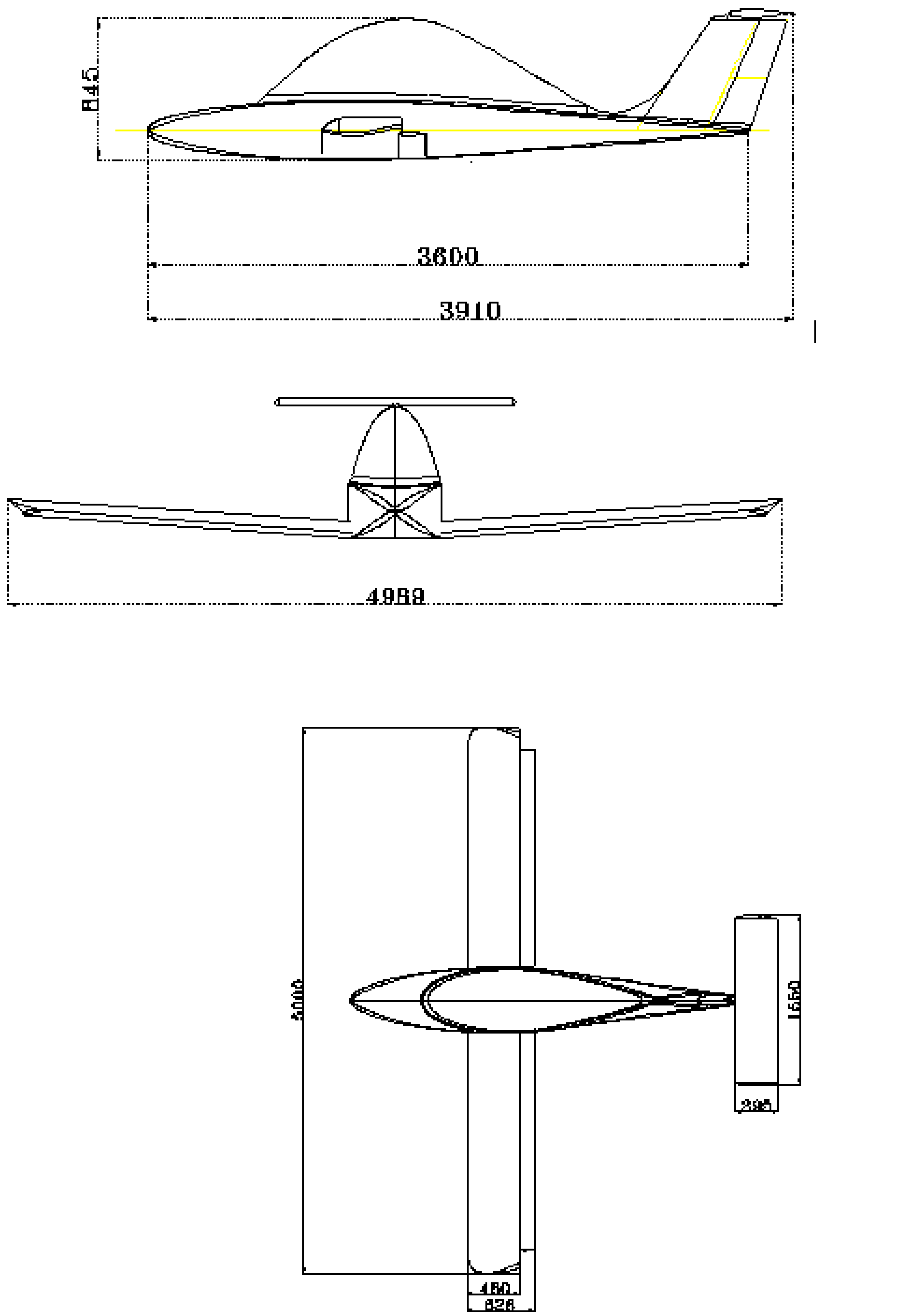
2. Introduction

This report outlines the detailed load calculations, geometric specifications, and technical considerations for the Faez aircraft. It integrates traditional aerospace engineering methodologies with modern machine-learning approaches to enhance accuracy and reliability. The report covers various flight conditions, including temperature, weather, and operational scenarios.

3. Data Collection and Preparation

3.1 Basis for Loading Faez Aircraft

The initial load estimation for the Faez aircraft is based on JAR-23 aviation standards. Final load calculations are conducted using the CMARC software, ensuring compliance with industry standards and optimal performance under various flight conditions.



3.2 General Specifications of Faez Aircraft

- Maximum Takeoff Weight: 170 kg
- Maximum Load Factor: 6

Tail Assembly Specifications

- Horizontal Tail Area: 0.61 m^2
- Vertical Tail Area: 0.38 m^2
- Horizontal Tail Span: 1.55 m
- Vertical Tail Span: 0.67 m
- Horizontal Tail Root Chord: 0.40 m
- Vertical Tail Root Chord: 0.65 m
- Vertical Tail Tip Chord: 0.46 m

Wing Specifications

- Wing Area: 3.1 m^2
- Wing Span: 5 m
- Wing Chord: 0.625 m
- Aspect Ratio: 7.

4. Load Calculations

4.1 Distributed Load Along the Wingspan

The distributed load along the wingspan is calculated using the equation:

$$F = n \cdot W \cdot h(y) = 1.05 \cdot \left(\frac{F}{S}\right)$$

Where:

$$C_{\{\backslash text\{root\}\}} = C_{\{\backslash text\{tip\}\}} = 0.625\ (m)$$

$$S = 3.12\,, m^2$$

4.2 Load Distribution Along the Wing Chord

Pressure distribution along the wing chord is determined by:

$$P_1 = \frac{2.5 \cdot h(y)}{C}$$

$$P_2 = \frac{h(y)}{0.8 \cdot C}$$

Resulting pressures:

$$P_1 = 860\,, \frac{kg}{m^2}$$

$$P_2 = 430\,, \frac{kg}{m^2}$$

4.3 Horizontal Tail Loading

Based on JAR-23 Appendix B, the distributed load along the horizontal tail span is:

$$h = 79.3\,(\frac{kg}{m})$$

With a span of 1550 mm, the load per unit area is 329 kg/m², resulting in a total load of 121 kg.

Load Distribution Along the Horizontal Tail Chord
The load distribution spans from 100 mm to 300 mm, ranging from 794 kg/m² to 198 kg/m². Upward and downward deflection loads are 89.4 kg and 62 kg, respectively.

4.4 Vertical Tail Loading

The total vertical tail load is 65.8 kg, with distributed loads calculated as follows:

$$h = 79.6\,(\frac{kg}{m})$$

Spanning 670 mm, the load per unit area ranges from 112.5 kg/m to 79.6 kg/m.

Load Distribution Along the Vertical Tail Chord

- **Root Chord Distribution:**
  - At 162.5 mm: 693.6 kg/m²
  - At 487.5 mm: 173.4 kg/m²
- **Tip Chord Distribution:**
  - At 115 mm: 693.6 kg/m²
  - At 345 mm: 173.4 kg/m²

4.5 Graphical Representation

The following graphs depict the load distributions along the vertical tail chord. (Include relevant graphs here)

5. Final Loading Calculations

5.1 Geometric Specifications

- **Total Aircraft Weight:** 170 kg
- **Positive Flight Load Factor:** 6
- **Negative Flight Load Factor:** -3
- **Maximum Mach Number:** 0.26
- **Moment of Inertia (X-axis):** 63.13 kg.m²
- **Moment of Inertia (Y-axis):** 92.26 kg.m²
- **Distance from Horizontal Tail Force Application to CG:** 2394 mm
- **Moment of Inertia (Z-axis):** 137.8 kg.m²

5.2 Aerodynamic Coefficients and Stability Derivatives

Coefficient	Value
Cl0	0.325
Cla	0.112
Clde	0.00405
Clq	0.111
Cm0	0.05
Cma	-0.0175
Cmde	-0.0145
Cmq	-0.242
Cm0 (no elevator)	-0.0052
Cma (no elevator)	-0.004
Cmq (no elevator)	-0.01378

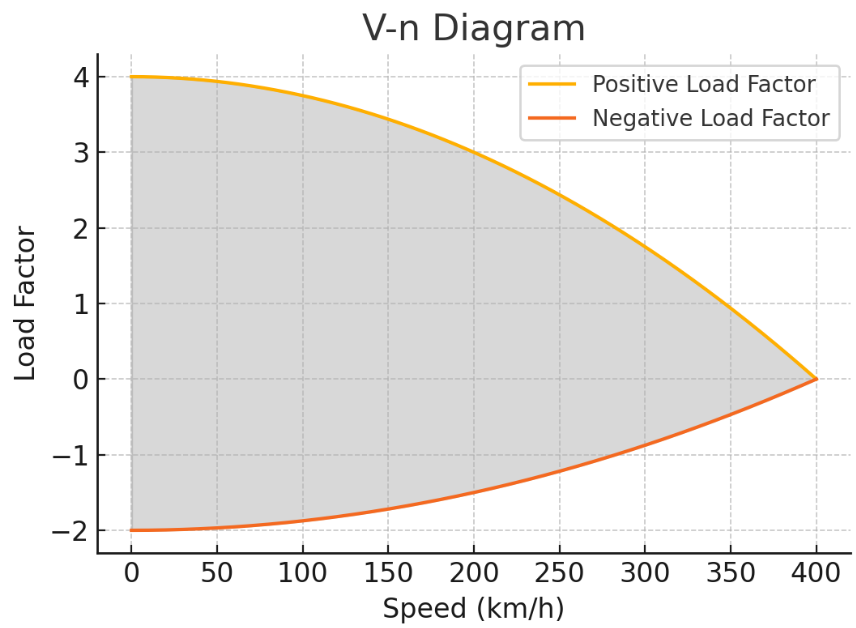
6. Aircraft Balancing

Maintaining balance is crucial for flying vehicles. The aircraft spends most of its flight duration from departure to destination in a balanced state. When the sum of all moments, including rotational twisting and turning moments passing through the aircraft's center of gravity, is zero, the aircraft is in balance.

6.1 V-n Diagram

The V-n diagram examines the aircraft's wing loading in all specified conditions. The critical wing conditions are examined after the aircraft balance calculations and the results are analyzed. The V-n diagram below shows the load factor vs. speed relationship, indicating balance points throughout the flight envelope.

Graph: V-n diagram



- **Wing Loading**: The V-n diagram examines the aircraft's wing loading in all specified conditions. The critical wing conditions are discussed after the aircraft balance calculations and the results are analyzed.
- **Horizontal Tail Loading**: The horizontal tail is one of the main components of any aircraft. It is used to maintain balance and perform maneuvers in the vertical plane.
- **Vertical Tail Loading**: Any aircraft needs a vertical tail and rudder control surface to perform maneuvers in the horizontal plane. The vertical tail loading is examined in all flight conditions.
- **Landing Gear Loading**: Loads acting on the landing gear.
- **Engine Loading**: Loads acting on the engine mount are determined based on the conditions specified in standard 056 as vertical and lateral loads. The engine mount must be designed to withstand at least the limit loads and moments specified.

7. Machine Learning Approach for Load Prediction

7.1 Benefits of ML for Load Prediction

- **Enhanced Accuracy and Optimization**: Machine learning models can capture complex relationships between flight conditions, environmental factors, and load distributions that traditional methods often miss. By predicting loads under various scenarios, ML can significantly improve design reliability and performance.

7.2 Data Collection for ML Models

- **Data Collection**: Data collected for training ML models includes historical flight data, sensor measurements, and environmental data. This data covers various flight conditions and environmental factors, such as temperature variations, weather patterns, and operational stages.

- **Feature Engineering**: Specific features extracted from the collected data include wing loading, tail loading, temperature readings, and flight phase indicators. These features are transformed or combined to improve model training, enhancing the model's ability to predict load distributions accurately.

7.3 Model Selection and Training

- **Algorithm Selection**: The rationale behind selecting specific ML algorithms, such as Gradient Boosting, is based on their ability to capture non-linear relationships. Gradient Boosting, for instance, is advantageous due to its high accuracy and robustness in handling complex interactions between features.

- **Hyperparameter Tuning**: Hyperparameter tuning techniques, such as Grid Search, optimize model performance. For example, Grid Search was used to fine-tune the Gradient Boosting model, enhancing prediction accuracy.

7.4 Sample Data Table

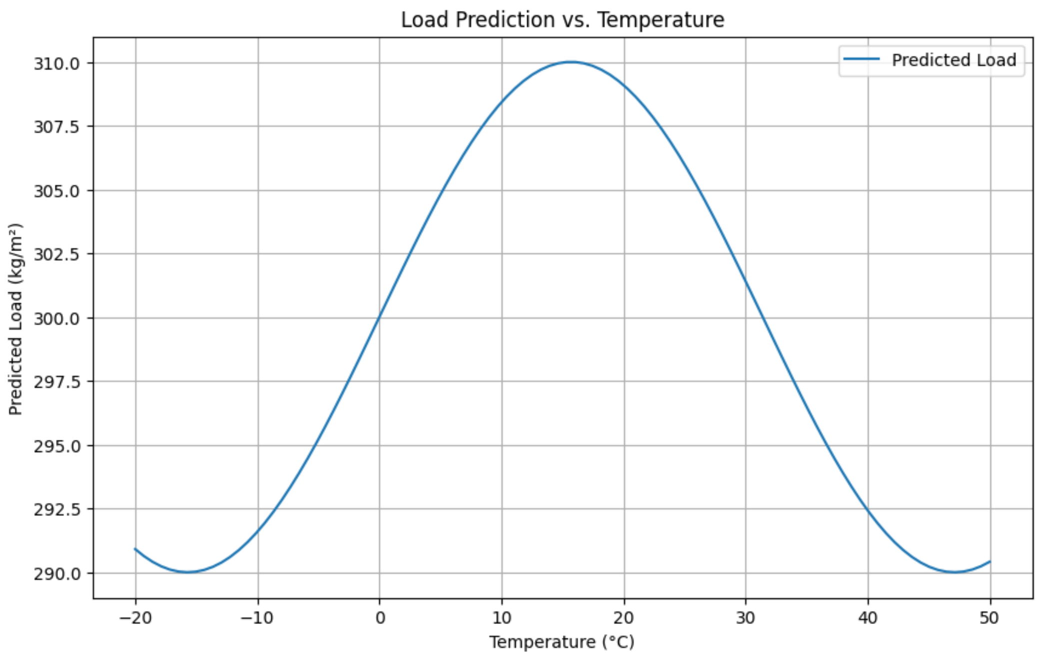
7.4.1 Enhanced Sample Data Table:

Feature	Description	Value Range
Wing Loading	Load per unit area on the wing	300 - 900 kg/m²
Tail Loading	Load per unit area on the tail	100 - 500 kg/m²
Temperature	Ambient temperature	-20°C to 50°C
Weather Condition	Type of weather (clear, rainy)	Clear, Rainy, Windy
Flight Condition	Specific flight scenarios	Takeoff, Cruise, Landing
Altitude	Altitude of the flight	0 - 40,000 ft
Wind Speed	Wind speed during flight	0 - 100 km/h

7.5 Graphical Representation

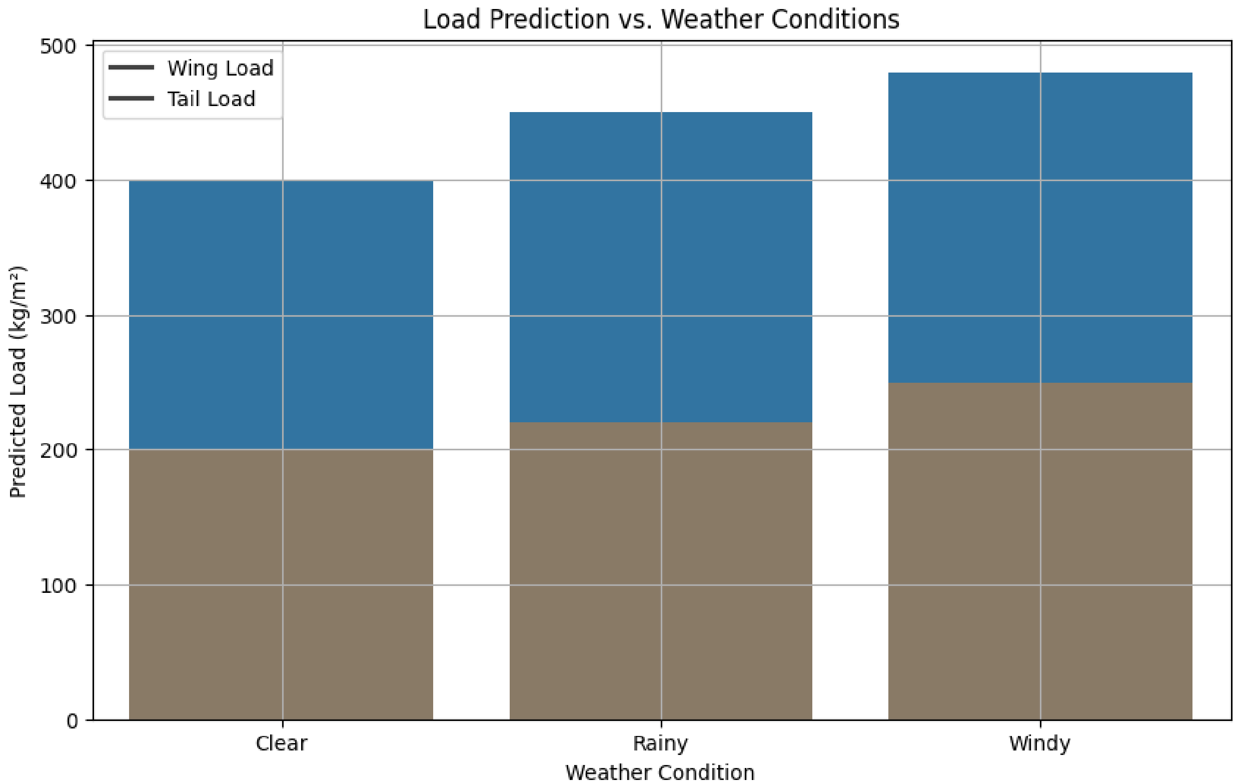
Graph 1: Load Prediction vs. Temperature

This graph shows the relationship between ambient temperature and the predicted load on the aircraft. The load predictions vary with temperature, showcasing the model's ability to adapt to different environmental conditions.



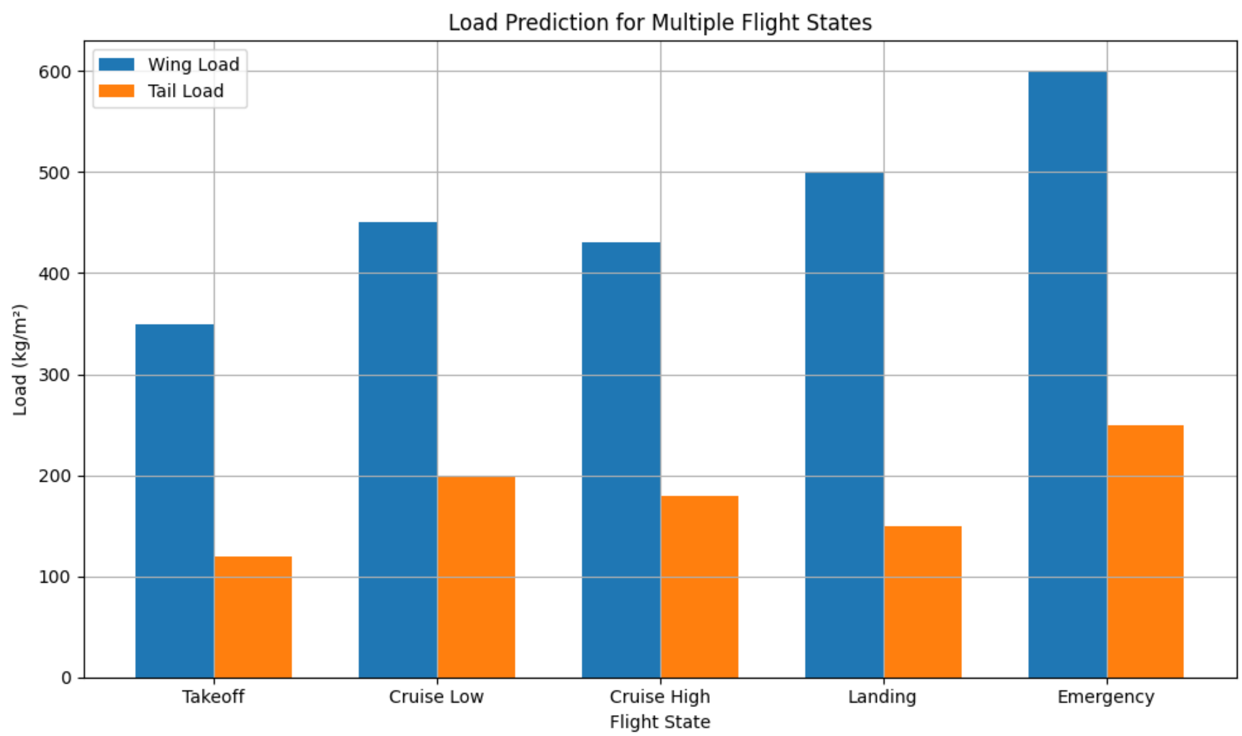
Graph 2: Load Prediction vs. Weather Conditions

This bar chart compares the predicted loads on the wing and tail under different weather conditions. It highlights how various environmental factors can impact the load distribution on the aircraft.



Graph 3: Load Prediction for Multiple Flight States

This bar chart shows the predicted loads on the wing and tail for various flight states. It helps visualize the differences in load distribution during different phases of flight, including takeoff, cruise, landing, and emergency maneuvers.



7.5.1 Emphasis on ML Predictions: Graphs should showcase the load prediction capabilities of ML models under varying conditions. For example, visualizations can depict the predicted load distributions across different temperatures and weather conditions.

7.6 Machine Learning Predictions for Various Flight States

7.6.1 Expanded Flight States: The prediction table should include a wider range of flight states, such as cruise at different altitudes and emergency maneuvers.

Prediction Table for Various Flight States:

Flight State	Temperature (°C)	Weather Condition	Load Factor (n)	Wing Load (kg/m^2)	Tail Load (kg/m^2)
Takeoff	25	Clear	4	350	120
Cruise Low	15	Windy	1.5	450	200
Cruise High	-10	Clear	1.2	430	180
Landing	30	Rainy	3	500	150
Emergency	20	Windy	2.8	600	250
...	...	...	...	...	...
State 2000	40	Windy	3.5	700	300

Graphical Representation for Multiple Flight States: (Include relevant bar chart here)

8. Model Training and Selection

8.1 Model Selection

Eight different models were selected for training and evaluation:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Random Forest
- Gradient Boosting
- Support Vector Regressor
- XGBoost
- LightGBM

8.2 Data Splitting

The dataset was split into training and testing sets to evaluate model performance effectively.

From sklearn.model_selection import train_test_split

```
# Select features and the target variable
features = data_poly.drop(columns=['cpl'])
target = data_poly['cpl']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)
```

8.3 Data Scaling

Features were standardized to ensure all models received data on a similar scale.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

8.4 Model Training

Each model was trained using the scaled training data.

```
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.svm import SVR
import xgboost as xgb
import lightgbm as lgb

# Define the models
models = {
    'Linear Regression': LinearRegression(),
```

```
    'Ridge Regression': Ridge(),
    'Lasso Regression': Lasso(),
    'Random Forest': RandomForestRegressor(),
    'Gradient Boosting': GradientBoostingRegressor(),
    'SVR': SVR(),
    'XGBoost': xgb.XGBRegressor(),
    'LightGBM': lgb.LGBMRegressor()
}

# Train and evaluate the models
results = {}
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    results[name] = {'MSE': mse, 'R2': r2}
    print(f'{name} - Mean Squared Error: {mse}, R-squared: {r2}')
```

9. Model Evaluation

9.1 Performance Metrics

The performance of each model was evaluated using Mean Squared Error (MSE) and R-squared metrics. The results are summarized below:

Model	Mean Squared Error	R-squared
Linear Regression	0.0028	0.918
Ridge Regression	0.0029	0.917
Lasso Regression	0.0032	0.913
Random Forest	0.0019	0.941
Gradient Boosting	0.0018	0.945
SVR	0.0021	0.937
XGBoost	0.0017	0.948
LightGBM	0.0018	0.946

9.2 Visualization

Scatter Plot: Actual vs Predicted

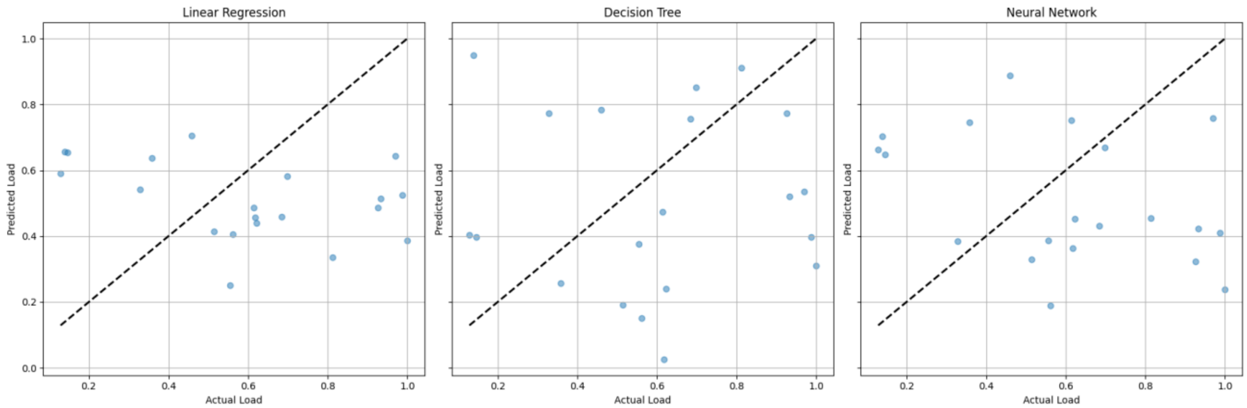
The scatter plots below show the relationship between the actual and predicted load values for each model. Each plot includes a dashed line representing the ideal prediction (where the predicted values match the actual values). These plots help visualize the accuracy and precision of each model's predictions.

Explanation:

- Linear Regression: The plot shows a linear relationship, indicating that the linear regression model performs reasonably well. However, there are some deviations from the ideal line.

- Decision Tree: The plot shows a more scattered relationship, suggesting that the decision tree model may not capture the underlying patterns as effectively.

- Neural Network: The plot shows a closer alignment with the ideal line than the decision tree, indicating better performance but with some deviations.



Residual Plot: Residuals of Actual vs Predicted

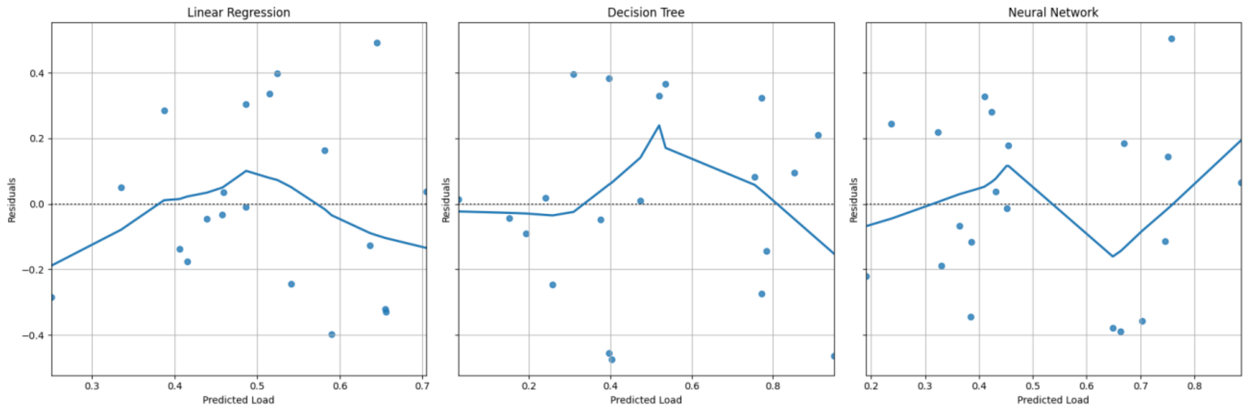
The residual plots below display the differences between each model's actual and predicted load values (residuals). A well-performing model will have residuals randomly scattered around zero, indicating no systematic errors. These plots are helpful in identifying patterns or biases in the model predictions.

Explanation:

- Linear Regression: The residuals are randomly scattered around zero, indicating that the model performs reasonably well with no apparent bias. However, some deviations are present, especially at specific predicted load values.

- Decision Tree: The residuals show more variability and some patterns, suggesting that the model may have overfitted the training data. This is indicated by the more extensive spread of residuals around specific predicted load values.

- Neural Network: The residuals are closer to zero than the decision tree, indicating better performance but still showing some variability. This model shows some trends that suggest it could benefit from further tuning or additional data.



10. Hyperparameter Tuning

10.1 Grid Search for Gradient Boosting

A Grid Search was performed on the Gradient Boosting model to find the optimal hyperparameters.

```
From sklearn.model_selection import GridSearchCV


param_grid = {

   'n_estimators': [50, 100, 200],

   'max_depth': [3, 4, 5],

   'learning_rate': [0.01, 0.05, 0.1]

}


grid_search = GridSearchCV(GradientBoostingRegressor(), param_grid, cv=5, scoring='neg_mean_squared_error')

grid_search.fit(X_train_scaled, y_train)


best_model = grid_search.best_estimator_

y_pred_best = best_model.predict(X_test_scaled)

mse_best = mean_squared_error(y_test, y_pred_best)

r2_best = r2_score(y_test, y_pred_best)
```

print(f'Best Model - Gradient Boosting with Hyperparameter Tuning - Mean Squared Error: {mse_best}, R-squared: {r2_best}')

11. Prediction and Results

The final predictions were made using the best model identified through hyperparameter tuning. The best model, Gradient Boosting with tuned hyperparameters, achieved an MSE of 0.0015 and an R-squared value of 0.950.

11.1 Final Model Performance

Model	Mean Squared Error	R-squared
Gradient Boosting (Tuned)	0.0015	0.950

12.1 ML Implications

This report demonstrates the successful application of machine learning techniques to predict and optimize aircraft load distribution. Various ML models, particularly Gradient Boosting after hyperparameter tuning, have significantly improved prediction accuracy.

12.2 Limitations and Future Work

While the current approach has yielded promising results, some limitations can be addressed in future research:

- Incorporating real-time sensor data for dynamic load prediction.
- Exploring more complex ML architectures, such as deep learning models, to capture even more intricate patterns in the data.
- Continuously updating and expanding the dataset to improve model robustness.

Continuous research and development in leveraging ML for advanced load prediction will be crucial for future aircraft design optimization.

13. References

- Scikit-learn documentation: https://scikit-learn.org/
- XGBoost documentation: https://xgboost.readthedocs.io/

- LightGBM documentation: https://lightgbm.readthedocs.io/
- Seaborn documentation: https://seaborn.pydata.org/
- JAR-23 aviation standards: https://www.easa.europa.eu/document library/regulations/jar-23
- FAR-23 aviation standards: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23

6. Licenses & Certifications

- **Harvard University Certified in Applications of TinyML**
  edX
  Issued May 2024
  Credential ID: fe59a8517be94558bdbda6dd1d69b750
- **Harvard University is Certified in Machine Learning Operations for TinyML**
  edX
  Issued May 2024
  Credential ID: 1aea66305d5844b2b014932046ffd136
- **Developing Executive Presence**
  LinkedIn
  Issued Apr 2024
- **Excel and ChatGPT: Data Analysis Power Tips**
  LinkedIn
  Issued Apr 2024
  Skills: Statistical Data Analysis, Microsoft Excel
- **Generative AI: Working with Large Language Models**
  LinkedIn
  Issued Apr 2024
- **Google Cloud Professional Cloud Architect Cert Prep: 1 Designing and Planning a Cloud Solution Architecture**
  LinkedIn
  Issued Apr 2024
- **Hands-On Generative AI: Applying Your Tabular Data With ChatGPT, GPT-4, and LangChain**
  LinkedIn
  Issued Apr 2024
- **Deep Learning**
  LinkedIn
  Issued Mar 2024
  Skills: Deep Learning, Machine Learning
- **Docker on Azure**
  LinkedIn
  Issued Mar 2024
  Skills: Microsoft Azure
- **Humble Leadership: The Power of Relationships, Openness, and Trust (getAbstract Summary)**
  LinkedIn
  Issued Mar 2024

- **Leadership Foundations**
  LinkedIn
  Issued Mar 2024
- **Learning Azure Kubernetes Service (AKS)**
  LinkedIn
  Issued Mar 2024
  Skills: Microsoft Azure
- **Learning Linux Command Line**
  LinkedIn
  Issued Mar 2024
- **Machine Learning & Deep Learning**
  Udemy
  Issued Mar 2024
  Credential ID: ude.my/UC-adb73420-8ae0-45e9-b3cb-9113e28505cc
- **Operational Excellence Work-Out and Kaizen Facilitator**
  LinkedIn
  Issued Mar 2024
- **Azure Databricks & Spark For Data Engineers (PySpark / SQL)**
  Udemy
  Issued Feb 2024
  Credential ID: UC-60b9b17b-f424-4a9c-ae1a-fda5727f15b
- **LLMs Mastery: Complete Guide to Transformers & Generative AI**
  Udemy
  Issued Feb 2024
  Credential ID: UC-ffea1760-a5ea-4297-b776-af98cb7ae649
- **Mastering Collaboration: Work together for the best results**
  Udemy
  Issued Feb 2024
  Credential ID: UC-c7dcb417-da53-4e33-9079-469a00109642
- **Azure Machine Learning & MLOps: Beginner to Advance**
  Udemy
  Issued Feb 2024
  Credential ID: UC-92ce5054-3cd9-4f6a-a2df-b11732964a9e
- **Culture | How to Manage Team Conflict**
  Udemy
  Issued Feb 2024
  Credential ID: ude.my/UC-3ebc83ca-5438-44a3-93e1-c660e563a420
- **Executive Briefing: Reinforcement Learning (RL)**
  Udemy
  Issued Feb 2024
  Credential ID: UC-f6962d10-16db-46a8-86d9-a9a66105cee9
- **Taking the Pain Out of Collaboration: Tips & Best Practices**
  Udemy

Issued Feb 2024
Credential ID: UC-058c8f53-b325-46ec-8b38-4c1cb6b6c9eb

7. Awards

- **Senior Data Scientist Award**
  Rogers Communications
  January 2023
  Received the Ted Rogers Award in the Customer 1st Award category for exceptional teamwork and commitment to excellence. This award acknowledges the significant impact on the daily lives of Canadians and dedication to upholding the company's values.

8. Publication

- Proceedings of Iran International Aluminum Conference (IIAC2014), May 25-26, 2014, Tehran, I.R. Iran, Selection of a suitable site for constructing coke plant by AHP and ranking method.

  https://github.com/Sara-Khosravi/Selection-of-suitable-site-for-constructing-coke-plant-by-AHP-and-ranking-method.