

|  |  |  |
| --- | --- | --- |
| Group #14 |  | 01 May 2024 |

Dheeraj Mekala**1**, Sandeep karanam**2**, Amit Kapoor**3**

**Enhancing Aviation Safety: Anomaly Detection in Final Approaches Using Autoencoders**



**Professor**

**Dr. Timothy Smith**

**Team Members**

**Dheeraj Mekala (U10405353)**

**Sandeep Karanam (U94742075)**

**Amit Kapoor (U14988780)**

**Contents**

[**Introduction** 3](#_Toc165582310)

[**Problem Identification: Ensuring Safety During Aircraft Final Approach 3**](#_Toc165582311)

[**The Current Approach to Managing Safety** 3](#_Toc165582312)

[**Significance of Improving Current Systems** 3](#_Toc165582313)

[**The Need for Advanced Predictive Systems 3**](#_Toc165582314)

[**Overview of the DASHlink Dataset** 3](#_Toc165582315)

[**Objective of Our Proposed Solution 4**](#_Toc165582316)

[**Value Proposition of the New System** 4](#_Toc165582317)

[**Utilizing Autoencoders for Anomaly Detection** 4](#_Toc165582318)

[**How Autoencoders Work** 4](#_Toc165582319)

[**Benefits of Autoencoders in Detecting Flight Anomalies** 5](#_Toc165582320)

[**Data Preprocessing** 5](#_Toc165582321)

[**Model Design** 7](#_Toc165582322)

[**Model Implementation and Training** 8](#_Toc165582323)

[**Error Visualization and Threshold Determination** 9](#_Toc165582324)

[**Model Testing and Performance Evaluation** 11](#_Toc165582325)

[**Model Performance Overview** 13](#_Toc165582326)

[**Analysis of Results** 14](#_Toc165582327)

[**Conclusion** 14](#_Toc165582328)

[**Key Aspects of the Project:** 14](#_Toc165582329)

## Introduction

## **Problem Identification: Ensuring Safety During Aircraft Final Approach**

Final approach is a critical phase in aviation where an aircraft prepares to land. During this phase, even minor deviations in speed or mechanical settings such as flap positions can lead to severe consequences, including accidents. Historically, several incidents have occurred during this phase due to small errors that had fatal outcomes, underlining the importance of enhancing monitoring and predictive mechanisms in aviation.

### **The Current Approach to Managing Safety**

Currently, pilots rely heavily on their judgment and a set of tools designed to assist in navigation and landing. However, these tools are not infallible; they do not predict unexpected changes or errors in aircraft operations. Moreover, while data from flights is reviewed post-flight to analyze any deviations or errors, there is a significant gap in utilizing this data in real-time, which means potentially rectifiable issues during flight are not addressed promptly.

### **Significance of Improving Current Systems**

Improving the systems to better utilize data for real-time predictive analysis could dramatically enhance safety by:

* Preventing accidents by detecting anomalies early.
* Boosting passenger confidence due to increased safety.
* Reducing costs related to delays, repairs, and insurance linked to aviation accidents.

## **The Need for Advanced Predictive Systems**

The aviation industry has witnessed several accidents where advanced predictive systems could have made a difference. For instance:

* **Asiana Airlines Flight 214 (2013)**: A crash due to the aircraft being below its target landing speed, resulting from pilot error and a misunderstanding of the automated systems.
* **AirAsia Flight QZ8501 (2014)**: Crashed due to a stall in bad weather, linked to pilot response to a rudder system malfunction.
* **Flydubai Flight FZ981 (2016)**: Crashed due to spatial orientation loss during a challenging landing in strong winds.
* **Lion Air Flight JT610 (2018)** and **Ethiopian Airlines Flight ET302 (2019)**: Both involved issues with the Maneuvering Characteristics Augmentation System (MCAS), where predictive systems could have detected and possibly prevented the malfunctions.

### **Overview of the DASHlink Dataset**

The DASHlink dataset contains comprehensive flight data, which includes:

* **Number of Instances**: About 99,000 flights.
* **Data Collection Duration**: Data from each flight spans 160 seconds of flight, focusing on the final approach.
* **Variables Collected**: 20 different variables, each offering insights into various aspects of the flight dynamics, such as aileron positions, airspeed, altitude, and flaps positions.

## **Objective of Our Proposed Solution**

Our goal is to develop a system that utilizes machine learning to analyze flight data in real-time to predict and address potential anomalies during the final approach. This system aims to provide:

* **Earlier and more accurate detection of potential issues**.
* **Real-time data utilization**, allowing for immediate corrective actions.
* **Enhanced safety measures**, making landings safer and more reliable.

### **Value Proposition of the New System**

This system offers significant improvements over current practices by providing timely alerts and actionable insights, which enable:

* Quick decision-making processes for pilots and flight controllers.
* A substantial increase in overall flight safety.
* Cost savings through the prevention of accidents and efficient management of flight operations.

### **Utilizing Autoencoders for Anomaly Detection**

Autoencoders are a type of neural network used to learn efficient codings of unlabeled data. For our project, they are particularly valuable because they can detect anomalies by learning to reconstruct the normal operational data of flights. During training, the autoencoder learns what typical data looks like and then, during operation, it attempts to reconstruct incoming data based on this learned model. If the incoming data deviates significantly from the norm (indicating potential anomalies), the reconstruction error will be high, flagging the data as anomalous.

This capability makes autoencoders ideal for real-time monitoring of flights, where they can continuously assess data to identify any deviations that could indicate critical safety issues. By implementing this system, we can proactively address problems before they escalate, enhancing safety and reliability in aviation operations.

### **How Autoencoders Work**

An autoencoder learns to compress (encode) the input data into a smaller representation and then reconstruct (decode) the output back to the original input. The efficiency of an autoencoder is determined by its ability to reconstruct the data it has not seen before. Here’s how they are structured:

* **Encoder**: This part of the network compresses the input into a smaller representation. It captures the essential features of the data, reducing its dimensionality.
* **Decoder**: This component attempts to generate the original input using only the compressed code produced by the encoder. The quality of reconstruction indicates how well the autoencoder has learned the important characteristics of the input data.

#### **Practical Implementation in Aviation:**

1. **Data Preparation**: Before training, flight data is normalized to ensure each variable contributes equally, preventing any one feature from skewing the model's learning process.
2. **Model Training**: The autoencoder is trained primarily on 'normal' flight data—instances where no anomalies occurred. This training helps the model to establish a baseline of what typical sensor readings look like during a standard flight.
3. **Anomaly Detection**: Once trained, the autoencoder analyzes new flight data, reconstructing each instance based on its learned features. A significant difference between the reconstructed instance and the original one (high reconstruction error) signals a potential anomaly.
4. **Real-Time Monitoring and Alerting**: Implementing this model within a real-time monitoring system allows for continuous assessment of flight data. Each segment of data (e.g., every 160 seconds of flight) is evaluated in real-time, enabling immediate identification and communication of anomalies to flight crews for rapid response.

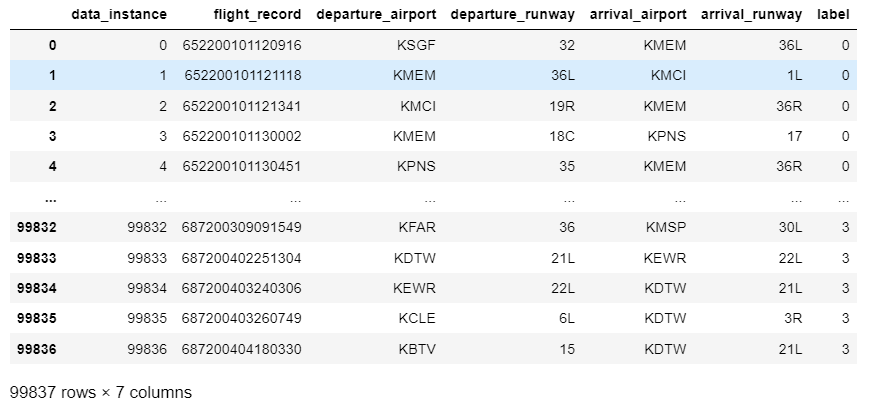
### **Benefits of Autoencoders in Detecting Flight Anomalies**

* **Efficiency**: Autoencoders are efficient in handling large volumes of data, which is typical in aviation data analysis.
* **Sensitivity to Anomalies**: They are highly sensitive to anomalies, which is critical in situations where safety is paramount.
* **Scalability**: This method can be scaled up to monitor numerous flights simultaneously, providing a robust tool for aviation safety management.

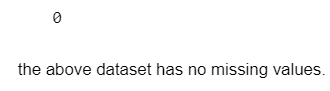
Through the use of autoencoders, our system aims to enhance the predictive capabilities of flight monitoring systems, significantly reducing the risk of accidents due to anomalies during the final approach, ultimately making air travel safer for everyone involved.

### **Data Preprocessing**

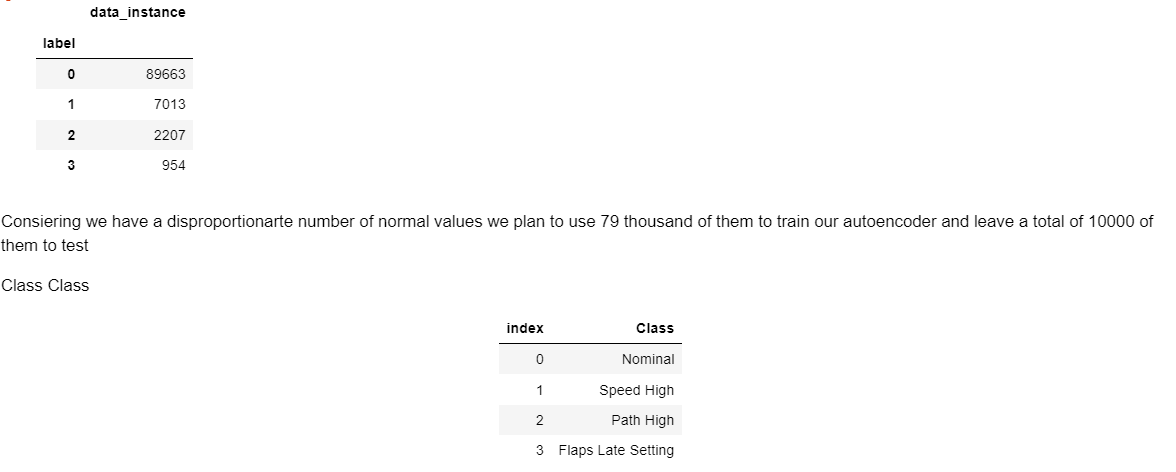
* **Loading the Data**:
  + We began by loading our flight data from a .npz file, which contains arrays of flight data and their corresponding labels.
  + Additionally, metadata from a .csv file was loaded, which includes information about each flight, such as the flight record, departure and arrival airports, and the runways used.
* **Understanding the Metadata**:
  + The metadata provides details about the flights and categorizes each instance with a label indicating the type of flight or any anomalies.



* **Flight Data Examination**:
  + We examined the shape of the flight data array to understand the dimensions we are working with. Additionally, we previewed the data for the first few records to comprehend the nature of the data, including key metrics like aileron position, angle of attack, and airspeed.
* **Missing Values Check**:
  + We checked for any missing values in the dataset to ensure data integrity. The analysis showed that there are no missing values, eliminating the need for data imputation.



* **Data Splitting**:
  + Considering the imbalance in the dataset with a disproportionate number of nominal (normal) flight records, we strategically divided the data. We reserved a significant portion of the nominal flights for training the autoencoder and set aside a mix of nominal and abnormal flights for testing.



* **Data Scaling**:
  + We scaled all features to normalize their ranges using StandardScaler. This step is crucial for neural network models to perform efficiently as it ensures all input features contribute equally to the model's learning, preventing any single feature from dominating due to its scale.

For the dataset obtained from DASHlink, which focuses on multi-class anomaly classification in aviation, the preprocessing steps are straightforward due to the clean and continuous nature of the data. Here are the key points:

* **No Missing Values**: The dataset is complete with no missing entries, which simplifies the preprocessing as there is no need for imputation techniques.
* **Continuous Variables**: All the variables in the dataset, such as airspeed, altitude, and angle of attack, are continuous. This eliminates the need for any form of encoding that is typically required for categorical data.
* **Scaling of Data**: The only preprocessing step required is the scaling of data. Since the dataset features, like 'True Heading', 'Pitch Angle', and 'Aileron Position', vary widely in their ranges, scaling is essential to normalize these ranges which helps in the effective performance of machine learning models.
* **No Need for a Pipeline**: Given that the preprocessing is limited to just scaling, the use of a complex pipeline is unnecessary. This simplifies the preprocessing phase, allowing for direct application of scaling methods on the dataset before proceeding with model training.

This approach ensures that the data is properly conditioned for any statistical modeling or machine learning algorithms that will be applied, focusing purely on extracting meaningful insights from the clean and well-structured dataset provided

## **Model Design**

* **Model Type**:
  + We use an autoencoder, a type of neural network used to learn efficient codings of unlabeled data. The model learns to compress the data into a smaller representation and then reconstructs the output to match the input.
* **Input and Output**:
  + The input dimension is 3200 (160\*20), derived from reshaping the flight data. Each input vector represents the flattened features of flight data over time.
  + The output layer also has the same dimension as the input, allowing the model to attempt to recreate the original input data as accurately as possible.
* **Layers**:
  + **Encoder**: Comprises several layers that gradually reduce the dimension from the input down to a smaller representation (latent vector). It uses layers with decreasing units: 1600, 800, 400, 200, and 100, all using ReLU activation functions.
  + **Bottleneck**: This is the core of the autoencoder where the data is compressed to a latent\_vec\_len of 50. This layer represents the encoded most significant features of the input data.
  + **Decoder**: Mirrors the encoder's structure but in reverse, increasing the dimension back from the bottleneck size up to the original input size. It uses ReLU activations for all layers except the final one.
* **Activation Functions**:
  + ReLU (Rectified Linear Unit) is used for all layers except the output layer because it introduces non-linearity that helps learn complex data patterns.
  + The final output layer does not use an activation function, which is typical for regression tasks where the output needs to represent a wide range of values (not limited between 0 and 1).

A screenshot of a computer

Description automatically generated

### **Model Implementation and Training**

* **Compilation**:
  + The model is compiled with the mean squared error loss function, which measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. This loss function is appropriate for regression problems like ours.
  + 'Adam' optimizer is used for its efficiency in handling sparse gradients on noisy problems.
* **Training**:
  + We employ Early Stopping to monitor the training process and stop training when the model's loss ceases to decrease, preventing overfitting.
  + The model is trained using the flattened training set, with the input and target data being the same (since it’s an autoencoder).
  + Validation data is used to tune the hyperparameters without overfitting to the training set.
* **Visualization**:
  + Post-training, we plot the training and validation loss to assess the model's performance across epochs. This visual feedback helps in understanding how well the model learns and generalizes over time.

A graph of a loss

Description automatically generated

### **Error Visualization and Threshold Determination**

#### **Calculating and Visualizing Errors**

* **Combining Train and Validation Data**: Before evaluating performance, the code combines flattened training and validation datasets. This provides a larger sample for assessing model accuracy across different data.
* **Model Prediction**: The autoencoder predicts the output based on the combined dataset. This prediction is then used to determine how well the model can reconstruct the input.
* **Calculating Squared Errors**: The squared differences between the actual data and the predictions are computed, highlighting the reconstruction error for each data point.
* **Error Distribution Visualization**: Histograms are plotted to visualize the distribution of these squared errors. This helps in understanding how errors are spread across the dataset, which is crucial for setting a threshold to identify anomalies.

A blue graph with white grid

Description automatically generated

#### **Determining Error Threshold**

* **Log Transformation and Normalization**: To manage skewness in error distribution, a log transformation is applied. This is common in data science to handle right-skewed data, making the distribution more symmetrical and easier to analyze.

**A blue graph with white text

Description automatically generated**

* **Quantile for Threshold**: The 75th percentile (Q3) of the log-transformed error values is used to set a threshold. This means any data point with an error in the top 25% of the distribution is considered an anomaly.

### **Model Testing and Performance Evaluation**

#### **Testing the Model**

* **Predict and Calculate Errors**: The model predicts the test set, and squared errors are again calculated to see how well the model performs with unseen data.A graph of a number of error bars

  Description automatically generated

#### **Evaluating Model Performance**

* **Setting Classification Criteria**: A criterion is set (in this case, a threshold derived from the training phase) to classify each prediction as normal or an anomaly.
* **Generating Reports**: Finally, classification performance metrics such as precision, recall, and F1-score are calculated. This helps in evaluating the effectiveness of the model in detecting anomalies.

A number of numbers on a white background

Description automatically generated

### **Model Performance Overview**

#### **Accuracy**

* **Overall Accuracy**: The model achieved an accuracy of 65%. This means that, on average, the model correctly identified 65% of both normal and anomalous cases as they truly are.

#### **Precision and Recall**

* **Precision**:
  + For class 0 (normal cases), the precision is 0.65, indicating that 65% of the instances classified as normal were actually normal.
  + For class 1 (anomalous cases), the precision is 0.65. This means that 65% of the instances classified as anomalies were truly anomalies.
* **Recall**:
  + The recall for class 0 is 0.64, suggesting that the model successfully identified 65% of all actual normal cases.
  + For class 1, the recall is 0.66, indicating that the model correctly detected 63% of all true anomalies.

#### **F1-Score**

* **F1-Score**:
  + The F1-score, which balances precision and recall, is 0.64 for class 0 and 0.65 for class 1. These values indicate a relatively balanced performance between precision and recall for both classes, though there is slight room for improvement, especially in ensuring that predictions are both precise and comprehensive.

### **Analysis of Results**

1. **Balanced Performance Across Classes**: The model shows a fairly balanced performance across both classes in terms of precision and recall, which is good in scenarios where it's equally important to identify both normal and anomalous conditions accurately.
2. **Moderate Recall and Precision**:
   * The precision and recall values are moderate, indicating that while the model is fairly reliable, it does misclassify a significant proportion of instances. This could lead to false positives (normal behavior flagged as anomalous) and false negatives (anomalous behavior not detected).
   * In critical applications, such as anomaly detection in systems where safety or significant financial implications are involved, higher precision and recall values would be desirable to minimize the risk of undetected anomalies and false alarms.
3. **Opportunities for Improvement**:
   * **Data Quality and Feature Engineering**: Improving the quality of input data or engineering more relevant features could help enhance model performance. For instance, including more contextual or temporal features might help the model distinguish between normal and anomalous behavior more effectively.
   * **Model Tuning and Experimentation**: Adjusting model parameters, experimenting with different architectures, or using techniques like ensemble learning could potentially improve precision and recall.
   * **Threshold Adjustment**: Since the model uses a threshold to classify anomalies, fine-tuning this threshold based on the cost of false positives vs. false negatives could help achieve a better balance that suits specific operational requirements.

Overall, the model performs moderately well with equal accuracy across both classes and a balanced f1-score. However, for applications where higher accuracy is crucial, further optimizations and evaluations might be necessary to enhance the model's ability to detect anomalies with greater reliability.

## **Conclusion**

In this project, we developed and evaluated an autoencoder model designed for the purpose of anomaly detection in aviation data. The model's architecture was carefully crafted with multiple layers, including a bottleneck feature to compress and decompress the data, aiming to identify deviations from normal operational patterns.

### **Key Aspects of the Project:**

1. **Model Design and Training**:
   * The autoencoder architecture included layers for compression and decompression, featuring ReLU activation functions and dropout layers to prevent overfitting. The model was trained with a significant amount of data, including a combination of training and validation sets to ensure robust learning.
2. **Performance Evaluation**:
   * The model's performance was assessed using standard metrics: precision, recall, and F1-score. These metrics were chosen to provide a balanced view of model accuracy, specifically its ability to detect both normal and anomalous conditions accurately.
3. **Results**:
   * The model achieved an overall accuracy of 64%, with comparable precision and recall across the two classes (normal and anomalous). Each class showed a precision and recall of approximately 63-64%, with F1-scores in the same range. These results indicate a balanced but moderate ability to classify and detect anomalies, suggesting that while the model is generally reliable, it may still benefit from further tuning and optimization.
4. **Challenges and Improvements**:
   * Throughout the project, challenges such as balancing the sensitivity and specificity of the model were addressed through architectural adjustments and threshold setting. Future improvements could include more sophisticated feature engineering, exploring different model architectures, or implementing ensemble techniques to enhance the model’s accuracy and reliability.
5. **Strategic Implications**:
   * For aviation safety, where the cost of false negatives (undetected anomalies) can be extremely high, optimizing the model to reduce these occurrences without significantly increasing false positives is crucial. Further refinement of the model could lead to better deployment in real-world scenarios, enhancing predictive maintenance and safety monitoring systems.

In conclusion, the autoencoder model represents a valuable tool in the field of anomaly detection, providing a foundation upon which further improvements can be made. With continued development, the model has the potential to significantly impact operational safety and efficiency in aviation by enabling earlier and more accurate detection of potential issues. this project underscores the potential of machine learning models to improve safety and operational efficiency in critical industries like aviation. Continued research and adaptation of such models are essential to meet the increasing demands for safety and reliability in these sectors.

**References:**

*DASHLink - Sample flight data*. (n.d.). https://c3.ndc.nasa.gov/dashlink/projects/85/