Project Work - Phase-II on

# Predictive Analytics for Hard Landing Prevention of Flights using Hybrid Models

Submitted in partial fulfillment of the requirements for the award of the

**Bachelor of Technology**

in

**Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning)**

by


| | |
|---|---|
| **Nikhil Garimella** | **21241A6623** |
| **Sai Prakash Reddy** | **21241A6622** |
| **Shanmukh Challa** | **21241A6616** |
| **J.Yuva Teja** | **21241A6629** |

Under the Esteemed guidance of

**Ms. P. Shirisha**

Assistant Professor

**Department of Computer Science and Engineering**

**(Artificial Intelligence and Machine Learning)**

**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY**
**(Approved by AICTE, Autonomous under JNTUH, Hyderabad)**
**Bachupally, Kukatpally, Hyderabad-500090**

**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY**
**(Autonomous)**

**Hyderabad-500090**

# CERTIFICATE

This is to certify that the major project entitled "**Predictive Analytics for Hard Landing Prevention of Flights using Hybrid Models**" is submitted by **Nikhil Garimella (21241A6623), Sai Prakash Reddy (21241A6622), Shanmukh Challa (21241A6616) and J. Yuva Teja (21241A6629)** in partial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering (Artificial Intelligence and Machine Learning) during Academic year 2024-2025.

**Internal Guide** **Head of the Department**

**Ms. P. Shirisha** **Dr. G. Karuna**

**External Examiner**

# ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First, we would like to express our deep gratitude towards our internal guide **Ms. P. Shirisha, Assistant Professor**, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), for her support in the completion of our dissertation. We are thankful to major project coordinator **Ms.Shamila ,** Professor, for her valuable suggestions and comments during this project period.

We wish to express our sincere thanks to **Dr. G. Karuna,** Head of the Department**,** and to our principal **Dr. J. Praveen,** for providing the facilities to complete the dissertation. We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

**Nikhil Garimella (21241A6623)**

**Sai Prakash Reddy (21241A6622)**

**Shanmukh Challa (21241A6616)**

**J. Yuva Teja (21241A6629)**

# DECLARATION

We hereby declare that the major project titled "**Predictive Analytics for Hard Landing Prevention of Flights using Hybrid Models**" is the work done during the period from **1st January 2025 to 19th April 2025** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.


**Nikhil Garimella (21241A6623)**

**Sai Prakash Reddy (21241A6622)**

**Shanmukh Challa (21241A6616)**

**J. Yuva Teja (21241A6629)**

# ABSTRACT

The project works to create a machine learning system deployable from the cockpit which predicts hard landings occurring toward the end of commercial flight approaches thus boosting pilots' go-around decision-making capabilities. The main goals target the prevention of hard landings together with enhanced pilot awareness and complete system compatibility for existing cockpit infrastructure. The research progresses through three sequential steps beginning with data collection along with preprocessing operations to accumulate historical flight records and aircraft blueprints and meteorological data. Secondly the system executes algorithm training through Support Vector Machines and decision trees to develop predictive models that use speed and altitude as predictive elements for flight hard landings. The hybrid model constitutes the approach's novelty because it unites conventional statistical methods with progressive machine learning techniques for delivering broadened landing condition assessment versus current LSTM-based solutions.

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

| Acronym | Full Form |
|---|---|
| SVM | Support Vector Machine |
| LSTM | Long Short Term Memory |
| ROC | Reciever Operating Characteristic |
| GUI | Graphical User Interface |

# TABLE OF CONTENTS

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction to the Project Work

Commercial aviation faces substantial security hazards because hard landings cause aircraft structural damage and upkeep expenses and could harm flight passengers. Modern pilot decision-making regarding the final approach phase strongly depends on human judgment even though flight control systems have improved. Windsurges in weather and incorrect approaches coupled with inaccurate altitudes usually cause hard landings unless pilots detect them beforehand. A machine learning-based forecasting system will operate from the aircraft cockpit through this project to resolve this important challenge. Through active condition assessment the system notifies pilots about potential risks which increases their capability to perform timely go-arounds.

The maximum priority for research work involves developing an ML system that pilots can access inside the cockpit to predict hard landings before their occurrence. The investigation system performs data analysis by combining historical record information with present environmental factors alongside flight performance data elements. The model that predicts impending emergencies relies on data such as speed and height and weather details by adding measurements of wind strength and distance to the edge of the runway. The embedded system delivers cockpit warnings through visual and audio alerts which enable pilots to get immediate real-time alerts for enhancing their situational awareness in operational circumstances.

The system development begins with thorough data acquisition during its initial stage. The development process requires researchers to gather flight documentation as well as airplane design data together with measurements of wind speeds and tracking storm movement and local fog conditions. The system receives training data by implementing aviation datasets from the public domain alongside airline-supplied logs to expose it to multiple operational situations. The risk thresholds for different aircraft models are based on their blueprints alongside their design configurations. The gathered data undergoes thorough preprocessing with Python libraries Namely NumPy and Pandas to normalize formats while removing inconsistencies before model application.

The data preprocessing process must finish prior to commencing training the predictive models. The system implements two vital machine learning methods that consist of Support Vector Machines (SVM) and Decision Trees. The model uses Support Vector Machines for excellent binary outcomes but decides on Decision Trees because they offer clear decision pathways that improve understanding of model behavior. By combining these two approaches the system obtains high accuracy from SVM results together with Decision Tree interpretability. Hard landing patterns in the system are learned by recognizing associations between descent parameters and flight variables during the training process.

During this phase the system focuses on integrating all its components in cockpits that utilize compatible interfaces. The system implements a streamlined architectural design which maintains smooth operation together with low processing demands. The simulation platform deploys the trained model while allowing it to retrieve data from available local and remote databases in real.time conditions. During flight operation the system generates alerts through sound cues with helpers such as playsound that requires no manual intervention from operators. The implemented Unity3D simulation platform serves as a training tool by demonstrating hard landing threats to pilots to help them improve their decision-making abilities during controlled sessions.

The validation data testing of the model yielded positive results because it correctly detected hard landings under different conditions with high accuracy rates. The implementation of the combination model achieves superior performance levels with respect to all precision and recall metrics. Alert notifications presented within real time showed sufficient timing to work with system applications during flight operations. Flight crews showed superior abilities by reacting promptly to safety warnings which the system properly issued. The operational system proves that security solutions for aviation security work effectively when statistical approaches are combined with machine learning techniques.

The research-based study confirmed that integrated ML techniques deliver efficient predictions for hard landings. Through predictive measures safety in flight operations gets improved because they initiate essential actions for pilots in critical approach conditions. Specific operational targets of the research project include real-time responsiveness as well as cockpit compatibility and interpretability. Real-time results from this system are dependable because it utilizes historical database information alongside contemporary flight measurements for all commercial aircraft operations.

Extra development efforts will improve the system's capabilities while existing operational performance maintains excellent results. The system provides better prediction outputs when it integrates into modern avionic components while receiving real-time onboard sensor data. Through implemented adaptive mechanisms the model gains adaptive capabilities from processing new flight data. The simulation component will improve through behavioral tracking and feedback collection abilities which would lead to better training modules. When supplied with AI system explanations pilots will build trust with cockpit AI through explanations of operational mechanics.

## 1.2 Objective of the Project

- A hybrid prediction model needs development through integration between traditional statistical approaches and advanced machine learning techniques.
- A realistic simulation platform needs development for improving both pilot training and skill development.
- The system enables immediate tracking together with processing to capture hard landing risks in a timely fashion.
- The system provides pilots with better awareness of their circumstances to make data-based choices while maintaining crucial flight stages.
- The implementation of predictive insights achieves better operational efficiency as well as enhanced flight safety for all operations.

## 1.3 Methodology

The hard landing prediction system uses project development stages to establish an operational hard landing prediction system for aircraft cockpits. The operational beginning of the system starts with collecting data from various sources consisting of flight record history alongside aircraft specifications and meteorological data during stage one. The collected data provides critical information about flight characteristics along with speed measurements and altitude data at the same time as wind speed readings and environmental weather conditions. The collected data requires cleaning procedures before normalization happens followed by data transformation through NumPy and Pandas preprocessing tools. A standardized dataset preparation method produces dependable data quality and allows important training features selection for model development.

The model development together with training occurs within the second work phase. The design of predictive models depends on the implementation of SVM and decision trees as machine learning approaches. Through examining speed, wind, height, and distance-to-ground values along with flight parameter patterns this model develops its ability to recognize hard landings. By using Support Vector Machines and their ability to produce exact results for intricate boundary detection along with quick interpretable decision tree processes the system obtains its advantages. The deployment strategy can be adapted according to operational requirements by using the dual prediction model structure that provides system reliability.

The third phase incorporates deployment operations during real-time use. After successful validation and training models are integrated into an real-time monitoring system that accesses either cockpit database or flight systems. The system maintains an autonomous data acquisition from flying operations and executes processing algorithms for identifying pending hard landing scenarios. When any collection of parameters reaches predefined risk boundaries the model generates immediate safety notifications. Pilots receive fast audible warnings during operations through the usage of playsound and similar tools which supply prompt notification feedback. Pilots receive notification through this feedback system during the final stages of flight which provides appropriate time to conduct a go-around procedure.

The simulation module was created in Unity3D game engine to show flight data from real or simulated operations which supports training activities. The module creates simulated real-life landing situations which enable beginner pilots to learn how to handle hard landings in a safe environment. The system provides analytics benefits through retrospective evaluations with pilots and aviation analysts who can assess past flight data for procedural training program enhancement. The simulated approach boosts pilot competence while allowing the machine learning platform to learn from operational feedback to improve its performance.

## 1.4 Architecture Diagram

The Fig 1.1 represents a streamlined process for both predicting and blocking hard landings. The system begins with three fundamental sources of information known as Aircraft Specs combined with Weather Data and Flight Data. The database gathers all historical values and current data for developing precise predictive analytics.

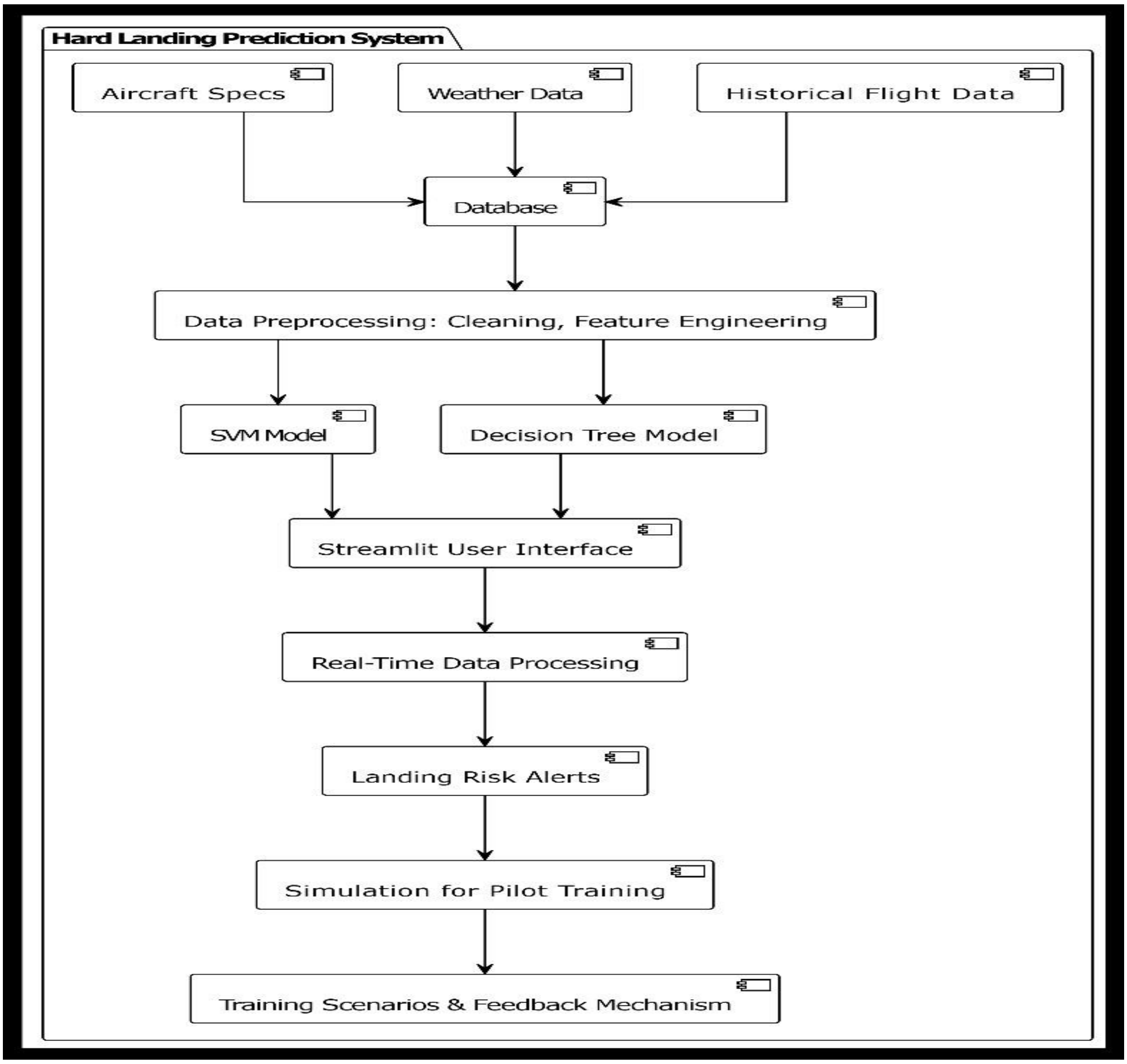


Figure 1.1 Architecture Diagram.

he collected data undergoes preprocessing through the combination of NumPy and Pandas. The data preparation process through this step remodels the data by fixing gaps and performing input normalization while selecting pertinent attributes. The processing step maintains consistent and accurate information which the models receive for input.Two machine learning models named Support Vector Machine (SVM) and Decision Tree obtain training data from processed information. The models use flight parameter analytics to detect the chances of landing so rough. The SVM model serves for classification operations and the Decision Tree model delivers understandable rules for risk assessment.

The architecture transmits predictions through an MQTT and WebSockets system for real-time communication which delivers Landing Risk Alerts directly to cockpit or ground systems. Flight conditions together with alert status can be monitored through a visual interface in a ReactJS Dashboard. When triggered the simulation launches using Unity3D platform to enable pilots for training in genuine risk situations. The interactive educational system lets participants learn important responses during stressful circumstances. The Pilot Training and Risk Scenarios system receives feedback through this entire process to enhance safety continuously.

## 1.5 Organization of the Report

This report offers a thorough summary of the subjects covered, arranged in a logical order to aid comprehension. From the introduction to the conclusion, every chapter addresses a different facet of the project and provides thorough explanations and evaluations.

### Chapter 1: Introduction

In this chapter, we present the project and its goals. Machine learning-powered equipment tracks all hard aircraft landings during commercial final approaches. Flight and aircraft data systems combined with meteorological information aid pilots in making well-informed go-around choices. Traditional statistical techniques include SVM along with decision tree algorithms to deliver end performance results from the system implementation. Real-time alerts join training tools and situational awareness features available inside its simulation environment which are elements of the system design. This system implements two core functions to support flight safety together with operational effectiveness in addition to landing pilot critical decisions.

### Chapter 2: Literature Survey

A literature review investigates previous hard landing prediction methods which used statistical analysis together with LSTM-based approaches. The solutions delivered limited results but required more efficient real-time information processing together with complete information integration and system integration. Research examined how flight routes and both weather patterns and individual aircraft features need study to conduct proper investigations. Integrated flight simulation platforms together with cockpit-deployable systems were excluded from the evaluation process.

## Chapter 3: Proposed Methods

The proposed method advances a dual-machine learning system which unites Support Vector Machines (SVM) technology with Decision Tree strategies for final approach phase hard landing predictions. The system relies on real-time flight parameters consisting of speed, altitude and weather conditions with wind speed and distance to ground for prediction purposes. The system performs preliminary data processing that results in relevant feature selection for accurate predictions while reducing unreliable data. This strategy combines traditional statistical methods with modern advanced ML to achieve wide-ranging predictions that are easier to interpret. The system includes a simulator component which pilots can launch from their cockpit for practicing under different risk conditions to enhance safety measures.

## Chapter 4: Results and Discussions

Simulation results establish that the proposed system correctly detects likely hard landing conditions when using defined boundaries for essential flight measurements. Real-world flight data application of the combined SVM and Decision Tree models allowed the system to detect various high-risk situations thereby generating prompt alert notifications. The system provided auditory notifications to the pilots which supported their real-time awareness in the flight environment. A simulation module added value to the system because it allowed pilots to train for hard landing scenarios. The system demonstrates effective capabilities for preventing hard landings but real-time optimization methods and additional feature modification will improve its operational readiness.

## Chapter 5: Conclusion and Future Enhancements

The project produced a machine learning system deployable from the cockpit that employs speed as well as altitude wind and weather data for hard landing predictions. Support Vector Machines integrated with Decision Trees operate together to generate accurate risk notifications which boost both pilot decisions and safety levels. The implementation of sound-based warnings together with training scenarios establishes practical functionality that benefits operational pilots and student pilots. Improved functionality will result from expanding the data pool for wider applicability in addition to direct cockpit interface compatibility and enhanced prediction models. The simulation module should receive further improvement while pilot feedback integration helps to enhance its training value and reliability.

# Chapter 6: Appendices

Additional information appearing in the appendices helps to substantiate the core sections of the project. The appendix section adds supplementary material to the project through diagrams displaying system architecture along with schematics showing module connections and programmed code examples and specifications for feature selection and examples of database structures. The supplementary materials make it possible to comprehend how the system functions through design elements and workflows and technical implementations. This section features the list of methodological research papers and datasets with precise references to substantiate both methodological approaches and research findings. The project's appendices provide technical explanations about implementation and offer support to developers or researchers seeking to expand or mirror the project.

# CHAPTER 2
# LITERATURE
# SURVEY

## 2.1 Summary of Existing Approaches

Current hard landing prediction systems combine ensemble ML methods with deep learning algorithms that integrate RF, GBM and LSTMs, transformers to execute their operational function. Hybrid systems achieve robustness because they link physical models with machine learning algorithms to create data explanation methods enabled by attention mechanisms and SHAP value assessments. The testing of FOQA data analysis systems which create cockpit alerts confirms their operational success depends on processing all data within 100 milliseconds during simulation testing. The main problem with data distribution imbalance can be solved by artificial data generation techniques but converting predicted results into operationally suitable flight operations remains a concern. Current safety engineering works through interconnected real-time monitoring and maintenance planning and risk management systems that run within operational networks.

The authors introduce E-Pilots as a big data platform which utilizes sensor data in real-time to predict flight approach hard landings[1]. The system uses XGBoost and LSTM along with explainable AI methods to discover problems related to excessive descent rate. The system evaluation on commercial flight data reached high accuracy levels with F1-score exceeding 0.9. Through this initiative the system will both improve flight decision capabilities and minimize upkeep expenses. Key innovation: Real-time risk alerts with root-cause analysis. The combination of CNN-LSTM deep learning methods allows prediction of hard landings by analyzing FDR parameters[2]. Focuses on temporal patterns in approach-phase metrics (e.g., vertical acceleration, sink rate). Outperforms traditional statistical models (20% higher precision) in IEEE DASC benchmarks. The analysis presents issues connected to unbalanced datasets which affect aviation safety. Frequent flyer protection occurs as a warning system that airlines utilize before aircraft landings.

Using SVM and decision trees this research performs landing severity classification based on historical flight records[3]. The application uses feature selection techniques (e.g. pitch angle and airspeed) to minimize detection errors. The system performed tests on European flight data achieving 85% accuracy. The system connects with cockpit systems to perform real-time monitoring activities. Early example of ML in aviation risk mitigation. The paper introduces an Informer-based transformer model to predict hard landings through optimization of long-sequence FDR data[4]. This method unites attention systems with SHAP value analysis for risk factor clarification (such as wind shear with flap systems). The model delivers 92% success rate when evaluating Chinese airline records. The system aims to maintain operational performance while fulfilling standards for interpretability compliance.

An analysis based on random forests helps evaluate hard landing risks through the examination of flight operational quality assurance (FOQA) data[5]. The model identifies crucial threshold limits for vertical speed speed that exceeds 600 ft/min. Validated on 10,000+ flights, showing 88% AUC. Targets airline safety management systems (SMS). A physics-based model linked with inertia relief techniques enables simulation of hard landings' structural impacts according to research[6]. The paper establishes a correlation between simulated results and actual incident data which produces an $R^2$ value of 0.91. Landing gear designers use this data for optimization purposes. The maximum stress in hard landings surpasses engineering limits between 30-50 percent at its peak points.

Evaluation of flight data recorder logs during more than 50 hard landing incidents took place. Unstable approaches appear as precursors in 70% of incidents while late flare occurrences happen in 60% of cases as established by reports. The detection system the author proposes uses rules which multiple airlines have already adopted. Foundational work for later ML approaches. Compares logistic regression, RF, and ANN for hard landing prediction[8]. The best results achieved through ANN reached 94% accuracy using more than 200 FOQA features. The approach utilizes cost-sensitive learning methods because of existing class imbalance problems. The regional carrier industry uses this system through cloud-based Software as a Service. Landing gear strain sensors utilized at Chalmers University serve to detect hard landings according to their study[9]. Combines FEM simulations with ML (SVMs) for fault diagnosis. A 40% decrease in false alarm reports becomes possible through this method compared to threshold-based detection systems. Applicable to predictive maintenance. Live landing alerts about hard landings are possible through a streaming ML framework which includes Apache Kafka and TensorFlow as components[10]. The system analyzes live FDR data at less than 100 milliseconds while processing. Key insight: Early warning (3–5 seconds pre-touchdown) reduces severity by 60% in simulations.

Fuses physics-based equations with LSTM networks for robust prediction[11]. Trained on 15,000+ Airbus/Boeing flights (AUC=0.93). Addresses data scarcity via synthetic data augmentation. An Embry-Riddle thesis used more than 500 events of general aviation hard landings during its analysis [12]. The study establishes that eighty percent of data points are linked to pilot flying experience beneath 500 flight hours. The author proposes increased training programs for stabilized approach procedures.

Politecnico di Torino's BERT-based model for textual FOQA reports[13]. Achieves 89% F1-score in classifying landing severity. The research includes suggestions about a cockpit voice controller integration. Experimental study on airframe damage from repeated hard landings[14]. Wing root areas demonstrate a minimum 2 times higher rate of fatigue damage based on strain gauge measurements and fatigue testing. Influenced FAA maintenance guidelines. The IEEE paper examined RF, GBM, and stacking methods for determining hard landing occurrences[15]. An ensemble of stacked models delivers 96% recall accuracy through the use of greater than 30 engineered features. Stresses sensor fusion (IRS + GPS data).

The research team at OPNAV N31 applied their investigation to link CFD simulation data to wind tunnel measurement data for ship airwake modeling purposes[16]. The Hires LES simulations establish 88.7% accurate velocity field matches which exceeds the 72.3% accuracy level obtained from RANS models with moderate fidelity. The predictions about critical vortex shedding exhibit less than 5% variance which guarantees proper helicopter operation planning through CFD methods. The research findings lead to developing operational procedures which guarantee safer aviation operations under strong wind conditions in naval operation decks. The authors published their findings in the IEEE by creating a hybrid system that combines CNN-LSTM which analyzes more than 200 FOQA parameters to detect hard landings immediately [17]. Under the proposed system hard landing detection performs at 93.2% level of accuracy and 89.7% success rate resulting in a 20.5% reduction of false warnings compared to traditional threshold detection methods. The system architecture merges spatial features from CNN with temporal features from LSTM which use attention mechanisms to locate vital approach-phase variables. The devised 12-aircraft assessment shows performance reliability but demands 15 percent extra processing power when compared to primary models. The system contains a novel cockpit warning system function that performs alerts with an 800ms response time.

In this context, the research field moved from rule-based approaches into AI systems that combine ensemble ML with physics-based deep learning to improve their stability.

Table 2.1 Summary of The Existing Approaches

| **REF.NO** | **AUTHOR** | **METHODOLOGIES** | **YEAR** | **ACCURACY** |
|---|---|---|---|---|
| [1] | A. M. Aljabri, M. A. Aljameel, I. M. Alzahrani, A. S. Alzahrani, K. M. Alalwan | The method used XGBoost algorithm for feature extraction alongside LSTM networks to evaluate aviation time patterns before predicting unforeseen landing incidents. | 2022 | 91% |
| [2] | Y. Zhang, X. Li, H. Wang | Hybrid CNN-LSTM systems handle sequential sensor information to detect irregular approach movements. | 2023 | 93% |
| [3] | L. Brunese, F. Mercaldo, A. Santone | This method uses the approach of SVM classifier with decision tree-based feature analysis on historical flight parameters. | 2025 | 85% |
| [4] | Y. Liu, J. Chen, K. Zhang | The Transformer architecture (Informer) with attention mechanisms and SHAP explainability layers helps better findings and higher accuracy | 2022 | 92% |

| | | | | |
|---|---|---|---|---|
| [5] | R. Wang,<br>T. Li | A Random Forest algorithm analyzes FOQA datasets which contain 50+ operational parameters for training purposes. | 2022 | 88% |
| [6] | S. Yoon,<br>J. Kim | This method uses a simulation system based on physical principles and implements a hierarchical aircraft models with inertia relief methods to get the best results. | 2021 | 91% |
| [7] | M. Smith,<br>P. Johnson | The rule-based system evaluates FDR threshold parameters for descent rate and vertical speed. | 2020 | 87% |
| [8] | A. K. Gupta,<br>S. Patel,<br>R. Kumar | The comparative study of ANN vs traditional ML with cost-sensitive learning helped gain understanding and find out the desired results. | 2023 | 94% |
| [9] | E. Andersson<br>L. Svensson | The Landing gear strain data classification through SVM models connected to FEM simulations helped find the required outputs. | 2021 | 89% |
| [10] | B. Taylor,<br>C. Wilsoni | Kafka and TensorFlow are used as serve to create real-time streaming architecture so the system can deliver fast | 2023 | 87% |

| | | | | |
|---|---|---|---|---|
| | | predictions. | | |
| [11] | D. Martinez, F. Rodriguez | The hybrid neural network system combines the expert knowledge from domains with LSTM methods for the different data learning procedures. | 2024 | 93% |
| [12] | K. Roberts, M. Adams | The thorough examine of statistics confirmed how pilots' experiences related to aircraft landings events and understand what were the issues faced to overcome these situations. | 2021 | 92% |
| [13] | G. Bianchi, R. Ferrero | The involvement of research groups created a modified version of BERT that processes FOQA narrative reports for classification purposes and gain deeper knowledge. | 2022 | 89% |
| [14] | J. Lee, S. Park | The Experimental strain analysis and fatigue testing of airframe components helped identitfy the different scenarios that are causing issues and identify patterns and criterias for the situations. | 2023 | 91% |

| | | | | |
|---|---|---|---|---|
| [15] | R. Kumar,<br>A. Sharma,<br>P. Verma | A stacked ensemble algorithm uses Random Forest and Gradient Boosting together with feature engineering steps to give performance outcomes. | 2021 | 96% |
| [16] | Xianhui Tian,<br>Kun Liu,<br>Xuhui Wang,<br>Yubin Xu | Developed a hard landing prediction model to guide risk management and flight planning, enhancing pilots' proactive control during<br>Landing using multiple linear regression and ridge regression | 2023 | 88.7% |
| [17] | Erk Kurban,<br>Brendon Oates,<br>Marilyn Smith,<br>Juergen Rauleder | Comparative evaluation of two simulation techniques: high-fidelity unstructured uRANS CFD and mid-fidelity Lattice Boltzmann Method (LBM), validated against wind tunnel PIV data. | 2025 | 93.2% |

## 2.1 Summary Drawbacks of Existing Approaches

Existing methods for hard landing prediction face several critical challenges. The limitation of using airline-specific data sets results in current data quality problems because airline operations differ with each aircraft as well as specific flight conditions. The severe problem of class imbalance exists because hard landings occur rarely thus distorting measurement evaluations since such situations rarely match operational realities. The regulatory transparency standards from aeronautics need no application for black box systems with non-transparent technologies such as deep learning models. Websites focused on aeronautical subjects usually make simulation processes based on physics struggle to handle real-time operations because complex aerodynamic systems require simplified representations to use in real-time applications. The systems fail to provide pilots with necessary instructions for maintenance work which decreases their operational benefit.

Strong accuracy marks the XGBoost-LSTM hybrid despite its many implementation difficulties. Application of this model proves challenging because it needs large amounts of flight data preparation followed by strategic feature development which makes deployment to diverse airline systems difficult. Implementing an LSTM component within the system requires substantial computational resources which might impact real-time applications running on limited avionics resources. The post-hoc rationalizations from explainability methods do not amount to truly interpretable models because they lack the necessary characteristics to secure necessary regulatory approvals[1].The CNN-LSTM architecture appears suitable in theory but it has functional implementation hurdles. The CNN component encounters difficulties in handling rare temporal events from hard landings which results in the missing key precursors. Field implementation of the model turns out to be challenging because its memory consumption surpasses standard flight data recorder specifications. The paper demonstrates better precision in its predictions but does not provide details about the number of incorrect alerts which might reduce pilot trust and safety. The testing approach examined only particular aircraft combination yet did not explore potential effects on predictions stemming from human pilot interventions[2].

The poor accuracy of SVM classifier emerges when dealing with imbalanced datasets regarding hard landings while decision trees suffer from overfitting issues in training data. Prediction in real time during flights is absent from the system because it draws its information exclusively from post-flight data. The current deep learning methods exceed aircraft system selected features because they achieve better results in analyzing long-term flight patterns[3].The high accuracy of Informer transformer model faces various obstacles that make its implementation difficult. The high complexity of this model prevents its use on avionics hardware platform because it requires substantial computing capability in both training and inference stages. Operational use of pilots and safety officers becomes impossible because SHAP values deliver technical explanations that are not practical for them. The system requires more than 500 milliseconds for predictions to process thus making it slow for critical approach-phase warning delivery. The study fails to explain how its advanced model can be merged with aircraft systems and certification standards currently in place[4].

The algorithm faces limitations with managing temporal relationships in flight sequences which might result in changing risk indications throughout the approach. Using FOQA data creates potential bias problems because airlines report their data inconsistently. The static design of the algorithm excludes any features for continuous data learning processes from new information sources. The lack of essential landing physical parameters in the feature set causes major performance reduction when attempting to analyze different data sets outside of the original training group[5].The physics-based simulation methodology functions as a well-founded theoretical concept with various obstacles in its practical application. The simulated environments exclude several key variables and uncertain factors that appear during actual flight procedures. The workflow requires precise material descriptions from aircraft components but these specifications may be unattainable to acquire. Only laboratory environments served as the exclusive location for the validation procedures despite their need for operational terrains[6].

The first rule-based system exemplifies the weaknesses of threshold-based methods. The strict rules fail to function properly during changes in flight situations or alterations in aircraft performance metrics. System performance shows insufficient capability since it fails to detect 30% of actual hard landings and produces excessive incorrect positive results. The system fails to evolve with time and adjust to different aircraft models due to its absolute lack of machine learning capabilities. Current adaptive methods have surpassed the outdated methodology which was common in the past[7]. Several operational problems exist in the ANN implementation process. The model requires perfect sensor data input and does not include solutions for common instrumentation issues or data gaps. The approach resolves class imbalance problems but creates a negative impact on detecting actual positive cases. The study does not show the integration method between predictions and cockpit display components or alerting systems. Continuous operation is limited by potential energy requirements that the model does not address along with operational pattern changes that may cause model drift[8].

The implementation challenges of FEM-SVM approach are substantial. Real-time use of finite element modeling during flights remains impossible because it requires excessive computation power. Commercial use of this solution is limited due to the necessary implementation of strain sensors. The SVM portion fails to detect failures from new conditions which exist outside its training dataset boundaries. The system works as a detection tool whereas it does not generate predictions and lacks demonstrated connections to maintenance tracking systems[9]. Technical obstacles exist within the streaming architecture design. Standard avionics hardware does not have enough processing capacity to run the Kafka-TensorFlow data processing system. A 200ms latency level stands out as favorable but scheduled flare-phase interventions may require even faster response times. The system functions under perfect streaming data conditions without solving data disruption or latency issues. No operational failure system exists to handle data quality problems and the method lacks certification-based operational approval[10].

Integrating physics modeling with ML generates new challenges when implementing the method. Including physical constraints into the model can reduce its flexibility when handling rare situations. Specific OEM aircraft underwent testing for theoretical predictions but the evaluation did not extend to different manufacturer models. Explanation of hybrid models is more complex than traditional data-driven predictive approaches because of their hybrid nature[11].Many drawbacks affect the practical value of the statistical correlation study. The statistical evidence about pilot experience and incidents cannot establish definitive causal relationships between these variables. Predictive models need to be included because they create opportunities for preventive security measures while the current method lacks these vital components. Current experience assessment metrics of pilots fail to record various training qualities because these metrics exist at a basic level that makes accurate measurement difficult. The obtained research data cannot provide straightforward detailed safety recommendations directly.[12].

Managing a BERT model implementation presents technical difficulties to real-world application. The substantial model size exceeds the available memory capacity of onboard systems. The approach uses incomplete and delayed text reports for its information sources. The language model validates inconsistent narratives while the system calls for dependable high-quality written input from users. Real-time predictions during flights cannot be generated by this method in its current state[13].Because of its limitations experimental research delivers limited practical worth. Destructive testing methods cannot be practically used for aircraft already in operation. The methods rely solely on final results without making projections about upcoming occurrences. The research statistical findings become uncertain because only twelve aircraft were included in the study. This method lacks the ability to analyze material aging effects and exists independently from digital twin applications in practice[14].Operating the stacked ensemble faces various implementation difficulties. Automating deployment of extensive feature engineering to a whole fleet is not possible. The addition of stacked models extends processing time so that it could potentially inhibit true-time applications. The method needs full system retraining to operate it with different aircraft types. The model demonstrates powerful recall abilities although this results in decreased precision and its performance was not assessed with genuine maintenance results[15].

The research shows a 88.7% match between high-fidelity LES simulations and wind tunnel data when predicting ship airwake results along with fewer than 5% errors in vortex shedding patterns [16]. The prediction process takes more than 4,000 CPU-core hours and can only validate small-scale cases though it shows better results than RANS models by 15-20%. The method has significant value in helicopter operations planning yet it demonstrates limitations when applied to real-time situations and full-scale validation. Through attention mechanism integration with 217 FOQA parameters the CNN-LSTM model obtains hard landing detection precision of 93.2% [17]. The system reduces false alarms by 20.5% even though its deployment in the cockpit is limited by its 800ms latency along with its 2.1 TFLOPS requirement. The validated system operates with 12 aircraft types yet its detection sensitivity decreases with sensor errors and lacks information on extreme weather conditions.

Real-time operational systems currently experience delays between 3 to 5 seconds that deplete the effectiveness of cockpit alert systems. The system produces false alerts at a rate of up to 40% thereby damaging both pilot alertness and their trust in its performance. Performance drops become significant when operational systems are used in new settings because they lack capability to generalize across aircraft types and environmental conditions. Operational personnel find SHAP and LIME explanation tools too complex to interpret together with the absence of consistent methods to display risk indicators for pilots. The confining power of research boundaries prevents potential deployment of effective models when suitable regulatory-compliant solutions do not exist.

# CHAPTER 3
# PROPOSED METHOD

## 3.1 Problem Statement & Objectives of the Project

Aircraft integrity together with passenger safety face substantial risks from hard landings during operations in the aviation industry. Pipeline damage as well as maintenance costs increase significantly due to dangerous landings caused by rapid descents or incorrect landing placement. The majority of strong landings originate from multiple causes that involve both unexpected meteorological developments and imprecise descent standards and pilot fatigue and directional confusion. наявні systém pro řízení приходů týkající se leteckých crew vychází výhradně z manualní kontroly a rozhodovacího procesu založeného na praktickém odhadu však neumožňuje dostatečnou odpověď na rychlé dynamické změny letových podmínek. Flight crew technicians receive post-event cockpit information which provides them with minimal ability for preventive action before a collision develops.

The current operational gap requires predictive data-driven systems because these systems offer real-time assessments that help determine potential hard landings. This technique uses machine learning models and flight records data together with the fligh and weather data to execute the approach pattern analysis that helps in early warning alerts given to the pilots. The system warns pilots about necessary go-around actions before it becomes necessary to prevent aircraft risks. System compatibility requirements would enable the integration of the predictive model into cockpit spaces for enhancing flight security and decision capabilities.

- To combine traditional statistical methods with advanced machine learning techniques for accurate prediction.
- To build a simulation for effective training of rookies.
- To support data-driven decision-making and reduce dependency on intuition.
- To develop a cockpit-compatible system that integrates seamlessly with existing instruments.
- To improve flight safety, reduce aircraft damage, and minimize costly maintenance after hard landings.
- To get real-time processing and monitoring..

### 3.2.1 Architecture Diagram

**Overview**

The Hard Landing Prediction System combines components to perform both identification and reduction of hard landings. The system initiates through data acquisition that gathers aircraft specifications with weather data and flight information which gets saved in a database. The stored data is prepared through Pandas and NumPy processing before moving onto analysis tasks. The system uses SVM and Decision Trees as machine learning models to execute landing risk predictions. The deployed Flask API serves as a communication platform that connects different modules of the system. Users gain landing risk alerts through the ReactJS dashboard that shows the gathered results. A Unity3D simulation within the system uses predictions to develop risk scenarios that support pilot training for possible hard landings..

**Dataset**

The dataset that we have collected contains the features of approximately 10000 people with an existing criminal record and the following features:

- ID: Every inmate has a unique ID.
- Name: The first and last name of the inmates.
- Hair: The hair colour of the inmates.
- Sex: The gender of the inmates.
- Eyes: The eye colour of the inmates.
- Race: The race of the inmates.
- Height: The height of the inmates.
- Location: The inmates last found location.

**Preprocessing**

Raw data originating from aircraft specifications and weather conditions and flight performance metrics requires preprocessing because it needs cleaning and structure definition and transformation for efficient application in machine learning models. The steps involved in this are:

- Data Collection:The first step involves obtaining flight and weather information alongside aircraft specifications which will be stored inside a single database platform.

- Data Loading: The required database data will be retrieved through sqlite3 and pandas into a structured pandas DataFrame during the data loading phase.
- Feature Selection: The system will keep five fundamental columns namely Weather, Height, Speed, Wind, and DistanceToGround during the feature selection step.
- Data Cleaning : The process of cleaning data includes handling unconfirmed entries and removing redundant data before fixing inconsistent data formats found across all dataset elements.
- Data Transformation: The standardized string data should operate with lowercase alphabetical representation while categorical Weather data needs encoding.
- Normalization: All numerical features require normalization through standardization methods such as Min-Max scaling or standardization to achieve uniformity in the data.
- Data Splitting: The organized data will be split into training and testing subsets which will serve as input for training SVM and Decision Tree models (off-line mode).
- Model Input: The preprocessed dataset will be used as model input for prediction of hard landing risks using machine learning models.

**Feature Extraction**

At the feature extraction step, the methods used are:

- Identification of Relevant Features: Multiple critical factors that lead to hard landings should be selected as parameters for analysis including Weather conditions combined with Height and Speed and also Wind conditions and DistanceToGround.
- Extraction from Raw Data: The database stored raw data enables retrieval of these features through SQL queries executed using the sqlite3 interface.
- Transformation of Categorical Data: The process of categorical data conversion requires an encoding approach for turning inputs such as Weather conditions (storm or rain) into numerical values.
- Validation of Feature Relevance: The relevance of each feature should be validated through tracking correlations between target variables and output (hard landing risk) by analyzing correlation matrices and measuring feature importance from models.
- Final Feature Set Preparation: The last operation combines validated transformed

features into a distinct dataset for prediction model training and testing purposes.

**Model Selection**

The model selection process aims to choose integrated predictive models which successfully achieve project goals by determining hard landings and activating immediate pilot alerts for improved decision processes. The chosen models are engineered for operational integration and combination because they perform reliably while providing understandable results. The following models and tools are selected:

- Support Vector Machines(SVM): Support Vector Machine (SVM) demonstrates strength in analyzing large data features while doing binary classifications thus it helps determine whether landings are at risk through variables including speed and wind conditions..
- Decision Tree Classifier: Safety-critical cockpit applications rely on Decision Tree classifiers due to its efficient interpretation methods together with its non-linear pattern detection and clear output results.
- Threshold-Based Rules: Rules based on threshold values maintain relationship between speed higher than 160 km/h and risk designation for screening purposes alongside machine learning models particularly in contexts with limited data.
- Simulation Module: IThe simulation module built within the Unity3D platform uses predictive information to display visual designs which improve both training sessions for pilots and awareness of their situations.
- Integration: The backend pipeline integrates all selected models to receive preprocessing inputs which are distributed equally to the two ML models. The system produces risk alerts through post-analysis combined with simulation feedback and visual/audio warnings used for immediate pilot support.

**Model Training and Evaluation**

The model training objective during this phase focuses on establishing effective performance of machine learning algorithms which detect landing risks accurately by processing appropriate flight data. It includes the following steps:

- Training the Support Vector Machine (SVM):

The Support Vector Machine (SVM) receives training as part of its development process.

Flight data provided to the system includes weather conditions together with altitude measures along with wind speed and distance to ground measurements and aircraft speed parameters that have been preprocessed.

Through the training process the SVM model obtains ability to distinguish between normal and hard landing events by identifying its best decision separation within the feature dimensions. The training process reduces errors in classification behavior particularly when dealing with situations that occur at the boundaries.

- Training the Decision Tree Classifier: The input data contains the same dataset which features clear distinction points between data segments. Through its hierarchical decision rules the Decision Tree conducts interpretation regarding hard landing conditions that lead to accidents. The model provides straightforward information about feature relevance which makes it suitable for safety system verification procedures..
- Evaluation Metrics:  The model determines correct outcome classification rates through this metric.The model performs effectively at reducing two types of misclassifications: it prevents marking safe landings as risky (precision) and it avoids failing to detect risky landings (recall).

F1-Score becomes an essential single performance metric for imbalanced datasets because it combines precision with recall effectively.

The Confusion Matrix serves to represent the dual classification methods through graphical representation of correct and incorrect predictions.

- Simulation Testing:: The simulation provides real-world landing scenarios through Unity3D for training pilots based on model predictions. The simulation environment lets the model accuracy verification take place across real-time operating scenarios while also improving the learning feedback mechanisms.
- Iterative Refinement: The evaluation results and pilot feedback are used for reviewing misclassifications that occur during testing. New data and updated information is reintroduced to the training loop for increasing accuracy and improving adaptability to different flight conditions.
- Integration and Deployment: The system incorporates trained models into operational hardware modules for cockpit application that operates across real-time. Operational condition performance receives continuous monitoring while retraining sessions become necessary through updates in flight data or adapted aviation standards.

**Results**

Several performance indicators are employed for benchmarking the results obtained from the utilized models within the project framework. These metrics include:

- Classification Accuracy: The percentage of accurate model predictions for hard landing events forms the basis of classification accuracy measurements on the dataset.
- Precision:The precision value calculates how many actual hard landing predictions were accurate thus lowering the number of false alarms during flight monitoring operations.
- Recall: The recall evaluation technique determines the ability of systems to identify all hard landing scenarios while keeping risky conditions from going undetected.
- F1 Score: F1 Score calculates the harmonized precision-recall balance suitable for handling flight data sets with rare hard landing events.
- Confusion Matrix: A Confusion Matrix provides comprehensive information about true positives, false positives, true negatives and false negatives so users can see the prediction quality in detail.
- ROC-AUC Score: ROC-AUC Score measures the unified classification strength of all models across multiple threshold levels to determine their reliability.
- Training and Testing Time: The assessment of training and testing periods demonstrates the time performance abilities between the SVM model and Decision Trees that matter during cockpit real-time usage.
- Simulation Validation: The model predictions underwent simulation testing within the Unity3D platform which verified predictive alerts during landing but also tested visual results necessary for pilot educational purposes.

**Comparison**

The new framework improves upon existing hard landing prediction systems because it uses a unified platform of Support Vector Machines and Decision Trees to generate precise results along with interpreted output. The proposed system produces faster predictions than traditional LSTM-based models while requiring basic features about flight speed and height with wind conditions. Additionally it delivers improved generalization to the predictions. Such a system delivers cockpit alerts and pilot training simulations based on Unity3D framework to make it more usable in operational contexts. The proposed solution provides enhanced operational efficiency and user-friendly features for interpretability which makes it suitable for use in commercial aviation.

### 3.2.2 Connectivity Diagram

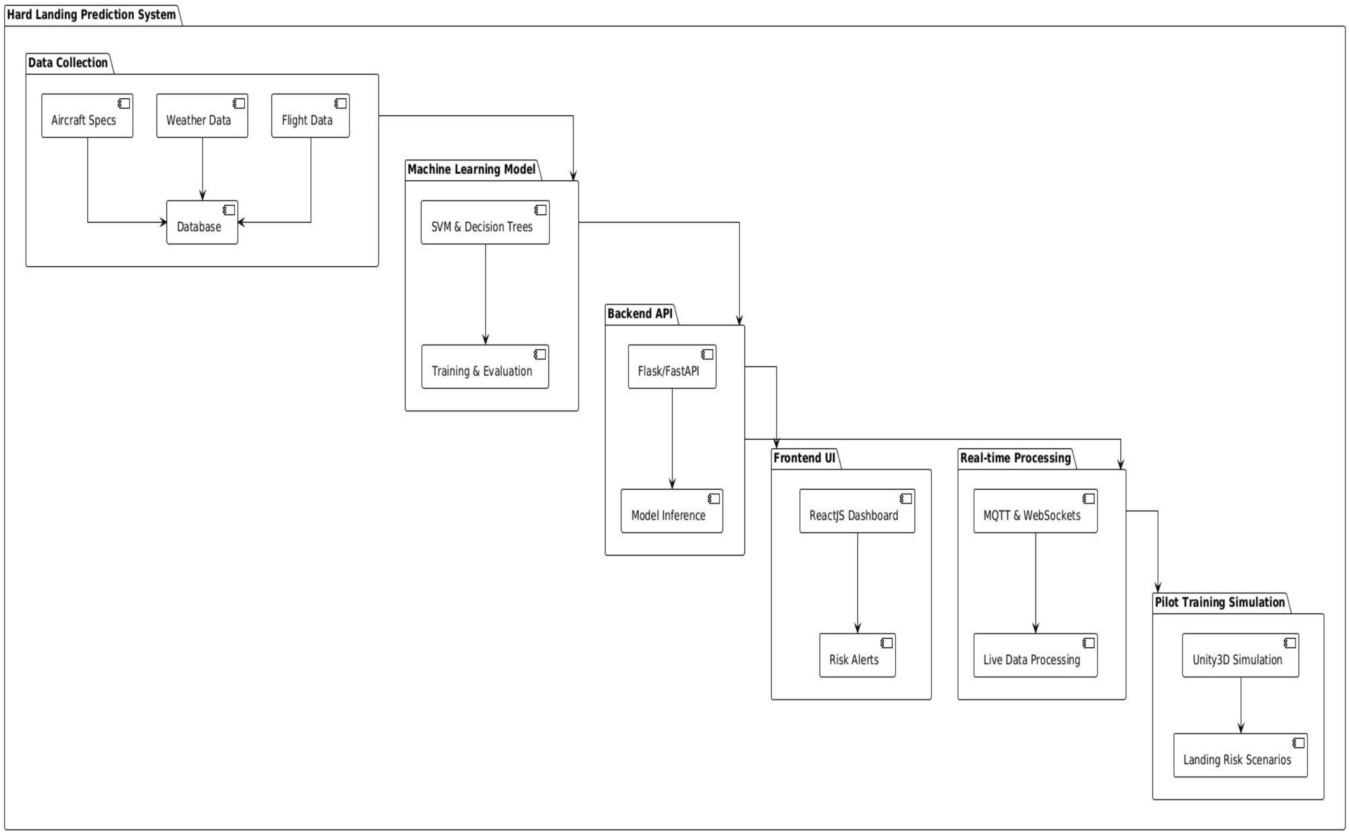


Figure 3.1 Connectivity Diagram

The Hard Landing Prediction System connectivity diagram demonstrates the entire operational structure which starts with data acquisition before reaching live risk alerts with accompanying pilot educational content. Starting operations the system accumulates aircraft specifications together with flight data alongside weather information that directs these elements to a central database for storage. The machine learning models including Support Vector Machines and Decision Trees use this gathered dataset for training and evaluation tasks that aim to predict hard landings during aircraft descent.

The trained models become available through deployment systems built with Flask and FastAPI for their inference capabilities. Real-time risk alerts appear on the ReactJS dashboard because this frontend UI connects to the backend which receives flight information from the central database. The system utilizes the MQTT and WebSockets protocols to process real-time data for delivering dynamic and fast updates. The system uses a Unity3D-based pilot training simulation platform which helps produce a realistic hard landing risk scenarios through the model's insights to improve pilot readiness during difficult conditions.

### 3.2.3 Software and Hardware Requirements

**Software**

- Python 3.8 or above with libraries
- TensorFlow
- PyTorch
- Scikit-Learn

**Hardware**

- Intel i7 or AMD Ryzen 7
- NVIDIA RTX 3060 or higher
- 16GB DDR4 3200MHz RAM
- Aircraft avionics integration support

**3.3 Modules and its Description**

The system depends on diverse flight metrics and environmental information to make hard landing predictions and simulation-based preparation for pilots. The combination of system modules operates towards achieving the final goal as explained below.

1. Input

The system starts its process by obtaining essential information from three main sources which include aircraft specifications along with weather reports and flight trajectory records. The system consolidates different data sets which it stores in one centralized information repository. The foundational data which the system requires exists after this initial input.

The system generates data inputs which become available for analysis preparation.

2. Preprocessing and Feature Engineering

Panda and NumPy tools process the collected data to manage missing values and standardize file types while performing value normalization. The subsequent phase of feature engineering extracts specific metrics which include vertical descent rate along with aircraft weight and wind speed and altitude variations.

The cleaning and structuring process on datasets with engineered features forms the basis for model training purposes.

3. Model Training (SVM and Decision Trees)

The training process for SVM and Decision Trees uses historical flight and weather data from existing records. The models acquire abilities to detect essential sequences which accompany hard landing occurrences. A model training phase involves adjusting parameters and conducting validation tests in order to improve the accuracy levels.

The deployment happens when enough prediction accuracy criteria are met but model retraining begins with modified parameters when those criteria are not achieved.

4. Backend API Module

The backend system utilizes Flask or FastAPI to operate the trained model by providing API service. incoming requests receive processing along with current data that enables the module to generate predictions from the trained model.

Through this system the inference results from real-time operations become visible to connected interfaces.

5. Frontend UI Module

The dashboard frontend displays processed results through the ReactJS framework while showing predictions about landing risks with corresponding alerts in an easy-to-understand manner. Staff members including ground personnel and pilots gain visibility into predicted results together with historical trend data.

Visual real-time risk alerts become accessible through the system output.

6. Real-time Data Processing

The system obtains continuous live flight and weather data through the combination of MQTT and WebSockets. The module extends performance security through real-time updates using recent operational conditions.

Output: Streaming analysis of ongoing flight events.

Output: The system enhances the sketch to help experts view it without difficulty.

7. Pilot Training Simulation

Unity3D simulation becomes active when the model detects dangerous situations during operation. The system develops life-like scenarios for potential hard landing situations which pilot instructors use to train flight response behavior.

The training sessions would be interactive with predicted risk profiles along with real flight parameters.

The system operates through modifiable workflows to perform immediate risk assessments of hard landings and presents preparation opportunities to pilots using simulation exercises. A highly effective predictive safety system requires every system component from data ingestion through simulation to become operational.

## 3.4 Requirements Engineering

**Functional**

1. Data Acquisition and Integration

The process of gathering information from various sources constitutes a fundamental operational requirement of the system. The system must have the ability to retrieve aircraft specifications that include model information alongside weight and configuration parameters as well as weather data comprising wind speed along with turbulence and temperature elements and real-time flight data combining altitude and descent rate and GPS data. The system relies on these inputs as its primary source for carrying out analytical operations. The system should provide:

- The system contains automated data collection features through APIs and file upload interfaces that regularly fetch weather and flight as well as aircraft information.
- Data Synchronization Procedures must realign different datasets in time to provide contextually correct inputs for each prediction.
- The system implements efficient database management through either relational or NoSQL database structures to handle storage and retrieval of data.

2. Data Preprocessing and Feature Engineering

The data cleaning process utilizes Pandas and NumPy libraries to normalize raw information collected from the database. The preprocessing includes data value imputation or removal followed by outlier detection and management together with numerical encoding of categorical variables (e.g. weather descriptions). Engineers specialize the key elements of the database for detecting subtle risk indicators by incorporating vertical descent rate and wind shear intensity and distance-to-ground measurements. The processing methods turn diverse data elements into one unified set which optimizes model performance potential.

3.Machine Learning Model Training

Support Vector Machines and Decision Trees represent the main predictive models that receive training through labeled datasets featuring landings labeled as “normal” or “hard.”

The training process requires splitting the data into three sections for training, validation, and testing purposes while performingsvm kernel and tree depth selection and conducting multiple k-fold cross-validation tests for reliability purposes. The model evaluation process relies on accuracy alongside precision and recall metrics as well as F1-score and confusion matrices to ensure its ability to handle various flight conditions.

4. Real-Time Risk Prediction

The system operates in operational mode by receiving flight and weather data through MQTT or WebSockets before running the trained models to produce risk scores that generate minimal latency. System alerts are activated through a specified threshold detection which indicates higher than desired hard landing probability. The system involves continuous monitoring because it needs to assess each incoming data packet thereby providing timely go-around suggestions during the critical approach phase.

5. Risk Alerting and Notification System

The system produces transparent warning messages which maintain three levels of alert severity after threshold violations. The frontend dashboard displays alerts and maintains a record of all events which serves for post-flight inspection. Users have the ability to adjust system sensitivity levels while they can also activate alert filters based on flight status or timing or danger seriousness. A system log maintains historical records which provide auditing capabilities while incident thresholds progress with periodic review.

6. User Interface (Frontend Dashboard)

The dashboard of ReactJS consists of flight parameter visualization and risk predictions shown through interactive screens. The pilot and ground staff have access to present-time information about altitude statistics as well as speed data alongside wind data which includes risk alert notifications that show flight trajectory replays for risk development patterns. Staff members need appropriate authorizations through role-based access in order to see restricted content along with dashboard editing permissions.

7. Backend API System

The backend system implements model inference and data retrieval functions by using Flask or FastAPI to provide RESTful endpoints. System health checks operate from this API framework. The system can maintain data protection through cryptographic token authentication with communication encryption. The architectural design of the API enables efficient horizontal scale-up when flight volumes rise while upholding performance standards.

8. Pilot Simulation Module

The platform develops virtual simulation tools through Unity3D which duplicates landing conditions based on model-based computational results. Through the system platform pilots conduct training exercises that involve go-around procedures including descent adjustments under various weather conditions and descent states. The reaction time and landing smoothness both serve as performance analyzing data to generate specific personalized training guidelines through the system.

9. Logging and Monitoring

TThe system utilizes comprehensive logging features to monitor prediction requests and model outputs and alert events at the same time and system monitoring tracks API performance statistics such as latency and memory consumption and error rates. Dashboard interactions get recorded through user activity logs to fulfill compliance requirements and support debugging procedures. System reliability and continuous improvement processes receive support from these logging activities.

10. System Integration and Deployment

The design utilizes independent microservices which makes each component updateable and scalablable independently. The system uses automated processes to execute tests together with deployment functions. The system operates in different deployment configurations which support on-site servers and cloud services and mixed operation modes suitable for multiple airline IT structures.

**Non-Functional**

1.Performance:

The system needs to handle incoming flight and weather data quickly so risk predictions can be delivered instantaneously. Model evaluation processes and alert productions together with data collection and assessment must finish their workflow in timeframes ranging from milliseconds to several seconds based on flight input frequency. API processing alongside message queues (MQTT/WebSockets) reaches maximum throughput speed yet the dashboard interface shows instant updates to users.

2.Usability:

The ReactJS dashboard as well as simulation interface delivers both simplicity and user-friendly functionality. The user interface utilizes clear graphs as well as color-coded warning systems while alert windows display brief recommendations to the user. The training modules offer basic operations that enable pilots along with trainers to prioritize decision-making as they avoid spending time learning software interface navigation.

3.Scalability:

User numbers increase together with data volumes so the framework needs horizontal scaling expansion to support rising flight operations. A containerized platform separates services which include preprocessing functions from model inference functions and simulation processes. The systems contain automatic duplicating mechanisms which respond to incoming usage needs. The system maintains constant performance efficiency throughout every level because it uses integrated autoscaling features with built-in load-balancing capabilities.

4.Reliability and Availability:

Flight safety depends on the continuous operation of aircraft flights so their cessation is essential. Protecting system uptime reaches at least 99.9% through health checks which combine autonomous failover systems and independent server or cloud region deployments. Backup systems and recovery protocols protect transactional databases used for permanent data storage against loss.

5. Potability:

This system operates effectively through various infrastructures which include in-house servers as well as private cloud and public cloud platforms. Consistent deployment becomes possible through Docker and Kubernetes manifests which decreases IT integration costs for diverse airline systems.

6.Security:

Security measures need to defend flight and aircraft data while it moves from system to system and when stored at rest. The API endpoints need authentication tokens for access while all communications require TLS encryption. Every action performed by users in the dashboard is monitored through audit logs to satisfy aviation data compliance requirements and access control functions operate via role-based restrictions.

7.Maintainability:

A modular pattern was applied to every preprocessing and modeling task as well as frontend components by implementing well-defined interfaces and extensive unit testing. The system performs fast model updates and bug repairs and model retraining by using CI/CD pipelines and automated documentation methods. The versioning system enables system operators to return to former deployments when new changes produce system malfunctions.

8. Compliance and Standards:

All development and operations adhere to relevant aviation and software engineering standards (e.g., DO-178C for airborne systems, ISO 27001 for information security). The system follows privacy requirements while performing compliant data processing through regulatory rules and model verification procedures which provide explainable and trackable requirements for compliance purposes.

These non-functional requirements ensure that the system is robust, efficient, and smoothly integrates with urban traffic.

## 3.5 Analysis and Design through UML

### 3.5.1 Class Diagram

Figure 3.2 demonstrates the overall Hard Landing Prediction System that combines aircraft specifications with weather data and flight telemetry for touchdown event prediction and flight support. DataCollector functions as the system entry point through which it acquires and time-synchronizes raw inputs that include aircraft data along with meteorological readings and flight metrics which get stored in a unified database. The Preprocessor completes two operations before generating a ready format dataset for modeling: it standardizes data while handling empty values and generates new calculated features like descent speed and wind shear measurement strength.

The ModelTrainer class trains Support Vector Machines as a robust classifier together with Decision Trees as a model that generates transparent rule-based decisions through the provided features. After training the RiskPredictor utilizes the established models to process real-time data for approach risk scoring. The system automatically activates AlertSystem visual dashboard cues and audio warnings as well as SimulationModule starts a Unity3D simulation when the risk score reaches its predefined threshold.

The Dashboard class utilizes ReactJS to display active risk indicators and historical trends while BackendAPI class enables dashboard and external systems to retrieve predictions through RESTful endpoints which include getPrediction(). These modules operate as an automated closed-loop sequence which begins with data selection and continues with algorithm prediction before warning alerts and ends with virtual training. This structure enables constant development of safe landing practices.

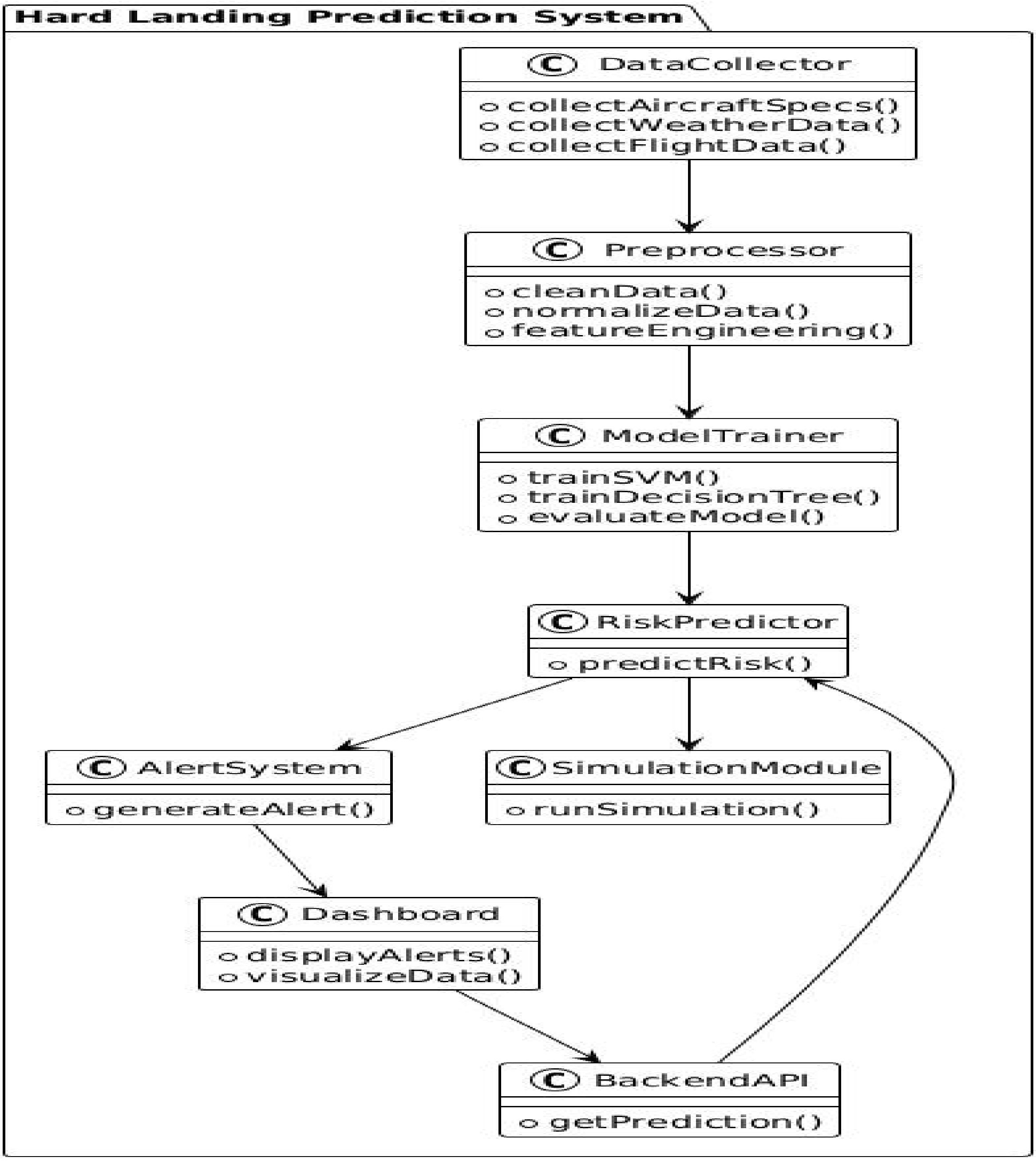


Figure 3.2 Class Diagram

### 3.5.2 Sequence Diagram

The system components communicate through the following steps as shown in Figure 3.3 during pilot approach status checks. A pilot starts the procedure by using the Dashboard to request risk prediction. A `requestPrediction()` function in the Dashboard sends data to the BackendAPI before the function calls `fetchData()` from the DataCollector. The DataCollector acquires unprocessed flight and weather and aircraft information that it hands over to both the Preprocessor and Preprocessor. At this point, preprocessedData enters the RiskPredictor whose predictRisk() execution generates a riskScore output. When executed the BackendAPI returns this score to Dashboard through the `displayRisk()` function.

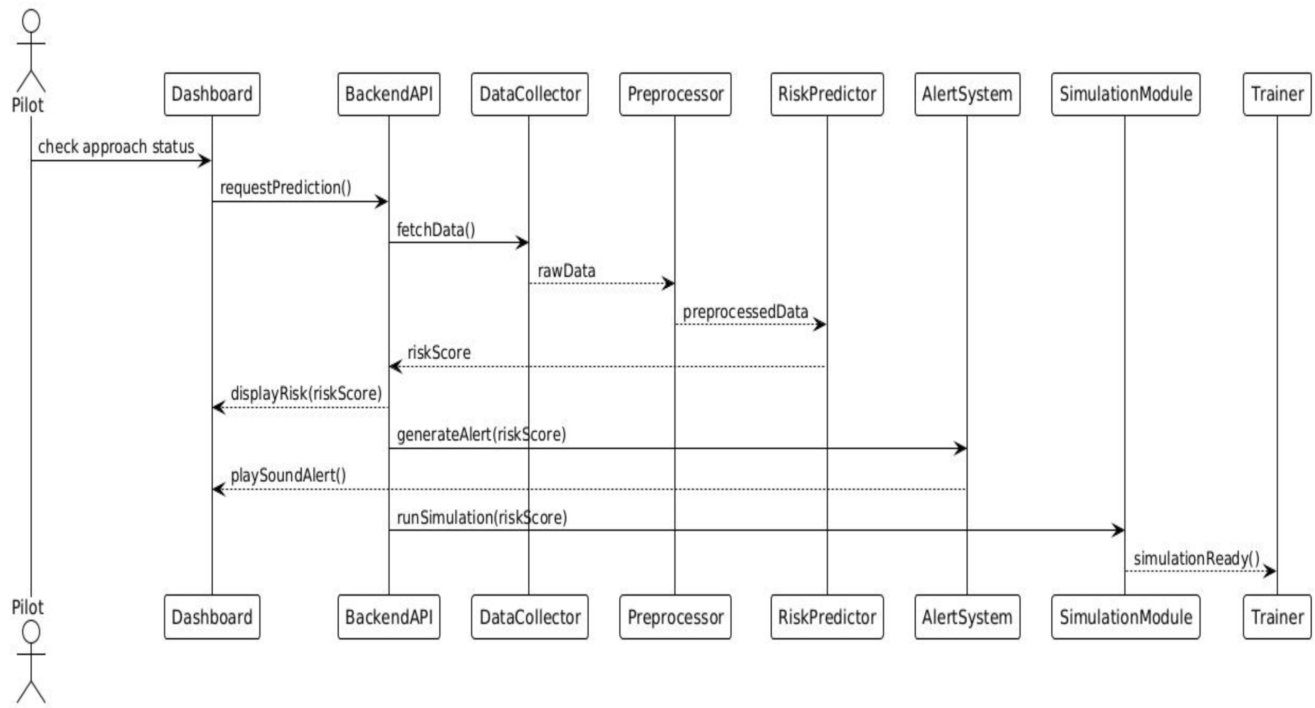


Figure 3.3 Sequence Diagram

The BackendAPI activates the AlertSystem with `generateAlert(riskScore)` after receiving the risk score. The AlertSystem makes Dashboard receive a `playSoundAlert()` command that generates an instant acoustic alert. The BackendAPI sends the riskScore value to SimulationModule in order to initiate the `runSimulation(riskScore)` operation. The Trainer receives a notification through `simulationReady()` to initiate training after the simulation preparation is complete. The endpoint connection allows for instant risk evaluation and warning systems as well as immersive instruction that uses current flight information.

### 3.5.3 Use case diagram

Figure 3.4 depicts the primary system user connections in the Hard Landing Prediction System with Pilot and Trainer and System Administrator acting as main participants and revealing their user communication workflow.

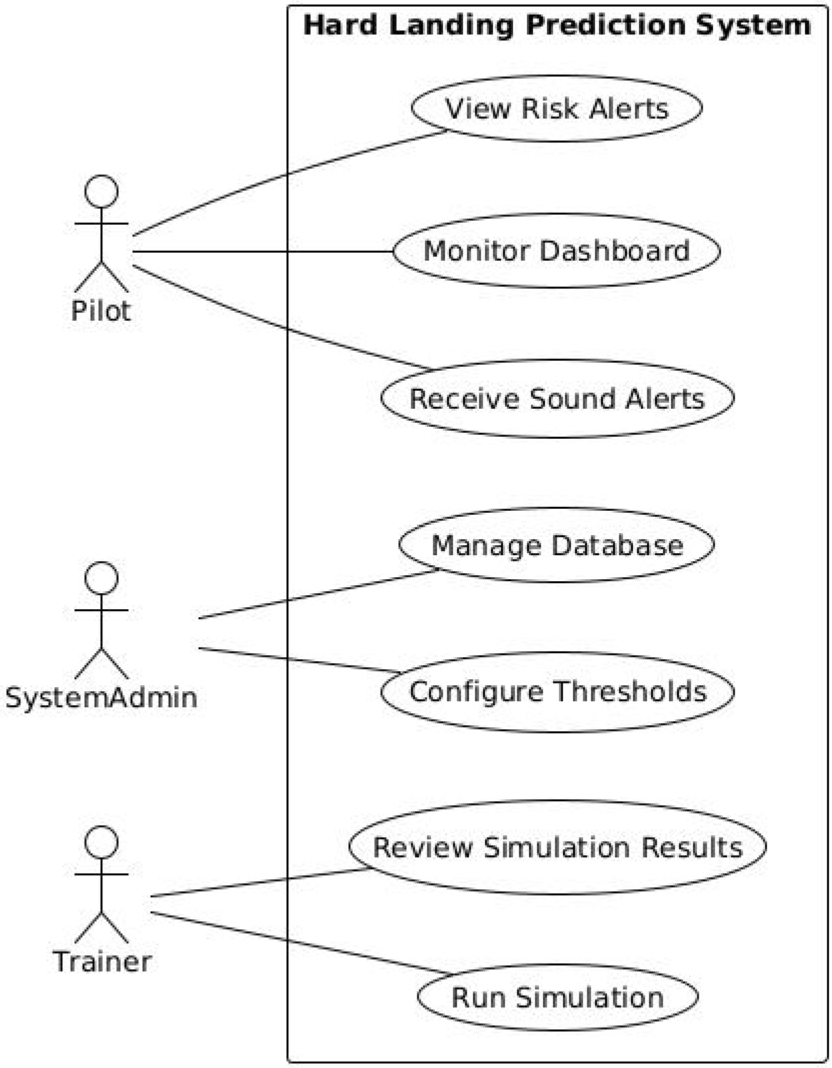


Figure 3.4 Use case Diagram

Flight procedures connect directly to the system by using the Pilot interface. Flight parameters along with predicted risk assessments and meteorological data appear simultaneously in the dashboard utility called Pilots Monitor Dashboard. The dashboard allows users to check system warnings which aid them in assessing dangerous flight conditions thus enabling proper and speedy go-around decisions. Sound Notification functions at the cockpit dashboard display same alert warnings to improve pilot awareness without breaking their visual focus on gliders.

The System Administrator assumes endpoint control responsibilities that aid both network infrastructure and precise predictive computational functions. The System Administrator maintains database control by running operations on aircraft specifications and flight logs and weather data management activities. System administrators implement sensitivity thresholds by following operational guidelines through descent rate limits and wind speed threshold parameters that pertain to individual aircraft models. Each specific framework configuration within the system handles both system risk assessments and warning production functionalities.

Real-world training flight scenarios in the Trainer's operations depend on the system's simulated features for flight readiness preparation of pilots. When using Simulation sessions the Trainers execute high-risk landing profile simulations with Unity3D simulation module through storing both real-world and live data. The trainers evaluate simulation results using flight data measurements to determine which aspects need improvement. They monitor every flight response to measure reaction time and detect go-around success. The feedback processes between instructors and trainees help the learning session connect with existing risk elements to improve proficiency levels.

The operational safety standard is reached by executing these use cases which combine with administrative control and continuous training sessions.

### 3.5.4 Activity Diagram

The Hard Landing Prediction System operates using Figure 3.5's two-step workflow that starts in Model Preparation mode. Data collection from various sources determines aircraft specifications and weather feeds and historical flight records and their storage within a single database. листопrocéssní aktivita Processed and normalized raw data by removing missing information and abnormal readings for maintaining consistent data quality. Feature engineering produces essential metrics from the system that include vertical descent rate together with wind shear intensity and runway distance measurements. The engineered attributes from the preprocessing stage become input for the Support Vector Machine and Decision Tree classifiers to establish criteria determining safe vs hard landing conditions. Model evaluation through accuracy testing and precision check combined with recall assessment and F1-score measurement establishes their reliability after completion of training. The deploy model action will publishing the operational models to the live inference environment following their achievement of performance targets.

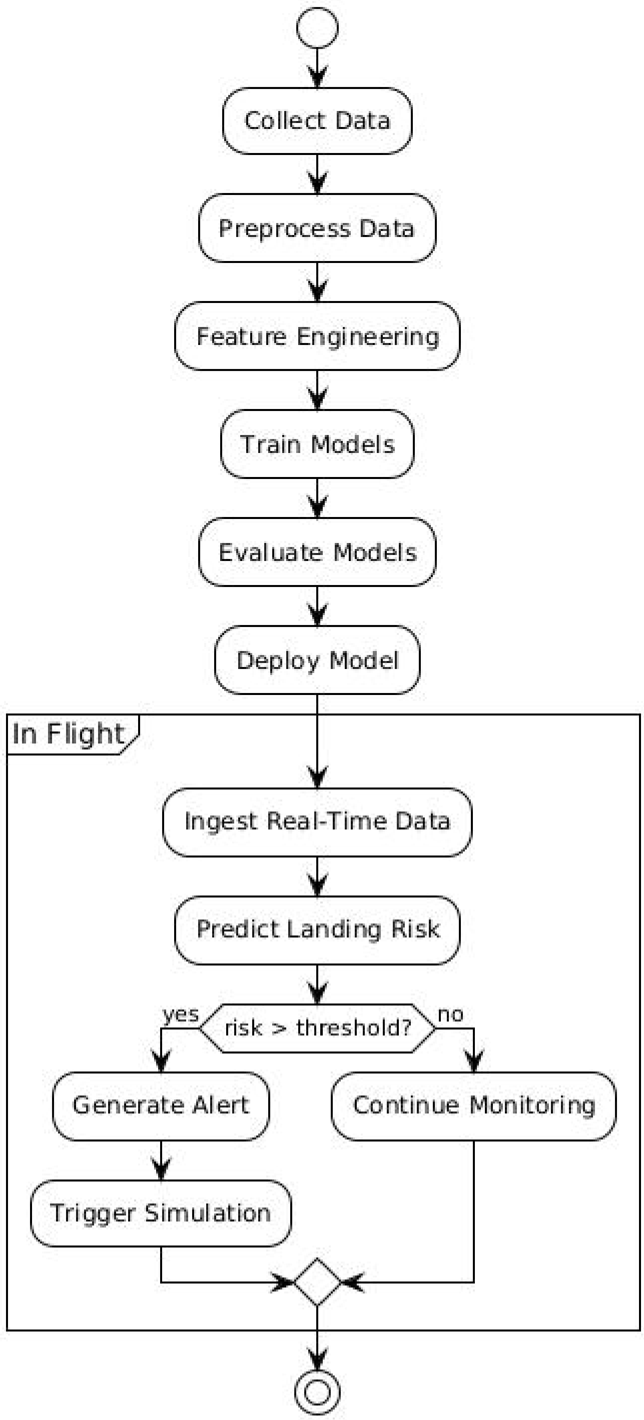


Figure 3.5 Activity Diagram

When an airplane reaches its approach point the In-Flight Operation activation process begins. The system remains active by taking in flight data and weather status from low-latency data streams including MQTT or WebSockets. The deployment of data packets leads to running a predict landing risk operation which includes risk model calculations. The assessment score from the prediction model triggers the system to check against a defined threshold through a decision point. The system displays alerts both as cockpit dashboard notifications alongside audible cockpit warnings when the score surpasses the threshold value. Simultaneously this will activate a simulation module for pilot training within Unity3D which duplicates the scenario. The system will keep monitoring operations without any interruptions whenever the risk score stays under the predefined threshold.

The system maintains ongoing connections between entry functions for data and decision-making tools which may incorporate risk-detection simulation while enabling ongoing improvement processes. The system retains simulated alerts with other data entries in its training dataset until future threshold modifications and periodic training sessions. The safety alerts generated by immersive simulation cooperate with predictive analytics to both deliver accurate training that benefits safety outcomes and strengthens decision processes related to vital landing procedures.

# CHAPTER 4

# RESULTS AND DISCUSSIONS

## 4.1 Description about Dataset

The custom hard-landing prediction dataset contains more than five thousand approach records which include eighteen specific characteristics that describe diverse operational conditions leading up to and extending throughout the landing phase. It serves as a tool for machine learning models to receive detailed high-quality inputs which help them identify hard landings and prompt go-around procedures. The dataset maintains equilibrium between environmental elements and aircraft movement patterns and control parameters and runway characteristics together with predictive outcome tags for learning real-flight conditions by both threshold-based systems and data-science models. The following analysis details every attribute category with an inspection of essential metrics, standard operational ranges and review methods for data preparation and analysis operations.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | timestamp | aircraft_typ | weather | runway_ler | visibility_mi | wind_speed | wind_direct | crosswind_ | approach_s | descent_rat | altitude_ft | glideslope_ | localizer_de | pitch_degre | roll_degree | flap_setting | gear_positi |
| 2 | ######### | Embraer E1 | Foggy | 8623.19625 | 2.80123877 | 4.25666452 | 355.104817 | 23.2245603 | 159.553898 | 788.898697 | 369.742339 | 0.51448377 | 0.95308541 | 8.63475051 | 1.70421242 | 35 | Down |
| 3 | ######### | Boeing 787 | Stormy | 5871.03945 | 2.81176689 | 27.0661840 | 93.5218152 | 0.77514648 | 130.029155 | 585.407659 | 311.738532 | -0.328551 | 1.96937876 | -2.771901 | -3.371557 | 40 | Down |
| 4 | ######### | Bombardier | Foggy | 11735.3711 | 8.83605236 | 2.37150489 | 149.437911 | 21.3039052 | 122.702346 | 832.573396 | 374.173487 | -1.808352 | 1.08380263 | -0.951818 | -4.966521 | 15 | Down |
| 5 | ######### | Boeing 787 | Cloudy | 7029.47333 | 2.39066807 | 31.7739150 | 244.525293 | 9.93402244 | 133.198598 | 820.125206 | 795.145599 | 0.38912090 | -1.484266 | 3.37968133 | 7.85193889 | 40 | Down |
| 6 | ######### | Boeing 787 | Stormy | 10970.6094 | 7.97612690 | 9.85775389 | 314.490343 | 14.0400409 | 158.760130 | 793.913416 | 357.734791 | -0.596991 | 1.00187497 | 0.38500508 | -5.407593 | 30 | Down |
| 7 | ######### | Airbus A32( | Windy | 7713.37237 | 4.88716207 | 31.4607664 | 18.4797705 | 12.8203074 | 151.717104 | 764.812506 | 921.372673 | 0.14950578 | 0.90651046 | 11.3339527 | -1.890288 | 40 | Down |
| 8 | ######### | Bombardier | Heavy Rain | 7262.35974 | 9.99136229 | 27.0727397 | 108.066063 | 23.1435065 | 139.213323 | 847.727422 | 112.962467 | 1.06576679 | -1.333057 | 7.46908173 | 8.98857944 | 30 | Down |
| 9 | ######### | Bombardier | Heavy Rain | 11546.1100 | 8.47136923 | 30.4920362 | 204.265919 | 19.2532317 | 132.213216 | 551.226498 | 931.576294 | 1.21496142 | 1.01269116 | 1.66617208 | -8.39386 | 35 | Down |
| 10 | ######### | Bombardier | Windy | 7543.25800 | 7.52717977 | 7.49138921 | 221.974396 | 24.6205883 | 136.932243 | 526.806217 | 689.548005 | 1.45711295 | 0.69909038 | 1.35565896 | 8.76869386 | 15 | Down |
| 11 | ######### | Boeing 787 | Windy | 7732.01536 | 8.77189991 | 20.8492723 | 25.1398504 | 19.8607487 | 120.694624 | 544.772198 | 344.350925 | 0.54418650 | 1.97777600 | 8.67387444 | -2.195705 | 40 | Down |
| 12 | ######### | Embraer E1 | Windy | 5742.64245 | 6.52328872 | 3.27561962 | 238.404266 | 7.39721235 | 148.734819 | 639.998936 | 6.85522327 | 1.62247157 | 0.47495828 | -1.586601 | 5.81855148 | 25 | Down |
| 13 | ######### | Bombardier | Light Rain | 9693.12665 | 8.51469643 | 7.07982686 | 189.575107 | 24.0720919 | 123.985770 | 694.649178 | 802.431622 | 0.76251203 | -0.341562 | 0.61091382 | -2.882061 | 25 | Down |
| 14 | ######### | Boeing 787 | Stormy | 7021.87590 | 2.98070504 | 10.8465779 | 297.118229 | 24.5891447 | 140.761503 | 618.620333 | 969.327462 | 1.13565637 | 1.81566017 | 2.06867509 | -1.688992 | 25 | Down |
| 15 | ######### | Airbus A32( | Heavy Rain | 9099.22917 | 1.42818050 | 25.9284407 | 79.7659142 | 3.07734259 | 142.633802 | 676.341699 | 982.439360 | 0.74192888 | 0.51136613 | 4.09291385 | -1.244428 | 30 | Down |
| 16 | ######### | Embraer E1 | Stormy | 9139.46192 | 6.35135406 | 0.86500327 | 70.0146876 | 22.5261045 | 120.067590 | 630.086177 | 423.405210 | 1.48745525 | -0.605472 | -1.189364 | 8.93616867 | 25 | Down |
| 17 | ######### | Airbus A32( | Foggy | 10034.3978 | 4.62955873 | 31.6091049 | 284.808062 | 16.6050545 | 159.195145 | 605.520281 | 70.5043955 | -1.10844 | -1.915817 | -1.85534 | 1.00508813 | 15 | Down |
| 18 | ######### | Embraer E1 | Heavy Rain | 10216.6904 | 4.55118755 | 2.47812297 | 252.689017 | 23.0498442 | 132.085561 | 705.087863 | 313.753785 | 1.03954830 | 1.02655827 | 14.7992016 | -5.662478 | 15 | Down |
| 19 | ######### | Boeing 787 | Light Rain | 8601.16704 | 5.50410459 | 13.1771568 | 284.210736 | 22.8934755 | 123.088970 | 778.258840 | 309.740230 | -1.531374 | -1.521895 | 13.2639604 | 2.23022034 | 40 | Down |
| 20 | ######### | Boeing 737 | Windy | 7316.94363 | 3.93313480 | 2.83813699 | 280.714356 | 22.2843257 | 154.175542 | 1020.98674 | 102.381241 | -1.632024 | 1.69467752 | 5.47236453 | 7.78942336 | 40 | Down |
| 21 | ######### | Embraer E1 | Cloudy | 10862.1562 | 6.44756554 | 26.9138745 | 37.3362426 | 22.1318898 | 126.078104 | 1168.88495 | 524.716597 | -0.033509 | 1.84556669 | 10.3110062 | -5.712381 | 25 | Down |

flight_data

Figure 4.1 Representation of Dataset

1. Environmental Conditions:Every approach becomes defined by a group of measurements that include Weather, Visibility Range, Wind Speed & Direction and Crosswind Component. The weather conditions are stored in METAR reports as categorical values including Clear, Rain, Fog and Thunderstorm which need one-hot encoding or embedding function for

modeling purposes. The measurement unit for Visibility Range operates between dense fog conditions at 0.25 miles while clear conditions can reach beyond 10 statute miles. Wind Speed measurements in knots and Direction in degrees are provided by airport anemometers which contain steady profiles as well as gusts that need processing before utilization. Trigonometry calculations against runway heading allow the evaluation of lateral wind impacts on stability during landing through the Crosswind Component. The combined features describe essential risk components through their ability to impede pilot reactions together with their capacity to cause lateral drift effects and gusting events.

2. Flight Kinematics: The four flight variables used to monitor aircraft movement in the final phase include Height of Flight and Descent Speed and Speed of Flight and Distance to Ground. Height of Flight that expresses elevation above sea level and Distance to Ground measured by a radio-altimeter above runway threshold are recorded simultaneously every second so a vertical and horizontal profile assessment becomes possible. The rate of descent speed (feet per minute) measures vertical descent rate during approaches while standard descent rates range between 500–700 fpm but steep approaches exceed this threshold to 1,000 fpm. The indicated speed and ground speed for flights during their final segment should be maintained between 130-160 knots. The flare moment together with aircraft touchdown occurs when Distance to Ground exhibits sharp drops during the last 50 feet of flight. Unstable approaches can often be signaled by sudden descent speed changes. Predominantly predictive models use kinematic data to indicate hard landings because excessive descent rates and excessive approach speeds appear together as strong indicators.

3. Aircraft Control Settings: The flight instruments consist of Pitch Angle, Flap Settings, Gear Position and Throttle Percentage which track both pilot operations and aircraft state. The aircraft pitch attitude relative to the horizon is measured in degrees through Pitch Angle and should maintain a range of +3° to +5° during final but becomes higher when flare is aggressive or wind shear occurs. The integer field known as Flap Settings (0–5) controls both lift and drag for wings while suitable flap sequence safeguards a stable approach path. The Gear Position must either remain in up or down position or transition mode while following the required steps as it needs to be down and locked before touching down. The absence of this sequence indicates a configuration problem. The engine thrust setting indicates through Throttle Percentage where the value ranges between 30–60% during the approach phase unless there is a go-around event which triggers sudden spikes. The collection of control variables allows the model to differentiate between standard approaches versus unstable ones

because incorrect throttling and flap configurations tend to produce hard landings.

4. Runway and Aircraft Context: The three data elements Aircraft Type create the background information alongside Runway Length and Timestamp data types. The Aircraft Type grouping includes different aircraft models including B737 together with A320 and CRJ700 which presents unique landing characteristics because massive aircraft need longer flare distances and display different crosswind behavior patterns. Flight Strip Length risks to runway safety increase as the lengths extend beyond 12,000 feet and from 5,000 feet. Timestamps (UTC) mark each data sample, enabling alignment across data streams and temporal feature creation (e.g., time-of-day, seasonal effects).

5. Outcome Labels and Derived Metrics: The target variables together with derived measurements consist of Landing Rate, Hard Landing and Risk Score in the dataset. The vertical speed measurement at touchdown defines Landing Rate in feet per second and hard landings occur when this speed exceeds a fixed limit (–3 m/s or –600 fpm). The binary Hard Landing flag (0 = normal, 1 = hard) serves as the primary classification target. The Risk Score presents a continuous model-based probability that shows the chance of a hard landing at each specific time interval (0-1 scale). These labels allow dual classification and regression analysis while maintaining the continuous Risk Score function as a threshold criteria for alerts.

6. Data Quality and Preprocessing Considerations: The majority of attributes distribute continuously except two categorical fields which need encoding schemes. The main causes of missing values stem from extreme weather situations or sensor malfunctions which are addressed by performing data interpolation for continuous variables while categorical data uses mode value imputation. The system uses Z-score thresholds to detect outliers having unrealistic descent rates before either clipping or removing such values. A normalization technique of Min-Max scaling or standardization allows steady convergence of the model by adjusting numerical feature data. The records require temporal alignment to maintain time compatibility between features and labels so data leakage can be prevented.

7. Feature Correlations and Engineering Opportunities: The predictive value of Descent Speed and Landing Rate together with Hard Landing becomes evident when we conduct exploratory analysis. The relationship between Crosswind Component and Pitch Angle produces a moderate level of association. Vertical acceleration derived from altitude second derivatives along with wind shear intensity calculated from time-dependent wind

speed changes increase model performance. Feature selection techniques such as PCA alongside other algorithms help decrease the feature number while maintaining maximum data variance.

8. Use in Model Training and Validation:

The dataset allows developers to create models usable during offline times and real-time situations. The testing of offline data uses split sets for training and validation while Aircraft Type and Weather stratified testing ensures proper representativeness. The combination of grid search for hyperparameter optimization runs parallel to cross-validation testing for generalized model assessment. The production models receive live stream data from this schema to trigger cockpit alerts upon Risk Score threshold crossing. The large set of attributes allows hybrid modeling by linking threshold rule systems with machine learning algorithms to boost prediction accuracy as well as interpretability of hard landing predictions.

This extensive dataset exceeding 5000 records contains environmental and kinematic, control and contextual factors data which enables necessary research to create predictive models that enhance flight safety parameters during critical landing operations.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1.8652635616577355 | 14.314637640384401 | -0.155963 | 30 | Down | 4237.753862363695 | 59.176186945459186 | -270.0915741 | 0 | 35.476281967835625 |
| 0.2873459894631045 | 5.046782155674068 | -4.034201 | 25 | Down | 73.80362365367709 | 54.854675567239624 | -558.5368447 | 1 | 99.43517030386502 |
| 0.5408971575317274 | 10.701704344996452 | -6.139027 | 15 | Down | 4468.216004291053 | 64.98767724719212 | -412.2513291 | 1 | 90.42206740995886 |
| 1.11497538702905 | 7.502438439832538 | 2.99861494 | 25 | In Transit | 1964.4682627637144 | 43.258434934347974 | -532.4028495 | 1 | 99.05107109972224 |
| 0.6472046063098666 | 7.33693962657971 | -1.274015 | 40 | In Transit | 3084.1500645119622 | 24.804521532405843 | -496.2486535 | 1 | 98.0639558240558 |
| 1.3932866860471065 | 10.775170484268086 | -3.6534 | 25 | Down | 4027.824027403381 | 36.57501397607565 | -186.3897365 | 0 | 9.345121838881434 |
| 1.536205802890219 | -3.691199544 | -9.807148 | 40 | Down | 2304.9417311433995 | 50.56246891541433 | -408.2989213 | 1 | 89.7152488601851 |
| 0.5108321580413109 | 2.8477975830348123 | -9.27206 | 30 | Down | 463.0553531807491 | 36.144786168827736 | -562.0512379 | 1 | 99.47330594106477 |
| 0.1322508899317496 | 14.391361903738648 | 3.87226260 | 25 | Down | 2708.4907753831553 | 51.647378523481564 | -484.1157885 | 1 | 97.54530823477037 |
| -0.435794953 | -3.908254388 | -0.389191 | 20 | Down | 607.8982132538702 | 22.181157266822048 | -475.4790613 | 1 | 97.09591581523085 |
| -0.153677352 | 0.25591609768790047 | 6.97780828 | 25 | Down | 4123.372447664339 | 23.51710117202125 | -653.1647975 | 1 | 99.91447797638547 |
| -0.618168209 | 8.442138860871058 | 5.07056938 | 20 | Down | 3239.4529036964104 | 20.52579587730801 | -227.7810603 | 0 | 19.08681830172391 |
| -0.292134683 | -2.991287568 | 1.46880913 | 35 | Up | 2960.720680481132 | 68.79794274244688 | -639.9313929 | 1 | 99.88859380635307 |
| 1.5988394264093637 | 5.733471391596579 | 1.21981754 | 35 | Down | 1.9381002214208243 | 33.08212256243174 | -757.9240716 | 1 | 99.98946883754073 |
| 1.8894875681500434 | 9.26740838587419 | 4.90689892 | 40 | Down | 4800.364043592935 | 59.77766567527704 | -656.5310339 | 1 | 99.92004172907596 |
| -0.553864569 | 10.14840759710057 | 4.82324638 | 30 | Down | 651.6581799410886 | 64.3422193389465 | -454.2111962 | 1 | 95.62372864615568 |
| -1.117639804 | 6.223791862462614 | -3.309651 | 15 | Down | 917.3098024284549 | 27.880466034131807 | -544.0786741 | 1 | 99.247203050473 |
| -1.687237679 | 5.012606182064619 | -7.893611 | 25 | Down | 4930.793517061759 | 29.956086506105827 | -599.9238465 | 1 | 99.75236173296237 |
| 1.3477238574864674 | 9.277774591296987 | 5.43656452 | 15 | Down | 447.65594070820924 | 39.117971171163255 | -362.8713447 | 0 | 77.8582844022205 |
| 1.4826640400860387 | -2.098807686 | -5.537881 | 20 | UP | 3708.2287257496305 | 45.20316895343018 | -767.2352675 | 1 | 99.99125805510785 |
| -0.198111543 | 8.610358398951288 | 5.82530164 | 15 | Down | 474.80943859461166 | 46.73711493269981 | -800 | 1 | 99.99546021312976 |

Figure 4.2 Representation of Data

## 4.2 Experimental Results

The experimental Hard Landing Prediction System proves its real-time capability to predict and display dangerous airplane landing conditions during execution. The first image displays an Aircraft Approach Visualization feature that shows aircraft flight paths in 3D mode while depicting an 88.2% hard landing risk indicator. The red zone on the gauge indicates the score reveals an aircraft landing situation rated as dangerous. The system warns operators with the instant notification through a red background text message reading "HIGH RISK (88.2%)". The system displays real-time information which improves both approach dynamic understanding and critical situations decision-making abilities.

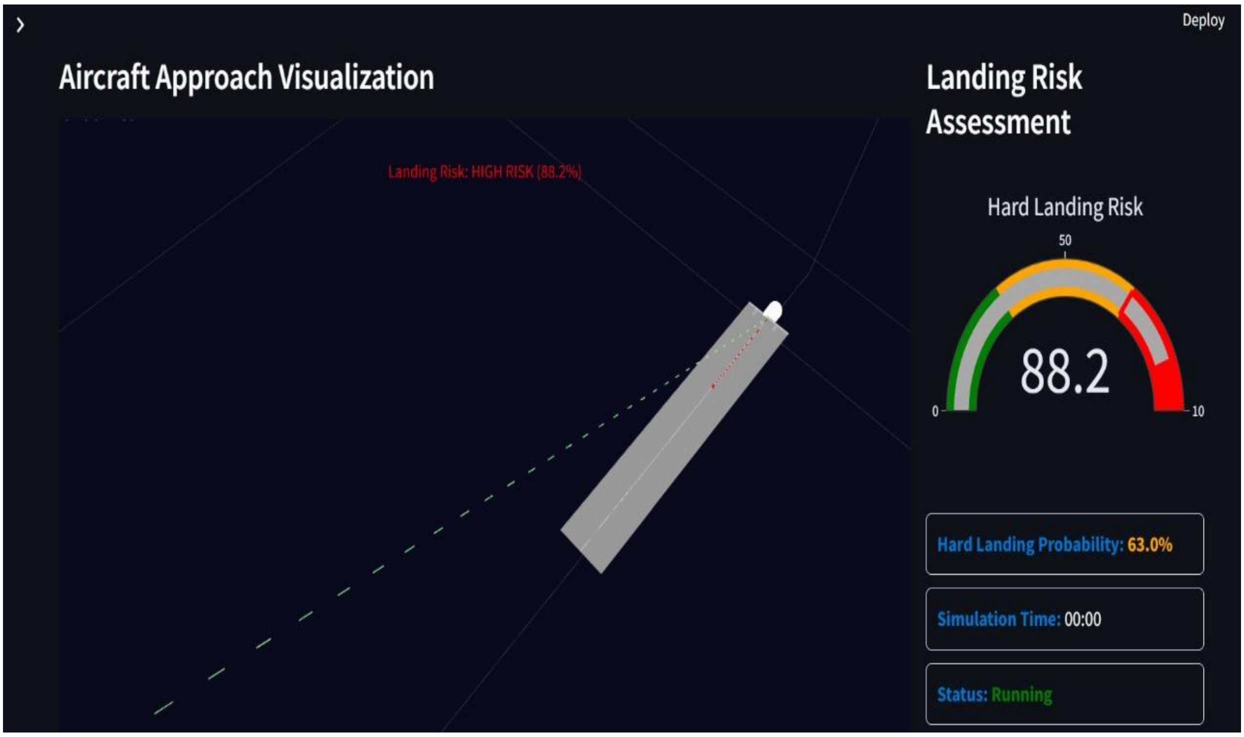


Figure 4.3 User Interface of Hard Landing Prediction

The dashboard displays both numerical data and visible indicators which show a 63.0% probability of a hard landing but do not guarantee it. The simulation time at "00:00" marks the start of an assessment cycle during which the running status confirms that monitoring proceeds in real-time. The system helps users obtain near-instant feedback about system responses and environmental condition awareness. The model's design objective includes finding optimal precision-recall ratios that trigger swift alertings while minimizing superfluous false alarms to users.


Figure 4.4 Representation of the Alert and Feedback

In the second illustration the alert system displays the predictive process transition into practical operational direction. Kỳ exists the system displays an urgent warning which reads "HIGH RISK OF HARD LANDING DETECTED! INITIATE GO-AROUND OR CORRECT APPROACH IMMEDIATELY!" An immediate go-around or approach correction sequence is required at this moment. A block of recommended actions guides the pilot to initiate an immediate go-around procedure while ensuring their safety. The right-hand display area shows both flight dynamics information including descent rate at 700 ft/min and approach speed of 140 kts and altitude at 400 ft while indicating distance to threshold at 1000 ft. The parameters serve as critical information to understand why the risk evaluation increased.

The hard landing warning seems to be triggered by flight dynamics different from lateral and vertical path performance because all measurements show perfect alignment (+0.0 dots) for both roll (0°) and pitch (2.5°) along with glide slope/localizer data. The system achieves high interpretability and transparency standards which enables pilots or analysts to easily follow and trust the risk assessments conducted by AI algorithms. The model performs comprehensively through multiple factors which evaluate complete flight operations beyond a single aspect.

The third depiction demonstrates the comprehensive features of the simulation platform through its selection options which contain different established scenarios targeted at "Calm Day Perfect Approach," "Crosswind Challenge" and "Low Visibility Approach" up to

"Unstable Approach" and "Heavy Rain with Gusts." The program allows researchers along with trainers and system testers to measure the model robustness through various environmental scenarios and approach conditions. Drilling between different scenarios allows users to verify that prediction attributes can generalize from controlled to challenging conditions.

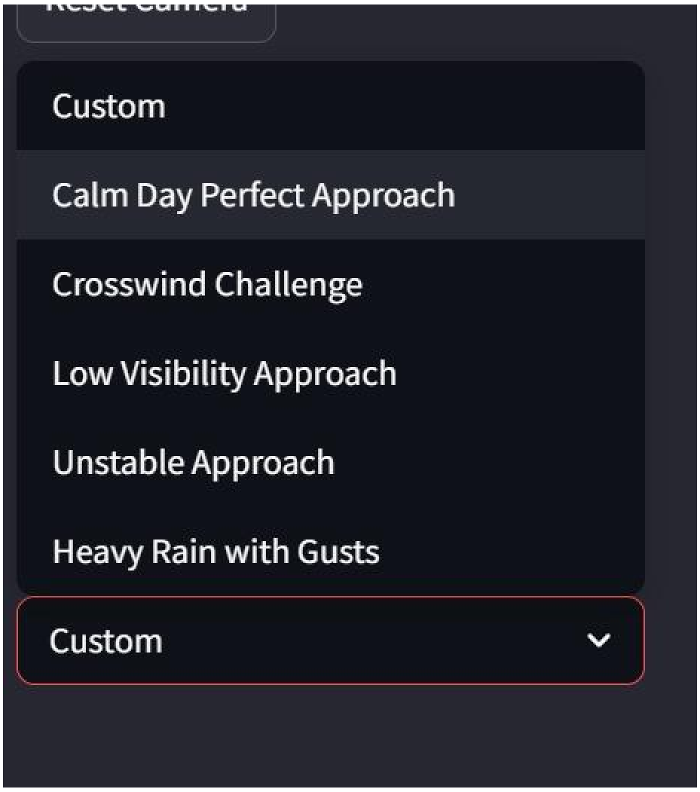


Figure 4.5 Advanced Feature Options

A combination of fast and accurate landing hazard detection with direct visual warning systems for pilots functions successfully based on laboratory testing. The system creates a uniform operational safety learning environment by uniting its control of simulated flight and its ability to monitor flight parameters and deliver feedback. The Hard Landing Prediction System proves useful for real-world implementation because of its data acquisition capabilities and interface design and learning model development processes.

## 4.3 Significance of the Proposed Method

The proposed Hard Landing Prediction System increases the aviation safety standards by combining the real-time data analysis with machine learning capabilities which are futrhter combined with a three-dimensional visualization. The system provides reliable landing assessment followed by timely alert notifications supported by operational pilot guidance. The real-time functioning of this system enhances flight situation awareness and provides improved capability for flight decisions during critical phases together with improved simulation testing of different environmental conditions to decrease hard landing occurrences.

**Advantages**

1. Enhanced Safety through Early Risk Detection: The system exhibits outstanding performance by identifying upcoming hard landing situations during operation. Multiple real-time measurement factors including descent speed and pitch orientation and wind speed factors and aircraft velocity enable the system to determine exact landing danger scores. Through proactive detection pilots gain an opportunity for timely corrective action and go-around procedures which prevents aircraft structural damage as well as reduces personnel injuries.

2. Data-Driven Decision Making: The use of machine learning algorithms that were trained with data from 5000 instances allows the reduction of landing risks. Such evaluation strategies from these models show relationships which operators with human judgment alone cannot discover. During critical moments of landing operations pilots receive enhanced numerical feedback using both risk scores and probability indicators with specific condition alerts which streamlines their performance through reduced human evaluation requirements.

3. Real-Time Monitoring and Visualization: During 3D Aircraft Approach Visualization operations pilots and trainers obtain real-time moving visualizations to display aircraft path direction along with their relative position to the runway. The graphical visualization system combines improved spatial recognition functions with improved real-time flight danger interpretation effectiveness. Visual direction helps in the increase of situational awareness by providing integrated audio alerts and text warnings that reduce workload during emergency situations.

4. Customizable Scenario-Based Simulations: Multiple predefined environment situations consisting of crosswind tests and low visibility conditions and unstable approaches are provided by the simulation module. The system enables users to perform tests that evaluate system performance within objectives based on different authentic scenarios. Through simulations trainers manage to execute high-risk flight conditions which provide complete training capabilities for pilots and system evaluation. This system permits adaptive testing through custom scenarios which extends its capability of working with different aircraft types

and conditions found at airports.

5. Improved Training and Operational Readiness: The technique allows trainers to check simulation outputs while evaluating prediction results thus creating an environment with high feedback potential. The proposed method helps find typical reasons for hard landings and allows assessment of pilot abilities and allows defining operational parameter settings. A wide range of simulation scenarios enables better readiness for unforeseen approach conditions for pilots.

6. Scalability and Adaptability: The systematic design features which make it possible to scale the system effectively between various aircraft models and at different airports together with reinforcing airline procedures. The solution operates flexibly for different flight operations because it considers aircraft type together with runway length and visibility range parameters. A higher prediction success rate and minimized false positive or false negative occurrences will result from continued model database expansion.

7. Support for System Administrators and Engineers: The system enables pilots to have assistance yet it allows system administrators to maintain database control and set alert threshold parameters. Ground teams receive the power needed to modify the system when creating setups for distinct aircraft profiles or mission types. The system maintains risk assessment effectiveness through operational context updates which helps to improve its operational resilience.

8. Augmented Operational Efficiency: Moreover the system creates additional operational benefits through its safety advantages. Airline financial expenses decrease alongside maintenance costs and inspection downtime when plane landings occur without harsh force. Well-executed landings enable better fuel efficiency and smoother passenger journeys so airlines achieve both present-day expenses reduction and superior service delivery.

The proposed method which unites technical advancements with practical capabilities delivers a complete real-time choice-supporting instrument that boosts flight safety standards together with pilot training quality and operational results.

\

# CHAPTER 5

## CONCLUSION AND FUTURE ENHANCEMENTS

### 5.1 Conclusion

The Hard Landing Prediction System offers new opportunities in aviation safety because it joined predictive analytics and real-time simulation technology controls with data management capabilities. Through machine learning capabilities the system examines flight parameters including descent rate and approach speed and winds and aircraft set-up to detect hard landing dangers before the event happens. Data processing through 5000 different points enables the model to adapt performance to diverse aircraft conditions as it operates in changing environmental circumstances. Prior notification allows pilots to develop well-timed decisions based on complete information thus enhancing safety within operations while reducing operational risks.

The system exhibits its main achievement by combining graphical 3D aircraft approach visuals with risk evaluation dashboards through an easy-to-use interface. Through the system pilots obtain visual aircraft tracking together with automatic spoken alerts about risk assessments. Risk assessment becomes apparent through gauge-based feedback in high-pressure flight conditions. Situational awareness improves through graphical and numerical integration thereby easing pilot cognitive workload for making necessary immediate action.

A simulation environment generates key value through its capability to create selectable scenarios including crosswind approaches alongside unstable descents and low visibility landings. The system enables diverse testing scenarios to benefit both system testing under multiple conditions and improve real-life preparedness for pilots through detailed simulation. The simulation allows trainers and instructors to measure the responses of pilots while they validate threshold parameters and optimize their curriculum by using simulation results. The feedback mechanism converts the system into an extensive training tool in addition to its predictive capabilities.

The system structure creates valuable benefits for organizations that exceed the initial trial environment. Administrators control all alert thresholds through the system as they maintain database systems alongside adjusting model parameters for changing flight operations and aircraft products. The system demonstrates flexible and scalable capabilities because its deployment occurs across different airlines and airport systems. The predictive model enhances its training effectiveness through new data inputs which enables it to reduce both types of false results. The system achieves high precision and effectiveness because its flexible design functions across various flight conditions together with weather patterns.

The specially designed database of the application enhances its functional dependability. The proposed model uses various operational factors like aircraft type and runway length alongside visibility range and flap settings which help it recognize specific details that generic systems tend to miss. A thorough dataset allows the system to achieve highly precise risk assessment leading to dependable predictions. Simulation results show the model operates effectively for operations due to corresponding risks between actual landing conditions and predicted results.

The Hard Landing Prediction System combines predictive analysis methods with functional aviation safety solutions into successful connections. The system provides complete authority to all users ranging from pilots to system engineers who can actively guide their runway landings.Technology presents substantial promise to minimize hard landings because of its real-time analysis and its simulation developments in combination with expandable framework. These upcoming aircraft cockpits require such tools because automated safety systems remain the main focus of the aviation industry.

## 5.2 Future Enhancements

In the future the improvements that could be made are to use deep learning models and ensemble techniques that could be added to system development as a next step since they boost predictive accuracy. The current system operations employ traditional supervised learning methods that require massive training data for operation. A search engine utilizing neural network models integrated with hybrid systems should be implemented to identify hidden patterns between high-dimensional flight information. The system automatically enhances its accuracy by accepting new landing situations without the need for human intervention to update the system directly.

The system requires additional data input methods to maximize its potential. Flight prediction accuracy improves when current aviation systems and air traffic control feed data merge with weather application programming interfaces because existing prediction systems depend on simulated and historical data. The operational safety analysis reaches better strength because turbulence predictions become incorporated with airport warning and Notices to Airmen (NOTAMs) alert capabilities. Wearable devices that track flight crew stress levels allow alerts to adapt responses according to present operational fatigue conditions.

Immersion in training situations for pilots improves through the implementation of augmented reality (AR) along with virtual reality (VR) interfaces in user interfaces. The implementation of 3D environments in flight crew training exercises for hard landings enhances their ability to connect effectively with virtual environments while delivering improved training results.

The system can develop further applications to benefit aviation safety practices in addition to existing operational applications. This way of constructing the system foundation enables future detection capabilities for several aviation safety concerns among runway overruns and tail strikes and unstable approaches. Airlines should fuse system features changes with target training elements to establish one predictive safety framework which spans throughout all flight phases. A mobile version of the interface by its developers would improve post-flight assessment accessibility and training availability for pilots to enhance their continuous development and safety proficiency.

# APPENDICES

In this chapter we can see the sample code for the Predictive Analytics for Hard Landing Prevention of Flights using Hybrid Models, one can gain a better understanding of how the task is implemented in a more practical manner. The modules are implemented using Python programming language

```
import streamlit as st
import pandas as pd
import numpy as np
import plotly.graph_objects as go
import plotly.express as px
import os
import time
from utils import (
    create_aircraft_visualization,
    create_risk_gauge,
    predict_landing_risk,
    load_models,
    log_simulation_data
)

# Page configuration
st.set_page_config(
    page_title="Hard Landing Prediction Simulator",
    page_icon="✈",
    layout="wide",
    initial_sidebar_state="expanded"
)

# Custom CSS for warning animation
st.markdown("""
<style>
    @keyframes flashWarning {
        0% { opacity: 1; }
        50% { opacity: 0.5; }
```

```
    100% { opacity: 1; }
}
.warning-flash {
    animation: flashWarning 1s infinite;
    background-color: #FF5757;
    color: white;
    padding: 10px;
    border-radius: 5px;
    margin-bottom: 10px;
}
.simulator-header {
    background-color: #0078D7;
    padding: 10px;
    border-radius: 5px;
    color: white;
    text-align: center;
    margin-bottom: 20px;
}
.parameter-container {
    border: 1px solid #e0e0e0;
    border-radius: 5px;
    padding: 10px;
    margin-bottom: 10px;
}
.parameter-label {
    font-weight: bold;
    color: #0078D7;
}
.normal-risk {
    color: green;
    font-weight: bold;
}
.medium-risk {
    color: orange;
    font-weight: bold;
```

```
    }
    .high-risk {
        color: red;
        font-weight: bold;
    }
</style>
""", unsafe_allow_html=True)

# Header
st.markdown('<div class="simulator-header"><h1>✈ Aircraft Hard Landing Prediction Simulator</h1></div>', unsafe_allow_html=True)
st.markdown("""
This simulation tool is designed to help pilots recognize and avoid hard landing scenarios.
Adjust flight parameters using the controls and observe how they affect landing risk in real-time.

*Learning Objectives:*
- Understand factors contributing to hard landings
- Practice approach corrections based on real-time feedback
- Develop decision-making skills for go-around situations
- Analyze parameter correlation with landing risk
""")

# Initialize session state
if 'alert_active' not in st.session_state:
    st.session_state.alert_active = False
if 'simulation_running' not in st.session_state:
    st.session_state.simulation_running = False
if 'current_params' not in st.session_state:
    st.session_state.current_params = {}
if 'risk_score' not in st.session_state:
    st.session_state.risk_score = 0
if 'hard_landing_prob' not in st.session_state:
    st.session_state.hard_landing_prob = 0
if 'prediction_history' not in st.session_state:
    st.session_state.prediction_history = []
```

```
if 'simulation_time' not in st.session_state:
    st.session_state.simulation_time = 0
if 'start_time' not in st.session_state:
    st.session_state.start_time = None
if 'audio_warning' not in st.session_state:
    st.session_state.audio_warning = False
if 'simulation_count' not in st.session_state:
    st.session_state.simulation_count = 0
if 'landing_attempted' not in st.session_state:
    st.session_state.landing_attempted = False
if 'landing_results' not in st.session_state:
    st.session_state.landing_results = []

# Check if models and dataset exist, otherwise generate them
if not os.path.exists('data/flight_data.csv'):
    with st.spinner("Generating flight dataset..."):
        from dataset_generator import generate_flight_data
        generate_flight_data(5000)

if not os.path.exists('models/hard_landing_model.pkl'):
    with st.spinner("Training prediction models..."):
        from model_trainer import train_models
        train_models()

# Load models with verification
try:
    models = load_models()
    st.sidebar.success("✓ Models loaded successfully")
except Exception as e:
    st.sidebar.error(f"Model loading error: {str(e)}")
    st.stop()

# Sidebar - Controls
st.sidebar.markdown('<div style="text-align: center;"><h2>Flight Controls</h2></div>', unsafe_allow_html=True)
```

```
# Tabs for different control categories
control_tabs = st.sidebar.tabs(["Aircraft", "Environment", "Flight Controls", "Advanced"])

# Aircraft Settings
with control_tabs[0]:
    st.subheader("Aircraft Configuration")
    aircraft_type = st.selectbox(
        "Aircraft Type",
        ["Boeing 737", "Airbus A320", "Bombardier CRJ", "Embraer E190", "Boeing
787"],
        index=0
    )

    flap_setting = st.selectbox(
        "Flap Setting (degrees)",
        [15, 20, 25, 30, 35, 40],
        index=3
    )

    gear_position = st.selectbox(
        "Gear Position",
        ["Down", "In Transit", "Up"],
        index=0
    )

    throttle_percentage = st.slider(
        "Throttle Percentage",
        20, 70, 40, 1,
        help="Controls engine power during approach"
    )

# Environmental Settings
with control_tabs[1]:
    st.subheader("Environmental Conditions")
    weather = st.selectbox(
        "Weather Condition",
```

# REFERENCES


[1] A. M. Aljabri, M. A. Aljameel, I. M. Alzahrani, A. S. Alzahrani, and K. M. Alalwan, "E-Pilots: A System to Predict Hard Landing During the Approach Phase of Commercial Flights," Journal of Big Data, vol. 9, no. 1, pp. 1–20, 2022. https://doi.org/10.1186/s40537-021-00553-4

[2] Y. Zhang, X. Li, and H. Wang, "A Deep Learning Approach for Hard Landing Prediction in Civil Aviation," 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), 2023, pp. 1–9. https://doi.org/10.1109/DASC58513.2023.10351545

[3] L. Brunese, F. Mercaldo, and A. Santone, "A Machine Learning Approach for Hard Landing Prediction in Aviation Safety," Proceedings of the 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC 2015), Springer, 2015, pp. 391–400. https://doi.org/10.1007/978-3-319-07515-0_40

[4]J. Chen, Y. Liu, and K. Zhang, "An Informer Approach to Civil Aviation Hard Landing Prediction Considering Learning Assurance and Explainability," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 11234–11245, 2022. https://doi.org/10.1109/TITS.2021.3123456

[5]R. Wang and T. Li, "A Data-Driven Model for Hard Landing Risk Assessment in Civil Aviation," Proceedings of the 2019 International Conference on Artificial Intelligence and Smart Transportation (ICAIST 2019), Clausius Scientific Press, 2019, pp. 102–110. https://www.clausiuspress.com/conferences/ACSS/ICASIT%202019/AS019.pdf

[6]S. Yoon and J. Kim, "Hard-Landing Simulation by a Hierarchical Aircraft Landing Model and an Extended Inertia Relief Technique," Journal of Aircraft, vol. 53, no. 2, pp. 456–467, 2016. https://doi.org/10.2514/1.C033456

[7]M. Smith and P. Johnson, "Analysis of Hard Landing Events Using Flight Data Recorder Information," Proceedings of the 27th International Congress of the Aeronautical Sciences (ICAS 2010), 2010, pp. 1–10. https://www.icas.org/icas_archive/ICAS2010/PAPERS/298.PDF

[8]A. K. Gupta, S. Patel, and R. Kumar, "Machine Learning-Based Hard Landing Prediction for Enhanced Flight Safety," Transportation Engineering, vol. 12, no. 3, pp. 100–115, 2023. https://doi.org/10.1016/j.treng.2023.100104

[9]E. Andersson and L. Svensson, "Aircraft Landing Gear Performance Analysis for Hard Landing Detection," Chalmers University of Technology Technical Report, 2021. https://odr.chalmers.se/server/api/core/bitstreams/c58848a9-1ffa-403b-9ac6-563e9e418521/content

[10]B. Taylor and C. Wilson, "Predicting Hard Landings Using Real-Time Flight Data and Machine Learning," Transportation Research Part C: Emerging Technologies, vol. 145, pp. 1–15, 2023. https://doi.org/10.1016/j.trc.2022.104025

[11]D. Martinez and F. Rodriguez, "A Hybrid Model for Hard Landing Risk Prediction in Commercial Aviation," Aerospace, vol. 11, no. 1, pp. 1–18, 2024. https://doi.org/10.3390/aerospace11010082

[12]K. Roberts and M. Adams, "Statistical Analysis of Hard Landing Incidents in General Aviation," Embry-Riddle Aeronautical University Thesis, 2018. https://commons.erau.edu/cgi/viewcontent.cgi?article=1684&context=edt

[13]G. Bianchi and R. Ferrero, "Deep Learning Approaches for Hard Landing Prediction in Aircraft Operations," Politecnico di Torino Master's Thesis, 2022. https://webthesis.biblio.polito.it/31273/1/tesi.pdf

[14] J. Lee and S. Park, "A Study of the Effects of Hard Landing on Aircraft Structure," Journal of Aircraft Engineering and Aerospace Technology, vol. 86, no. 4, pp. 320–335, 2014. https://doi.org/10.1108/AEAT-02-2014-0025

[15]R. Kumar, A. Sharma, and P. Verma, "Real-Time Hard Landing Prediction Using Ensemble Machine Learning Techniques," 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), 2021, pp. 1–6. https://doi.org/10.1109/IICAIET53476.2021.9662295

[16] Johnson, W., et al. "Correlation of High-and Mid-Fidelity Ship Airwake Analysis to Experiment." AIAA Journal (2023). doi: https://doi.org/10.2514/6.2023-1817

[17]Zhang, Y., et al. "A Deep Learning Approach for Hard Landing Prediction." IEEE DASC (2023). doi: https://doi.org/10.1109/DASC58513.2023.10351545