Gümüşhane University Journal of Science

GUFBD / *GUJS* (2025) 15(3): 797-808 doi: 10.17714/gumusfenbil.1676270 Research Article

AnoSense: Edge computing for real-time flight anomaly detection by using embedded deep neural networks

AnoSense: Gömülü derin sinir ağları kullanarak gerçek zamanlı uçuş anormalliklerinin tespiti için uç birim hesaplama

Hatice Vildan DUDUKCU*[1], Murat TASKIRAN[1], Nihan KAHRAMAN[1]

[1]*Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü, 34220, İstanbul*

• Received: 14.04.2025 • Accepted: 01.08.2025

Abstract

Autonomous systems, including unmanned aerial vehicles and commercial airplanes, are increasingly integrated into modern aircraft to minimize pilot errors while enhancing flight control. Ensuring flight safety requires accurate detection of anomalies in sensor data that causes error. This study, AnoSense, proposes an autoencoder-based deep neural network designed to detect anomalies in an unmanned aerial vehicle. AnoSense processes 20 flight sensor parameters to identify irregularities that could compromise operational safety. The model is trained and evaluated using NASA's DASHlink anomaly data set, achieving 97.07% precision, outperforming conventional deep learning methods. Additionally, AnoSense is optimized for deployment on resource-constrained edge devices, with implementation and performance validation conducted on a Raspberry Pi. The experimental results demonstrate the feasibility of real-time flight anomaly detection on embedded systems, making AnoSense a promising solution to improve aircraft safety through edge computing.

Keywords: Anomaly detection, Deep neural networks, Edge computing, Embedded systems, Flight data

Öz

Otonom sistemler — insansız hava araçları ve ticari uçaklar da dahil olmak üzere — modern hava araçlarında giderek daha fazla kullanılmaktadır. Bu sistemler, pilot hatalarını en aza indirmeyi ve uçuş kontrolünü geliştirmeyi amaçlamaktadır. Uçuş güvenliğini sağlamak için, hatalara yol açabilecek sensör verilerindeki anormalliklerin doğru şekilde tespit edilmesi büyük önem taşır. Bu çalışmada AnoSense adı verilen bir sistem önerilmektedir. AnoSense, insansız hava araçlarında meydana gelebilecek anormallikleri tespit etmek amacıyla geliştirilmiş, autoencoder tabanlı bir derin sinir ağı modelidir. Sistem, uçuş güvenliğini riske atabilecek düzensizlikleri belirleyebilmek için 20 farklı uçuş sensörü parametresini analiz eder. Model, NASA'nın DASHlink anormallik veri seti kullanılarak eğitilmiş ve test edilmiştir. Yapılan değerlendirmeler sonucunda %97.07 doğruluk oranına ulaşmış ve geleneksel derin öğrenme yöntemlerinden daha iyi bir performans sergilemiştir. Ayrıca AnoSense, donanım kaynakları sınırlı olan uç birimlerde çalışacak şekilde optimize edilmiştir. Gerçek zamanlı performansı doğrulamak amacıyla Raspberry Pi üzerinde uygulanmış ve başarılı sonuçlar elde edilmiştir. Elde edilen deneysel sonuçlar, gömülü sistemler üzerinde gerçek zamanlı uçuş anormalliği tespiti yapılabileceğini göstermektedir. Bu yönüyle AnoSense, uç bilişim teknolojisi kullanarak hava aracı güvenliğini artırmaya yönelik etkili ve umut vadeden bir çözüm sunmaktadır.

Anahtar kelimeler: *Anomali tespiti, Derin sinir ağları, Uç bilişim, Gömülü sistemler, Uçuş verileri*

1. Introduction

Today, autonomous systems are widely utilized in various aircraft, such as unmanned aerial vehicles and airplanes, to help with flight controls and minimize pilot mistakes. Autonomous systems can automatically perform various tasks and provide flight control on aircraft without human intervention. Flight safety may be jeopardized by unexpected situations or abrupt changes in the aircraft. However, autonomous anomaly detection increases flight safety and operational efficiency through the early detection of these changes and occurrences in the aircraft.

*Hatice Vildan DUDUKCU; vdudukcu@yildiz.edu.tr

The use of autonomous systems in aircraft has been one of the interesting areas for researchers since the early 2000s (Cini & Griffith, 1999; Nonami, 2007). Although the first studies on autonomous systems included the aim of helping the user of the aircraft, today autonomous systems aim to fulfill the tasks without the need for any user. While in the past any error that might have occurred could be compensated by the pilot, today a system that can calculate these errors with high performance and take precautions has become mandatory. One of the most important parts of autonomous systems is the analysis progress of the data received from the sensors placed on the aircraft and the early detection of any anomaly. In recent years, the high performance of Deep Neural Networks (DNNs) in solving various problems has led researchers to implement DNNs in their proposed methods for anomaly detection. Although the performance of classical deep learning methods in anomaly detection using flight sensor data on aircraft has reached a certain level, it is still not a fully resolved problem. Therefore, various DNN-based solutions are offered by researchers in the current literature for anomaly detection. In this study, the AnoSense method, which includes an autoencoder (AE)-based hybrid DNN architecture, is proposed for flight anomaly detection.

AnoSense, a DNN method optimized for operating on low-power development boards, was proposed to detect aircraft touchdown anomalies. The applicability of the proposed method to edge applications was evaluated using the DASH-link multi-class anomaly classification dataset, which was made available to researchers by the National Aeronautics and Space Administration (NASA). Furthermore, the edge device onboard tests are conducted using a Raspberry Pi development board. The following is a summary of the contributions and outputs of this study:

- Studies on the multi-class anomaly problem, which has rarely been studied in literature, were examined, and their details were shared with researchers.
- AnoSense, a new method for anomaly detection, was developed and tested using the DASHlink multi-class anomaly classification data set containing aircraft touchdown anomalies.
- The developed method was optimized to run on the development board, and its usability was tested for low-power edge applications.

After mentioning recent research closely related to the topic, the suggested framework and its use are discussed in this study. The paper concludes with conclusions and suggestions following the section that presents the outcomes of various applications.

2. Related works

Upon examining the literature on aerial vehicle studies, it has been observed that there are three main issues; flight monitoring, energy modeling, and remote sensing (Dudukcu et al., 2023). These issues mainly focused on preventing possible damage to aerial vehicles and optimizing power consumption. The most important action for preventing prospective damage to aircraft is detecting anomalies in flight sensors. Early action brought about by anomaly detection can greatly help reduce financial losses associated with aerial vehicles (Yang et al., 2023). However, anomalies are infrequent events, leading to an imbalance between normal and abnormal data. This imbalance poses various challenges in training deep learning models, as the algorithms may become biased towards the majority class, often overlooking the minority instances. Therefore, some studies, such as (Ma et al., 2024), investigate the impact of diverse data resampling techniques on deep learning-based anomaly detection. Their findings suggest that oversampling methods generally outperform undersampling and hybrid approaches, emphasizing the importance of generating more data for minority classes to mitigate imbalance effects. In another study (Pezzicoli et al., 2025) to address the class imbalance in anomaly detection teacher-student perceptron model was solved through replica theory, and a foundation for developing more robust models was offered.

Many different methods have been proposed by researchers to address the challenge of anomaly detection in aerial vehicles. In the study carried out by Lu et al. in 2017, temperature data in Unmanned Aerial Vehicles (UAVs) engines were recorded using DS18B20 sensors, and it was used to detect UAVs' abnormal operation temperatures with reinforcement method, which is one of the popular methods among researchers recently (Lu et al., 2017). Consequently, the proposed solution prevents financial loss. One of the other methods used for anomaly detection is the kernel principal component analysis (KPCA) method. The KPCA was used by (Yong et al., 2017) to detect sensor anomalies in UAVs where the tests were carried out by the researchers for two different anomaly cases that were artificially added to the data obtained from the flight simulator. On the other

hand, (Liu et al., 2018) used the data obtained from Micro-Electro-Mechanical system (MEMS) sensors to test the competence of the KPCA method in anomaly detection. In (Wang et al., 2019) anomaly detection was attempted for multiple sensor data and Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), one of the Recurrent Neural Networks (RNNs) that achieves very high performance in time series prediction, is used. The architecture consisting of LSTM is first trained with nominal data and then future data is predicted with the pre-trained model. Unusual results were determined as anomalies and high anomaly detection performance was achieved with the proposed method. In (Lian et al., 2022), the Inverse Reinforcement Learning (IRL) method was proposed for the detection of different types of anomalies. This study is similar to the study conducted by (Wang et al., 2019) in terms of training and testing process. A Digital Signal Processor (DSP) and Field-Programmable Gate Array (FPGA) based system, that monitors the features such as altitude, speed, temperature, and humidity and reports anomalies have been proposed by (Chunhui et al., 2020).

According to the literature, anomaly detection studies are similar to each other nearly for all aircraft anomaly problems. In (Nanduri & Sherry, 2016), an anomaly detection system using LSTM and Gated Recurrent Unit (GRU) (Chung et al., 2014) architectures were proposed and 9 out of 11 anomalies in the test data set were detected with high sensitivity. Another study using RNNs, (Cao et al., 2021) proposed an aircraft track anomaly detection system using a combination of Multidimensional Outlier Descriptor (MOD) and the Bi-directional Long-Short Time Memory network (Bi-LSTM). It has shown that the proposed system is sufficient for aircraft track anomaly detection with a test accuracy of 96% obtained in experimental studies. The Recursive Least Squares method was used by (Keipour et al., 2019) to perform real-time anomaly detection, and it was observed that 86.36% accuracy was achieved in the tests with 22 flight data. In (Lee et al., 2020), a real-time and data-driven anomaly detection system was developed using support vector regression models. The performance of the proposed method was tested with artificially created anomaly scenarios. As a result of the experimental studies, the proposed method achieved both 38.2% decrease in the overall computational time and 37% decrease in the Root Mean Square Error (RMSE). Furthermore, (Dangut et al., 2023) has proposed an approach that relies on auto-encoder and bidirectional gated recurrent unit networks to address the challenge of predicting exceedingly rare failures in aircraft predictive maintenance modeling. The results of this investigation showed a notable increase in precision by 18%, recall by 5%, and G-mean values by 10% when contrasted with other traditional DNN techniques. Lastly, (Memarzadeh et al., 2023) introduced a semi-supervised active learning approach for anomaly detection in aviation, addressing the limitations of fully unsupervised methods by leveraging large amounts of unlabeled data alongside a small set of expert-labeled samples. The results showed that the proposed framework achieves reliable performance when only the 1% of the data is labeled.

According to research that has been reviewed recently, there is a rapid increase in anomaly detection methodologies in aerial vehicles. Particularly, it has been tried to prevent great material and human losses by detecting the anomalies in the sensor outputs in aircraft. Within the scope of this study, an autoencoder-based hybrid classification method has been proposed to detect flight data anomalies for commercial aircraft. Detailed information and the results of the experimental studies are shared with the researchers in the following sections.

3. Proposed framework

Anomaly detection, or the identification of events that deviate from the norm, is a critical issue in the aviation industry for promptly detecting problems and taking appropriate action. A review of existing studies highlights key gaps, such as computational complexity and challenges in edge deployment. Deep learning-based methods often require high computational power, making real-time deployment on low-power edge devices challenging. Additionally, most models are trained in high-performance computing environments, lacking efficient deployment strategies for low-power embedded systems like Raspberry Pi. Therefore, the aim of this paper is first to explain the proposed methodology, AnoSense, a DNN model for anomaly detection, and then to deploy this model on low- power and low-memory edge devices. The main contents of this study, AnoSense architecture, and edge device deployment steps are explained in detail in the following subsections.

3.1. Overview

AnoSense is proposed for aircraft final approach anomaly detection using flight sensor data. The various steps of this study and the AnoSense are given in Figure 1. Furthermore, the details can be listed as follows:

- As a first step in the proposed method, preliminary analysis and preprocessing were performed on the anomaly data accessed from NASA's DASHlink website. The four-class data was rearranged as binary classed data as anomaly and nominal labels and the input data was normalized before the neural network input step.
- After the preprocessing steps, data were used in both training and testing of the AE-based hybrid DNN method, AnoSense. The DNN architecture was optimized with the Grid Search algorithm (Liashchynskyi & Liashchynskyi, 2019) for various layers, filter, and hyper-parameter combinations.
- After the AnoSense model was trained, its performance was evaluated using accuracy, precision, recall, and F- score metrics on validation and test data. The trained model was then optimized in terms of size and memory before being deployed to the selected Raspberry Pi edge device.
- In the last step, the performance of the proposed AnoSense method was also tested on the Raspberry Pi board and the real-time operation of the model on edge devices was evaluated.

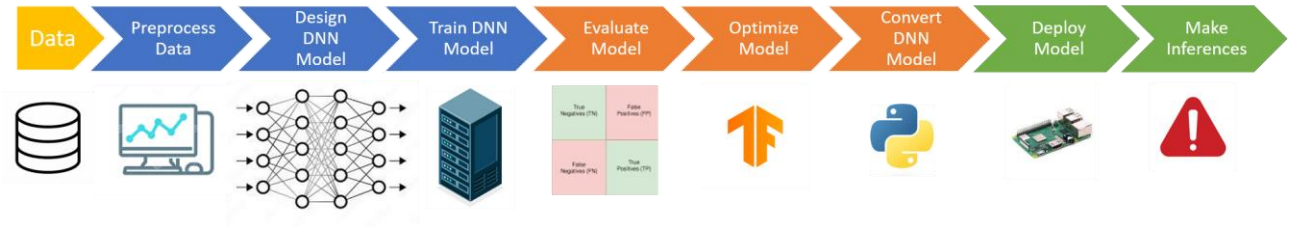

Figure 1. Overview of the study and the proposed AnoSense method.

3.2. Multi-layer perceptron

The Multi-Layer Perceptron (MLP) is a type of artificial neural network characterized by a fully connected architecture. In deep learning, MLPs can incorporate multiple hidden layers, forming a Fully Connected Neural Network (FCNN), where each neuron is connected to all neurons in the preceding layer. In this architecture, input data is propagated through the hidden layers, processed by activation functions, and subsequently passed to the next layer. Training is generally carried out using the backpropagation algorithm, which employs a loss function to evaluate the discrepancy between the predictions of the model and the actual labels. The backpropagation iteratively updates the network's weights and biases to minimize this error. Research indicates that MLPs serve as a simple yet effective benchmark for time series classification in deep neural network applications (Wang et al., 2017).

3.3. Convolutional neural networks

The Convolutional Neural Network (CNN) is a widely used deep neural network technique commonly used to extract features in classification tasks. They employ convolutional operations to capture hierarchical spatial features, reducing computational complexity compared to fully connected networks. Unlike traditional MLPs, where each neuron in a layer is connected to all neurons in the next layer, CNNs exploit the spatial structure of data by introducing localized connections and weight sharing, enhancing feature learning efficiency.

By applying filters, Convolutional Neural Networks (CNNs) progressively combine simpler patterns to form more complex representations. CNNs improve performance by using filters while requiring less preprocessing compared to other classification methods. Convolution is utilized in at least one of its layers instead of the standard matrix multiplication (Albawi et al., 2017).

3.4. Autoencoders

Autoencoders (AEs) are a type of DNNs designed for data representation via compression and subsequent reconstruction. They have two fundamental components: an encoder and a decoder. The encoder acquires a feature representation of the incoming data and compresses the input data into a lower-dimensional latent representation. In contrast, the decoder uses this latent representation to reconstruct a feature vector that closely approximates the original input.

Autoencoders can be classified into undercomplete and overcomplete types based on the size of the latent space relative to the input dimension. An undercomplete autoencoder compresses input data using an encoder layer that has fewer neurons than the dimensionality of the input data. The undercomplete autoencoder is chosen to highlight essential data features while omitting irrelevant information due to its superior generalization abilities. An overcomplete autoencoder features an encoder layer with more neurons than the input data. The encoder increases the dimensionality of the input data. Overcomplete autoencoders retain multiple data attributes, enabling them to capture intricate details and relationships.

3.5. Model construction

In this study, the aircraft final approach anomaly detection problem is considered as a binary classification task, where machine learning is used to categorize input data into one of two distinct classes. The AnoSense method was introduced for binary classification of flight anomalies.

The proposed AnoSense classification method employs a hierarchical deep learning architecture that integrates two consecutive autoencoder structures: a Convolutional Autoencoder followed by an MLP Autoencoder. The model architecture of the AnoSense method can be seen in Figure 2. It begins with an input layer that processes samples with 160 time-steps and 20 different sensor readings, then passes through a Convolutional Autoencoder with filter sizes of 32-16-16-32, respectively. In this structure, the encoder begins with a 32-filter convolutional layer that extracts spatial features from the input. This is followed by a 16-filter layer, leading to a compact latent representation. This latent space serves as a compressed encoding of the original input. The decoder then reverses this process, expanding the feature dimensions from 16 to 32, reconstructing the input from the compressed latent space. This decoding helps to identify anomalies by comparing reconstructed outputs with original inputs.

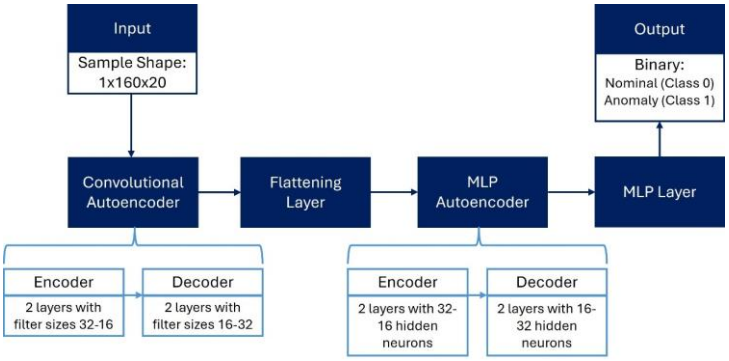

Figure 2. Proposed AnoSense model architecture.

A flattening layer follows the Convolutional Autoencoder, converting the multi-dimensional feature maps into a one-dimensional vector representation. This transformation allows the extracted spatial features to be fed into the MLP Autoencoder, which has four layers with number of neurons 32-16-16-32, respectively. This transition from convolutional to fully connected layers enable the model to learn both spatial and high-level abstract patterns, improving feature extraction and classification performance. Finally, a fully connected MLP that uses Rectified Linear Unit (ReLU) as an activation function processes the extracted features, leading to a binary classification output.

3.6. Edge deployment

Edge computing is an approach where data processing and analysis are performed on resource-constrained, low-power, and low-memory devices where data is generated or collected. Machine learning at the edge intends to perform this process locally and therefore more quickly than machine learning typically runs on servers in huge data centers. In real-time systems, edge computing offers minimal latency and quick response times. The aim of the study was to perform anomaly detection using sensor data from aircraft, and an edge operation was examined since it is essential to quickly address problems on the control card when anomalies are discovered. For this purpose, anomaly detection was also performed using the trained AnoSense model on the Raspberry Pi development board, and the performance of the method on resource-limited and small-sized control boards was evaluated.

The Raspberry Pi 3 Model B card used in the comparison study has a quad-core 1.2 GHz ARM Cortex-A53 processor and includes 1 GB RAM internal memory. Since the Raspberry Pi has limited processing power, the sparsity of the AnoSense model was increased by pruning, and the large weights in the model were modified. The board is configured to employ deep learning frameworks to handle the libraries and dependencies that the model intended to run on the Raspberry Pi will require. Due to the Raspberry Pi's memory limitations, predictions were made sequentially, and input sensor data was transferred to the card in specific batches to avoid memory problems throughout the test. Accuracy, precision, recall, and F-score were used in the testing on the Raspberry Pi to assess the classification outputs based on the input data, as well as to assess the method's effectiveness. During the training process pruning was incorporated to enhance the performance of the models in edge applications. Pruning is a method used in neural network training to enhance efficiency or decrease the model's size by intentionally eliminating specific features, such as connections or neurons.

The implementation of the AnoSense model on the Raspberry Pi does not include any online learning capability. The model deployed to the edge device operates solely in inference mode, and it uses the parameters learned during the offline training phase and does not update based on new data encountered during deployment. This decision was made to preserve the real-time performance, memory efficiency, and computational feasibility on the edge device. As a result, while the system can detect anomalies in real time, it does not adapt to new patterns or environmental changes autonomously. This limitation was accepted as a design trade-off in favor of a lightweight, fast-response edge solution.

4. Experiments and results

In this study, using commercial aircraft flight sensors' data, the classification of flight anomalies was carried out. For this purpose, first, AnoSense, an AE-based deep learning method, was proposed for anomaly detection, and its performance was examined compared to other DNN methods in literature. The DNN models were initially trained on a computer using the Python environment, then performance tests were carried out on both the Python environment and Raspberry Pi board with 5-fold cross-validation. In the next step, to evaluate the applicability of the proposed AnoSense method in real aircraft, the trained model was deployed to an edge device, and anomaly detection performance was also tested on this device. This section begins by providing comprehensive details about the utilized data set. The experimental test setup that was necessary for evaluating the suggested approach was then described. Finally, the experimental study results were evaluated using various evaluation metrics and shared with the researchers.

4.1. Data set

The Curated 4 Class Anomaly Detection Data Set (Matthews, 2022), a publicly available data set sourced from NASA's DASHlink website, is used to train and test the proposed framework AnoSense. This data set, which includes real data collected on a regional jet flying in commercial service over a three-year period, is submitted to the literature to minimize potential problems in aviation and enhance overall flight safety.

The anomaly data set includes 20 different flight variables (Table 1) recorded from the flight sensors in a 160-second window on the final approach before touchdown for 99837 flights.

Table 1. Flight data variables in the anomaly detection data set.

Control Surface Variables	Positional Variables	Flight Sensor Variables		Other Flight Variables
Aileron position (Left, Right)	Roll angle	Altitude	Core speed	Selected Course
Elevator position	Drift Angle	Airspeed	Angle of attack	Glideslope deviation
Flap position	Pitch angle	Total Pressure	Vertical acceleration	Selected Heading
Rudder position		Wind Speed		Localizer deviation
				True heading

The flight variables include control commands (e.g., aileron, rudder, elevator positions), spatial orientation (e.g., roll, pitch, drift angle), flight metrics (e.g., altitude, airspeed, angle of attack), and navigational deviations (e.g., glideslope and localizer). These features reflect the aircraft's stability and compliance with expected flight paths. Therefore, their inclusion strengthens the model's potential to identify and classify operational anomalies in real time, especially when deployed on edge hardware with limited computational resources. There are three different anomaly classes and one nominal class in the data set and the distribution of these classes is given in Table 2.

Table 2. Distribution of the original data set class samples.

Class	Number of samples
Nominal (Normal Operation)	89663
Speed High	7013
Path High	2207
Flaps Late Setting	954

In the original data set, three distinct types of anomalies are provided along with nominal flight data. In this study, these three different anomaly types in the data set were assigned to a single class and the anomaly detection was carried out as binary classification (nominal/anomaly). This simplification was preferred to reduce the model complexity and computational burden, which is particularly important for edge-based deployment on resource-constrained devices like Raspberry Pi. Moreover, due to the significant class imbalance across the anomaly types, combining them helped mitigate potential bias and improved the stability of the model during training.

The modified dataset, labeled as 0 (nominal) and 1 (anomaly), was split into stratified folds for the 5-fold cross-validation process, ensuring balanced representation of both classes in each subset. Each stratified fold contains training, validation, and test sets. Data set preparation in this study is also visualized in Figure 3.

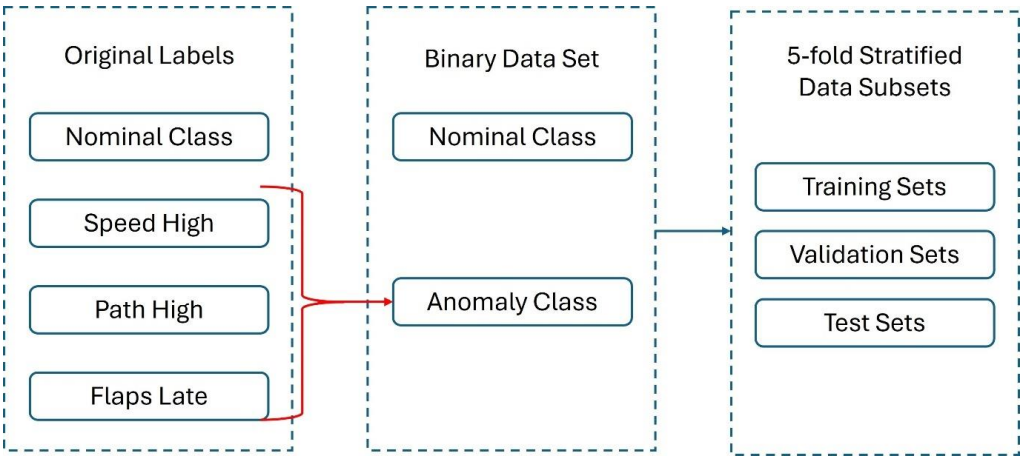

Figure 3. Organizing and dividing the data set for experiments

4.2. Experimental setup

AnoSense was first trained with a commercial aircraft sensor dataset, and then its performance was compared with classical deep learning methods in the Python environment. Later, this model was also tested on the selected edge device, and its anomaly detection performance was examined.

In the first step, the flight data was first divided into 5-stratified folds as mentioned in the 4.1. Data Set subsection. During the training phase, the model underwent training using the training subsets, which consisted of 70% of the data from each stratified fold. In addition, training validations were conducted by partitioning

20% of the training data. To compare different DNN approaches, the trained models were assessed using test subsets that were specific to their respective folds within the Python environment.

AnoSense was compared with LSTM, GRU, Temporal Convolutional Network (TCN) (Bai et al., 2018; Lea et al., 2016, 2017), and CNN, which are methods used in time series problems frequently, for anomaly detection. The 5-fold cross-validation accuracy, precision, recall, and F-score metrics show the performance of the proposed AnoSense method's success and it is proved that AnoSense can be used in edge deployments. After the performance evaluation, pruning was applied to the network to finalize the proposed method. These performances were examined using validation and test sets on the Python environment and Raspberry Pi board.

4.3. Experimental results

In this study, the DNN model created for AnoSense was initially trained on a computer using the Python environment, then performance tests were carried out on both the Python environment and edge device (Raspberry Pi). Accuracy, precision, recall, and F-score metrics (Hossin & Sulaiman, 2015) were used for model performance evaluations for both environments. In the experiments, model assessments were done using 5-fold cross-validation. Table 3 shows the experimental results of the AnoSense method before pruning, performed on the Python environment. The results of these tests demonstrate that AnoSense models outperform LSTM, GRU, CNN, and TCN models in the Python environment, by achieving accuracy score of 97.43%. Upon analyzing the classification results using the metrics presented in Table 3, it becomes evident that precision, recall, and F-score exhibit lower values in comparison to the accuracy score. The reason for this is that the analyzed anomaly dataset is an unbalanced dataset consisting of a significant proportion of nominal classes.

Table 3. Performance metrics (%) of the AnoSense model without pruning on Python environment using validation data sets (Best results are shown in bold).

Method	Accuracy	Precision	Recall	F-Score
LSTM	96.17	83.57	77.80	80.51
GRU	96.47	86.50	77.45	81.70
CNN	96.78	87.44	79.87	83.46
TCN	96.96	87.05	82.33	84.60
AnoSense	**97.43**	**89.28**	**85.04**	**87.10**

Table 4. Distribution of the original data set class samples.

Model	Size (KB)
AnoSense without pruning	1453
AnoSense with pruning	497

To validate the findings on Raspberry Pi, the model sizes and the binary classification performance of the AnoSense model were analyzed for the base and pruned model. The effect of the pruning process on the model size can be seen in Table 4. The comparison of the model performance metrics using the pruned final AnoSense model is also given in Figure 4.

It can be seen that the proposed AnoSense method provides higher anomaly detection performance for flight sensor data, compared to other classical DNN methods explained in the Experimental Setup subsection. The AnoSense edge application has 97.32% accuracy before pruning and 97.07% after pruning for binary classification of the anomalies. Although there is a slight decrease in anomaly detection performance after pruning, the high reduction in model size (Table 4) of the final method is preferred because it provides better deployment to resource-constrained devices.

5. Conclusion

Anomaly detection in aerial vehicles plays a crucial role in ensuring flight safety by identifying sensor abnormalities that could lead to critical failures. This study introduced AnoSense, an autoencoder-based deep

learning model, designed for anomaly detection in aircraft. The proposed method addresses key challenges in literature, particularly those related to computational complexity, real-time deployment, and model optimization.

AnoSense achieves 97.07% accuracy, surpassing conventional anomaly detection approaches such as LSTM, GRU, and traditional deep learning methods. Unlike previous works that focus primarily on high-performance computing environments, AnoSense is optimized for low-power edge devices, making it a flight anomaly detection model successfully deployed on a Raspberry Pi. To further improve computational efficiency, model pruning techniques were applied, reducing the model size from 1453 KB to 497 KB while maintaining high anomaly detection performance. This enables inference without compromising accuracy, a critical advancement for edge computing applications in aviation. Additionally, AnoSense was validated on real-world flight sensor data from NASA's DASHlink dataset, demonstrating its reliability and robustness in practical aviation environments.

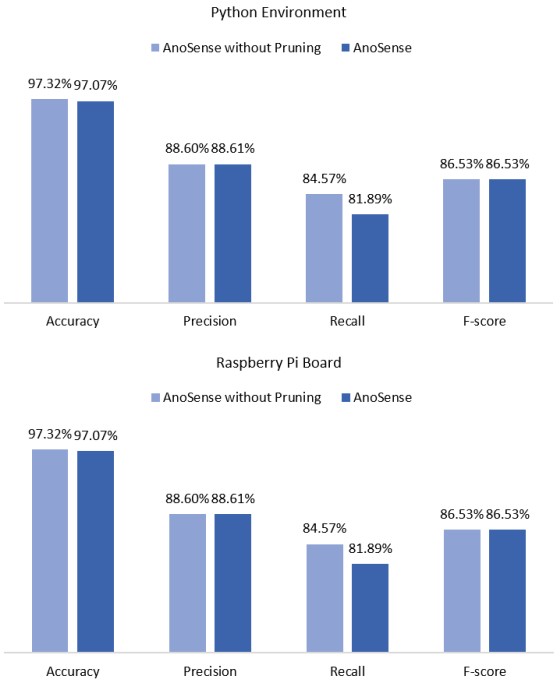

Figure 4. Performance metrics of the final AnoSense model on Python environment and development board using test data sets.

To benchmark, AnoSense was compared with several state-of-the-art studies from the literature. Notably, (Nanduri & Sherry, 2016) employed a hybrid RNN architecture with LSTM and GRU models, detecting 9 out of 11 flight anomalies with a perfect precision, 0.818 recall, and an F-score of 0.89. (Cao et al., 2021) achieved 96 % accuracy and 97.36 % recall using a MOD-Bi-LSTM architecture on flight track data. In contrast, our framework achieves comparable detection performance with 97.07% accuracy, while being optimized for deployment on resource-constrained hardware such as Raspberry Pi, highlighting its balance of accuracy and efficiency. Compared to the reviewed literature, the proposed AnoSense framework distinguishes itself through its emphasis on lightweight deployment and edge compatibility, particularly using resource-constrained devices like the Raspberry Pi. While models such as those by (Cao et al., 2021) demonstrated high accuracy in centralized environments, their architectures may require significant computational resources, limiting real-time applicability in embedded platforms. Moreover, unlike previous studies that often relied on either synthetic anomalies or limited flight conditions, AnoSense was trained and tested on a real-world, dataset, and optimized for binary classification to ensure fast and reliable anomaly detection at the edge. This prioritization of efficiency, without significant compromise on accuracy, and its modular design for future extension to multi-class classification and fault localization, position AnoSense as a scalable and robust alternative in real-time aerial anomaly detection.

Despite its advancements, AnoSense has certain limitations. The model is trained on a specific set of flight sensor datasets, and its performance may vary when applied to different aircraft models or previously unseen environmental conditions. To address this, future work will focus on expanding the training dataset with more diverse aircraft models and flight scenarios, including adverse weather and varied geographical contexts, to improve robustness and transferability. While model pruning effectively reduces size and computational cost, it results in a minor accuracy reduction of approximately 0.25%. Further optimization strategies, such as quantization, could be explored to achieve a better balance between efficiency and accuracy. Additionally, AnoSense relies on labeled anomaly data for training, which may not always be available in real-world applications. To overcome this, unsupervised or self-supervised learning approaches could be integrated to detect previously unknown anomalies. The deployment of AnoSense on a Raspberry Pi demonstrates feasibility for real-time anomaly detection; however, high-frequency detection tasks may still pose performance challenges. Future implementations could investigate hardware accelerators such as FPGA to further enhance real-time processing capabilities.

The current binary classification strategy in AnoSense merging all anomaly types into a single class was deliberately chosen as the initial phase of a broader, multi-stage anomaly detection framework. This simplification allows for computationally efficient real-time deployment on edge hardware such as the Raspberry Pi, where fast anomaly detection is crucial. In future works, the model will be extended to recover multi-class distinctions, particularly focusing on identifying the root causes of critical mechanical faults to meet the deeper expectations of anomaly interpretation and fault localization.

Moreover, the original dataset exhibits a significant class imbalance across different anomaly types, which influenced the decision to merge anomalies into a single class for binary classification. This approach supported training stability and enabled efficient edge deployment. However, this simplification is intended as an initial phase of a broader framework. Future work will also aim to restore multi-class distinctions and explore advanced methods that can effectively handle imbalanced classes, thus supporting more informative anomaly detection in aviation applications.

A key challenge also lies in ensuring the portability and optimization of the AnoSense framework across a wide range of embedded platforms beyond the Raspberry Pi. These platforms may differ significantly in terms of computational power, memory capacity, and energy efficiency. Deploying the model effectively on heterogeneous edge hardwarerequires careful consideration of the trade-offs between latency, accuracy, and resource usage. Addressing this challenge is essential to make a scalable and robust solution for real-world anomaly detection tasks in aviation and UAV systems, where hardware diversity is a common constraint.

By overcoming the limitations of existing approaches, AnoSense represents a significant step forward in anomaly detection for aerial vehicles. Its high detection accuracy, lightweight deployment, and validated real-world performance make it a promising solution for enhancing aircraft safety through real-time, edge-based anomaly detection. Future work will address these limitations and extend the model's applicability to other aviation and UAV anomaly detection scenarios.

Author contribution

H.V.D contributed to the conceptualization of this study, methodology, experimentation and writing. M.T. and N.K. contributed to the conceptualization, supervision, writing and review of this study.

Declaration of ethical code

The authors of this article declare that the materials and methods used in this study do not require ethics committee approval and/or legal-special permission.

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 international conference on engineering and technology (ICET)*, 1–6.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Cao, Y., Cao, J., Zhou, Z., & Liu, Z. (2021). Aircraft track anomaly detection based on mod-bi-lstm. *Electronics*, *10*(9), 1007.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*. Chunhui, W., Zhou, J., Yuanhang, W., Shi, Z., Chuangmian, H., & Yunfan, Y. (2020). An anomaly detecting system for power system of four-rotor uav. *2020 International Symposium on Autonomous Systems (ISAS)*, 109–114.

Cini, P. F., & Griffith, P. (1999). Designing for mfop: Towards the autonomous aircraft. *Journal of Quality in Maintenance Engineering*, *5*(4), 296–308.

Dangut, M. D., Jennions, I. K., King, S., & Skaf, Z. (2023). A rare failure detection model for aircraft predictive maintenance using a deep hybrid learning approach. *Neural Computing and Applications*, *35*(4), 2991-3009.

Dudukcu, H. V., Taskiran, M., & Kahraman, N. (2023). Uav sensor data applications with deep neural networks: A comprehensive survey. *Engineering Applications of Artificial Intelligence*, *123*, 106476.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, *5*(2), 1.

Keipour, A., Mousaei, M., & Scherer, S. (2019). Automatic real-time anomaly detection for autonomous aerial vehicles. *2019 International Conference on Robotics and Automation (ICRA)*, 5679–5685.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 156–165.

Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. *European conference on computer vision*, 47–54.

Lee, H., Li, G., Rai, A., & Chattopadhyay, A. (2020). Real-time anomaly detection framework using a support vector regression for the safety monitoring of commercial aircraft. *Advanced Engineering Informatics*, *44*, 101071.

Lian, B., Kartal, Y., Lewis, F. L., Mikulski, D. G., Hudas, R., Wan, Y., & Davoudi, A. (2022). Anomaly detection and correction of optimizing autonomous systems with inverse reinforcement learning. *IEEE Transactions on Cybernetics*.

Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A big comparison for nas. *arXiv preprint arXiv:1912.06059*.

Liu, L., Liu, M., Guo, Q., Liu, D., & Peng, Y. (2018). Mems sensor data anomaly detection for the uav flight control subsystem. *2018 IEEE SENSORS*, 1–4.

Lu, H., Li, Y., Mu, S., Wang, D., Kim, H., & Serikawa, S. (2017). Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE internet of things journal*, *5*(4), 2315–2322.

Ma, X., Zou, H., He, P., Keung, J., Li, Y., Yu, X., & Sarro, F. (2024). On the influence of data resampling for deep learning-based log anomaly detection: Insights and recommendations. *IEEE Transactions on Software Engineering*.

Matthews, B. (2022). Curated 4 class anomaly detection data set. https://c3.ndc.nasa.gov/dashlink/resources/1018

Memarzadeh, M., Matthews, B., Templin, T., Sharif Rohani, A., & Weckler, D. (2023). Semi-supervised active learning

for anomaly detection in aviation. *Journal of Aerospace Information Systems*, *20*(4), 181–194.

Nanduri, A., & Sherry, L. (2016). Anomaly detection in aircraft data using recurrent neural networks (rnn). *2016 Integrated Communications Navigation and Surveillance (ICNS)*, 5C2–1.

Nonami, K. (2007). Prospect and recent research & development for civil use autonomous unmanned aircraft as uav and mav. *Journal of system Design and Dynamics*, *1*(2), 120–128.

Pezzicoli, F., Ros, V., Landes, F., & Baity-Jesi, M. (2025). Class imbalance in anomaly detection: Learning from an exactly solvable model. *arXiv preprint arXiv:2501.11638*.

Wang, B., Wang, Z., Liu, L., Liu, D., & Peng, X. (2019). Data-driven anomaly detection for uav sensor data based on deep learning prediction model. *2019 Prognostics and System Health Management Conference (PHM-Paris)*, 286–290.

Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. *2017 International joint conference on neural networks (IJCNN)*, 1578–1585.

Yang, L., Li, S., Li, C., Zhang, A., & Zhang, X. (2023). A survey of unmanned aerial vehicle flight data anomaly detection: Technologies, applications, and future directions. *Science China Technological Sciences*, *66*(4), 901–919.

Yong, D., Yuanpeng, Z., Yaqing, X., Yu, P., & Datong, L. (2017). Unmanned aerial vehicle sensor data anomaly detection using kernel principle component analysis. *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 241– 246.