ISSN 1648-7788 / eISSN 1822-4180

VILNIUS TECH
Vilnius Gediminas Technical University

AVIATION

2024
Volume 28
Issue 3
Pages 175–187
https://doi.org/10.3846/aviation.2024.22173

DEEP LEARNING-BASED PROACTIVE FAULT DETECTION METHOD FOR ENHANCED QUADROTOR SAFETY

Mehmet OZCAN ✉, Cahit PERKGOZ

Department of Computer Engineering, Faculty of Engineering, Eskisehir Technical University, Eskisehir, Turkiye

Article History:
- received 13 April 2024
- accepted 6 August 2024

Abstract. The early detection of faults in advanced technological systems is imperative for ensuring operational reliability and safety. While there is a growing interest in using artificial intelligence for fault detection, current methodologies often exhibit limitations in utilizing comprehensive system information and sensor data. Hidden faults within collected data further highlight the need for advanced analysis techniques.

This study introduces a novel deep learning-based framework designed to predict faults and extract insights from complex system datasets. The model, consisting of LSTM-autoencoder and BiLSTM classification components, effectively reduces feature dimensions, thereby enhancing fault detection accuracy. The autoencoder's latent layer identifies prominent features across various dimensions, while BiLSTM classification conducts bi-directional analysis using these features from both healthy and faulty states, facilitating early fault detection.

Experimental results demonstrate the model's efficacy, achieving an accuracy of 79.48% in predicting incipient faults 30 seconds before a serious malfunction occurs. This underscores the significant potential of the proposed framework in enhancing operational safety and reliability in complex systems. Moreover, the study emphasizes the importance of leveraging comprehensive data and advanced analysis techniques for early fault detection.

Keywords: quadrotor fault prognostics, early fault detection, deep learning, autoencoders, LSTM.

✉Corresponding author. E-mail: *mehmet_ozcan@eskisehir.edu.tr*

1. Introduction

In the modern age, advanced technological devices have become central components of daily life, fundamentally shaping interactions with the world. Among these innovations, quadrotors, also known as quadcopters or drones, have gained significant popularity due to their diverse applications ranging from transportation to search and rescue missions (Belcastro et al., 2017; Zhao et al., 2018). However, the development and maintenance of quadrotors, equipped with complex sensors, navigation systems, and communication devices, often entail considerable costs. Emerging demands for enhanced safety, and cost-effective maintenance necessitate the development of innovative preventive maintenance and inspection protocols, alongside improvements in design and manufacturing processes (Afshari & Pourtakdoust, 2018).

Quadrotors, comprising intricate systems, are particularly sensitive to mechanical failures, environmental conditions, and unpredictable behaviors. Along with the nonlinear relationships, aerodynamic inefficiencies in the design, rotor interactions generating undesired lateral forces, and complex control algorithms designed to stabilize and guide the vehicle further complicate the system. Despite advancements in control mechanisms, experiencing control loss and crashes are still commonly observed. While some failures occur suddenly, others are incipient faults that intensify over time, eventually leading to a crash by affecting the subsystems. The consequences of quadrotor crashes often result in irreparable damage and significant financial losses, underscoring the imperative for effective early fault detection mechanisms.

In the field of unmanned aerial vehicles (UAVs), numerous studies have investigated both traditional and artificial intelligence (AI) techniques for diagnosing and detecting faults (Ignatovich et al., 2013; Yasniy et al., 2024). These studies typically concentrate on identifying specific sensor and equipment failures (Puchalski & Giernacki, 2022). Model-based approaches, such as the use of Kalman filters, rely on mathematical models to eliminate noise from sensor data (Hajiyev, 2016; Liu et al., 2016; Vural & Hacızade, 2016; Wang et al., 2019). While effective in detecting existing faults, these methods require an accurate system model and comprehensive observation of each component, which is challenging to achieve.

Deep learning methods, on the other hand, rely on processing flight data from various sensors, including vibration, acoustic emissions, speed, acceleration, current, and temperature. For instance, vibration data are widely used for diagnosing faults in bearings and gearboxes, while acoustic emission data detect faults in bearings and gears, especially in low-speed operating conditions. Despite the success of these methods in specific fault scenarios, they often neglect the thorough detection of early-stage faults that lead to quadrotor crashes, indicating a need for additional exploration in this domain.

This study aims to bridge this gap by introducing a proactive fault detection method based on deep learning specifically tailored for quadrotors. The approach seeks to analyze comprehensive flight data from all sensors in the system, including gyroscopes, accelerometers, magnetometers, temperature sensors, GPS sensors, and current sensors, to enable early detection by revealing confidential and useful information. The novelty of the proposed framework lies in its combination of long short-term memory (LSTM)-autoencoder and bidirectional LSTM (BiLSTM) classification components, which effectively reduce feature dimensions and enhance early fault detection accuracy. The autoencoder's latent layer identifies prominent features across various dimensions, while the BiLSTM classification conducts bidirectional analysis using these features from both healthy and faulty states, facilitating early fault detection.

The experimental results showcase the model's ability, successfully predicting the onset of faults with a 79.48% accuracy, a half-minute prior to critical system failures. The development of an early warning system utilizing the model for potential faults holds promising prospects for enhancing the safety and reliability of quadrotor operations and other complex setups. Additionally, the study highlights the role of using expansive datasets and cutting-edge analytic methodologies in the early identification of potential faults. In summary, this article contributes to the field by:

- Developing a deep learning-based model for early detection of incipient faults in quadrotors.
- Enhancing the model's efficiency through deep learning-based dimension reduction methods.
- Creating a dataset by collecting real flight data and applying the model to these real quadrotor flights.

The rest of this paper is organized as follows. In Section 2, studies on fault detection in aircraft are introduced and discussed. Section 3 explains the model and its processes. Section 4 presents the experimental results and findings. Finally, section 5 concludes the paper with a summary of the key points and suggestions for future research.

2. Literature review

Studies on fault detection in quadrotors are categorized into model-based, data-based, and hybrid approaches, focusing on the detection of failures in one or several sensors and components (Puchalski & Giernacki, 2022). Each of these approaches has distinct strengths and limitations which are crucial to understand for advancing the field.

Model-based methods rely on mathematical models, including linear or nonlinear models, to detect faults. The most effective methods involve the use of Kalman filters to eliminate noise from sensor data (Hajiyev, 2016; Liu et al., 2016; Vural & Hacızade, 2016; Wang et al., 2019). While model-based approaches have shown success in detecting existing faults, comprehensive detection of all malfunctions in a quadrotor requires observation of each component and an accurate system model. A significant limitation is their inability to model the complex, nonlinear dynamics of quadrotors precisely, which leads to inaccurate fault detection and diagnosis under varied operating conditions. A common issue in model-based methods is the simulation-to-reality gap. Models often fail to capture real-world complexities, leading to discrepancies between simulated results and actual performance (Zhang et al., 2023). This limitation impacts the reliability of fault detection in real-world scenarios.

Data-based methods utilize historical and real-time data from various sensors. Deep learning techniques, such as convolutional neural networks (CNN) and LSTM, are commonly employed. For instance, Zhang et al. (2023) used a deep learning model for quadrotor propeller fault diagnosis, achieving notable accuracy. However, the reliance on extensive labeled datasets for training is a significant limitation. The collection and labeling of such comprehensive datasets are time-consuming and costly. Additionally, data-based methods struggle with generalizing to new, unseen fault types not represented in the training data. These methods rely heavily on the quality and availability of sensor data, and incomplete, noisy, or biased datasets significantly degrade the performance of fault detection algorithms. Furthermore, obtaining labeled data for every possible fault condition is impractical (Puchalski & Giernacki, 2022). Besides, advanced machine learning techniques, such as deep neural networks, require substantial computational resources for training and real-time operation. This requirement limits their applicability in resource-constrained environments, such as onboard UAV systems (Bondyra et al., 2022).

Hybrid approaches combine model-based and data-based methods to utilize the strengths of both. For example, Anidjar et al. (2023) proposed a framework using deep learning for anomaly detection in UAV sound waves, which integrates model predictions with real-time data analysis. While hybrid methods show promise in enhancing fault detection accuracy, they inherit the limitations of both constituent approaches, such as the need for accurate models and extensive data. Both model-based and data-based methods often struggle to generalize to new, unforeseen fault types. This limitation is particularly critical in dynamic environments where new fault modes can emerge (Liu et al., 2020).

Deep learning methods rely on processing flight data, where they are either integrated into the model or processed independently. Data are collected continuously from various sensors existing on a system, including vibration, acoustic emissions, speed, acceleration, current, axes of movement, and temperature. For example, vibration data find broad application in diagnosing faults in bearings and gearboxes. Acoustic emission data, especially in environments with low-speed operating conditions and low-frequency noise, have the potential to detect faults and deformations in bearings and gears. Instantaneous speed data are typically used for motor fault diagnosis because they have high resistance to external influences. Current or voltage data play an important role in diagnosing faults in electrical components on the systems. Therefore, existing AI based studies are roughly categorized based on the applicable machine learning methods and the components related to obtained data.

Beginning with Zhang et al. (2023), their deep learning model effectively addresses the simulation-to-reality gap in quadrotor propeller fault diagnosis. Through the use of newly identified features and domain adaptation techniques, the model achieves commendable accuracy, marking a significant advancement in the field. Similarly, Pose et al. (2023) contribute to the domain with a neural network-based method for estimating propeller damage. Their approach exhibits satisfactory performance in extrapolating from limited data, with the added advantage of quick prediction times, implying its potential for onboard application. Another method proposed by Liu et al. (2020) is for identifying propeller damage using CNN and transfer learning techniques, achieving high accuracy rates. Iannace et al. (2019) construct a classification model for detecting unbalanced blades in quadrotor propellers using artificial neural networks (ANN), while Yang et al. (2021) focus on detecting early minor faults in propellers using deep residual shrinkage network.

Numerous studies also aim to identify faulty components through acoustic data analysis. Bondyra et al. (2022) present fault detection and isolation systems offering high detection rates and precise localization of single actuator faults. Their work demonstrates practical applications in onboard acoustic data acquisition and processing. Similarly, Anidjar et al. (2023) propose a framework for detecting anomalies in sound waves emitted from UAVs using deep-learning methods. Their approach demonstrates high accuracy in anomaly detection with fewer parameters, making it suitable for real-time applications.

Guo et al. (2018) propose a transformative approach by utilizing CNN to extract features from residual signals of UAV sensor faults, yielding promising outcomes in fault detection. Their method utilizes short-time Fourier transform to convert residual signals from the sensors into time-frequency maps, which are then analyzed using a CNN to extract features and implement fault diagnosis. Olyaei et al. (2018) introduce a novel algorithm using color images from time-frequency-amplitude graphs for fault classification through deep neural networks, achieving satisfactory accuracy in fault identification, particularly in sensor and actuator fault scenarios. Fu and Che (2021) present a fault diagnosis model for UAVs, combining conformal Fourier transform and an improved self-organizing feature map neural network. Their method enhances fault diagnosis and pattern recognition capabilities for efficient clustering and fault diagnosis. Huang et al. (2021) introduces a novel fault detection and classification method integrating time-frequency analysis and deep learning technologies. Initially, randomly generated datasets undergo transformation into the time-frequency domain using short-time Fourier transform, resulting in time-frequency graphs. Subsequently, these graphs are utilized to train a deep network, enabling rapid and accurate classification of fault types.

Zheng et al. (2021) employ a compound fault dictionary and machine learning models for enhanced fault diagnosis. The method enables the labeling of flight data with multiple fault modes corresponding to simultaneous single faults. Altinors et al. (2021) focus on fault diagnosis of brushless DC motors in UAVs, achieving high accuracy using decision tree, support vector machines (SVM), and k-nearest neighbor (KNN) algorithms, with the added advantage of real-time operation on embedded systems.

Several studies have proposed innovative methods using vibration data. One approach involves (Ozkat et al., 2023) detecting the remaining useful life of UAVs using collected vibration data and long short-term memory (LSTM) to forecast future mean peak frequency values. Another study (Chen et al., 2021) utilize wavelet analysis and gate recurrent units for fault detection using motor vibrations or wind-induced noise and effectively analyzes sensor data and detects faults using residuals and threshold-based techniques. Additionally, a novel method (Jiang et al., 2015) is proposed for detecting and identifying rotor faults in quadrotors using airframe vibration signals. This method employs three-level wavelet packet decomposition and ANN to analyze vibration signals and design a fault diagnostic system, validated through experimental data collected from quadrotor hovering experiments.

In Ouadine et al. (2020) and Jing and Pebrianti (2016), authors propose fault detection methods combining model based approaches with ANN, and demonstrate effectiveness in fault classification and identification. Jing et al. (2017) develop a fault detection algorithm using Kalman filter and ANNs for quadrotors, achieving decent detection accuracy with minimal time delay.

Al Younes et al. (2016) introduce a fault-tolerant diagnosis and control approach for sensors in quadrotors, to address bias-type sensor faults by introducing an intelligent output-estimator, during real flight experiments. Ai et al. (2021) contribute a novel fault diagnosis scheme, utilizing wavelet packet translation for fault feature extraction, based on the optimized deep forest algorithm. Erfanian and Ramezani (2022) discusses the successful utilization of the bidirectional LSTM (BiLSTM) algorithm for detecting actuator faults in quadrotors. By simulating the quadrotor model and employing BiLSTM algorithm, the study successfully detects actuator faults through data extraction and network training, demonstrating the potential of BiLSTM in fault detection applications for quadrotor systems.

3. Methodology

Historically, fault diagnosis in quadrotors heavily relied on human expertise, but artificial intelligence algorithms, also known as intelligent fault diagnosis (IFD), have enabled quadrotors to automatically identify, classify, and potentially predict their own fault conditions. This not only reduces the need for human intervention but also equips operators to proactively manage quadrotor health, thereby ensuring secure and uninterrupted operations.

However, as mentioned earlier, previous studies have focused on detecting faults for one or a few components and on detecting faults as they occur. In this study, a framework has been established for detecting faults that cause a fall in quadrotors after a certain time. Overview of the framework is illustrated in Figure 1. The framework is comprised of three main stages: data preparation, dimensionality reduction and fault prediction. After preparing a large amount of sensor data collected from various flights for use in the mentioned methods, an autoencoder using LSTM is initially trained to both eliminate unnecessary information and reduce dimensionality. Subsequently, the encoded data containing more useful information from the latent layer is fed into a classification algorithm using BiLSTM to make early predictions of faults. The proposed structure has been detailed step by step in a top-down strategy throughout the following.

Although data processing appears to be the first step in Figure 1, it is necessary to provide information about the data used in this study. Firstly, it should include flight data resulting in crashes so that the algorithm learns from those incidents as well. However, such data are very limited compared to healthy ones. There are numerous studies that inject synthetic values emulating faults into flight data, thereby simulating various fault events in this manner (Chen et al., 2021; Erfanian & Ramezani, 2022; Guo et al., 2018; Huang et al., 2021; Jing & Pebrianti, 2016; Jing et al., 2017). The flight of a UAV is carried through highly nonlinear systems; therefore, failure detection models constructed on such data do not perform well under real flight circumstances. Furthermore, studying early failure detection on such data is not possible since there is not a symptom leading to a breakdown. Hence, these issues led us to search for natural crash events in quadrotors.

Accidents involving quadrotor operators often prompt the sharing of experiences on online blogs to exchange information among peers facing similar challenges. For this study, data were systematically collected and analyzed from these online sources, focusing on incidents where the cause of the accident was uncertain. During the selection process, data from 10 flights that experienced crashes were obtained from online blogs, and the data were organized to exclude information beyond the start point of uncontrolled descent. Additionally, collaboration with a team conducting quadrotor experiments on campus provided another set of 10 flights.

Consequently, a dataset comprising 20 flights was compiled; these flights were classified into 10 as faulty (crash-related) and 10 as healthy. The faulty flights, sourced from online blogs, had a total duration of 126 minutes and 16 seconds, while the healthy flights from campus experiments had a total duration of 70 minutes and 25 seconds.

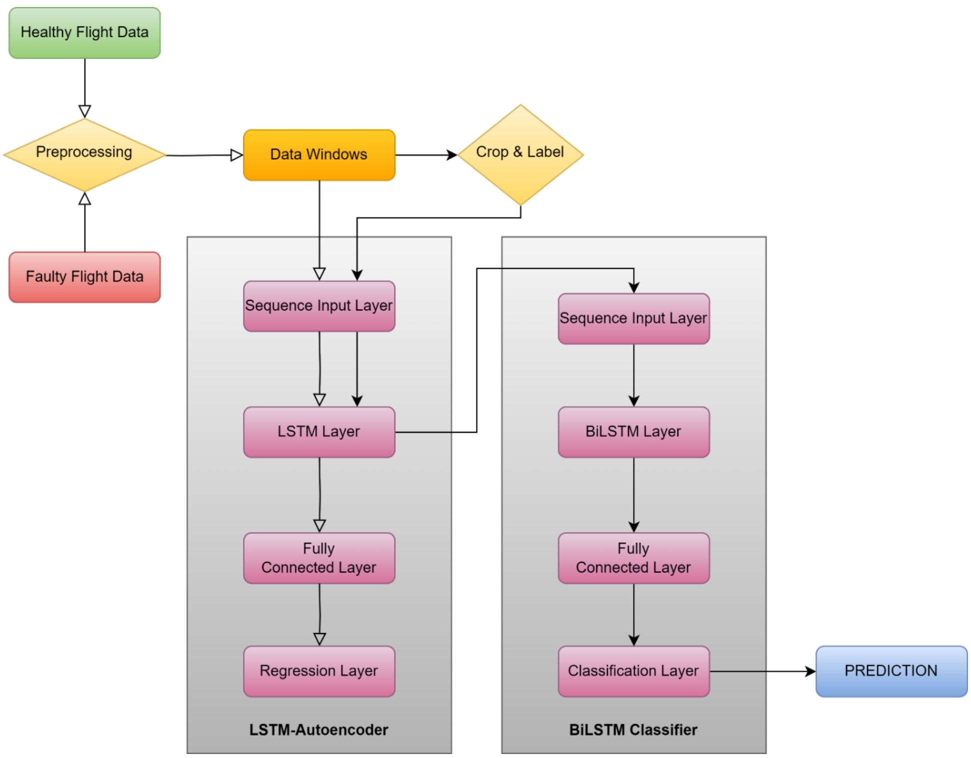


Figure 1. Proposed framework in this study

The collected data included log records containing measurements from various sensors onboard the quadrotors. To ensure the validity of early fault detection, careful selection prioritized potentially useful data from these logs, focusing on features that could provide insights into flight health and fault detection. This process involved filtering out irrelevant, unnecessary, or redundant sensor readings that could complicate fault detection.

From the extensive sensor data available, 50 features across different sensors were identified and prioritized as attributes for the dataset. The selected features encompass a wide range of parameters including estimated orientation (roll, pitch, yaw), geographic coordinates (latitude, longitude), vehicle performance metrics (gyro drift error, altitude, climb rate), controller gains, throttle and servo inputs/outputs, battery status, velocity components, accelerometer data, and more. This comprehensive dataset, combining insights from real-world accident reports shared online and controlled sensor data from campus experiments, serves as the foundation for analyzing and improving early fault detection mechanisms in quadrotor operations.

When feeding data with varied distributions directly into an ANN, it can lead to increased error rates due to the network's sensitivity to the scale and distribution of input data which cause a slower convergence. To mitigate this issue, the data should undergo normalization first. Normalization transforms the data so that each feature (sensor data in this case) has a similar scale and distribution. For this purpose, z-score normalization was applied to the data. Z-score normalization reorganizes the sensor data to have a mean of 0 and a standard deviation of 1. This adjustment ensures that each sensor's data points are expressed in terms of their relationship to the mean and standard deviation of the entire dataset. Normalizing the data in this way reduces the variability that could cause the ANNs to prioritize features with larger numerical ranges, potentially skewing the learning process. Consequently, normalized data helps the ANN to learn more effectively by ensuring that all input features contribute equally to the model's training, ultimately improving its accuracy and convergence rate.

After normalization process, the data is divided into small data windows (segments) by the sliding window method. Experimenting various sizes, window size is decided as 10 which refers to 1 second. Changing shift sizes are also experimented and it is determined as 5 which refers to 0.5 second long. Note that size of each data sample becomes 500 (10 × 50).

The next stage involves extracting meaningful information from the data. Here, dimensionality reduction was applied to all available data using a deep learning method proven to be successful for complex systems. This process is advantageous for eliminating redundant information and enhancing the accuracy of diagnostic outcomes. To achieve this, an LSTM-autoencoder structure was proposed.

An LSTM-autoencoder is a neural network architecture that integrates the principles of LSTM networks and autoencoders, specifically effective in handling sequential data. When combined, they form a powerful model for reconstructing sequence-to-sequence data and extract features of lower dimensionality from the latent layer.

LSTM represents a significant advancement in neural network architecture, often seen as an extension of recurrent neural networks (RNN). While RNN offer a form of "short-term memory," allowing them to utilize recent information for ongoing tasks, LSTM takes this a step further by introducing the concept of "long-term memory." Unlike RNN, which retain information only from a certain point in time, LSTM maintains a comprehensive record of all past information, making it accessible to the current neural node. This extended memory capacity enables LSTM to capture and utilize context and dependencies over more extended sequences, making them especially powerful in handling sequential data and tasks that require a broader understanding of past information.

In a standard LSTM unit, which is illustrated in Figure 2, several key components (gates) work together to enable its extended memory and precise information flow control. These components include the cell, an input gate, an output gate, and a forget gate. The cell retains values over extended time intervals, effectively functioning as the long-term memory repository. Meanwhile, the input gate, output gate, and forget gate work in coordination to manage the flow of information into and out of the cell. The cell state maintains a record of the network's long-term memory, encompassing a list of past information. In contrast, the previous hidden state serves as a form of short-term memory, capturing the network's output from the preceding time step. Finally, the input data carries the current time step's input value, allowing the LSTM to process and incorporate the most recent information into its memory and computations. In the LSTM architecture, the three primary gates, which allow the network to effectively process sequential data by maintaining and updating the network's memory over time, are described below.

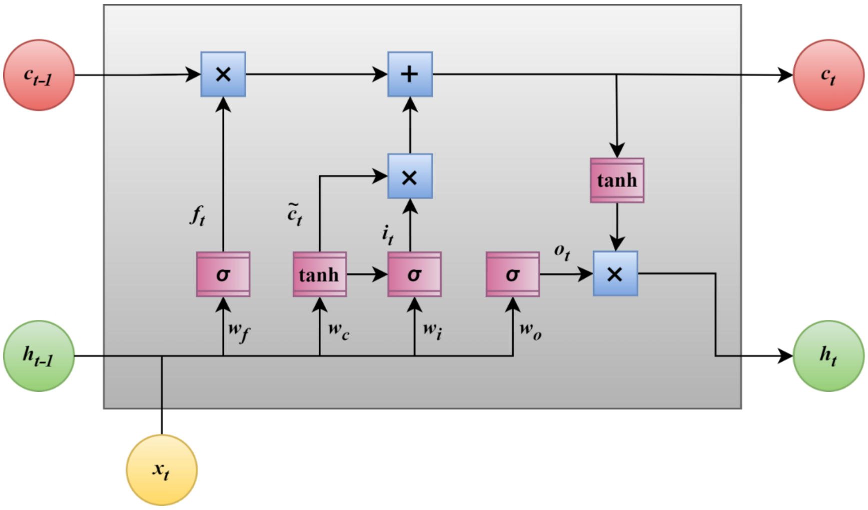
Step 1: Forget Gate

The forget gate determines which pieces of information within the cell state should be preserved or discarded. It generates the outputs using a sigmoid activation function by evaluating both the previous hidden state and the new input data. These outputs, represented as f_t, tend to be closer to 1 for relevant information and closer to 0 for less relevant data. The mathematical representation of this process is given in Equation 1, which involves weights w_f, bias b_f, and the concatenation of the previous hidden state h_{t-1}, and the current input x_t.

$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] - b_f\right). \tag{1}$$

Step 2: Input Gate

The input gate serves a dual purpose. First, it assesses whether incoming information, including the previous hidden state and new input data, is worth retaining in the cell



Note: the variables depicted in the figure are represented in equations 1-6, with their explanations provided in the corresponding paragraphs.

Figure 2. LSTM unit (source: created by authors)

state. Subsequently, it determines the extent to which the new data will be added to the cell state. To achieve this, the input gate undergoes two processes. The first involves calculating a new information vector $\tilde{c}_t$, using the hyperbolic tangent (tanh) activation function as follows:

$$\tilde{c}_t = \tanh\left(w_c\left[h_{t-1}, x_t\right] + b_c\right). \quad (2)$$

The second process identifies the components of the new input using a sigmoid-activated network, resulting in an i_t vector as shown in the following equation:

$$i_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] - b_i\right). \quad (3)$$

These two processes are combined through pointwise multiplication as depicted in Equation (4) to regulate the magnitude of the new information. This combined vector is then added to the cell state, updating the network's long-term memory.

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t. \quad (4)$$

Step 3: Output Gate

In the final step, the output gate determines the new hidden state. It applies a sigmoid activation function to the previous hidden state and the current input data as shown in Equation (5) to obtain the filter vector o_t.

$$o_t = \sigma\left(w_o\left[h_{t-1}, x_t\right] - b_o\right). \quad (5)$$

The cell state is passed through a tanh activation function to constrain its values to the range of [–1, 1]. This squished cell state is then multiplied pointwise with the filter vector o_t to produce the new hidden state h_t as follows:

$$h_t = o_t * \tanh\left(c_t\right). \quad (6)$$

Both the new cell state c_t, and the new hidden state h_t are outputs of this unit. These values are propagated to the next LSTM unit in the sequence, with c_t becoming the previous cell state for the next unit, and h_t serving as the previous hidden state.

An autoencoder is a neural network that copies values from the input layer to the output layer. That is, the data given as input to the neural network is reconstructed in the output layer. The most significant feature that distinguishes autoencoders from feed-forward ANN is that the input dataset and the output dataset are the same, resulting in the number of neurons in the output layer being equal to the number of neurons in the input layer. In an autoencoder model, the number of neurons in the input layer is usually higher than the number of neurons in the hidden layers. Autoencoders (Figure 3) is broken down into three key components: encoding, decoding, and the optimization process aimed at minimizing reconstruction loss.

Initially, the autoencoder begins its operations by receiving input data, which includes all the relevant features, into its input layer. This information is then transmitted to the hidden layer(s) through a mathematical process, which is represented by Equation (7).

$$y_i = f\left(\sum_{j=1}^{n} x_j w_{ji}\right), \quad (7)$$

where x_j represents the value associated with the j-th neuron in the input layer, y_i represents the value transferred to the i-th neuron in the hidden layer, n corresponds to the total number of neurons within the input layer, w_{ji} denotes the weight that connects the j-th neuron in the input layer to the i-th neuron in the hidden layer, and finally, f represents the activation function applied to each neuron. These components collectively contribute to the encoding process, a pivotal step in the functioning of the autoencoder.

Following the encoding stage, the obtained values are then transmitted to the output layer, denoted as $x'j$, for further processing (decoding), as demonstrated in Equation (8).

$$x'_j = f\left(\sum_{i=1}^{m} y_i w_{ji}\right), \quad (8)$$

where j is the neuron index number within the output layer, the i is the neuron index number residing in the hidden

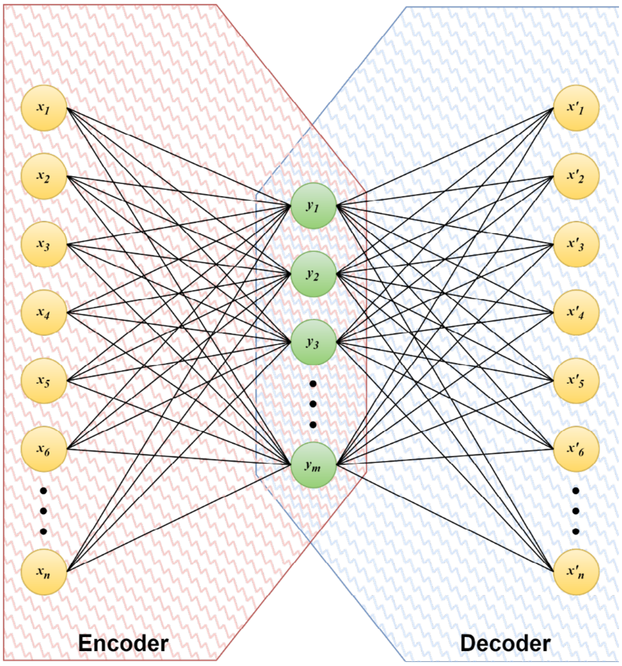


Figure 3. Autoencoders (source: created by authors)

layer (represented as y_i), w_{ji} is the weight connecting the i-th neuron of the hidden layer to the j-th neuron of the output layer, m is the number of neurons situated within the hidden layer, and f is the activation function. Equation (8) encapsulates the essence of this transfer and transformation process, which ultimately yields the final values in the Autoencoder's operation.

At the core of the autoencoder model lies a fundamental objective: to ensure that the obtained output closely resembles the value initially present in the input layer. To achieve this, the optimization process involving weight calculation and continuous updates comes into play. These weight adjustments are driven by the backpropagation algorithm, a crucial operation of ANN's training process. This algorithm works diligently to minimize the square of the difference between these two values, as elucidated in Equation (9). In essence, the backpropagation algorithm tunes the network's parameters, enabling it to learn and replicate the essential patterns and features in the input data, facilitating the reconstruction of the original input, which is a central goal of the autoencoder model.

$$min\left(\sum_{j=1}^{n}\left(x'_j - x_j\right)^2\right). \quad (9)$$

The LSTM-autoencoder model, combines LSTM networks and autoencoders for dimensionality reduction in time-series data (Lazzara et al., 2022; Said Elsayed et al., 2020). Initially, the original dataset is transformed into time sequences containing fixed-length windows of data. The LSTM encoder unfolds these sequences, processing them with multiple LSTM cells capturing crucial information from each time step. The encoded representation is then propagated through a decoder to reconstruct the original data sequence. Using LSTM-autoencoder, meaningful and reduced features are obtained obtained from the latent layer of LSTM-autoencoder, subsequently enhancing the performance of the classification layer.

In the final stage, a classification layer was employed to determine the system's health. During this phase, various machine learning techniques such as ANN (Fu & Che, 2021; Iannace et al., 2019; Jiang et al., 2015; Jing & Pebrianti, 2016; Jing et al., 2017; Ouadine et al., 2020; Pose et al., 2023), Random Forests (Bondyra et al., 2018), SVM (Altinors et al., 2021), KNN (Altinors et al., 2021), Decision Trees (Altinors et al., 2021), Fuzzy C-means (FCM) Clustering (Wei et al., 2020), Ensemble Learning (Duncan Imbassahy et al., 2020) can be utilized. However, considering the data structure and its suitability for time series data, a BiLSTM-based deep learning layer was implemented.

The conventional LSTM network primarily focuses on past data context while overlooking future context. Using LSTM to regenerate inputs in time series data is an effective strategy for autoencoders. However, the classification performance can be enhanced even further by employing bidirectional data processing for classification. Given the prominent temporal dependencies within time series sensor data, this approach falls short in fully capturing the data's intricacies. To address this limitation and incorporate both past and future temporal information, a BiLSTM structure is adopted for fault detection. This specialized architecture processes the sequence data in dual directions, both forward and backward, utilizing two hidden layers. Consequently, it combines the insights from both directions into a single output layer. The structure of the BiLSTM network is presented in Figure 4, showcasing its bidirectional data flow and enhanced capacity to comprehend the temporal dynamics of the input data.

BiLSTM network processes input sequences, represented with x_i, in both forward and backward directions with LSTM units. In the forward pass, data flows conventionally from the sequence's start to the end, capturing past dependencies. Simultaneously, in the backward pass, data is processed in reverse, from end to start, enabling the network to grasp future context. Each LSTM unit within these two passes maintains hidden states, representing learned information from both past and future contexts.

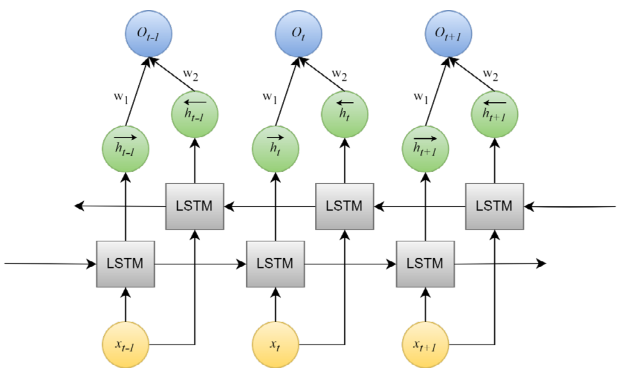


Figure 4. BiLSTM structure (source: created by authors)

These forward ($\overrightarrow{h_t}$) and backward ($\overleftarrow{h_t}$) hidden states are then merged as in Equation 10, where g represents ReLU function, creating a unified representation encompassing bidirectional context.

$$O_t = g\left(w_1\overrightarrow{h_t} + w_2\overleftarrow{h_t}\right). \quad (10)$$

This merged representation O_t is fed into an output layer tailored to the specific task, such as classification or prediction. During training, the model minimizes the loss between its predictions and actual targets using back-propagation and optimization techniques. During inference, the BiLSTM uses its bidirectional understanding to make predictions or generate output, proving particularly effective in tasks where considering both past and future context is critical.

In this study, the BiLSTM classifier is utilized for detecting incipient faults in quadrotors. To enhance fault detection using BiLSTM classification, the importance of integrating dimensionality reduction techniques to optimize the structure's efficiency is emphasized, particularly in terms of complexity. Furthermore, this strategic reduction of unnecessary inputs is paramount in refining the quality of the outcomes.

As a result, a structure consisting of three stages was introduced: data processing, LSTM-autoencoder, and BiLSTM classifier. The adoption of the LSTM-autoencoder unit is advocated as a viable approach with its success on time-based data. The primary goal of the first phase of the autoencoder is to represent the flight data with a low margin of error by utilizing the robust framework of the LSTM-autoencoder. As a crucial next step, the intention is to discard the decoder component of this architecture, thus unveiling a latent layer. Rather than directly employing the original data, this strategy hinges on the extraction of outputs from the latent layer, which serves as the critical intermediary between raw data and BiLSTM classifier. In order to provide real-time alerts upon the detection of a fault during a flight, real-time monitoring of the flight is necessary, and the data from the flight needs to be processed in a segmented manner rather than as a whole. In the proposed framework, data generated during the flight is processed in a segmented fashion to perform fault detection. As an output, the BiLSTM classifier classifies the segments into two categories: healthy and faulty. BiLSTM is particularly useful for tasks where understanding the complete context is essential with its ability to capture information from both past and future contexts simultaneously which sets it apart from traditional LSTM and other deep learning methods. Since each data segment contains sequential data and needs to be decided whether faulty or not, BiLSTM classification is utilized for the fault detection process.

To evaluate the performance of the proposed LSTM-autoencoder model integrated with the BiLSTM classifier, several evaluation metrics were employed. The metrics provided below (Equations (11), (12), (13), and (14)) offer a comprehensive understanding of the model's effectiveness in identifying healthy and faulty flights.

Accuracy: Accuracy is the ratio of correctly predicted instances to the total instances. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100, \quad (11)$$

where:

- TP (True Positives): The number of correctly predicted faulty flights.
- TN (True Negatives): The number of correctly predicted healthy flights.
- FP (False Positives): The number of healthy flights incorrectly predicted as faulty.
- FN (False Negatives): The number of faulty flights incorrectly predicted as healthy.

Precision: Precision is the ratio of correctly predicted positive (faulty) observations to the total predicted positives. It is particularly important in scenarios where the cost of false positives is high. It is calculated using the formula:

$$Precision = \frac{TP}{TP + FP}. \quad (12)$$

Recall (Sensitivity): Recall is the ratio of correctly predicted positive (faulty) observations to all observations in the actual class. It is crucial for detecting faults, ensuring that as many actual faults are identified as possible. It is defined as:

$$Recall = \frac{TP}{TP + FN}. \quad (13)$$

F1-score: The F1-score is the weighted average of Precision and Recall, providing a balance between the two. It is especially useful when there is an uneven class distribution, as in cases where faults (1) are rare compared to healthy states (0). It is calculated as:

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (14)$$

The model was evaluated using these metrics to ensure a comprehensive assessment of its performance in distinguishing between healthy and faulty flights. The selected metrics contribute to a precise evaluation of the model's capabilities, particularly in accurately identifying faults while minimizing false alarms.

4. Results

A major drawback in detecting faults in quadrotors is the highly imbalanced nature of the data. The likelihood of a healthy flight is much more common than that of a crash. Even when a crash event occurs, there is only a slight chance of having sufficient pre-failure flight time for analysis. Machine learning algorithms struggle to accurately predict outcomes with such data. In this study, data totaling 126 minutes and 16 seconds from 10 different flights that experienced crash events were collected from blogs, along with 70 minutes and 25 seconds of data from 10 different healthy flights conducted on our campus. A dataset was created by consolidating all this data.

For the model to detect incipient faults during flights, these faults need to be introduced to the model. To achieve this, the flight segments created must be labeled as either healthy or faulty before being provided to the system. In this study, the aim is to detect incipient malfunctions 30 seconds before the start of an uncontrolled descent. Accordingly, in flights where accidents occurred, the flight segments from 30 seconds before the start of the descent until the moment the descent begins are labeled as faulty. To ensure an equal distribution of the data, in flights without accidents, the last 30 seconds of the flight are selected and labeled as healthy.

In the pursuit of fault detection with the BiLSTM classifier, emphasis has been placed on dimensionality reduction to enhance the efficiency of the structure in terms of time complexity and achieve more refined outcomes by eliminating unnecessary input features. During fault detection, instead of using the original data directly, outputs obtained from the latent layer of the data fed into the LSTM-autoencoder structure were utilized to feed the BiLSTM classifier network depicted in Figure 1. This approach began with LSTM-autoencoder experiments.

To retain all pertinent information from the flight data during the LSTM-autoencoder phase, the objective is to minimize the error while reconstructing the flight data. To achieve this, a structured model was developed incorporating sequence input layer, LSTM layer with sigmoid gate activation and tanh state activation, fully connected layer, and regression layer. Training employed the Adaptive Moment Estimation (ADAM) optimizer with a learning rate of 0.01, utilizing a batch size of 50 over 200 epochs. The loss function employed was Mean Squared Error (MSE), augmented by L2 regularization to enhance generalization. A series of experiments were conducted using both healthy and faulty data. Whole data is partitioned as 80% train, 10% test and 10% validation which is vital to avoid overfitting. To evaluate the performance of networks with different neuron counts in the latent layer, nine different networks were created by gradually increasing the neuron count in the latent layer from 5 to 45 in increments of 5.

Figure 5 depicts the graph of actual and autoencoder-reconstructed values for normalized pitch and altitude signals obtained from the respective sensor. The employed autoencoder in this experiment features a latent layer with 20 neurons. Likewise, each set of sensor data was examined with varying neuron counts in the latent layer, and the Root Mean Squared Error (RMSE) values for each flight are presented in Table 1. The overall average RMSE for the entire dataset is 0.3898.

The results demonstrate that the flights having crash (C1-C10) and healthy flights (H1-H10) are represented precisely by LSTM-autoencoders particularly when the number of neurons in hidden layer is near to the feature count. The autoencoder's error rate decreases with an increase in the number of neurons within the hidden layer. This observation indicates that precise dimensionality reduction is achieved by keeping the neuron count above a certain number where significant information is not lost.

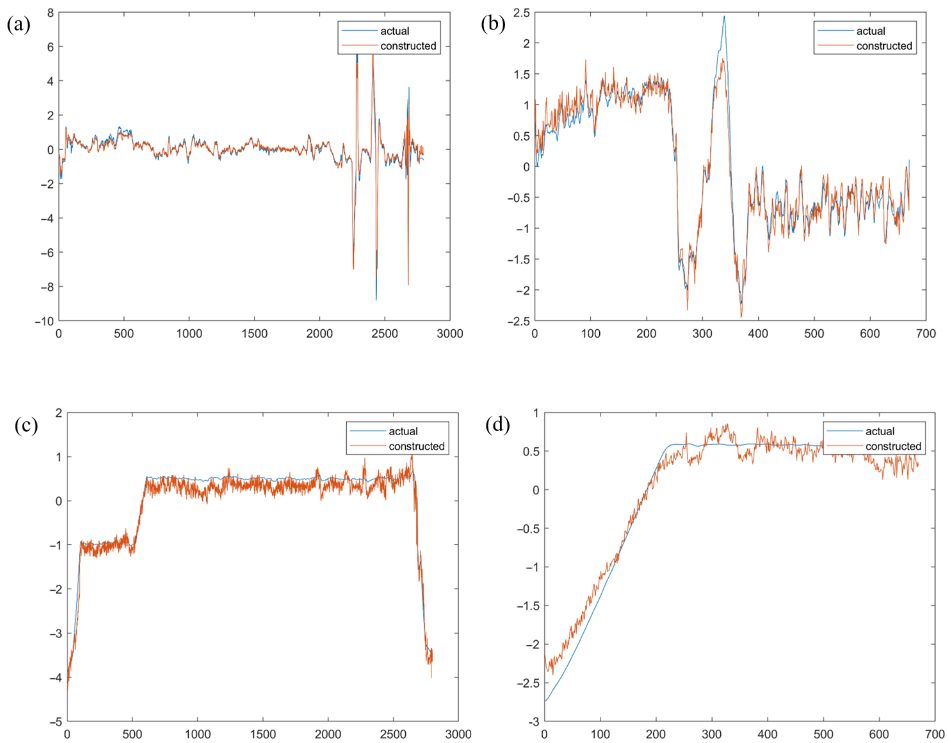


Figure 5. Normalized actual signals and their representatives reconstructed with LSTM-autoencoder having 20 neurons in latent layer: a) pitch signal of a healthy flight; b) pitch signal of a faulty flight; c) altitude signal of a healthy flight; d) altitude signal of a faulty flight

Table 1. Root mean squared errors of trained LSTM-autoencoder on flight data with varying neuron counts in latent layer

Flights	Instance	5	10	15	20	25	30	35	40	45
C1	10897	0.7667	0.6614	0.5494	0.4809	0.3996	0.3142	0.2293	0.1444	0.0637
C2	6507	0.6946	0.5649	0.4957	0.4105	0.3469	0.2706	0.1903	0.1034	0.0409
C3	5523	0.8009	0.6981	0.608	0.512	0.4429	0.3416	0.2623	0.1607	0.0654
C4	7459	0.721	0.5932	0.4834	0.4055	0.3385	0.264	0.1777	0.1017	0.0446
C5	3790	0.7723	0.6876	0.5638	0.4559	0.3779	0.3069	0.2315	0.1258	0.07
C6	1057	0.7163	0.5892	0.5254	0.465	0.3973	0.3182	0.2352	0.1312	0.0635
C7	10987	0.7709	0.5838	0.5207	0.4412	0.3573	0.2737	0.1853	0.1017	0.0461
C8	25118	0.6701	0.5294	0.4584	0.3956	0.3167	0.247	0.1759	0.1054	0.0457
C9	4230	0.784	0.6733	0.6055	0.5404	0.4719	0.3791	0.2894	0.1611	0.0847
C10	670	0.8147	0.6528	0.586	0.5177	0.4303	0.3576	0.2348	0.1291	0.0587
H1	1935	0.7887	0.6783	0.587	0.5005	0.4091	0.333	0.2318	0.1463	0.0597
H2	1624	0.7801	0.671	0.5843	0.5004	0.4227	0.3157	0.2393	0.1501	0.066
H3	2071	0.7997	0.6391	0.5636	0.4629	0.3731	0.2864	0.207	0.1273	0.062
H4	1974	0.7837	0.6508	0.5636	0.4759	0.391	0.3136	0.2334	0.1372	0.0592
H5	2648	0.7425	0.6284	0.5415	0.4727	0.3868	0.2968	0.2252	0.1022	0.0458
H6	8057	0.7995	0.6639	0.5756	0.4973	0.3989	0.3159	0.235	0.1258	0.0473
H7	7388	0.7532	0.5937	0.4992	0.4016	0.3328	0.2506	0.1997	0.125	0.0532
H8	9254	0.7395	0.616	0.5203	0.438	0.3632	0.2702	0.2047	0.1258	0.052
H9	2806	0.7531	0.6454	0.5634	0.4872	0.4186	0.3455	0.2545	0.1393	0.0625
H10	3341	0.773	0.6427	0.5575	0.4724	0.3664	0.2866	0.2171	0.1244	0.0581
Average:		0.7612	0.6331	0.5476	0.4666	0.3870	0.3043	0.2229	0.1283	0.0574

After the dimensionality reduction phase, the subsequent phase entails preemptive fault detection through the classification network. To assess the efficacy of BiLSTM in fault detection, a series of experiments were conducted using a BiLSTM classifier network. This network architecture includes, a sequence input layer, a BiLSTM layer with 30 hidden units (using tanh as the state activation function and sigmoid for gating), a fully connected layer, and a classification layer. Training was performed using the ADAM optimizer with a learning rate of 0.01, a batch size of 50, and a maximum of 100 epochs. Cross-entropy was used as the loss function, and L2 regularization was applied to prevent overfitting. To determine the optimal number of latent variables for dimensionality reduction, nine previously created LSTM-autoencoder networks were used, excluding their decoder components. The effectiveness of the LSTM-autoencoder in dimensionality reduction for fault detection was evaluated through this approach. During the experimentation phase, a specialized 10-fold cross-validation methodology was employed across 20 flights. Each fold consisted of two distinct flights, one healthy and one faulty. Within each fold, half of the data was allocated to the validation set to mitigate overfitting, with a validation patience set at 7, meaning that the training will be stopped if the validation accuracy does not increase for 7 consecutive epochs. Nine networks were generated based on variations in the number of neurons in the latent layer of the LSTM-autoencoder network integrated into the BiLSTM classifier network. Each network underwent the experiment four times, and the average classification performance results, using different metrics, are presented in Figure 6.

Based on the results, fault detection achieved an accuracy of 73.57% when performed without dimensionality reduction. In subsequent experiments involving dimensionality reduction, it was noted that reducing the number of neurons in the latent layer from 45 to 20 led to an increase in classification accuracy, rising from 74.83% to 79.48%. However, a further reduction in the number of neurons to 15 resulted in a significant decrease in classification success, dropping to 70.48%. Continuing the reduction to even fewer neurons led to a further decline in classification success, reaching 67.95%.

Precision and recall follow similar trends, reinforcing the observations from the accuracy metric. Precision begins at about 0.70 with 45 neurons and increases to a peak of 0.84 at 20 neurons. This trend highlights that an optimal number of neurons enhances the model's ability to correctly identify faulty flights (true positives) while minimizing false positives. Similarly, recall starts at 0.68 with 45 neurons, peaks at 0.76 at 20 neurons, and then declines with further reduction in neurons. This pattern indicates that the model's ability to identify all actual faults is maximized with an optimal latent layer size.

The F1-score mirrors the trends observed in the other metrics since the dataset was balanced by gathering last 30 seconds from both healthy and faulty flights. Starting from 0.70 with 45 neurons, it peaks at approximately 0.76 at 20 neurons, and then decreases as the number of neurons is further reduced. The peak F1-score at 20 neurons

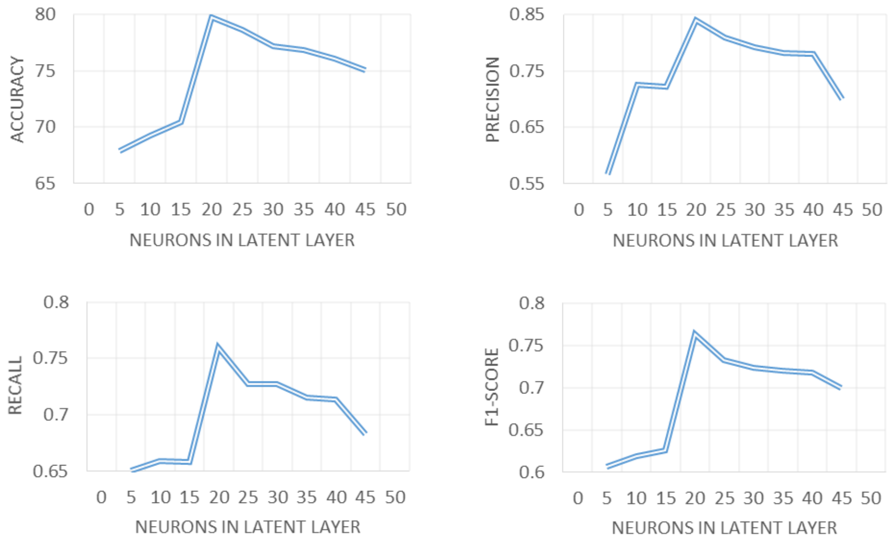


Figure 6. Performance results of classification

signifies the most balanced performance, combining both high precision and recall.

The findings illustrate that excessive information is introduced into the system when the neuron count in the latent layer is high, resulting in a detrimental impact on classification performance. Conversely, a reduction in the neuron count leads to improved classification success, signifying the elimination of unnecessary or redundant information and an augmentation in the proportion of essential data. Overall, the results underscore the critical role of dimensionality reduction in enhancing performance. Nonetheless, when attempting further dimensionality reduction, a noticeable decline in classification success emerges, primarily due to the loss of fault-descriptive information.

5. Conclusions

A deep learning-based framework is proposed herein to anticipate and mitigate potential crashes in quadrotors, enabling safe landings facilitated by an early warning system. This study distinguishes itself from previous works through the comprehensive utilization of all available sensor data, ensuring no relevant information is overlooked. Despite the recognition that some sensor data may be unnecessary or redundant, dimensionality reduction was conducted before the classification process to eliminate such redundancy. This step enhances fault detection accuracy and reduces complexity. The model comprises two primary components: 1) LSTM-autoencoder for both feature dimension reduction and computation of optimal reconstruction errors associated with each quadrotor flight, 2) BiLSTM classification for learning incipient faults from both healthy and faulty flights during training.

The proposed model was applied to quadrotor flight dataset created by gathering data collected from both web and the flights carried out in the university campus. The experimental results show that incipient faults are efficiently detected by the model with an average accuracy of 79.48%, 30 seconds before triggering serious malfunctions. This underscores the potential of the model to enhance both the safety and security of quadrotor flight operations. Moreover, the model's flexibility, with adjustable input and neuron sizes, makes it highly applicable to other systems.

Adapting the model to fault detection applications in a wide range of complex systems will be a research path in the future. It is believed that the results of this study are quite relevant for further expansion of the potential in this area through new studies. A prevalent challenge in fault detection lies in the scarcity of data containing information about faults compared to healthy operational data, leading to an unbalanced distribution of data. The data were balanced first, hence, experiments were conducted with a limitation as a significant portion of healthy flight data could not be fully utilized. Thus, this study will be redirected towards optimizing the utilization of healthy flight data and enhancing fault detection success by adapting methods employed for imbalanced data. This adaptation will be a focus of upcoming research to ensure a more comprehensive and effective fault detection approach.

Acknowledgements

Authors would like to express our gratitude to the Sensor Array Signal Processing Laboratory and Prof. Dr. Tansu Filik at Eskisehir Technical University for generously providing the flight data used in this research.

Funding

This work was supported by the Eskisehir Technical University Scientific Research Projects Commission (No. 21DRP053).

Author contributions

Mehmet OZCAN and Cahit PERKGOZ designed the research. Mehmet OZCAN processed the data, performed the simulation and drafted the paper. Cahit PERKGOZ revised and finalized the paper.

Disclosure statement

All the authors declare that they have no conflict of interest.

Data availability statement

The data and codes that support the findings of this study are openly available in fault_detection repository at https://github.com/mzcan8/fault_detection

References

Afshari, S. S., & Pourtakdoust, S. H. (2018). Probability density evolution for time-varying reliability assessment of wing structures. *Aviation*, *22*(2), 45–54. https://doi.org/10.3846/aviation.2018.6010

Ai, S., Shang, W., Song, J., & Cai, G. (2021). Fault diagnosis of the four-rotor unmanned aerial vehicle using the optimized deep forest algorithm based on the wavelet packet translation. In *2021 8th International Conference on Dependable Systems and Their Applications* (*DSA*). IEEE. https://doi.org/10.1109/DSA52907.2021.00085

Al Younes, Y., Rabhi, A., Noura, H., & El Hajjaji, A. (2016). Sensor fault diagnosis and fault tolerant control using intelligent-output-estimator applied on quadrotor UAV. In *2016 International Conference on Unmanned Aircraft Systems* (*ICUAS*). IEEE. https://doi.org/10.1109/ICUAS.2016.7502557

Altinors, A., Yol, F., & Yaman, O. (2021). A sound based method for fault detection with statistical feature extraction in UAV motors. *Applied Acoustics*, *183*, Article 108325. https://doi.org/10.1016/j.apacoust.2021.108325

Anidjar, O. H., Barak, A., Ben-Moshe, B., Hagai, E., & Tuvyahu, S. (2023). A stethoscope for drones: Transformers-based methods for UAVs acoustic anomaly detection. *IEEE Access*, *11*, 33336–33353. https://doi.org/10.1109/ACCESS.2023.3262702

Belcastro, C. M., Klyde, D. H., Logan, M. J., Newman, R. L., & Foster, J. V. (2017). Experimental flight testing for assessing the safety of unmanned aircraft system safety-critical operations. In *17th AIAA Aviation Technology*, *Integration*, *and Operations Conference*. ResearchGate. https://doi.org/10.2514/6.2017-3274

Bondyra, A., Gasior, P., Gardecki, S., & Kasinski, A. J. (2018). Development of the sensory network for the vibration-based fault detection and isolation in the multirotor UAV propulsion system. In *Proceedings of the 15th International Conference on Informatics in Control*, *Automation and Robotics* (*CINCO 2018*) (Vol. 2, pp. 102–109). Scitepress Digital Library. https://doi.org/10.5220/0006846801120119

Bondyra, A., Kołodziejczak, M., Kulikowski, R., & Giernacki, W. (2022). An acoustic fault detection and isolation system for multirotor UAV. *Energies*, *15*(11), Article 3955. https://doi.org/10.3390/en15113955

Chen, B., Peng, Y., Gu, B., Luo, Y., & Liu, D. (2021). A fault detection method based on enhanced GRU. In *2021 International Conference on Sensing*, *Measurement & Data Analytics in the era of Artificial Intelligence* (*ICSMD*). IEEE. https://doi.org/10.1109/ICSMD53520.2021.9670769

Duncan Imbassahy, D. W., Costa Marques, H., Conceição Rocha, G., & Martinetti, A. (2020). Empowering predictive maintenance: A hybrid method to diagnose abnormal situations. *Applied Sciences*, *10*(19), Article 6929. https://doi.org/10.3390/app10196929

Erfanian, A. M., & Ramezani, A. (2022). Using deep learning network for fault detection in UAV. In *2022 8th International Conference on Control*, *Instrumentation and Automation* (*ICCIA*). IEEE. https://doi.org/10.1109/ICCIA54998.2022.9737206

Fu, J., & Che, G. (2021). Fusion fault diagnosis model for six-rotor UAVs based on conformal Fourier transform and improved self-organizing feature map. *IEEE Access*, *9*, 14422–14436. https://doi.org/10.1109/ACCESS.2021.3052317

Guo, D., Zhong, M., Ji, H., Liu, Y., & Yang, R. (2018). A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors. *Neurocomputing*, *319*, 155–163. https://doi.org/10.1016/j.neucom.2018.08.046

Hajiyev, C. (2016). An innovation approach based sensor fault detection and isolation. *IFAC-PapersOnLine*, *49*(17), 420–425. https://doi.org/10.1016/j.ifacol.2016.09.072

Huang, J., Li, M., Zhang, Y., Mu, L., Ao, Z., & Gong, H. (2021). Fault detection and classification for sensor faults of UAV by deep learning and time-frequency analysis. In *2021 40th Chinese Control Conference* (*CCC*). IEEE. https://doi.org/10.23919/CCC52363.2021.9550141

Iannace, G., Ciaburro, G., & Trematerra, A. (2019). Fault diagnosis for UAV blades using artificial neural network. *Robotics*, *8*(3), Article 59. https://doi.org/10.3390/robotics8030059

Ignatovich, S., Menou, A., Karuskevich, M., & Maruschak, P. (2013). Fatigue damage and sensor development for aircraft structural health monitoring. *Theoretical and Applied Fracture Mechanics*, *65*, 23–27. https://doi.org/10.1016/j.tafmec.2013.05.004

Jiang, Y., Zhiyao, Z., Haoxiang, L., & Quan, Q. (2015). Fault detection and identification for quadrotor based on airframe vibration signals: a data-driven method. In *2015 34th Chinese Control Conference* (*CCC*). IEEE. https://doi.org/10.1109/ChiCC.2015.7260639

Jing, C. S., & Pebrianti, D. (2016). Fault detection and identification in Quadrotor system (Quadrotor robot). In *2016 IEEE International Conference on Automatic Control and Intelligent Systems* (*I2CACIS*). IEEE. https://doi.org/10.1109/I2CACIS.2016.7885281

Jing, C. S., Pebrianti, D., Qian, G. M., & Bayuaji, L. (2017). Fault detection in Quadrotor MAV. In *2017 7th IEEE International Conference on System Engineering and Technology* (*ICSET*). IEEE. https://doi.org/10.1109/ICSEngT.2017.8123422

Lazzara, M., Chevalier, M., Colombo, M., Garcia, J. G., Lapeyre, C., & Teste, O. (2022). Surrogate modelling for an aircraft dynamic landing loads simulation using an LSTM AutoEncoder-based dimensionality reduction approach. *Aerospace Science and Technology*, *126*, Article 107629. https://doi.org/10.1016/j.ast.2022.107629

Liu, L., Ma, Y., Xu, B., Xiang, C., & Yang, X. (2016). Fault detection and isolation based on UKFs for a novel ducted fan UAV. In *2016 IEEE International Conference on Aircraft Utility Systems* (*AUS*). IEEE. https://doi.org/10.1109/AUS.2016.7748049

Liu, W., Chen, Z., & Zheng, M. (2020). An audio-based fault diagnosis method for quadrotors using convolutional neural network and transfer learning. In *2020 American Control Conference* (*ACC*). IEEE. https://doi.org/10.23919/ACC45564.2020.9148044

Olyaei, M. H., Jalali, H., Noori, A., & Eghbal, N. (2018). Fault detection and identification on UAV system with CITFA algorithm based on deep learning. In *Iranian Conference on Electrical Engineering* (*ICEE*). IEEE. https://doi.org/10.1109/ICEE.2018.8472529

Ouadine, A. Y., Mjahed, M., Ayad, H., & El Kari, A. (2020). UAV quadrotor fault detection and isolation using artificial neural network and Hammerstein-Wiener model. *Studies in Informatics and Control*, *29*(3), 317–328. https://doi.org/10.24846/v29i3y202005

Ozkat, E. C., Bektas, O., Nielsen, M. J., & la Cour-Harbo, A. (2023). A data-driven predictive maintenance model to estimate RUL in a multi-rotor UAS. *International Journal of Micro Air Vehicles*, *15*. https://doi.org/10.1177/17568293221150171

Pose, C., Giribet, J., Torre, G., & Marzik, G. (2023). Neural network-based propeller damage detection for multirotors. In *2023 International Conference on Unmanned Aircraft Systems* (*ICUAS*). ResearchGate. https://doi.org/10.1109/ICUAS57906.2023.10156355

Puchalski, R., & Giernacki, W. (2022). UAV fault detection methods, state-of-the-art. *Drones*, *6*(11), Article 330. https://doi.org/10.3390/drones6110330

Said Elsayed, M., Le-Khac, N.-A., Dev, S., & Jurcut, A. D. (2020). Network anomaly detection using LSTM based autoencoder. In *Q2SWinet'20: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks* (pp. 37–45). ACM Digital Library. https://doi.org/10.1145/3416013.3426457

Vural, S. Y., & Hacızade, C. (2016). Sensor/actuator fault detection, isolation and accommodation applied to UAV model. *Journal of Aeronautics and Space Technologies*, *9*(2), 1–12.

Wang, B., Liu, D., Peng, Y., & Peng, X. (2019). Multivariate regression-based fault detection and recovery of UAV flight data. *IEEE Transactions on Instrumentation and Measurement*, *69*(6), 3527–3537. https://doi.org/10.1109/TIM.2019.2935576

Wei, Y., Wu, D., & Terpenny, J. (2020). Robust incipient fault detection of complex systems using data fusion. *IEEE Transactions on Instrumentation and Measurement*, *69*(12), 9526–9534. https://doi.org/10.1109/TIM.2020.3003359

Yang, P., Geng, H., Wen, C., & Liu, P. (2021). An intelligent quadrotor fault diagnosis method based on novel deep residual shrinkage network. *Drones*, *5*(4), Article 133. https://doi.org/10.3390/drones5040133

Yasniy, O., Mytnyk, M., Maruschak, P., Mykytyshyn, A., & Didych, I. (2024). Machine learning methods as applied to modelling thermal conductivity of epoxy-based composites with different fillers for aircraft. *Aviation*, *28*(2), 64–71. https://doi.org/10.3846/aviation.2024.21472

Zhang, W., Tong, J., Liao, F., & Zhang, Y. (2023). Simulation-to-reality UAV fault diagnosis with deep learning. In *arXiv preprint arXiv:2302.04410*. Cornell University.

Zhao, Z., Wang, X., Xu, J., & Yu, J. (2018). A performance evaluation algorithm of stochastic hybrid systems based on fuzzy health degree and its application to quadrotors. *IEEE Access*, *6*, 37581–37594. https://doi.org/10.1109/ACCESS.2018.2838149

Zheng, K., Jia, G., Yang, L., & Wang, J. (2021). A Compound fault labeling and diagnosis method based on flight data and BIT record of UAV. *Applied Sciences*, *11*(12), Article 5410. https://doi.org/10.3390/app11125410