REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ANOMALY DETECTION IN UNMANNED AERIAL VEHICLES: A COMPREHENSIVE STUDY OF HYBRID DEEP NEURAL NETWORK METHODS FOR EDGE-BASED APPLICATIONS

Hatice Vildan DÜDÜKÇÜ

DOCTOR OF PHILOSOPHY THESIS
Department of Electronics and Communications Engineering
Program of Electronics

Supervisor
Assoc. Prof. Dr. Nihan KAHRAMAN

Co-supervisor
Asst. Prof. Dr. Murat TAŞKIRAN

February, 2024

REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ANOMALY DETECTION IN UNMANNED AERIAL VEHICLES: A COMPREHENSIVE STUDY OF HYBRID DEEP NEURAL NETWORK METHODS FOR EDGE-BASED APPLICATIONS

A thesis submitted by Hatice Vildan DÜDÜKÇÜ in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY** is approved by the committee on 07.02.2024 in Department of Electronics and Communications Engineering, Program of Electronics.

Assoc. Prof. Dr. Nihan KAHRAMAN
Yildiz Technical University
Supervisor

Asst. Prof. Dr. Murat TAŞKIRAN
Yildiz Technical University
Co-supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Nihan KAHRAMAN, Supervisor
Yildiz Technical University _____

Prof. Dr. Tülay YILDIRIM, Member
Yildiz Technical University _____

Assoc. Prof. Sadiye Nergis TURAL POLAT, Member
Yildiz Technical University _____

Prof. Dr. Günhan DÜNDAR, Member
Boğaziçi University _____

Prof. Dr. Ender Mete EKŞİOĞLU, Member
İstanbul Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Anomaly Detection in Unmanned Aerial Vehicles: A Comprehensive Study of Hybrid Deep Neural Network Methods for Edge-Based Applications supervised by my supervisor, Assoc. Prof. Dr. Nihan KAHRAMAN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Hatice Vildan DÜDÜKÇÜ

Signature

This work was supported by Yildiz Technical University Scientific Research Projects Coordination Unit under project number FBA-2021-4671.

Dedicated to my family...

ACKNOWLEDGEMENTS

I would like to thank my esteemed supervisor, Assoc. Prof. Dr. Nihan KAHRAMAN, and my co-supervisor, Assoc. Prof. Dr. Murat TAŞKIRAN, for their time, guidance, suggestions, contributions, and invaluable support throughout the course of my thesis study. I would also like to thank the members of my thesis monitoring committee, Prof. Dr. Günhan DÜNDAR and Assoc. Prof. Sadiye Nergis TURAL POLAT, for their insightful comments, suggestions, and willingness to share their knowledge and experiences with me during my studies. I am also grateful to my examining committee members Prof. Dr. Tülay YILDIRIM and Prof. Dr. Ender Mete EKŞİOĞLU for their advice and insightful comments on the thesis.

I would like to thank Yildiz Technical University Scientific Research Projects Coordination Unit for supporting my thesis studies under project number FBA-2021-4671.

I would also like to thank all my friends for their unwavering support, unfailing belief, and encouragement throughout my studies. Finally, I would like to extend my most profound thanks and convey my sincere appreciation to my mother Fatma Ayşe DÜDÜKÇÜ, my father Muammer DÜDÜKÇÜ, and my sister Merve Zeynep DÜDÜKÇÜ for their love, well wishes, and all their assistance. I am forever grateful to them for the love and support that they have consistently shown me throughout the years.

Hatice Vildan DÜDÜKÇÜ

TABLE OF CONTENTS

LIST OF SYMBOLS

σ	Sigmoid Function
tan	Tangent Function
$\odot$	Hadamard product
$\sum$	Summation
R^2	Coefficient of Determination
∞	Infinity

LIST OF ABBREVIATIONS

1D	One-Dimensional
3D	Three-Dimensional
ADS	Air Data System
AE	AutoEncoder
ALFA	AirLab Failure and Anomaly
ANN	Artificial Neural Network
Bi-LSTM	Bi-directional Long-Short Time Memory network
BLE	Bluetooth Low Energy
CNN	Convolutional Neural Network
CS2S	Causality-Sequence-to-Sequence
CV	Cross-Validation
DNN	Deep Neural Network
DoF	Degrees of Freedom
FC	Fully Connected
FCNN	Fully Connected Neural Network
FN	False Negative
FP	True Negative
FPGA	Field-Programmable Gate Array
FPR	False Positive Ratio
FPV	First Person View
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GRU	Gated Recurrent Unit

HDIN	Hull Drone Indoor Navigation
IMU	Inertial Measurement Unit
IoT	Internet of Things
k-NN	k-Nearest Neighbor
LR-TCN	Temporal Convolutional Network with Leaky Rectified Linear Unit
LS-SVM	Least Squares Support Vector Machine
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical Systems
Mid-Air	Montefiore Institute Aerial Images and Records
MLP	Multi-Layer Perceptron
MOD	Multidimensional Outlier Descriptor
MSE	Mean Squared Error
NASA	National Aeronautics and Space Administration
NIR	Near Infrared
PCA	Principal Component Analysis
PQAT	Pruning Preserving Quantization Aware Training
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RMSE	Root-Mean-Square Error
RNN	Recurrent Neural Network
RTK	Real-Time Kinematic
SMA	Simple Moving Average
SoC	System-on-Chip
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TCN	Temporal Convolutional Network

TCN-SMA	Temporal Convolutional Network with Simple Moving Average
TN	False Positive
TP	True Positive
TPR	True Positive Ratio
UAV	Unmanned Aerial Vehicle
UMDC	UAV Mosaicking and Change Detection

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Anomaly Detection in Unmanned Aerial Vehicles: A Comprehensive Study of Hybrid Deep Neural Network Methods for Edge-Based Applications

Hatice Vildan DÜDÜKÇÜ

Department of Electronics and Communications Engineering
Doctor of Philosophy Thesis

Supervisor: Assoc. Prof. Dr. Nihan KAHRAMAN
Co-supervisor: Asst. Prof. Dr. Murat TAŞKIRAN

Unmanned Aerial Vehicle (UAV) usage has increased significantly in the past few years all over the world, with major benefits in the transportation, entertainment, military, and industrial sectors. Improving the dependability and safety of UAVs requires early identification of flight anomalies and continuous monitoring of sensor data in order to identify malfunctioning components in the system. The present work focuses on detecting anomalies in edge-based applications of UAVs using a hybrid deep neural network approach.

This study offers a novel approach that is well-suited for the utilization of onboard systems in UAVs for detecting anomalies in flight sensor data. To achieve this objective, the data collected from UAV sensors was analyzed, and a classification method was developed to identify abnormal flying patterns. Additionally, a regression method was also proposed to detect anomalies in UAV sensors. The effectiveness of these methods was assessed by utilizing both existing UAV datasets from the literature and a dataset specifically generated for the study. Furthermore, as part of the study, data obtained via flight sensors has been collected and labeled. Subsequently, this dataset was utilized to assess both classification and regression techniques and it served as a way to analyze the proposed hybrid approach that combines a classification method for detecting flight anomalies with a regression method for sensor analysis. In order to assess the effectiveness of the hybrid method on power-constrained devices with limited memory resources, a Raspberry

Pi development board was utilized as the testing platform. In addition, a model update feature, which made it possible to conduct continuous model training on the edge, was developed and evaluated.

When the outcomes of the study were analyzed, it was determined that the recommended AnoSense classification and TCN-SMA regression methods demonstrated improved performance in terms of anomaly detection and sensor data prediction than the other compared classical deep neural network models. In addition, the hybrid method that has been presented is capable of efficient anomaly detection, analysis of each instantaneous sensor data, and labeling of anomaly data points. Additionally, it has been demonstrated that the hybrid anomaly detection approach that has been presented can be successfully used for applications involving UAVs that have limited resources, where it is of the utmost importance to have the ability to detect anomalies onboard and provide remedial control commands promptly. As a result of the thesis, a hybrid method for detecting anomalies in UAV sensors has been devised, and its effectiveness as a real-time solution for anomaly detection has been validated by the findings.

Keywords: Unmanned aerial vehicle, deep neural network, anomaly detection, edge-based application

ÖZET

İnsansız Hava Araçlarında Anomali Tespiti: Uç Birim Tabanlı Uygulamalar için Hibrit Derin Sinir Ağı Yöntemlerinin Kapsamlı Bir Çalışması

Hatice Vildan DÜDÜKÇÜ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Doktora Tezi

Danışman: Doç. Dr. Nihan KAHRAMAN
Eş-Danışman: Dr. Öğr. Üyesi Murat TAŞKIRAN

İnsansız Hava Aracı (İHA) kullanımı son yıllarda dünya çapında önemli ölçüde artmış ve ulaşım, eğlence, askeri ve endüstri sektörlerinde büyük fayda sağlamıştır. İnsansız hava araçlarının güvenilirliğini ve güvenliğini artırmak sistemdeki arızalı bileşenlerin tespitini ve bunun için uçuş anomalilerinin erken teşhis edilmesini ve sensör verilerinin sürekli izlenmesini gerektirir. Mevcut çalışma, hibrit derin sinir ağı yaklaşımı kullanılarak insansız hava araçlarının uç birim tabanlı uygulamalarında anomalilerin tespit edilmesini amaçlamaktadır.

Bu çalışma, insansız hava araçlarının anormal uçuş sensör verilerini tespit etmek için yerleşik sistemlerin kullanımına uygun yeni bir yaklaşım önermektedir. Bu amaca ulaşmak için insansız hava araçlarının sensörlerinden toplanan veriler analiz edilmiş ve anormal uçuş şekillerini tespit etmek için bir sınıflandırma yöntemi geliştirilmiştir. Buna ek olarak uçuş sensörlerindeki anormalliklerin tespiti için ise bir regresyon yöntemi önerilmiştir. Bu yöntemlerin etkinliği, hem literatürdeki mevcut uçus sensör veri kümeleri hem de bu çalışma için özel olarak oluşturulmuş bir veri kümesi kullanılarak değerlendirilmiştir. Ayrıca çalışma kapsamında uçuş sensör verileri toplanarak etiketlenmiş ve bu veri seti hem sınıflandırma hem de regresyon tekniklerini değerlendirmek için kullanılmıştır. Ayrıca uçuş anormalliklerini tespit etmek için bir sınıflandırma yöntemini, sensör analizi için bir regresyon yöntemiyle birleştiren hibrit yaklaşımı analiz etmek için de bu veri seti kullanılmıştır. Hibrit yaklaşımın sınırlı bellek kaynaklarına sahip, gücü

kısıtlı cihazlar üzerindeki etkinliğini değerlendirmek amacıyla ise Raspberry Pi geliştirme kartı test platformu olarak kullanılmıştır. Ayrıca uçta sürekli model eğitimi yapılmasını mümkün kılan bir model güncelleme yöntemi de geliştirilerek test edilmiştir.

Çalışmanın sonuçları analiz edildiğinde, önerilen AnoSense sınıflandırma ve TCN-SMA regresyon yöntemlerinin, anormallik tespiti ve sensör veri tahmini açısından diğer klasik derin sinir ağı modellerine göre daha iyi performans gösterdiği belirlenmiştir. Ek olarak önerilen hibrit yöntemin, anomali tespiti, anlık sensör veri analizi ve anomali veri noktalarının etiketlenmesinde başarılı olduğu görülmüştür. Ayrıca, önerilen hibrit anomali tespit yaklaşımının, anomali tespiti ve düzeltici kontrol komutları sağlama becerisinin son derece önemli olduğu sınırlı kaynaklara sahip hava araçlarını içeren uygulamalarda başarıyla uygulanabileceği gösterilmiştir. Tez sonucunda İHA sensörlerindeki anormalliklerin tespitine yönelik hibrit bir yöntem geliştirilmiş ve bu yöntemin anormallik tespitinde gerçek zamanlı bir çözüm olarak etkinliği bulgularla doğrulanmıştır.

Anahtar Kelimeler: İnsansız hava aracı, derin sinir ağı, anomali tespiti, uç birim tabanlı uygulama

1
INTRODUCTION

An Unmanned Aerial Vehicle (UAV), more often known as a drone, is a kind of aircraft that does not need the presence of human pilots or crew members. In recent years, there has been a significant rise in the use of UAVs (drones) worldwide, which has resulted in significant benefits in various fields, including industry, entertainment, military, and transportation. Nonetheless, the efficacy and dependability of these UAVs are contingent upon the quality of their underlying software systems. Therefore, anomaly detection software that is capable of operating with limited power and memory resources and can be integrated into compact development cards is crucial for small-scale and indoor UAVs.

Small and indoor drones are typically constrained by their restricted energy and processing capabilities. Therefore the need arises for a specilized software on these drones to accurately identify anomalies in their flights. Anomaly detection software that can function with minimal power consumption and restricted memory usage is an optimal solution for these drones. These type of anomaly detection algorithms usually examines a variety of sensor data to identify patterns of nominal activity and identify any anomalies during flight. These algorithms will enable drones to be used more safely, efficiently and effectively.

This thesis presents the development and implementation of a tailored anomaly detection system specifically designed for small-sized drones. The objective is to promptly identify anomalies in flight sensors, with limited power consumption and computational costs, particularly for small indoor drones.

1.1 Literature Review

The utilization of Unmanned Aerial Vehicles (UAVs) has had a notable rise in popularity in recent years, resulting in a substantial increase in research efforts on this subject, which is well documented in the literature. Drones present numerous possibilities for many different industries. For instance within agricultural sector,

drones have the potential to enhance operational efficiency and save expenses through their utilization in activities such as commercial spraying [1, 2], plant damage detection [3, 4], measurement [5, 6], fertilization [7, 8] and pollination [9, 10]. In health applications, drones can be used for the transport of healthcare supplies and medicines [11, 12] and in emergency applications, they can be used for the provision of health supplies, food, and medication to individuals affected by calamities [13–16]. UAVs can also be used in defense industry for the purpose of border surveillance [17, 18], identification and elimination of a specific objective [19, 20], and in the mapping field for the extraction of 3D images of lands [21, 22]. Also in the transportation sector, they can be used for traffic monitoring [23, 24] or as transportation UAVs [25, 26]. In recent years, there has also been a significant rise in the utilization of drones in the entertainment business for filming action movies [27, 28], as well as the deployment of light swarm drones for celebratory events [29, 30]. Table 1.1 also provides a concise overview of the many application areas where UAVs are commonly utilized.

Table 1.1 Common application areas for UAVs.

Industry	**Application Areas**
Health	Logistics of medical supplies and pharmaceuticals [11, 12] Distribution of supplies, food, and medicine to people who are victims of disasters [13–16]
Agriculture	Application of pesticides [1, 2] Identification of plant damage [3, 4] Measurement [5, 6] Fertilization [7, 8] Pollination [9, 10]
Transport	Transportation UAVs [25, 26]
Defense	Border surveillance [17, 18] Identification and elimination of a specific target [19, 20]
Mapping	3D image extraction of lands [21, 22]
Traffic Applications	Traffic Monitoring [23, 24]
Entertainment	Shooting action scenes [27, 28] Usage of light swarm drones in celebrations [29, 30]

The issue of the rapidly growing UAV business and its widespread use across several domains has garnered significant attention within academic interest. The acquisition of data from various sensors during the flight of UAVs has prompted researchers to explore the fields of flight monitoring, energy modeling, and remote sensing. These studies facilitate the extraction of flight-related information, enhance system control optimization, and simplify the process of gathering visual data. More studies are also now being undertaken to address concerns about safety and flight control. The taxonomy of UAV applications using Deep Neural Network (DNN) techniques in the existing literature may be categorized into three primary groups: flight monitoring, energy modeling, and remote sensing applications (Figure 1.1) [31].

Flight monitoring applications aim to evaluate time-series sensor data collected

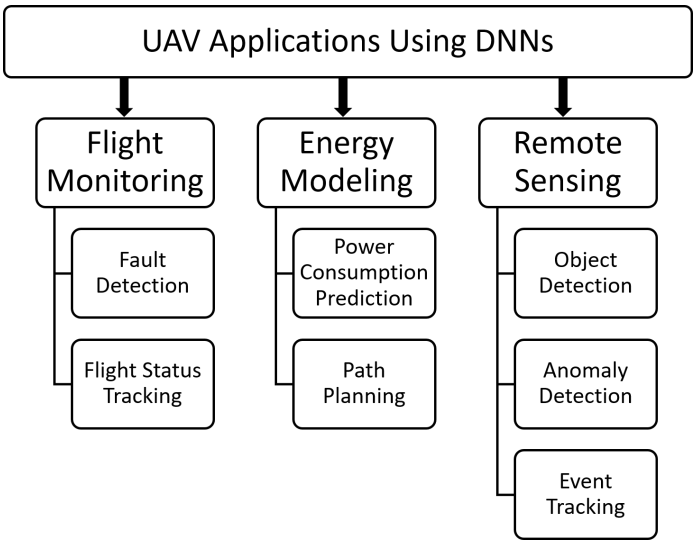

Figure 1.1 The taxonomy of UAV DNN application areas in the literature.

from UAV flight records to monitor flight status, detect faults, and identify anomalies. The domain of energy modeling and planning often involves conducting research on monitoring the battery condition of UAVs, as well as estimating and optimizing flight route planning. These studies mostly rely on the use of sensor or image data. Studies on UAV remote sensing aim to identify and analyze abnormalities within a given observed region by using image or sensor data collected during UAV flights. Figure 1.2 also presents the attributes of the data used in the main application areas of UAV DNN applications.

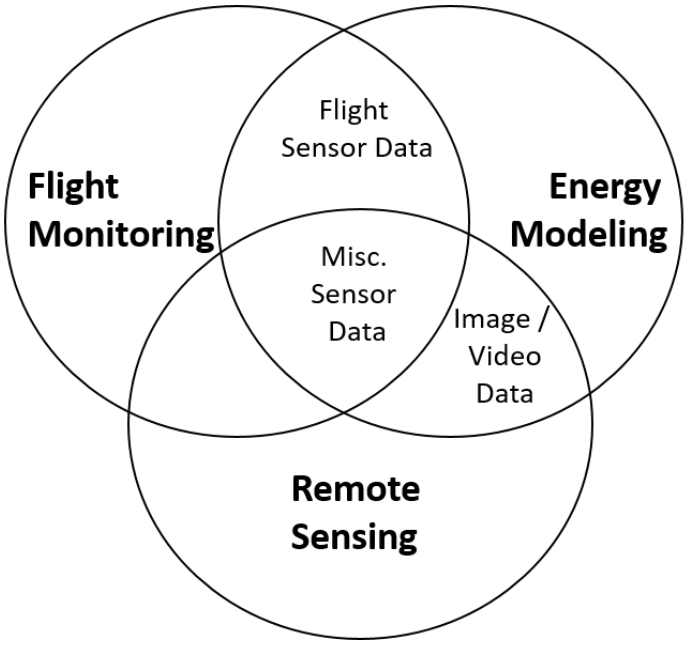

Figure 1.2 Data types for different UAV DNN application fields.

This study aims to perform fault detection in order to effectively monitor a system and ascertain the timing, kind, and location of an error. In literature,

the identification of incorrect sensor data may be achieved using two approaches: direct pattern recognition models or the calculation of deviations from the predicted values. Some studies, however, integrate the identification of faults and anomalies to address issues originating from both external sources and internal system-related factors [32].

Research conducted on fault and anomaly detection for UAVs often uses a combination of several sensor data, including Inertial Measurement Unit (IMU), Global Positioning System (GPS), motor speed, and battery sensor readings. These data sources are utilized to continuously monitor and identify any abnormalities or malfunctions occurring during UAV missions. While fault detection can be performed by using various classical machine learning methods [33–48], it is seen that DNN methods are generally used in recent studies [49–53].

When analyzing research that utilizes DNNs, it is seen that these studies typically collect anomaly data during UAV flights through the utilization of various sensors such as the Global Positioning System (GPS), Inertial Measurement Unit (IMU), Air Data System (ADS), and occasionally these data are subjected to feature extraction via the application of the Short-Time Fourier Transform (STFT) technique. Moreover, the utilization of Convolutional Neural Networks (CNN) can be employed for the goal of extracting features from the aforementioned acquired data. The results indicate that this approach has proficient fault diagnosis capabilities, as substantiated by its favorable performance in flight testing [50]. Similarly, when CNN was used to automatically extract features from raw simulated sensor data, learn, and then support anomaly detection it also gave successful results for anomalies affecting different components of the IMU, such as accelerometer only, accelerometer–gyroscope, and accelerometer–gyroscope-orientation [51].

Moreover in other studies, it is seen that the Long Short-Term Memory (LSTM) method is also recommended for UAV sensor anomaly detection, apart from CNN. In [49], a prediction model with LSTM architecture is trained using the training data set containing normal data, and future data prediction is performed. Then, results that were far from the predicted value within a certain threshold range were determined as anomalies. As a result of the experimental results performed with point anomalies in the north velocity and pneumatic lift velocity of UAV, it was seen that only 1 of the 30 anomalies was misclassified and the proposed method was successful. Similarly, [52] was also able to predict failure with an average accuracy of 93% and was able to detect failures with 100% accuracy, using LSTM on the AirLab Failure and Anomaly (ALFA) data set. Furthermore, in [53], to detect UAV battery anomaly, the causality-sequence-to-sequence (CS2S) type deep

learning model, which is also a successful method in time series, was used. This model is a DNN method that includes two LSTM layers, defined as encoder and decoder, and as a result of this study, it was seen that CS2S gives better results compared to other conventional prediction-based and reconstruction-based models for the detection of UAV battery anomalies.

The utilization of Temporal Convolutional Network (TCN), a CNN method that offers a comprehensive architecture to systematically gather both low-level and high-level data, has also experienced significant growth in anomaly detection for UAV sensors in recent years. The reason for this is the exceptional efficacy of TCN in detecting anomalies and predicting time series problems [54, 55]. In their work, You et al. hypothesized that operating environment variability might lead to decreased performance of deep learning models used for anomaly detection in UAVs. To address this issue, they conducted an experiment where the trained models were fine-tuned to the specific region of operation. By employing TCN they have demonstrated that the suggested approach enhances model performance and lowers the rate of false detection [56].

It is noted that the research on anomaly detection exhibit similarities across all aviation anomaly issues. The study by Nanduri et al. [57] introduced a system for detecting anomalies using LSTM and Gated Recurrent Unit (GRU) architectures. The system successfully recognized 9 out of 11 anomalies in the test data set with a high level of sensitivity. The architecture described in the paper was based on the empirical work of Chung et al. [58]. A recent research [59] introduced a novel approach for detecting anomalies in airplane tracks. The suggested method combines the Multi-dimensional Outlier Descriptor (MOD) with the Bi-directional Long-Short Time Memory Network (Bi-LSTM). The experimental studies have demonstrated that the suggested system is capable of detecting anomalies in aircraft track with a test accuracy of 96%. The Recursive least squares method was also utilized for real-time anomaly identification [48]. The experimental results demonstrated a remarkable accuracy of 86.36% when applied to a data set including 22 flight records. Additionally in [60], the researchers developed an anomaly detection method that operates in real-time and is based on data-driven support vector regression models. Artificial anomalies were generated to assess the effectiveness of the suggested strategy and the proposed technique yielded a significant reduction in both the total computing time (38.2% decrease) and the Root-Mean-Square Error (RMSE) value (37% decrease) based on the experiments.

Although studies generally focus on offline algorithm development using data sets containing sensor data, some researchers have also tried to offer solutions for

online fault detection in UAVs using various hardware systems [61–64]. In [61] and [62], researchers developed an online anomaly detection algorithm running on a Field-Programmable Gate Array (FPGA), that is based on the Least Squares Support Vector Machine (LS-SVM), demonstrating that the system can be used for real-time anomaly detection. However, in [64], in which anomaly detection was carried out, the researchers used the Raspberry Pi as the processor and carried out the real-time system operation by communicating with supervisory control.

As can be seen, when fault detection studies are examined, anomalies or faulty components are detected by modeling sensor data on UAV onboard systems. Although there are many studies in this field, it is clear that future studies in this area are necessary due to the technology's tendency towards fully autonomous UAVs, and the importance of fault detection for autonomy.

Due to the rapid development of UAV technologies, the decrease in their costs, and the increasing demand for them today, researchers have turned to studies on the use of UAVs in various fields. Upon examination of fault detection studies, it is evident that there is a shift towards prioritizing onboard fault and anomaly detection in UAVs rather than relying on data processed by ground controllers or supervisory centers [65–67]. This shift is driven by the increasing demand for fully autonomous flights. It is anticipated similar to [65] that dynamic thresholding will be used as well as various DNNs such as Recurrent Neural Network (RNN) and CNN variations in future studies to solve the problems in this area. In addition, similar to the [66], it will become common to run fault and anomaly detection algorithms in real-time within the development boards that can be used both in drone control and data processing in the future. To mitigate the anticipated challenges on computing cost, processing speed, and reaction time outlined in the examined research, academics may direct their attention toward the decrease of model inference time on development boards.

1.1.1 Types of UAV Sensor Data

Current UAVs are frequently outfitted with a diverse range of sensors, which are selected based on their intended use. These sensors facilitate the execution of a multitude of activities by UAVs. The following is a compilation of several types of sensors that are frequently employed in UAVs, along with the corresponding data they collect. This section presents an overview of several sensor types that are often employed in UAVs, along with the corresponding data acquired from these sensors.

- **Imaging Sensors:** Imaging sensors, namely cameras, are employed to take

photos or record videos on UAV flights. These cameras can take images using many methods, encompassing but not restricted to color, thermal, or hyper-spectral approaches.

- **Inertial Measurement Unit Sensors:** The Inertial Measurement Unit (IMU) sensors are electronic devices utilized to measure speed, acceleration, and orientation on aerial vehicles. IMU sensors utilize Micro-Electro-Mechanical Systems (MEMS) technology to transform the acceleration and rotational velocity of the vehicle into electrical signals. The accelerometer, which is a component of the IMU, measures the acceleration, or the rate of change in velocity, of the vehicle. It usually has a three-axis (x, y, z) accelerometer which is used to track the three-dimensional movement. Acceleration is an important parameter that plays a crucial role in the determination of a vehicle's velocity and position. The gyroscope, which is another component of the IMU, is a sensor that measures the rotation speed and orientation of the vehicle. Additionally, this sensor also possesses a three-axis (x, y, z) reading capability, which is utilized to determine the vehicle's orientation and track its changes. The magnetometer, which also serves as a three-axis IMU sensor, measures the magnetic field around the vehicle. This sensor helps the vehicle determine geographic north, which is then used for mapping or calculating the vehicle's orientation.

- **Positioning Sensors:** Positioning sensors are components used to determine a vehicle's geographic location. The Global Positioning System (GPS) is a highly prevalent positioning sensor that utilizes signals transmitted by satellite networks to ascertain the precise geographic coordinates of a vehicle on Earth. Altitude sensors, as another sort of positioning sensor, serve the purpose of assisting aircraft and UAVs in determining their vertical position, often known as altitude.

- **Ambient Sensors:** UAVs can employ a variety of environmental sensors in order to monitor and analyze environmental variables while performing various tasks. These sensors are often used for the purpose of measuring, monitoring and gathering data related to environmental conditions during the course of a flight. One of the most common types of sensors in this category is wind sensors, which are utilized to detect both the speed and direction of wind. This data aids in maintaining flight stability and ensuring that the UAV adheres to its predetermined trajectory. Pressure sensors are also a common environmental sensor that is used to monitor changes in air pressure and provide information regarding altitude. Therefore, they can be employed for the purpose of monitoring changes in flight altitude and barometric pressure.

1.1.2 UAV Sensor Data Sets

UAVs, have increasingly found applications in several fields in recent times. Data is now gathered and assessed during flights using a diverse range of equipment, including cameras, hyperspectral sensors, flight sensors, and ambient sensors. Various data sets, particularly encompassing image and flight sensor data, are readily accessible in the literature for academics to utilize in the creation of novel methodologies. The accessible and widely used data sets for UAVs in the literature are presented in this section to give a brief overview to researchers.

The data sets utilized in UAV research may be classified into three groups according to the sorts of data they contain. These are sensor-based data sets, image-based data sets, and data sets including both. For the first category, sensor-based data sets, the AirLab Failure and Anomaly (ALFA) Data Set [48, 68] is one of the widely known data sets. The ALFA data set contains data collected from autonomous flights for fault and anomaly detection studies. The data set includes processed data collected from autonomous flights, including scenarios such as sudden engine and actuator failures. Furthermore, raw data from fully autonomous, autopilot-assisted, and manual flights with different error scenarios are also included in the data set. The data set also includes failure type and failure time ground truth for each scenario. Another sensor-base data set, the In-flight Positional and Energy Use Data Set [69, 70] includes flight data collected using the DJI Matrice 100 quadcopter. This data set was created to be used in energy consumption measurement studies of small UAVs. For data collection, a small package delivery drone was directed to autonomously take off, fly on a specific route, and land. During flight operations on different routes, data was collected for parameters such as the vehicle's inertia states, wind speed, and power consumption. This data set contains flight information collected from various onboard sensors such as GPS, IMU, anemometer, voltage, and current sensors.

Data sets in the second category consist of image-based data. These include: Mini-drone Data Set [71], Multi-Target Detection Data Set [72], VisDrone Data Set [73, 74], HighD Data Set [75], UAV Mosaicking and Change Detection (UMDC) Data Set [76], WeedMap Data Set [77], and Agriculture-Vision Data Set [78].

Mini-drone data set was created for drone-based surveillance studies, and it consists of video images taken in a parking lot with a mini drone Phantom 2 Vision+. Data set contents are clustered into three categories such as normal, suspicious, and illegal behavior. In this data set normal content includes images such as walking, driving, and parking, suspicious content includes, people acting in a suspicious manner, and illegal content includes images of incorrectly parked vehicles, theft,

and fighting. Multi-target detection data set [72] includes various videos recorded by a camera mounted on a delta-wing airframe. The videos in the data set contain multiple target UAV images with various appearances and shapes. The ground truth information of the targets in the data set is annotated manually. The VisDrone data set [73, 74] is an image data set containing ground truth for various important computer vision tasks collected by the AISKYEYE team at Tianjin University Machine Learning and Data Mining Laboratory in China. The data set contains images of different environments, objects and densities. Images in the data set were collected using various drones, in different scenarios, and in various weather and lighting conditions. Although this data set was first published for a challenge in 2018, it was updated in the following years and various versions continued to be presented to researchers. The HighD data set is a data set containing traffic and road images recorded on German highways using drones. Although the main purpose of creating the data set is the security verification of autonomous vehicles, it can also be used for tasks such as traffic and driver analysis. UAV mosaicking and change detection (UMDC) data set consists of 50 aerial video sequences collected at a flight height between 6 and 15 meters in different environments. The data set includes videos with and without the presence of vehicles, persons, and objects.

WeedMap data set [77], which consists of sugar beet field images collected using drones, includes various plant and weed images. In this data set, images taken by UAVs were collected using two multispectral camera sensors and the data has 5 and 4 raw image channels. Agriculture-Vision, a large-scale aerial farmland image data set collected for use in semantic segmentation of agricultural fields, includes image data collected from high-resolution Red Green Blue (RGB) and Near-Infrared (NIR) channels.

The data sets in the last category comprise both sensor-based and image-based data. Mid-Air Data Set [79], UZH-FPV Drone Racing Data Set [80, 81], Blackbird Data Set [82, 83], Au-air Data Set [84], Hull Drone Indoor Navigation (HDIN) Data Set [85], and INSANE Data Set [86] are some of the examples for this category.

Montefiore Institute Aerial Images and Records (Mid-Air) Data Set is a synthetic data set for low-altitude drone flights. The data set contains a large amount of synchronized data corresponding to flight logs collected from multi-modal vision sensors and flight sensors mounted on the UAV. These data set algorithms, in which data recorded several times in different climatic conditions in the same place are added, also offer the opportunity to train for resistance to visual changes. The data set includes images from the RGB camera and flight records from various simulated positioning sensors such as accelerometer, gyroscope, and GPS. UZH-FPV Drone

Racing data set presented as a visual-inertial odometry data set, the UZH-FPV Drone Racing data set was collected with a First-Person View (FPV) quadrotor racing aircraft equipped with sensors and flown by an expert pilot. The data set includes camera images, IMU data, and ground truth data collected by a laser tracker. The Blackbird unmanned aerial vehicle data set is an indoor flight data set created to evaluate agile perception. The data set includes flight data collected from sensors such as IMU, motor speed, and motion capture sensors. In addition, camera images for each flight in the data set were photorealistically rendered in various environments for potential use in future high-speed fully-autonomous drone racing research. The Au-air data set is a multi-modal UAV data set for low-altitude traffic surveillance collected for the use of object detection in aerial images. In this data set, frames are also labeled with information such as time, GPS, IMU, altitude, and linear velocity. The HDIN data set collected for indoor image-based navigation research includes image samples and mission tags obtained from UAV onboard sensors. The data set includes data collected from a front-facing camera, 3-axis accelerometer, 3-axis gyroscope, and magnetometer sensors. INSANE data set contains multimodal data collected from a Mars analog mission. The UAV used for data collection includes multiple sensors, including a navigation camera, a stereo camera, Real-Time Kinematic (RTK), Global Navigation Satellite Systems (GNSS), and IMU. The data set includes six Degrees of Freedom (DoF) outdoor, indoor-outdoor transitions, and indoor data ground truths.

The accessible and widely used data sets for UAVs in the literature are presented in Table 1.2. The chronological order of these data sets is illustrated in Figure 1.3.

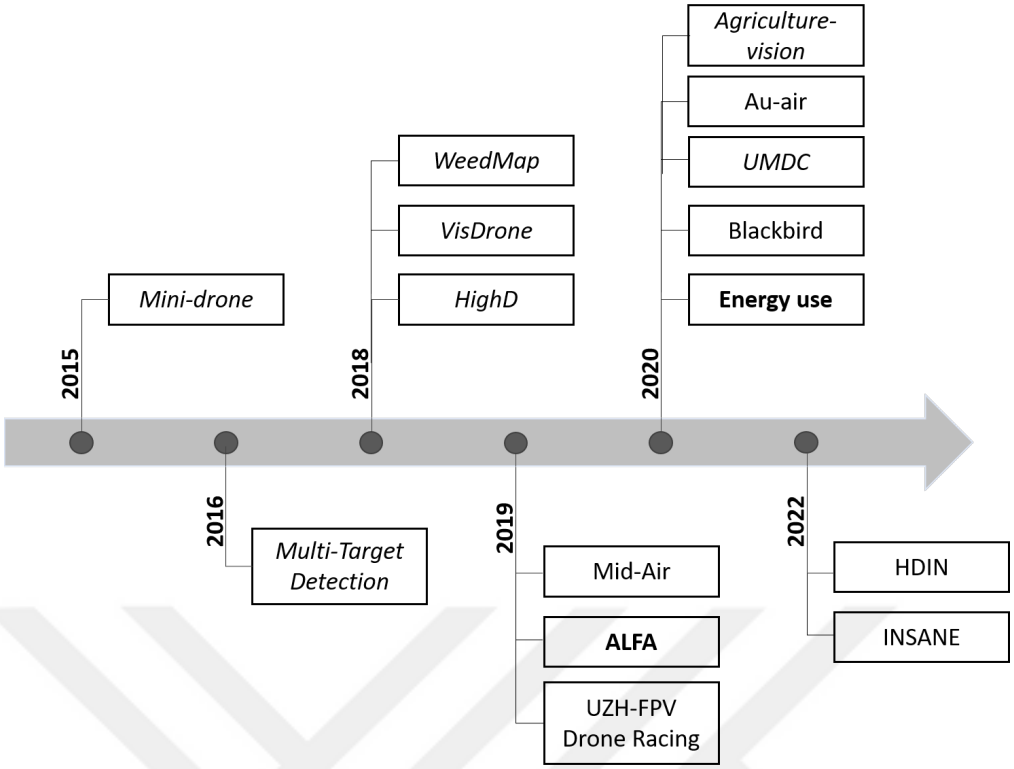

Figure 1.3 The data sets used for UAV studies. The sensor data sets are displayed in bold, the image data sets are displayed in italics, and the data sets that contain both sensor and image data are displayed regular.

Table 1.2 Data set of unmanned aerial vehicles (UAVs) published from 2015 to 2022.

Data Set	Year	Data Type	Included Sensors	Specifications
Mini-drone [71]	2015	Image data	Camera	38 different contents with a duration of 16-24 seconds
Multi-Target Detection [72]	2016	Image data	Camera	50 video sequences of 70250 frames
WeedMap [77]	2018	Image data	Camera	18,746 image files of sugar beet fields
VisDrone [73, 74]	2018	Image data	Camera	288 video clips formed by 261,908 frames and 10,209 static images
HighD [75]	2018	Image data	Camera	Six different locations, 147 hours and more than 110,500 vehicles
Mid-Air [79]	2019	Sensor and image data	RGB camera, IMU, GPS	Synthetic data set, more than 420,000 frames
ALFA [48, 68]	2019	Sensor data	Airspeed, velocity, position, IMU	47 sequences with a total of 79 minutes autonomous flights
UZH-FPV Drone Racing [80, 81]	2019	Sensor and image data	Event camera, RGB camera IMU, laser tracker	27 sequences, with more than 10 km of distance
Blackbird [82, 83]	2020	Sensor and image data	Photorealistic virtual camera IMU, motor speed	110 hours of flight data from 168 flights
In-flight Positional and Energy Use [69, 70]	2020	Sensor data	Wind, GPS, IMU, current, voltage	195 flights with additional 14 recordings
UMCD [76]	2020	Image data	Camera	50 video sequences
Au-air [84]	2020	Sensor and image data	Camera, GPS, IMU, altitude, velocity	More than 2 hours of video with 32,823 labelled frames
Agriculture-Vision [78]	2020	Image data	RGB and Near Infra-red (NIR) cameras	21,061 farmland images
HDIN [85]	2022	Sensor and image data	RGB camera, IMU	48 trajectories for steering and 15 trajectories for collision
INSANE [86]	2022	Sensor and image data	Navigation camera Stereo camera RTK GNSS, IMU	26 flights with a total distance of 2.5km

1.2 Objective of the Thesis

The assessment of flight state in autonomous UAVs relies on the functionality of onboard intelligent systems, making the monitoring of their condition a critical issue. The early detection of flight anomalies and the monitoring of sensor data are crucial for detecting faulty elements in the system, and for enhancing the reliability and safety of UAVs. This work aims to devise a suitable method for detecting anomalies in flight sensor systems in UAVs. With this proposed method, autonomous UAVs will possess the ability to detect irregularities that arise from sensor data, while flight control systems can efficiently execute suitable corrective measures in reaction to these irregularities. The objective of this thesis study can be succinctly summarized as follows.

- To provide an academic contribution to the existing body of literature on the subject of UAV anomaly detection problems by proposing a new hybrid approach that utilizes both classification and regression methods.
- To implement a periodic update function for the DNN models to enable continuous learning in the system.
- Optimizing the proposed flight sensor anomaly detection algorithm for the use of small drones.
- To implement an example edge application to demonstrate the usability of the proposed hybrid DNN-based anomaly detection algorithm on small-sized drones.

1.3 Hypothesis

The hypothesis of this thesis is that DNN-based solutions can be supplied for small drones and employed on control cards with power and memory constraints for flight sensor data anomaly detection for autonomous drones. Therefore this study explains the design and execution of a customized anomaly detection system specifically tailored for autonomous small drone use. The primary goal of the proposed system is to rapidly identify deviations in flight sensors while minimizing computational expenses to enable its application in compact drone control boards for swift and efficient control responses.

To achieve the intended objective, the first step was to review existing literature on the analysis of UAV sensor data. Subsequently, a decision was made to employ a

hybrid approach that combines both classification and regression methods for the edge application.

In the study, firstly, UAV sensor data was examined and a classification method was proposed to detect flight anomalies. The proposed classification method's accuracy was evaluated by using anomaly data sets sourced from existing literature, as well as utilizing the data set created within this study. Similarly, a regression method was also proposed to detect UAV sensor anomalies, and the performance of this method was evaluated using both the UAV data sets in the literature and the data set created within the study. The final methodology proposed in the study is a hybrid approach that integrates the classification method for identifying flight anomalies and the regression method for sensor analysis.

The next phase of the study involved an evaluation of the performance of the hybrid approach on power-constrained and low-memory edge application devices. During this stage, development boards were employed to integrate the proposed method for edge application, and a model update feature enabling ongoing model training on the edge was developed.

2
DEEP NEURAL NETWORKS AND EDGE COMPUTING FOR TIME SERIES ANALYSIS

In recent years, there has been a widespread preference for using deep learning models to address time series prediction problems. The high performance of Deep Neural Network (DNN) models in time series prediction has encouraged academics to commit considerable effort to this area and propose innovative architectural frameworks. Given this context, the subsequent sections go into great detail about the commonly used DNN methods, the steps needed to analyze data, the metrics used to evaluate models, and the applicability of these methodologies within edge computing frameworks.

2.1 Deep Neural Networks

Deep Neural Networks (DNNs) are a category of machine learning methods that include complex mathematical learning algorithms to emulate human learning processes. In DNNs, which consist of multi-layer Artificial Neural Networks (ANNs), each layer is responsible for performing a range of tasks, including complicated operations such as representation and abstraction. These operations enable the extraction of distinct features from the input data. As the data traverses each layer, simple operations are performed on the data and the results are then transferred to the next levels. Subsequent layers inside the network progressively prioritize higher-level features compared to their preceding counterparts, until the network generates the output. The intermediate layers between the input and output layers in DNNs are referred to as hidden layers. This distinction sets DNNs apart from conventional ANNs, which normally comprise just one or two hidden layers. In contrast, DNNs are characterized by their utilization of several hidden layers, often numbering in the dozens. [87].

The selection of DNN architectures for applications often varies according to the particular problem that researchers are focused on. The subsequent sections provide

an in-depth explanation of the DNN methodologies employed for the analysis of time-series sensor data.

2.1.1 Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a classic type of artificial neural network that features a fully connected architecture. In deep learning, MLP can be used to incorporate several hidden layers to form a Fully Connected Neural Network (FCNN), in which each neuron is connected to all neurons in the previous layer, as depicted in Figure 2.1. MLPs are commonly employed for applications like classification and regression. Inputs are primarily passed to neurons in hidden layers, then processed by an activation function and forwarded to the subsequent layer. In this architecture, training is generally carried out using an optimization algorithm called backpropagation. This process uses a loss function to evaluate the degree to which the model's predictions match the actual labels and then performs backpropagation to update the weights and biases to minimize this error. Some research suggests that MLPs serve as a simple yet robust benchmark for time series classification using deep neural networks [88].

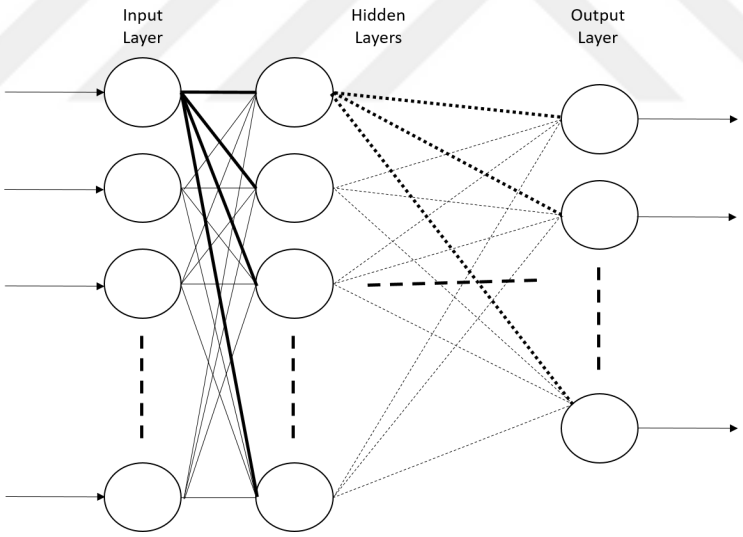

Figure 2.1 A fully connected deep neural network.

2.1.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a type of deep learning approach frequently utilized to process sequential input data. RNNs differ from other DNN approaches by not only considering the current input at a particular time step but also taking into account the information that has been previously provided to the network. RNNs can retain information from previous inputs and generate outputs

by evaluating the relationship between past and current data. However, in basic RNNs, some data-related parameters may disappear from the system memory after a specific number of iterations. Various improved RNN architectures such as Long Short-Term Memory (LSTM) [89] and Gated Recurrent Units (GRUs) [58, 90] have been proposed to solve this problem, also called the vanishing gradient problem [91].

Long Short-Term Memory (LSTM): LSTM is a type of RNN architecture that was introduced to the field of deep learning by Hochreiter and Schmidhuber in 1997 [89]. In contrast to conventional feed-forward neural networks, LSTM networks have feedback connections, enabling them to handle not just present input but also sequential data effectively. This architecture incorporates gates within its cells to effectively control the information flow within its memory, enabling the processing of sequential data over time. An LSTM cell has three gates called an input gate, an output gate, and a forget gate as shown in Figure 2.2. These gates play a crucial role in regulating the information flow within the cell. The input of a cell at a given time is x_t, the previous cell memory is c_{t-1} and the output of the previous cell is h_{t-1}. The cell has two different outputs called c_t and h_t. The input and forget gates given in equations (2.1) and (2.2) control the cell state, which is the memory transmitted through time, given in equations (2.4-2.5). The output gate result given in equation (2.3) gives the output of the cell, also called the hidden state, as seen in equation (2.6).

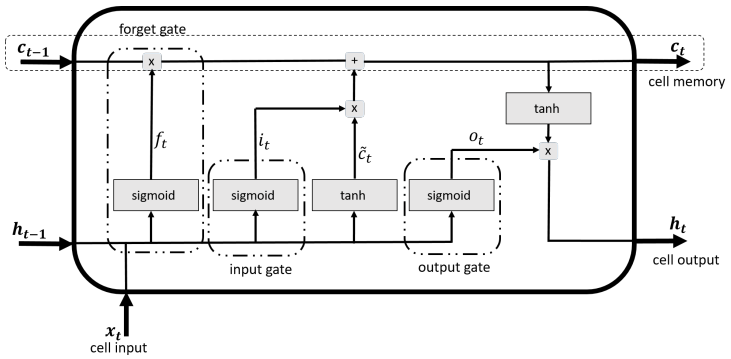

Figure 2.2 Operations in an LSTM unit.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.1}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.3}$$

$$\tilde{c}_t = tan(W_c x_t + U_c h_{t-1} + b_c) \tag{2.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_{t-1} \tag{2.5}$$

$$h_t = o_t \odot tan(c_t) \tag{2.6}$$

Gated Recurrent Unit (GRU): GRU, similar to LSTM, is an enhanced version of the RNN structure designed to address the vanishing gradient problem [90, 92]. GRU unit possesses a reduced number of parameters and computational requirements in comparison to the LSTM model due to the absence of an output gate. A GRU cell consists of two gates called reset and update gates as shown in Figure 2.3. These elements are responsible for regulating the flow of information inside the unit similar to LSTM. The input of a cell at a given time is x_t and the output of the previous cell is h_{t-1}. The calculated cell output is called h_t and also acts as the hidden state. The result of the hidden state is calculated by equation (2.9-2.10) using update (2.7) and reset gates (2.8).

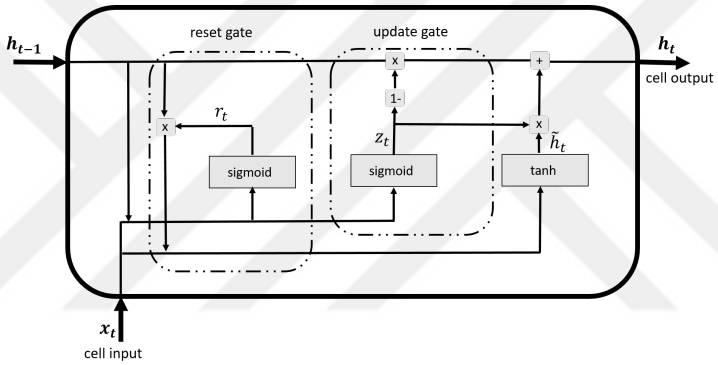

Figure 2.3 Operations in a GRU unit.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{2.7}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2.8}$$

$$\tilde{h}_t = tan(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{2.9}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{2.10}$$

2.1.3 Convolutional Neural Networks

The Convolutional Neural Network (CNN) is a popular DNN method frequently used for feature extraction in classification tasks. CNNs leverage the inherent hierarchical structure present in the data by modifying the architecture of MLPs where each neuron in a given layer is linked to all neurons in the subsequent layer. By employing filters, CNNs can combine increasingly intricate patterns by utilizing

smaller and more simple patterns. CNNs optimize filters (or kernels) through machine learning and use less preprocessing compared to other classification algorithms. In at least one of its layers, convolution is employed instead of generic matrix multiplication [93]. Temporal Convolutional Networks (TCNs) with dilated causal one-dimensional (1D) convolutional layers are among other variations of CNNs that are being employed in academic research nowadays [94, 95].

Temporal Convolutional Network (TCN): TCN is a deep learning architecture that employs dilated, causal 1D convolution layers specifically designed for processing 1D input. This architectural design was initially employed for video-based action segmentation [94]. It has dilated CNN layers, which are capable of computing low-level characteristics. The utilization of dilated convolutional layers within the TCN framework enables the extraction of features from extended time sequences in a more effective manner when compared to RNNs. This is achieved by the incorporation of a large receptive field by dilation. The TCN architecture, as seen in Figure 2.4, is composed of a residual block that incorporates dilated convolutional layers, normalization layers, activation layers, and dropout layers. Additionally, there can be an optional 1x1 convolutional layer that is employed when the dimensions of the residual input and output differ. Within dilated convolutional layers, the computation of the relevant hidden unit value for each hidden vector is done by the convolution of the input values with the appropriate filter. The subsequent hidden layer vector is generated by utilizing the computed hidden vector and dilated calculations. The last dilated layer yields the output vector.

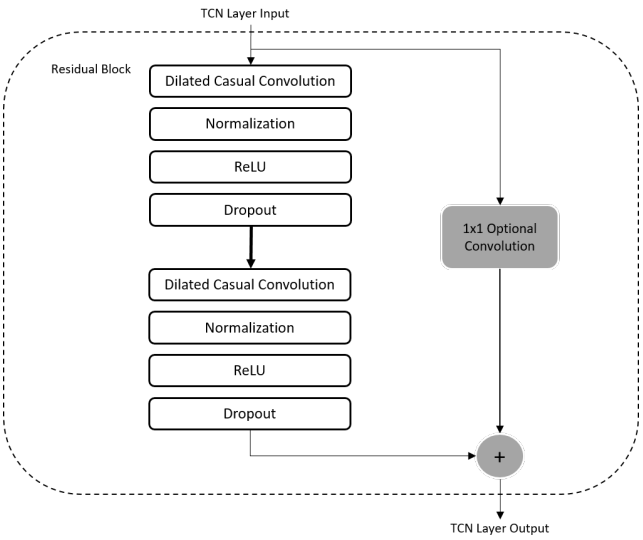

Figure 2.4 TCN architecture.

2.1.4 AutoEncoders

AutoEncoders (AEs) are a type of DNNs and are used to represent data by compressing it and then restructuring it. They are composed of a network that includes both an encoder and a decoder. The encoder network acquires knowledge about the feature vector of the data and produces a probability distribution. In contrast, the decoder network utilizes the aforementioned probability distribution to generate a feature vector that closely approximates the initial input [96]. AEs are generally used for feature extraction in UAV data in addition to data compression and reconstruction.

Autoencoders are classified into two categories based on the number of neurons in the encoder layer as undercomplete and overcomplete autoencoders. An undercomplete autoencoder is defined as having an encoder layer with a lesser number of neurons compared to the dimensionality of the input data and the encoder reduces the dimensionality of the input data by compressing it. Undercomplete AE often exhibits enhanced generalization capabilities, making it a preferred choice for emphasizing important features of the data while minimizing unnecessary details. An overcomplete autoencoder is characterized by an encoder layer that has a higher number of neurons than the amount of the input data. In other words, the encoder expands the input data into a higher dimensional encoding. Hence overcomplete autoencoders can be used to capture finer details and relationships in the data because the autoencoder tends to preserve many details of the data.

Furthermore, AEs also differ based on the fundamental DNN structure they employ. Fully connected AE is the most basic type of autoencoder and it consists of fully connected layers with matching dimensions between the encoder and decoder layers. These autoencoders aim to use the least complicated representation of the input data. Whereas convolutional AEs are specifically designed to be used in multi-dimensional data such as image data. These networks extract features using the CNN architecture and perform more efficient dimensionality reduction.

2.2 Main Steps of Sensor Data Analysis

UAVs are equipped with onboard control systems that include fundamental flight sensors, as well as supplementary sensors that frequently differ depending on the intended use of the UAV. Therefore the objectives and fundamental procedures of studies differ depending on the type of sensor data acquired from the UAV and the flight's objective. In studies involving the utilization of time-series sensor data acquired from UAVs, the first step involves the application of preprocessing

techniques such as data cleaning and transformation. Subsequently, the feature selection process occurs, followed by the use of a selected regressor or classifier for data modeling. These primary steps are shown in Figure 2.5.

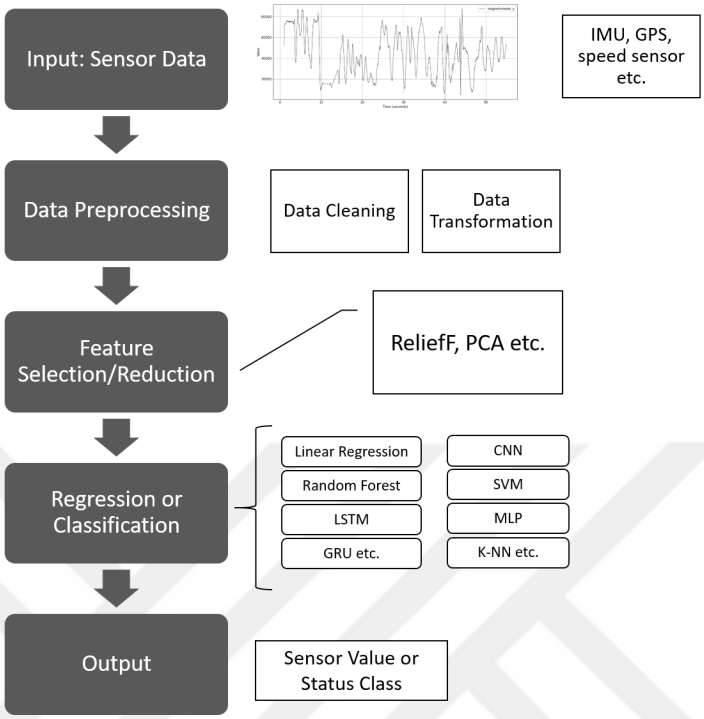

Figure 2.5 Block diagram of the sensor data analysis main steps.

- **Data Acquisition:** At this step, input data is obtained with the onboard sensors, in accordance with the application's purpose. The sensor data selected as input usually includes various IMU outputs such as a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer data. Furthermore, a range of supplementary sensors, including the GPS, air pressure sensors, wind speed sensors, and altitude sensors, can also be employed. The aforementioned data is documented in flight logs as time series throughout UAV flights.

- **Data Preprocessing:** Sensor data, typically collected as univariate or multivariate time series, includes a substantial amount of information that unfortunately appears unintelligible upon initial inspection. Incomplete or unordered timestamps, missing values, outliers, and noise are frequent issues in relation to this category of data. To effectively address such issues, it is essential to implement a range of preprocessing techniques on the time series data prior to model development. Time series data is frequently found in an unorganized state, with timestamps that may not be appropriately arranged for the accurate analysis of sensor data. Therefore, the initial procedure to be

undertaken is sorting the timestamps. Data cleaning is also a crucial step in the data processing pipeline, where missing data is added, and inaccurate data is corrected, repaired, or removed from the data set. This step can also include smoothing out noisy data [97]. Following the cleaning process, it is necessary to carry out certain transformations to turn the data into appropriate forms for the following analysis. This particular stage often involves collecting and combining all relevant data in a uniform manner, followed by the normalization and scaling of the data to a common range. This process enhances the accuracy of feature comparison [98]. After these steps, typically, the data set is prepared for the following feature selection step.

- **Feature Selection:** Feature selection is a crucial step in regression or classification tasks, which involves the selection of the most important parameters. The features selected at this phase will then be used for the training of DNN models. The inclusion of more features during the training phase leads to an escalation in computational costs, while certain characteristics have a negligible impact on the system's output. In this step supervised methods such as ReliefF [99] or unsupervised methods such as Principal Component Analysis (PCA) [100] can be utilized [101].

- **Regression/Classification:** Following the preprocessing and feature selection on the sensor data, it is required to conduct accurate model training using a machine learning technique that aligns with the desired application. Then, using the trained model, regression or classification operations will be performed. Regression can be summarized as predicting the system output value by matching the given inputs. For this purpose, linear regression [102] and random forest [103] have been widely used, and especially in recent years, RNNs such as LSTM [49] and GRUs [104] are also used frequently. Similarly, classification can be defined as the prediction of a categorical label at the output from the given inputs, and for this purpose, in addition to widely used CNNs [105], classifiers such as Support Vector Machines (SVMs) [62], MLP [106] and k-Nearest Neighbors (k-NNs) [33] are also used.

- **Output:** Similar to model training preparations, the relevant sensor test data must first be preprocessed before being input into the trained model. In the model output, the projected value or class is generated as the result of regression or classification, respectively.

2.3 Edge Computing

Edge computing refers to a technique in which the processing and analysis of data occur on devices that have limited resources, such as low-power and memory-capacity development boards. These devices are situated near the source of data generation or collection. The concept of edge computing in machine learning aims to execute this process locally, resulting in faster processing times compared to the conventional approach of running machine learning algorithms on servers located in large-scale data centers. Edge computing in real-time systems generally provides low latency and rapid reaction times to systems. And the importance of edge computing in drone control may also be considered from several different aspects.

- Drones often require rapid response times, which necessitate real-time control. Edge applications enable the placement of computing resources directly on the drone control board, eliminating the latency caused by data transmission between the drone and a central server.
- Drones can gather sensitive information, therefore ensuring the security and privacy of this data is of utmost importance. Edge computing allows for data processing to occur directly on the drone, minimizing security vulnerabilities associated with data transfer.
- Drones frequently gather high-resolution images and sensor data. Transmitting this volume of data to central servers requires large bandwidths. Utilizing edge applications specifically for flight sensor data monitoring helps in minimizing network traffic.

2.4 Evaluation Metrics

Evaluation metrics are used to assess and quantify the effectiveness of the systems developed to address a certain problem. To assess the success of deep learning models designed to address challenges pertaining to UAVs, a range of evaluation metrics can be used. The metrics, which may differ depending on the specifics of the problem, may be classified into two groups based on their application in either classification or regression problems. DNNs provide an output through the execution of several mathematical operations. If the resulting output is in the form of categories, the problem is referred to as classification, and if the resulting output is in the form of numerical values, the problem is referred to as regression.

Classification problems are problems in which the prediction of the class labels

given for the input data is performed. In these applications, the number of output classes may vary, for example, there may be two classes in anomaly detection studies (binary classification), and there may be more than two classes in applications such as object recognition (multi-class classification) [107, 108]. On the other hand regression, refers to the process of constructing a model using previous data to generate predictions about new data that has not been previously observed in the data set. In the context of these applications, the resulting output is continuous as opposed to being categorical. Therefore, it is not feasible to measure accuracy for a regression model. The evaluation of the performance of these models is conducted based on the magnitude of inaccuracy observed in their predictions [109–114]. Table 2.1, provides a comprehensive overview and explanation of the main evaluation metrics often used in DNN performance evaluations. It also includes the mathematical formulations associated with each metric. Additionally, in the following list, the metrics used in this study are explained in detail.

Table 2.1 Common metrics used in DNN performance evaluations.

Metric	Problem Type	Definition	Formulation
True Positive Ratio (TPR)	Classification	Probability that an actual positive will test positive	$\frac{TP}{TP+FN}$
False Positive Ratio (FPR)	Classification	Probability that a positive result will be given when the true value is negative	$\frac{FP}{FP+TN}$
Accuracy	Classification	How often a classifier classifies correctly	$\frac{TP+TN}{TP+FP+TN+FN}$
Precision	Classification	How many correctly predicted cases are actually positive	$\frac{TP}{TP+FP}$
Recall	Classification	How many of the true positives are correctly classified	$\frac{TP}{TP+FN}$
F-score	Classification	Measure of the accuracy of a performed test	$2 \times \frac{Precision \times Recall}{Precision + Recall}$
Mean Squared Error (MSE)	Regression	Gives an absolute number of how the predicted results differ from the actual values	$\frac{1}{n}\sum_{i=1}^{n}(predicted_i - measured_i)^2$
Root Mean Square Error (RMSE)	Regression	A measure of how the prediction errors are spread out	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(predicted_i - measured_i)^2}$
Mean Absolute Error (MAE)	Regression	The sum of the absolute prediction errors to the number of samples	$\frac{1}{n}\sum_{i=1}^{n}\lvert predicted_i - measured_i\rvert$
Mean Absolute Percentage Error (MAPE)	Regression	Intuitive interpretation of relative error	$\frac{1}{n}\sum_{i=1}^{n}\frac{\lvert predicted_i - measured_i\rvert}{measured_i}$
Coefficient of Determination (R^2)	Regression	The difference between the samples in the data set and the predictions made by the mode	$1 - \frac{\sum_{i=1}^{n}(predicted_i - measured_i)^2}{\sum_{i=1}^{n}(predicted_i - \frac{1}{n}\sum_{i=1}^{n}predicted_i)^2}$

- **Accuracy:** Accuracy measures how often a classifier classifies correctly and

is calculated by dividing the number of correct predictions by the total number of predictions. When any model gives a high accuracy rate, that model can be considered to perform very well, but for unbalanced data sets this assessment is not accurate. In classification problems, the classes of data are often unbalanced. For this reason, a more detailed examination of the classification results is carried out by using different evaluation metrics such as precision and recall.

- **Precision:** Precision is a metric that shows how many correctly predicted cases are actually positive. For a class, precision is defined as the number of correctly predicted positives divided by the number of all predicted positives and is useful when false positives are a higher concern than false negatives.

- **Recall:** Recall shows how many of the true positives are correctly classified. For a class, recall is defined as the number of correctly predicted positives divided by the number of all actual positives. It is a useful metric when false negatives are more important than false positives, for example where false alarms are not important, but true positives should not be overlooked.

- **F-score:** F-score is a measure of the accuracy of a performed test. It gives a general idea of the precision and recall criteria in classification and takes its maximum value when precision and recall metrics are equal. It is calculated as the harmonic average of precision and recall and takes a value between 0 and 1.

- **Confusion matrix:** Confusion matrix is a table used in classification problems, consisting of four different parameters, True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), of predicted and actual values. The confusion matrix is used to calculate other evaluation metrics such as accuracy, precision, and recall. A confusion matrix is given in Figure 2.6, other classification metrics can be calculated using the values given in this matrix.

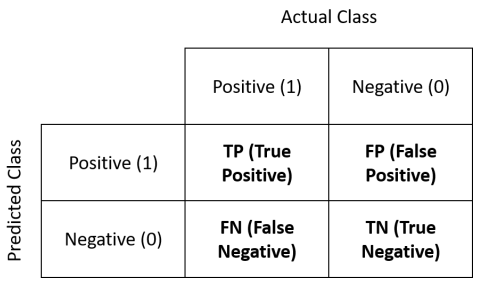

Figure 2.6 General confusion matrix notation.

- **Root Mean Square Error (RMSE):** Root Mean Squared Error (RMSE) is the standard deviation of the prediction errors, and it is calculated by taking the square root of the mean of the squared errors. RMSE is a measure of how spread out these prediction errors are and how concentrated the data is around the regression line. Taking the square root of the error in this criterion ensures that the error and predicted data units are the same for the RMSE, unlike the MSE. This causes the MSE metric to be preferred for model training and the RMSE metric for model performance evaluation. The effect of each error on the RMSE is proportional to the size of the squared error; therefore, larger errors have a large effect on the metric, making the RMSE sensitive to outliers.

- **Coefficient of Determination (R-squared, R^2):** The coefficient of determination is the difference between the samples in the data set and the predictions made by the model, and it works by measuring the amount of variance in the predictions. The closer the R^2 value is to 1, the higher the model success. Although MAE and MSE are metrics that give results depending on the data set they are used, the R^2 score is an evaluation metric independent of the data set features. Since MSE, MAE, RMSE metrics can range from zero to $+\infty$, a single value of these does not provide much information about the performance of the regression, but R^2 metric only results in a high score if the majority of the elements of a ground truth are correctly predicted.

3
FLIGHT DATA OF AERIAL VEHICLES

The objective of this work is to identify anomalies in drone sensor data through the utilization of several DNN methodologies. The proposed method for this purpose is a hybrid approach that combines both classification and regression techniques. The study used a range of sensor data sets to evaluate the performance of the proposed methods. The anomaly data collected during commercial flights was utilized to evaluate the accuracy of classification methods, while the drone flight sensor data set was used to evaluate the performance of regression methods. Moreover, within the scope of the study, data acquired during UAV flights was gathered and labeled using our UAV sensor setup. Afterward, this data set was also used for the evaluation of classification and regression methods, in addition to serving as a means to evaluate the proposed hybrid approach. This chapter provides an overview of the data sets utilized in this thesis from the existing literature as well as the data sets collected during the study.

3.1 Curated 4 Class Anomaly Detection Data Set

This study utilizes the Curated 4 Class Anomaly Detection Data Set [1] [115], which is publically accessible and obtained through National Aeronautics and Space Administration's (NASA) DASHlink website, to implement the proposed methodologies for the classification of anomalies in-flight data. The dataset presented in this study comprises authentic data obtained from a regional jet operating in commercial service for three years. Its purpose is to contribute to the existing body of literature by addressing potential issues in aviation and improving overall flight safety.

The anomaly dataset comprises 20 different flight parameters (see Table 3.1) that were collected from flight sensors during a 160-second timeframe in the final approach before landing, encompassing a total of 99837 flights. The data set

[1] https://c3.ndc.nasa.gov/dashlink/resources/1018/

consists of three distinct anomaly classes and one nominal class. The distribution of these classes may be observed in Table 3.2.

Table 3.1 Flight data variables in the curated 4-class anomaly detection data set.

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

Table 3.2 Distribution of the curated 4-class anomaly detection data set class samples.

Class	Number of samples
Nominal	89663
Speed High	7013
Path High	2207
Flaps Late Setting	954

In this study, the data set included three separate anomaly types, which we consolidated into a singular class for anomaly detection. The anomaly detection process included performing binary classification, specifically categorizing instances as either nominal or anomalous. The altered dataset, which includes two classes, either class 0 (nominal) or class 1 (anomaly), has been divided into stratified folds to facilitate the execution of 5-fold Cross-Validation (CV) procedures during the study. Within each stratified fold, there are distinct sets for training and testing purposes. The process of preparing the data set for this investigation is also depicted in Figure 3.1.

3.2 In-flight Positional and Energy Use Dataset

The study utilized the in-flight positional and energy use data set of package delivery quadcopter UAVs [69, 116] as one of its primary data sets. This particular data set was employed to evaluate the accuracy of the regression methods presented in the study. This data set was generated with a DJI Matrice 100 (M100) drone, which collected data simultaneously from onboard sensors.

The data set incorporates a variety of sensors, including wind sensors, GPS and IMU sensors, as well as current and voltage sensors. Battery power consumption data can also be obtained by the sensor data readings in the dataset [70, 117]. It contains a total of 196 flights that were conducted under various operational

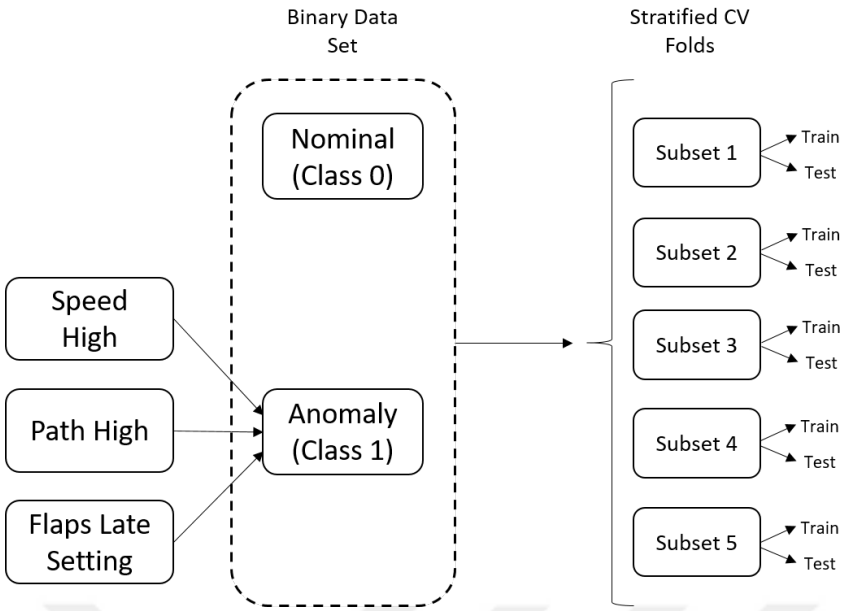

Figure 3.1 Organizing and dividing the curated 4-class anomaly detection data set for experiments.

settings. Furthermore, an additional 13 records in the data set were conducted to evaluate the auxiliary power and hover conditions of the drone. A total of 175 flight data recordings have been selected for this study due to their collection from an identical path. The parameters utilized from this data set in this study are provided in Table 3.3.

Table 3.3 Sensor parameters utilized from the in-flight positional and energy use dataset.

Flight Sensor Variables	Battery Sensor Data	Other Parameters
Wind speed Vertical speed Wind angle Airspeed-x Airspeed-y	Power consumption	Payload Air density

3.3 Collected UAV Anomaly Data Set

The performance evaluation of the suggested hybrid technique was conducted using the UAV flight sensor dataset that was specifically constructed for this study. To achieve the intended objective, the Corby CX014 Zoom Voyager Drone was utilized alongside the Arduino Nano 33 BLE Sense Rev2 [118] development board, which was added to the drone.

The Nano 33 BLE Sense Rev2 board is a Microcontroller Unit (MCU) that has the capability to establish a connection using Bluetooth®Low Energy (BLE), in

addition to being equipped with a range of integrated sensors. The proximity, motion, temperature, humidity, and audio sensors are some of these sensors. The board is equipped with a 9-axis IMU that includes a Three-Dimensional (3D) accelerometer, gyroscope, and magnetometer. With this configuration, the MCU can detect orientation, motion, and vibrations, and this information can then be transmitted over a BLE connection. In this study, data was gathered using 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer on this board during the flights and then labeled accordingly. A visualization of the quadcopter and sensor board configuration employed for data collection is presented in Figure 3.2.

Flights conducted to acquire sensor data include four flight scenarios, such as normal flight, rollover, drift, and sudden drop. The data collection was conducted indoors to mitigate the influence of external environmental factors on this small drone flight sensor data set. During the process of data collection, the movement of the UAV was recorded using IMU sensors at intervals of 100 milliseconds. These recorded data points were transmitted to the computer using a BLE connection. After this stage, the take off and landing data records from the flights were not included in the records. Subsequently, the obtained flight data was partitioned into segments of 20 time steps, each consisting of 2 seconds of data. These segments were then assigned the corresponding label from four different status labels. Upon the completion of data collection, flight segments that had missing data were removed from the data set in the initial analysis, and then the necessary pre-processing steps were carried out.

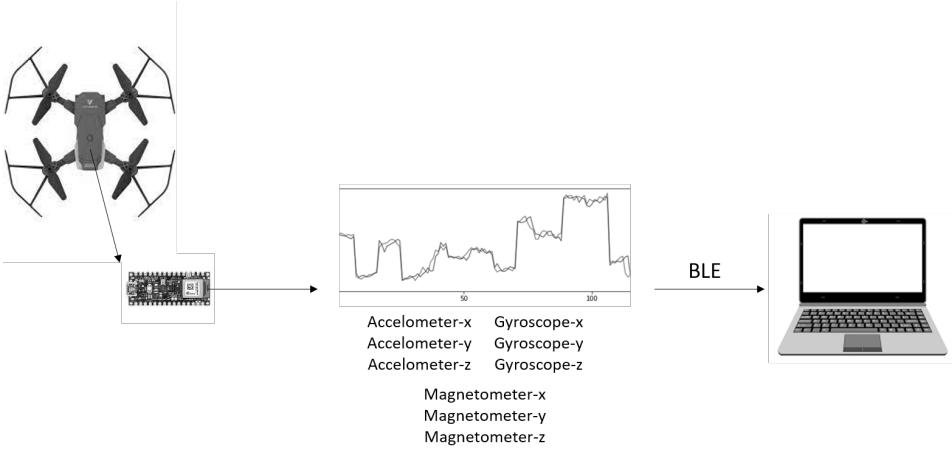

Figure 3.2 UAV flight data collection setup.

This study's collected data set included three different anomaly types and a nominal class (Table 3.4) similar to the 4-Class Anomaly Detection data set from DASHlink. This data set was similarly organized as a binary classification data set for anomaly detection by categorizing flight segments as either normal or anomalous. The

Table 3.4 Distribution of the collected UAV data set class samples.

Class	Number of samples
Nominal	2480
Rollover	539
Drift	599
Sudden drop	759

modified data set, consisting of two classes, either class 0 (nominal) or class 1 (anomaly), has been divided into stratified folds to implement 5-fold CV techniques throughout the study. In each stratified fold, distinct subsets are allocated for training and testing. The methodology employed for data set preparation in this inquiry closely resembled that of the prior classification data set and is depicted in Figure 3.3.

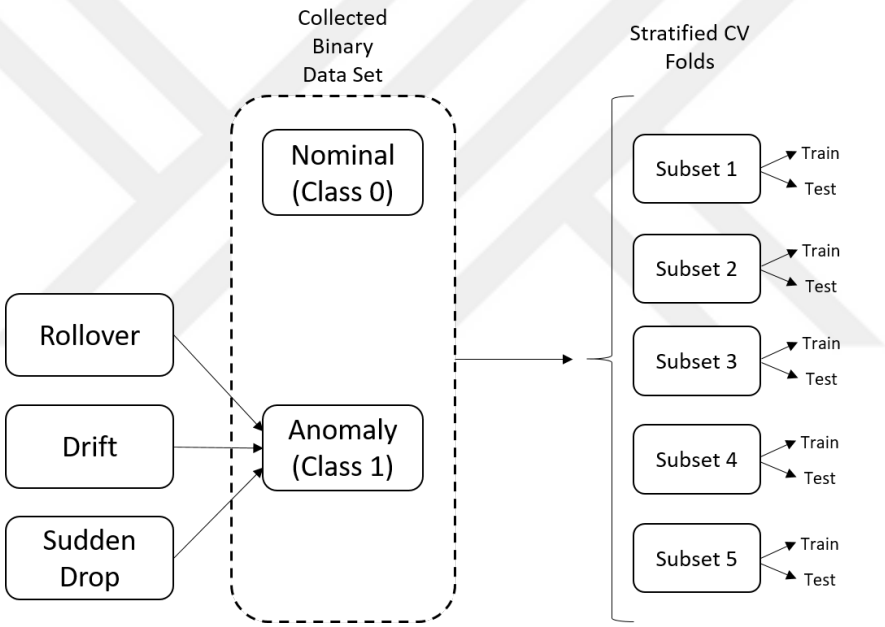

Figure 3.3 Organizing and dividing the collected UAV data set for experiments.

4
SENSOR DATA ANOMALY DETECTION WITH DEEP NEURAL NETWORKS

UAVs, employed in many applications, are required to detect anomalies in flight sensor data to operate autonomously. A flight anomaly happens when the UAV deviates from its pattern or encounters an unexpected event. The identification of flight anomalies is frequently achieved through continuous monitoring of various parameters, such as GPS data, airspeed, and altitude. Flight anomalies encompass several occurrences, such as rapid changes in airspeed, position, or altitude. This study aims to enhance the security and performance of UAV-based applications by developing a new hybrid approach that can effectively detect anomalies in UAV flights. The overall framework of this study is presented in Figure 4.1 and it presents a novel approach that integrates classification and regression methods for anomaly classification, data analysis, and data monitoring to detect and identify flight anomalies in UAV flights. Following, the discussion of the various experiments conducted on classification and regression approaches, and a thorough explanation of the proposed hybrid method is provided in the next sections.

4.1 Sensor Data Anomaly Detection Using Classification

An autoencoder-based DNN called AnoSense is proposed in the thesis for detecting anomalies in aircraft flight segments. In experimental studies, two different data sets are used to evaluate the model performance of the proposed method AnoSense, using two experimental setups.

The proposed AnoSense classification method employs an AE architecture comprising convolutional and MLP layers. The filter and neuron sizes for the layers of the AnoSense model can be seen in Figure 4.2. The sigmoid function was chosen as the activation function for the output layer of the model, while binary cross-entropy and Adam were found to be the optimal loss function and optimizer, respectively.

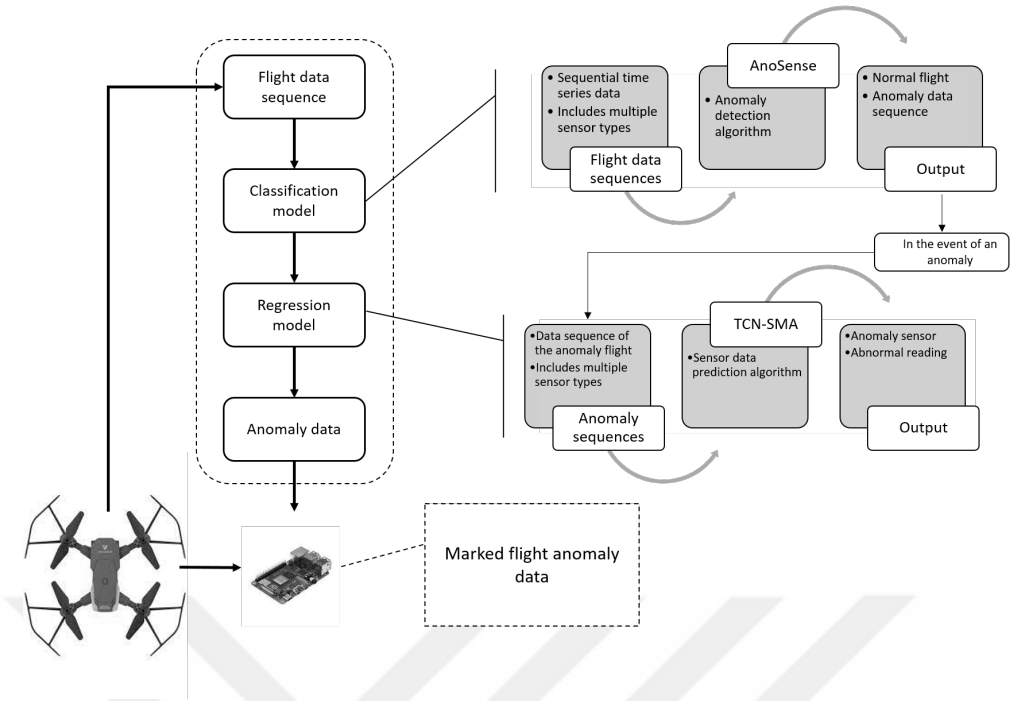

Figure 4.1 Overview of the proposed hybrid anomaly detection framework.

The DNN model for the classification stage of the proposed hybrid approach was chosen based on several experimental studies. This study utilized the curated 4-class anomaly detection data set and the collected UAV anomaly data set for classification model experiments. These datasets were first organized as binary data sets, as described in earlier chapters, and were then employed in the experiments. Two different experimental setups were used for the two datasets. The following provides a comprehensive explanation of the experimental setups as well as the experimental results of the classification tests.

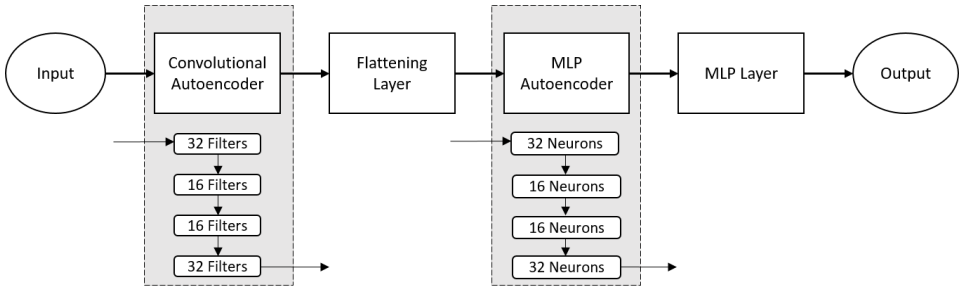

Figure 4.2 AnoSense method model architecture.

4.1.1 Classification Experiments on the Curated 4-Class Anomaly Detection Data Set

The first setup is specifically tailored for research conducted on the curated 4-class anomaly detection data set, as seen in Figure 4.3. The data set used in this setup

includes flight data consisting of 20 different flight parameters for a 160-second time frame for each sample. In addition, the data for each sample was labeled with binary output labels as nominal and anomaly. During classification, data with the shape of (160x20) is given as input to each DNN model, and an output label is predicted.

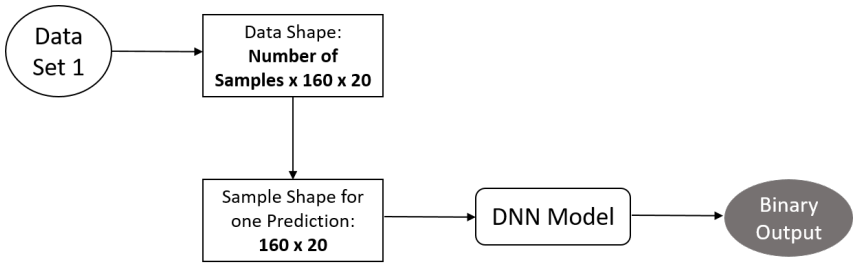

Figure 4.3 Block diagram of the experimental setup for the curated 4-class anomaly detection data set (Data set 1 refers to the curated 4-class anomaly detection data set.).

In the first experimental setup, AnoSense was compared with LSTM, GRU, CNN, and TCN, which are commonly employed methodologies in the context of time series problems, specifically for anomaly detection. AnoSense was first trained with the training subsets for the experiments, and then its performance was compared with classical DNN methods in the Python environment using the test subsets. 5-fold cross-validation was used for the model evaluations using the stratified subsets explained in the previous chapters. During the training phase, the model was trained using the training subsets, which is 70% from each stratified fold. Additionally, training validations were performed by separating 20% of this training data. For the comparison of various DNN methods, the trained models were evaluated by employing test subsets specific to their particular folds within the Python environment.

The performance of the proposed AnoSense was evaluated using the 5-fold cross-validation accuracy, precision, and F-score metrics. The data in Table 4.1 demonstrates that the proposed AnoSense method exhibits superior performance across all evaluation metrics.

Here, the performance metrics of the AnoSense models was examined using 5-fold cross-validation. The results of these tests demonstrate that AnoSense models outperform LSTM, GRU, CNN, and TCN models in the Python environment, by achieving an accuracy of 97.47%. In addition, the AnoSense model was also the best performing method in all metrics with an average score of 98.12% precision, 98.08% recall and 97.62% F-score value in these tests.

Table 4.1 Test results for the 5-fold cross-validation performance metrics of the DNN models using the curated 4-class anomaly detection data set (Best results are shown in bold.).

Method	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
LSTM	96.80	87.17	80.50	83.70
GRU	96.55	86.06	79.02	82.37
CNN	96.52	88.41	75.85	81.64
TCN	97.12	88.18	82.88	85.43
AnoSense	**97.47**	**98.77**	**95.36**	**97.03**

4.1.2 Classification Experiments on the Collected UAV Anomaly Data Set

Following the initial experiments which demonstrated the success of the AnoSense method on imbalanced anomaly data sets, the performance of the proposed method was further evaluated on the data set created during this study. The data set collected and used in this setup includes flight data consisting of nine different flight parameters for 20 time steps for each sample. In addition, the data for each sample was also labeled as binary outputs: nominal and anomaly. The input data shapes for this setup are also given in Figure 4.4

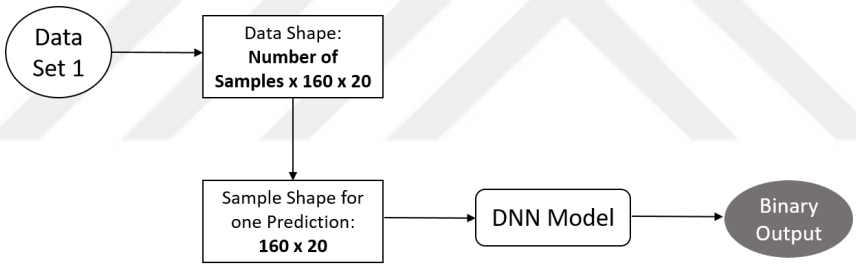

Figure 4.4 Block diagram of the experimental setup for the collected UAV anomaly data set (Data set 3 refers to the collected UAV anomaly data set.).

For experiments in this setup, the data set was initially divided into stratified folds to enable 5-fold cross-validation. Subsequently, each fold was further separated into training and test subsets, maintaining a ratio of 70:30. During the training process, a portion equivalent to 20% of the training sets was assigned for validation purposes. AnoSense was again compared with LSTM, GRU, CNN, and TCN methods by evaluating the trained models using test sets specific to their particular subsets within the Python environment. The performance of the proposed AnoSense was evaluated using the 5-fold cross-validation accuracy, precision, recall and F-score metrics and the results are given in Table 4.2. The experimental results on the collected data set also show that the proposed AnoSense method exhibits superior performance across all evaluation metrics for anomaly detection.

When the results of the experiments performed on both data sets for classification

Table 4.2 Test results for the 5-fold cross-validation performance metrics of the DNN models using the collected UAV anomaly data set (Best results are shown in bold.).

Method	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
LSTM	96.18	97.38	97.38	96.98
GRU	95.94	96.95	96.69	96.53
CNN	96.43	97.05	96.48	96.66
TCN	96.31	97.24	97.14	96.90
AnoSense	**96.65**	**98.12**	**98.08**	**97.62**

model selection are examined, it is seen that for the curated 4-class anomaly detection data set, the proposed AnoSense method has a 97.47% accuracy score and a 97.03% F-score, which is especially important when using imbalanced data. In addition, the proposed method also gives a 96.65% accuracy score and a 97.62% F-score for the tests conducted on the collected UAV anomaly data set. These results show that for the average performance metrics, AnoSense method exhibits superior classification performance for both of the utilized anomaly datasets. Especially for the collected UAV anomaly data set it shows a considerable improvement for the precision, recall, and F-score metrics. Consequently, this method was selected as the classifier in the first step of the hybrid approach.

4.2 Sensor Data Anomaly Detection Using Regression

In this study, a methodology is employed that combines the use of Temporal Convolutional Network with Simple Moving Average (TCN-SMA), for flight sensor anomaly detection [119, 120]. In experimental studies, two different data sets are used to evaluate the model performance of the proposed method, TCN-SMA, using two different experimental setups for two different flight sensor data sets. The proposed method is evaluated first on the in-flight positional and energy use data set of package delivery quadcopter UAVs and then on the collected UAV anomaly data set.

TCN is an architecture consisting of a CNN layer with dilated convolution, activation, normalization, and dropout layers. The proposed approach employs the TCN together with the Simple Moving Average (SMA) algorithm to enhance the predictive capabilities of DNN, models and improve the accuracy of anomaly detection. In the first experimental setup for the regression, another similar approach, the Temporal Convolutional Network with Leaky Rectified Linear Unit (LR-TCN) was also tested for more in-depth performance analysis [119]. The LR-TCN is a modified method obtained by replacing the Rectified Linear Unit

(ReLU) activation function, of the TCN architecture, with the Leaky Rectified Linear Unit (Leaky ReLU) function. Both the TCN and LR-TCN architectures are given in Figure 4.5 for better comparison.

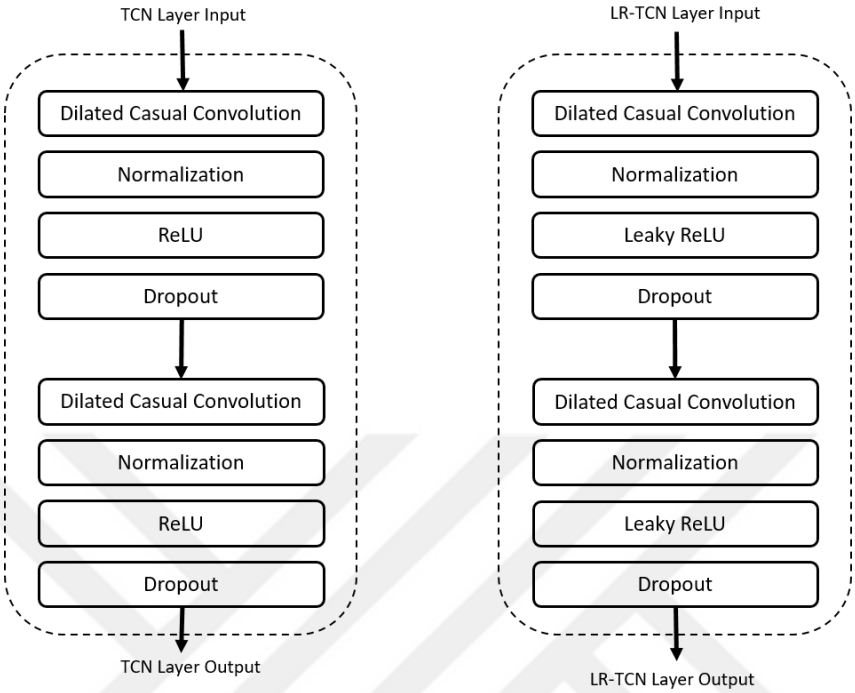

Figure 4.5 Side-by-side comparison of TCN and LR-TCN architectures.

ReLU, which is frequently used in the artificial neural network activation layers, is a function defined as the positive part of a neuron's input. On the other hand, Leaky ReLU has a small slope for negative values, unlike the ReLU function [121]. The definitions of ReLU and Leaky ReLU activation functions are also given in Equations 4.1 and 4.2, respectively.

$$f(x) = \begin{cases} x & \text{if x > 0} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

$$f(x) = \begin{cases} x & \text{if x > 0} \\ 0.01x & \text{otherwise} \end{cases} \tag{4.2}$$

Since Leaky ReLU has a small negative slope it can be used as an attempt to solve, the dying ReLU problem, which is the case where the ReLU neurons are disabled and there is only 0 output for any input [122]. Hence, by also employing this activation function, it was aimed to compare the prediction performances of conventional TCNs with the LR-TCN.

Two additional preprocessing processes, normalization, and SMA, were employed on the input data for the conducted regression experiments. Normalization has been carried out to transform the data so that the scale is similar for each feature [98]. This process improves the performance and training stability of the neural network models. First of all, each data type was normalized to be within the range of zero and one, and all the remaining calculations were performed on the normalized data. The equation of the normalization used in this study for x_i, a random parameter A of the data set is given in Equation 4.3.

$$Normalized\, x_i = \frac{x_i - A_{minimum}}{A_{maximum} - A_{minimum}} \tag{4.3}$$

The second method used for data preprocessing, SMA, is a calculation used in statistics to analyze data points by averaging a series of different subsets of the complete data set [123]. The first element of the SMA is derived by averaging the first subset of the sequence of numbers, as shown in equation 4.4. Then the moving average calculation continues throughout the series by excluding the first number of the series and including the next value in the subset. In the experiments, for all DNN methods, both normalized sensor data and the data obtained with SMA in addition to the sensor data are given as input to the networks, as seen in Figure 4.6, and the effect of the SMA on the prediction performance is examined.

$$SMA_i = \frac{x_i + x_{i-1} + .. + x_{i-n+1}}{n} \tag{4.4}$$

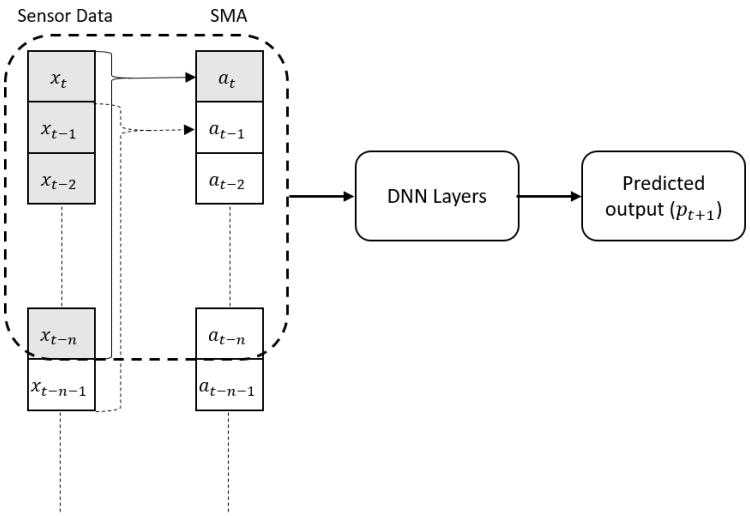

Figure 4.6 Using simple moving average as the DNN inputs.

4.2.1 Regression Experiments on the In-flight Positional and Energy Use Data Set

The first data set utilized for regression model experiments has nine distinct features for flight data. The research conducted on this dataset focused on studying battery power consumption prediction, aligning with the dataset's intended use in the literature. Four distinct input types were employed in this experimental setup to investigate the optimal performance of sensor data prediction for battery power consumption. These input types are outlined below, and the data shapes for these input types are given in Figure 4.7 in detail.

1. Eight types of flight parameters, wind speed, vertical speed, airspeed-x, airspeed-y, wind angle, angle of attack, air density, and payload, were selected as DNN inputs, and instantaneous battery power consumption data of the UAV was determined as output.
2. Battery power consumption data was given as input to DNN, and the next battery power consumption data output was predicted.
3. Eight flight parameters and the SMA values obtained by using the 10-time-step rolling window were given as input to the DNNs, and the battery power consumption was predicted as output.
4. Battery power consumption data and the SMA values obtained by using the 10 time-step rolling window of these sensor data were used as DNN input data and battery power consumption data was tried to be predicted.

For all four different input cases, using the last 10 time-step data as the history window, instantaneous battery power consumption was predicted. At the first tests, flight data were split into two, where 105 of it was used for training and the remaining was used for testing. 20% of the training flights were used for validation. The rest of the data set was used for model performance tests. In addition to the TCN and LR-TCN networks, different input data types were also used with other different DNN methods, LSTM and GRU, and the performance of the proposed method was compared with these methods. Furthermore, to compare the results in more detail, each method was trained using both simple (1-layer) and stacked (3-layer) architectures. For each model training, Adam is used as the optimizer, and mean squared error is used as the loss function, to train all models. The models with the lowest training loss were preserved to be utilized in the testing procedure after each DNN model had been trained.

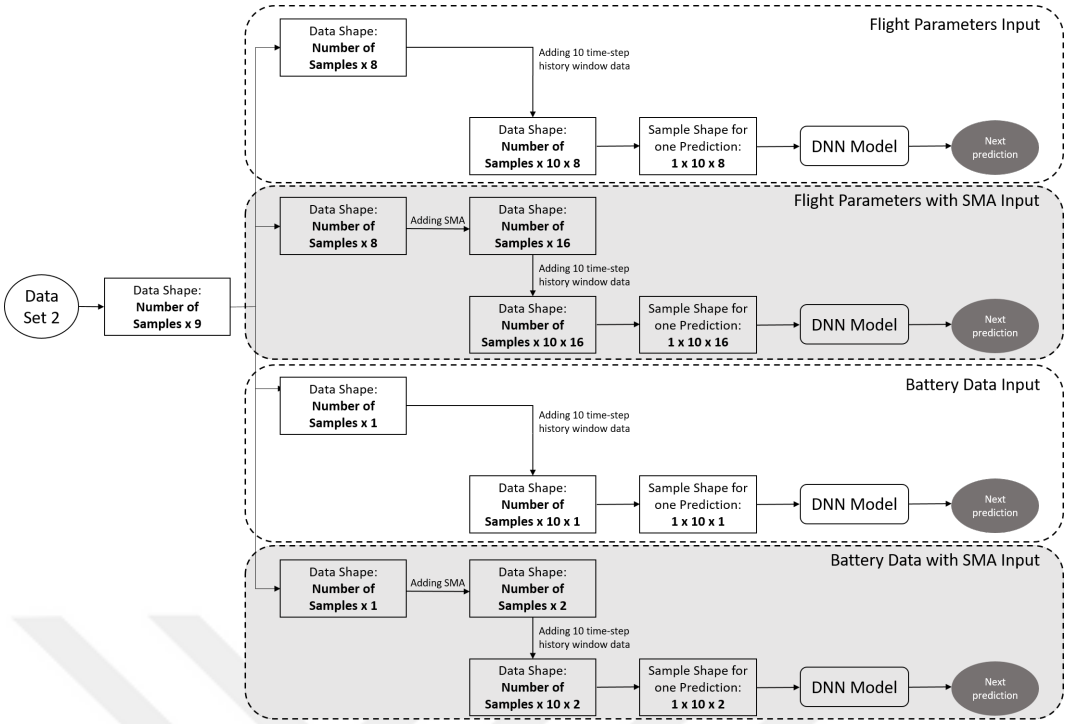

Figure 4.7 Block diagram of the experimental setup for the in-flight positional and energy use data set (Data set 2 refers to the in-flight positional and energy use data set.).

Table 4.3 Test results of four simple DNN methods using four different input types (Best results are shown in bold.).

Input Type	DNN Type	RMSE	MAE	MAPE	R^2
Flight Parameters	Simple LSTM	0.0696	0.0490	0.2240	0.3041
	Simple GRU	0.0708	0.0501	0.2236	0.2805
	Simple TCN	0.0722	0.0506	0.2296	0.2522
	Simple LR-TCN	0.0647	0.0450	0.2166	0.3986
Battery Sensor	Simple LSTM	0.0519	0.0361	0.1388	0.6133
	Simple GRU	**0.0489**	**0.0341**	**0.0756**	**0.6573**
	Simple TCN	0.0522	0.0378	0.1029	0.6083
	Simple LR-TCN	0.0504	0.0358	0.0917	0.6348
Flight Parameters with SMA	Simple LSTM	0.0688	0.0486	0.2144	0.3272
	Simple GRU	0.0699	0.0500	0.2115	0.3052
	Simple TCN	0.0697	0.0497	0.2199	0.3094
	Simple LR-TCN	0.0663	0.0463	0.2172	0.3743
Battery Sensor with SMA	Simple LSTM	0.0492	0.0345	0.0845	0.6557
	Simple GRU	0.0503	0.0347	0.1196	0.6405
	Simple TCN	0.0496	0.0347	0.0891	0.6498
	Simple LR-TCN	0.0496	0.0351	0.0884	0.6497

When the results given in Table 4.3 and Table 4.4 were compared, it was seen that increasing the number of layers did not make a significant contribution to the model prediction performance. Furthermore, it has been observed that model performance decreases in stacked structures when the battery sensor is used together with SMA as input. Therefore, stacked DNNs were excluded from further experiments because

Table 4.4 Test results of four stacked DNN methods using four different input types (Best results are shown in bold.).

Input Type	DNN Type	RMSE	MAE	MAPE	R^2
Flight Parameters	Stacked LSTM	0.0674	0.0473	0.2152	0.3476
	Stacked GRU	0.0670	0.0479	0.2139	0.3559
	Stacked TCN	0.0657	0.0463	0.2170	0.3800
	Stacked LR-TCN	0.0595	0.0402	0.1989	0.4912
Battery Sensor	Stacked LSTM	0.0493	0.0341	0.1198	0.6508
	Stacked GRU	0.0486	**0.0340**	0.0990	0.6616
	Stacked TCN	**0.0485**	**0.0340**	0.0914	**0.6623**
	Stacked LR-TCN	0.0497	0.0348	0.0932	0.6451
Flight Parameters with SMA	Stacked LSTM	0.0673	0.0479	0.2084	0.3564
	Stacked GRU	0.0675	0.0482	0.2102	0.3526
	Stacked TCN	0.0661	0.0466	0.2079	0.3786
	Stacked LR-TCN	0.0629	0.0442	0.1991	0.4371
Battery Sensor with SMA	Stacked LSTM	0.0507	0.0349	0.1186	0.6342
	Stacked GRU	0.0517	0.0359	0.1087	0.6193
	Stacked TCN	0.0533	0.0381	0.1272	0.5967
	Stacked LR-TCN	0.0507	0.0353	**0.0902**	0.6342

of their elevated computational cost with no discernible improvement in model performance.

When the results given in Table 4.3 are analyzed in more detail, it was observed that the LR-TCN algorithm exhibited a lower RMSE value compared to the traditional DNN models, for three of the four input types. In addition, the LR-TCN model exhibited superior predictive performance in comparison to the other three DNN methods, across all evaluation metrics used. addition to the performance of the LR-TCN method, it is seen that the use of battery sensor data as input gives a higher R^2 of 0.6348 than the use of flight sensor data with the R^2 of 0.0.3986. Furthermore, it was observed that adding the data obtained by SMA for both input types increases the prediction test performance by at least 0.0143 for the RMSE metric.

However, since no definitive result could be obtained about the LR-TCN DNN model performance in these initial tests, additional experiments were performed by changing the training and test set ratio to 70:30 and these results were also examined. For these experiments, TCN and LR-TCN models were also compared with LSTM, GRU and CNN methods. The test results of the simulations performed for the given DNN models using the drone flight sensor data set are given in Table 4.5. Upon analysis of the simulation test results, it is seen that the use of battery sensor data as input gives the lowest RMSE value of 0.6348 than the use of 0.0473 when predicted using the TCN method together with the SMA algorithm. Furthermore, the TCN-SMA method demonstrated better predictive performance compared to the other DNN methods, for all employed evaluation metrics.

Table 4.5 Test results of four DNN methods using four different input types for the drone flight sensor data set (Best results are shown in bold.).

Input Type	DNN Type	RMSE	MAE	MAPE	R^2
Flight Parameters	LSTM	0.0613	0.0402	0.2377	0.4477
	GRU	0.0611	0.0397	0.2402	0.4510
	CNN	0.0624	0.0414	0.2408	0.4266
	TCN	0.0806	0.0550	0.2874	0.0457
	LR-TCN	0.0717	0.0491	0.2647	0.2447
Battery Sensor	LSTM	0.0485	0.0340	0.0951	0.6545
	GRU	0.0485	0.0340	0.0945	0.6546
	CNN	0.0502	0.0343	0.1491	0.6289
	TCN	0.0485	0.0336	0.1180	0.6541
	LR-TCN	0.0718	0.0497	0.2667	0.2412
Flight Sensor with SMA	LSTM	0.0612	0.0402	0.2371	0.4487
	GRU	0.0613	0.0400	0.2399	0.4481
	CNN	0.0610	0.0410	0.2380	0.4520
	TCN	0.0771	0.0543	0.2620	0.1248
	LR-TCN	0.0487	0.0339	0.1182	0.6507
Battery Sensor with SMA	LSTM	0.0481	0.0337	0.0972	0.6593
	GRU	0.0481	0.0337	0.0967	0.6602
	CNN	0.0477	0.0333	0.1003	0.6654
	TCN	**0.0473**	**0.0329**	**0.0762**	**0.6710**
	LR-TCN	0.0478	0.0334	0.0870	0.6638

The drone flight sensor data set battery power consumption prediction experiments yielded the findings that more successful prediction results can be obtained if the measurements of the battery sensor are used as input data rather than flight sensor data. These experiments also showed that using one sensor data together with their SMA values can improve the prediction performance. Hence, in the subsequent experiments, only one sensor data was employed for that specific sensor data prediction. Lastly, these experiments demonstrated the success of the TCN-SMA method on sensor data prediction with the best RMSE value of 0.0473.

4.2.2 Regression Experiments on the Collected UAV Anomaly Data Set

Following the initial experiments which demonstrated the success of the TCN-SMA method on sensor data prediction, the performance of the proposed method was further evaluated on the data set created during this study, using the last 5 time-step data as the history window. The block diagram and the input data shapes for the experimental setup of the collected UAV anomaly data set are given in Figure 4.8 in detail.

The experimental results on the collected data set (Tables 4.6 and 4.7) show that the

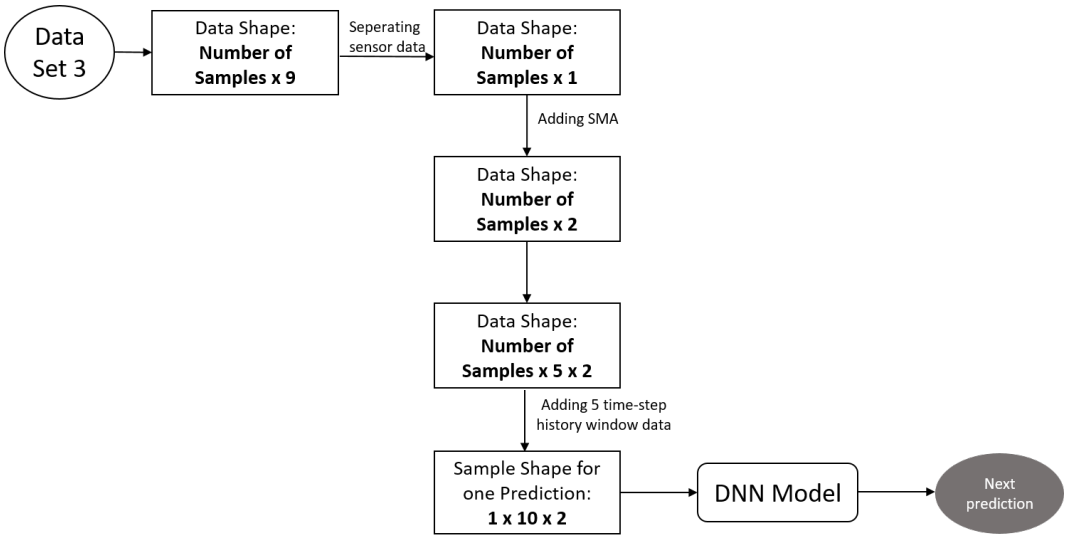

Figure 4.8 Block diagram of the experimental setup for the collected UAV anomaly data set (Data set 3 refers to the collected UAV anomaly data set.).

proposed TCN-SMA method, similar to the other drone flight data set experiments, exhibits superior performance compared to other DNN methods for sensor data prediction. Also compared to the LR-TCN method, the TCN method was found to be more easily deployed to an edge device. Thus, UAV anomaly detection was further investigated utilizing the suggested TCN-SMA method.

Table 4.6 RMSE test results of three DNN methods using collected flight data together with SMA (Best results are shown in bold.).

Sensor Data Type	LSTM	GRU	CNN	TCN	LR-TCN
Accelerometer-x	**0.6149**	0.6420	0.8150	0.6867	0.8012
Accelerometer-y	0.1039	0.1037	0.1005	**0.0892**	0.0953
Accelerometer-z	0.0848	0.0828	0.0705	**0.0600**	0.0659
Gyroscope-x	0.0746	0.0719	0.0708	**0.0671**	0.0704
Gyroscope-y	0.0813	0.0794	0.0743	**0.0689**	0.0738
Gyroscope-z	0.0752	0.0749	0.0739	**0.0719**	0.0730
Magnetometer-x	0.1017	0.1006	0.0982	**0.0891**	0.0953
Magnetometer-y	0.0587	0.0559	0.0541	**0.0524**	0.0533
Magnetometer-z	0.0806	0.0753	0.0735	**0.0681**	0.0739
Average	0.1417	0.1429	0.1590	**0.1393**	0.01558

4.3 Sensor Data Anomaly Detection Using Hybrid Approach

The primary aim of this research is to improve the security and performance of UAV-based applications by developing novel methodologies and warning systems that can accurately identify anomalies in UAV flights. The study's overall concept can be seen in Figure 4.1, presenting a methodology that combines anomaly

Table 4.7 MAE test results of three DNN methods using collected flight data together with SMA (Best results are shown in bold.).

Sensor Data Type	LSTM	GRU	CNN	TCN	LR-TCN
Accelerometer-x	0.0665	0.0633	0.0705	**0.0516**	0.0610
Accelerometer-y	0.0749	0.0748	0.0715	**0.0627**	0.0679
Accelerometer-z	0.0618	0.0611	0.0489	**0.0405**	0.0438
Gyroscope-x	0.0439	0.0435	0.0427	**0.0412**	0.0432
Gyroscope-y	0.0577	0.0550	0.0530	**0.0441**	0.0487
Gyroscope-z	0.0511	0.0506	0.0499	**0.0484**	0.0492
Magnetometer-x	0.0720	0.0696	0.0672	**0.0589**	0.0657
Magnetometer-y	0.0403	0.0381	0.0373	**0.0363**	0.0370
Magnetometer-z	0.0535	0.0516	0.0509	**0.0488**	0.0509
Average	0.0580	0.0564	0.0546	**0.0481**	0.0519

classification, data analysis, and data monitoring methodologies to detect and identify flight anomalies in UAV flights.

This study proposes a hybrid approach that utilizes the AnoSense method for binary classification to determine the presence or absence of anomalies in the flight segments. The TCN-SMA technique is then used to detect anomalous sensor readings inside the flight segments that have been classified as anomalies. The flow chart depicting the hybrid approach proposed for anomaly detection is presented in Figure 4.9.

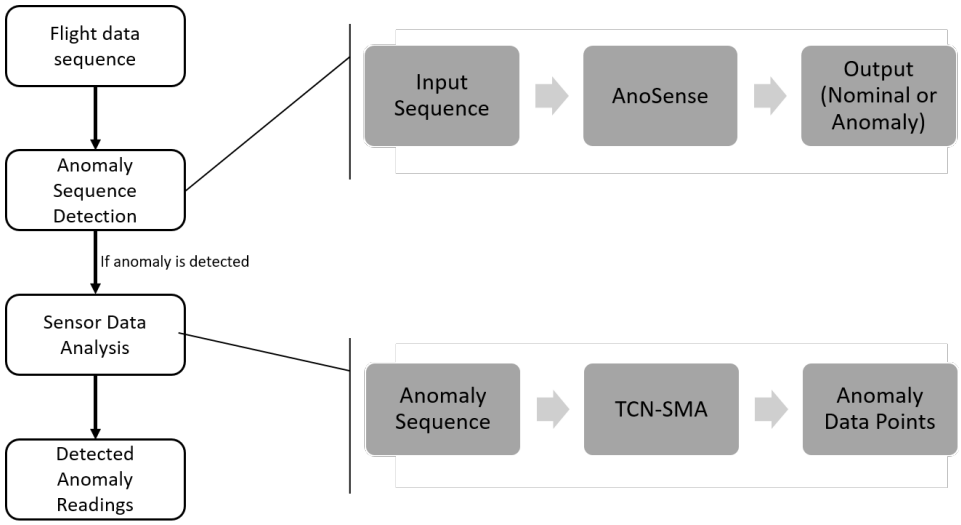

Figure 4.9 The flow chart of the hybrid anomaly detection method.

The hybrid method in this study involves the first processing of flight data sequences using AnoSense models that have been trained specifically for binary classification. This processing step aims to establish whether the sequence belongs to the anomaly or nominal class. In the subsequent phase, if the data sequences pertain to the anomaly category, individual sensor data is analyzed. The analysis of sensor data is

conducted utilizing the trained TCN-SMA prediction models. After that, anomaly identification was carried out by analyzing the prediction error [124]. Prediction error was calculated as the difference between the sensor data values predicted by the model and the sensor data values measured from the sensors on the UAV, and differences greater than the threshold error value determined were marked as anomalies. The threshold value for each sensor was determined by using the Mean Absolute Error (MAE) achieved during individual sensor prediction model training. The steps to perform anomaly detection are detailed in Fig. 4.10.

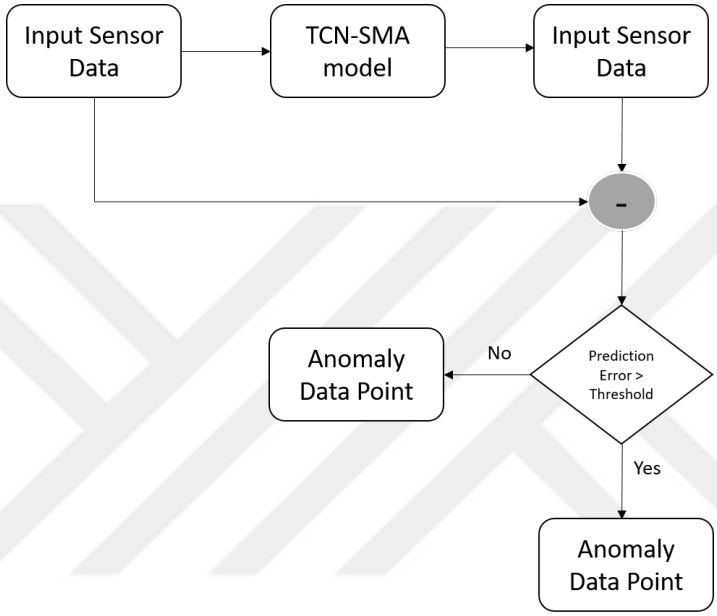

Figure 4.10 Flow chart illustrating the proposed method for anomaly detection.

Figures 4.11-4.12 present data sequences that include data on nominal flights and anomaly flights with the indicated anomaly points for the accelerometer data respectively. Additionally, data sequences depicted in Figures 4.13-4.14 include data on nominal flights and anomaly flights with the indicated anomaly points for the gyroscope data respectively. And finally Figures 4.15-4.16 present data sequences that include data on nominal flights and anomaly flights with the indicated anomaly points for the magnetometer data respectively.

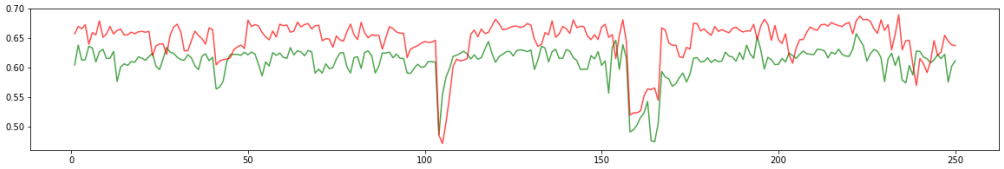

Figure 4.11 Nominal flight accelerometer data with the actual readings and predicted values.

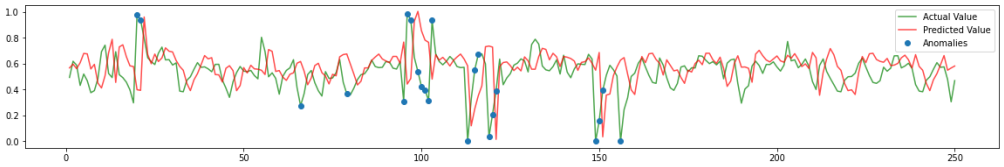

Figure 4.12 Anomaly flight accelerometer data with the actual readings and predicted values including detected anomaly points.

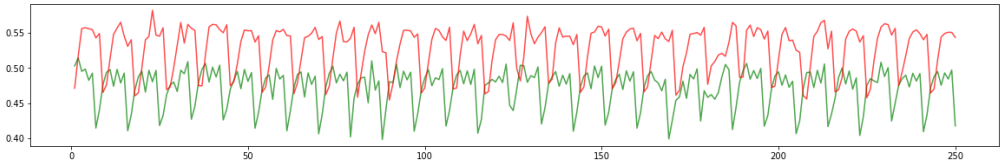

Figure 4.13 Nominal flight gyroscope data with the actual readings and predicted values.

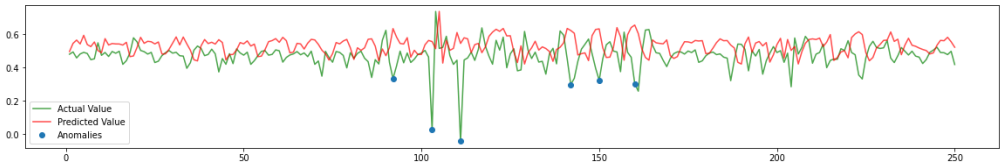

Figure 4.14 Anomaly flight gyroscope data with the actual readings and predicted values including detected anomaly points.

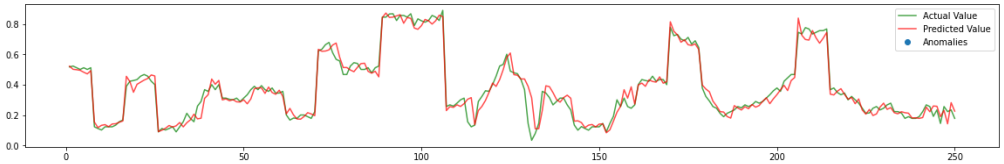

Figure 4.15 Nominal flight magnetometer data with the actual readings and predicted values.

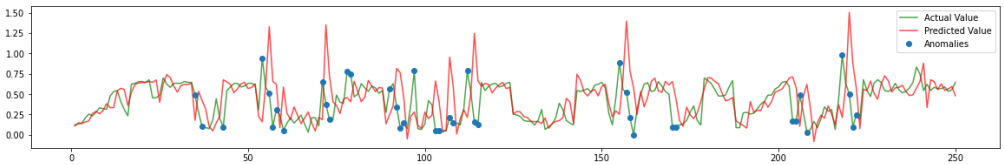

Figure 4.16 Anomaly flight magnetometer data with the actual readings and predicted values including detected anomaly points.

5
SENSOR DATA ANOMALY DETECTION ON EDGE

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 study aimed to perform anomaly detection on the sensor data from UAV, and edge operation was examined since it is essential to quickly address problems on the control card when anomalies are discovered. For this purpose, Espressif ESP32-DevKitC [125] Microcontroller Unit (MCU) and Raspberry Pi 3 Model B [126] development were selected for experimental studies and the operation of the proposed methods on these resource-limited and small-sized control boards was evaluated.

5.1 Evaluating Model Performance and Functionality on the ESP32

ESP32 is a low-cost, low-power System-on-Chip (SoC) microcontroller with integrated Wi-Fi and Bluetooth. The ESP32 is highly favored by a wide range of users because of its notable attributes, including internet connectivity, low power consumption, and functionality. It is used in many different application areas, from Internet of Things (IoT) applications to smart home systems. Furthermore, in recent years, it has also started to be used for machine learning applications utilizing optimized DNN models.

To evaluate the device's performance, in first experiments, DNN models were trained in a Python environment for the regression problem of sensor data prediction. The models was then optimized for compatibility with the chosen

ESP32 MCU. In order to achieve this objective, experiments were conducted utilizing a portion of the UAV dataset collected during this study. The data used was divided into training and test sets, with a ratio of 70:30. In these initial experiments, LSTM, GRU and MLP methods were used.

For the model deployment to the hardware, first of all, all trained DNN models were optimized for on-device machine learning and model performances in were compared with original DNN models. The results for these evaluations are given in Tables 5.1, and 5.2 respectively for evaluation metrics RMSE and R^2.

Table 5.1 RMSE test results comparison for three DNN models before and after optimization (Best results are shown in bold.).

Sensor Data Type	Before Optimization			After Optimization		
	LSTM	GRU	MLP	LSTM	GRU	MLP
Accelerometer-x	0.0535	0.0481	**0.0474**	0.0514	0.0481	**0.0473**
Accelerometery	**0.0629**	**0.0629**	0.0631	0.0629	0.0607	**0.0604**
Accelerometer-z	0.0942	**0.0909**	0.0915	0.0942	**0.0902**	0.0910
Magnetometer-x	0.0119	0.0078	**0.0074**	0.0076	**0.0073**	0.0074
Magnetometer-y	0.0095	**0.0077**	0.0078	0.0094	**0.0076**	0.0080
Magnetometer-z	0.0229	0.0215	**0.0212**	0.0229	0.0214	**0.0212**

Table 5.2 R^2 test results comparison for three DNN models before and after optimization (Best results are shown in bold.).

Sensor Data Type	Before Optimization			After Optimization		
	LSTM	GRU	MLP	LSTM	GRU	MLP
Accelerometer-x	0.7788	0.0481	**0.8262**	0.7955	0.8214	**0.8268**
Accelerometery	**0.8288**	**0.0629**	0.8277	0.8288	0.8409	**0.8423**
Accelerometer-z	0.7586	**0.0909**	0.7725	0.7586	**0.7786**	0.7747
Magnetometer-x	0.9841	0.0078	**0.9938**	0.9936	**0.9941**	0.9938
Magnetometer-y	0.9960	**0.0077**	**0.9972**	0.9960	**0.9974**	0.9971
Magnetometer-z	0.9799	0.0215	**0.9827**	0.9799	0.9824	**0.9827**

After the simulations carried out in the Python environment, the MLP model was chosen primarily for model deployement tests, for its lightweight network architecture and computational costs. The MLP model trained with magnetometer-y data, has been deployed on the development board. After the deployment, prediction tests were performed on the board and the results were compared with the models before and after optimization (Table 5.3).

The LSTM model trained with magnetometer-y data, has also been deployed on the development board and prediction tests results performed on the board were compared with the models before and after optimization (Table 5.4).

	RMSE	R^2-score
Before Optimization	0.0078	0.9972
After Optimisation	0.0080	0.9971
Deployed Model	0.0111	0.9944

Table 5.3 MLP model test results comparison for magnetometer-y data.

	RMSE	R^2-score
Before Optimization	0.0095	0.9960
After Optimisation	0.0094	0.9960
Deployed Model	0.0108	0.9947

Table 5.4 LSTM model test results comparison for magnetometer-y data.

The experiment's findings indicate that both the trained MLP and LSTM models are also capable of making sensor predictions with a slight increase in error on the ESP32 SoC. Nevertheless, the excessive processing time required for sensor data on the board, even with the utilization of these relatively lightweight models, indicates that employing this card for the suggested more complex architectures is not suitable for a real-time solution. For this reason, it was decided to continue testing other proposed models and methods with Raspberry Pi, which is a development board with relatively higher memory and processing capacity, while still being a low power and low cost solution.

5.2 Evaluating Model Performance and Functionality on the Raspberry Pi

The Raspberry Pi 3 Model B [126] development board used in the comparison study has a quad-core 1.2 GHz ARM Cortex-A53 processor and includes 1 GB RAM internal memory. 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.

To validate the findings on Raspberry Pi, nine types of flight data that were obtained during this study were utilized, and hybrid method validation was carried out. At this point, a total of 20 time-step flight segments with a train and test ratio of 70:30 were employed. The DNN models was 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. At this stage pruning, quantization and model compression approaches were included during the training process to optimize the performance of the models in edge applications. Pruning is a technique employed in the training of neural networks to improve efficiency or reduce the size of the model by deliberately removing certain features, such as connections or neurons, where as quantization is a technique used to reduce the amount of intricacy in the weights and activations inside a neural network. In this study, the values were represented using 8-bit integer precision for quantization, instead of using full precision floating point numbers and the Pruning Preserving Quantization Aware Training (PQAT) was used for the integration of both pruning and quantization techniques throughout the model training.

Firstly, the model sizes and the binary classification performance of the AnoSense models were analyzed for the base, PQAT, and compressed models and the results were given in Table 5.5. Tables 5.6 and 5.7 provides information regarding the anomaly and nominal classification accuracy and F-score performance of 2-second flight segments of these models on the Python environment.

Table 5.5 Effect of pruning on model size (kilobayt) for all 5-fold cross-validation binary classification models.

Data Subset	Base Model	PQAT Model	Compressed Base Model	Compressed PQAT Model
Subset 1	511.79	393.22	166.99	45.72
Subset 2	506.78	448.09	166.93	45.59
Subset 3	511.12	407.53	166.96	46.12
Subset 4	509.17	402.57	166.98	45.57
Subset 5	511.05	384.38	166.99	46.17

Table 5.6 Accuracy test results (%) for binary classification of the flight sequences using collected UAV anomaly data set on Raspberry Pi.

Data Subset	Base Model	PQAT Model	Compressed Base Model	Compressed PQAT Model
Subset 1	97.72	94.06	97.72	94.14
Subset 2	97.11	91.78	97.11	91.70
Subset 3	97.03	93.00	97.03	92.85
Subset 4	97.49	94.22	97.49	94.14
Subset 5	98.02	95.13	98.02	94.98

Table 5.7 F-score test results (%) for binary classification of the flight sequences using collected UAV anomaly data set on Raspberry Pi.

Data Subset	Base Model	PQAT Model	Compressed Base Model	Compressed PQAT Model
Subset 1	97.32	92.66	97.32	89.96
Subset 2	96.65	89.53	96.65	89.43
Subset 3	96.51	91.27	96.51	91.06
Subset 4	97.03	92.90	97.03	92.80
Subset 5	97.68	94.07	97.68	93.88

When Table 5.5 is examined, it is seen that the compressed PQAT model size for each subset decreases between 90.97% and 91.07% compared to the base models. However, when the anomaly classification performances in Tables 5.6 and 5.7 are examined, it is seen that there is a performance decrease only between 3.10%-5.57% for the accuracy metric and 3.89%-7.56% for the F-score metric. Here, it can be seen that as a result of model optimization, there is a significant reduction in both model size and computational load, with a minimal decrease in performance.

Furthermore, Table 5.8 presents the accuracy score comparison for the binary classification conducted on the Python environment and Raspberry Pi board using the compressed PQAT model. Additionally, Tables 5.9 and 5.10 provides information regarding average accuracy, precision, recall and F-score metrics and the confusion matrix of the test results, which may be used for further analysis.

Table 5.8 Accuracy scores (%) comparison for the binary classification on the Python environment and Raspberry Pi board using the compressed PQAT model.

Data Subset	Python environment	Raspberry Pi Board
Subset 1	94.14	94.06
Subset 2	91.70	91.70
Subset 3	92.85	92.92
Subset 4	94.14	94.14
Subset 5	94.98	94.98

Table 5.9 5-fold cross-validation average test results for binary classification of the flight sequences using collected UAV anomaly data set on Raspberry Pi.

Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
93.56	99.68	85.44	91.99

Table 5.10 Confusion matrix for binary classification of the flight sequences using collected UAV anomaly data set on Raspberry Pi.

	Actual Positive (1)	Actual Negative (0)
Predicted Positive (1)	736	8
Predicted Negative (0)	71	499

After classification model, the model sizes and performances of the TCN-SMA regression model were also analyzed for the base, and compressed PQAT, and these results were given in Table 5.11. It is crucial to emphasize that, at this time, the model performance tests were exclusively carried out using the nominal flight data for the purpose of analyzing prediction performance. Tests of sensor data prediction were also run on Raspberry Pi, and the RMSE and MAE values of those tests were presented in Tables 5.12 and 5.13.

Table 5.11 Effect of pruning on model size (kilobayt) for TCN-SMA regression models.

Sensor Data Type	Base Model	Compressed PQAT Model
Accelometer-x	264.51	135.42
Accelometer-y	247.44	135.34
Accelometer-z	256.20	135.33
Gyroscope-x	226.81	135.27
Gyroscope-y	226.46	135.25
Gyroscope-z	229.60	135.27
Magnetometer-x	257.93	135.35
Magnetometer-y	247.90	135.35
Magnetometer-z	242.55	135.32

Table 5.12 Comparison of the RMSE values for sensor data prediction of the flight sequences using collected UAV anomaly data set on the Python environment and Raspberry Pi board.

Sensor Data Type	Base Model	Compressed Model on Python Environment	Compressed Model on Raspberry Pi
Accelometer-x	0.6867	0.6280	0.6280
Accelometer-y	0.0892	0.0918	0.0918
Accelometer-z	0.0600	0.0632	0.0632
Gyroscope-x	0.0671	0.0673	0.0673
Gyroscope-y	0.0689	0.0697	0.0697
Gyroscope-z	0.0719	0.0725	0.0725
Magnetometer-x	0.0891	0.0894	0.0894
Magnetometer-y	0.0524	0.0535	0.0535
Magnetometer-z	0.0681	0.0705	0.0705

Table 5.13 Comparison of the MAE values for sensor data prediction of the flight sequences using collected UAV anomaly data set on the Python environment and Raspberry Pi board.

Sensor Data Type	Base Model	Compressed Model on Python Environment	Compressed Model on Raspberry Pi
Accelometer-x	0.0483	0.0418	0.0418
Accelometer-y	0.0630	0.0653	0.0653
Accelometer-z	0.0388	0.0428	0.0428
Gyroscope-x	0.0413	0.0430	0.0430
Gyroscope-y	0.0446	0.0468	0.0468
Gyroscope-z	0.0486	0.0494	0.0494
Magnetometer-x	0.0614	0.0619	0.0619
Magnetometer-y	0.0362	0.0378	0.0378
Magnetometer-z	0.0483	0.0499	0.0499

When Table 5.11 is examined, it is seen that the compressed PQAT model size for each subset decreases nearly half compared to the base models. However, when the regression performances in Tables 5.12 and 5.13 are examined, after model optimization, similar to binary classification, there is only a minimal decrease in

performance.

Following the prediction tests using just nominal flight data, additional tests were run on the Raspberry Pi with data that included both nominal and anomaly data, and the change in the evaluation metrics were examined in Tables 5.14 and 5.15.

Table 5.14 Comparison of the RMSE values for sensor data prediction of the flight sequences using collected UAV anomaly data set on the Python environment and Raspberry Pi board.

Sensor Data Type	With Nominal Data	With Added Anomaly Data
Accelometer-x	0.6280	0.4266
Accelometer-y	0.0918	0.1009
Accelometer-z	0.0632	0.1450
Gyroscope-x	0.0673	0.0694
Gyroscope-y	0.0697	0.1226
Gyroscope-z	0.0725	0.0628
Magnetometer-x	0.0894	0.2603
Magnetometer-y	0.0535	0.1660
Magnetometer-z	0.0705	0.2538

Table 5.15 Comparison of the MAE values for sensor data prediction of the flight sequences using collected UAV anomaly data set on the Python environment and Raspberry Pi board.

Sensor Data Type	With Nominal Data	With Added Anomaly Data
Accelometer-x	0.0418	0.3156
Accelometer-y	0.0653	0.0614
Accelometer-z	0.0428	0.0806
Gyroscope-x	0.0430	0.0404
Gyroscope-y	0.0468	0.1128
Gyroscope-z	0.0494	0.0423
Magnetometer-x	0.0619	0.1526
Magnetometer-y	0.0378	0.0923
Magnetometer-z	0.0499	0.1491

Based on the acquired results from Tables 5.14 and 5.15, it is evident that the RMSE values are greater in the tests conducted using a combination of nominal and anomaly data, as opposed to the regression tests conducted only with nominal flight data. The observed rise in this error metric during anomaly conditions facilitates the identification of anomaly points through the computation of prediction error.

Following the selection of the most optimal models based on individual model testing, a thorough assessment of the hybrid anomaly detection algorithm for Raspberry Pi was carried out, demonstrating the effective implementation of the proposed method in its entirety (Figure 5.1).

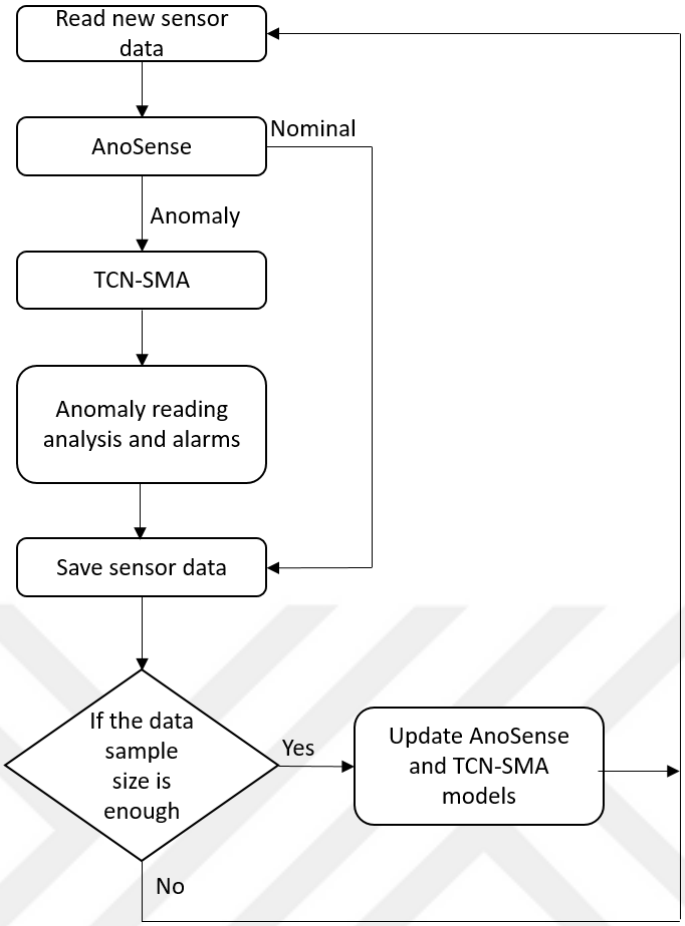

Figure 5.1 Overall framework of the hybrid algorithm employing the model update feature executed on the board.

The proposed hybrid algorithm is capable of processing in the form of a 2-second time series sequence and conduct anomaly detection and sensor analysis on the received data sequence. Finally, an online learning aspect was incorporated into the system to enhance the model performance on edge. Throughout this phase, the process of fine-tuning is executed on all DNN models employed in the hybrid anomaly detection approach. This fine-tuning is conducted utilizing sensor data that has been acquired at regular intervals throughout flights. It is important to note that the recorded data is subsequently erased after this procedure to prevent any memory issues on the Raspberry Pi board. The resultant current model was stored in the board memory and then employed for further examination. The overall framework of the hybrid algorithm employing the model update feature executed on the board is also presented in Figure 5.1 for enhanced comprehension.

6
CONCLUSION

This thesis aims to provide a novel approach that is appropriate for the utilization of onboard technologies to identify flight anomalies in UAVs. The present work first involves an examination of sensor data collected from UAVs, followed by the development of a classification methodology aimed at detecting abnormalities during flights. Subsequently, a regression-based method was introduced to detect abnormalities in individual UAV sensors. Ultimately, a hybrid approach encompassing both classification and sensor analysis using regression was proposed to detect anomalies for edge applications on the Raspberry Pi development board.

The study used three different flight sensor data sets to evaluate the performance of the proposed methods. First, the curated 4-class anomaly detection data set was utilized to evaluate the accuracy of classification methods. The study analyzed the performance metrics of the AnoSense models using 5-fold cross-validation. The results of these tests demonstrate that AnoSense models outperform LSTM, GRU, CNN, and TCN models in the Python environment, by achieving an accuracy of 97.47%. In addition, the AnoSense model was again the best performing method in all metrics with an average of 98.77% precision, 95.36% recall and 97.03% F-score in this data set tests.To examine the performance of the proposed AnoSense method on the UAV data set collected in the study, the method was again compared with LSTM, GRU, CNN and TCN methods. The 5-fold cross-validation average results of these tests demonstrate that AnoSense models, once again, outperform LSTM, GRU, CNN, and TCN models in the Python environment, by achieving an accuracy of 96.65%. Furthermore, the AnoSense model was also the best performing method in all metrics with an average score of 98.12% precision, 98.08% recall and 97.62% F-score values.

The initial step in conducting the regression approach trials was utilizing in-flight positional and energy use data set. The final results obtained in the experiments carried out using this data set show that the TCN-SMA method proposed for sensor data prediction performs better compared to other methods such as LSTM, GRU,

CNN and LR-TCN, with the lowest RMSE value of 0.0473. Similarly, it gave the best results for the all other examined metrics, MAE, MAPE and R^2-score, with 0.0329, 0.0762, respectively. with values of 0.6710. Furthermore, it yielded the best results for the other evaluation metrics, MAE, MAPE, and R^2-score, with values of 0.0329, 0.0762, and 0.6710, respectively.

Moreover, within the scope of the study, data acquired during UAV flights was collected and labeled. Afterward, this data set was also used for the evaluation of classification and regression methods, in addition to serving as a means to evaluate the proposed hybrid approach that integrates the classification method for identifying flight anomalies and the regression method for sensor analysis.

The next phase of the study involved an evaluation of the performance of the proposed approach on a power-constrained and low-memory device. During this stage, a Raspberry Pi development board was employed to integrate the proposed method for edge application, and an online learning methodology enabling ongoing model training on the edge was developed and tested. The results obtained from these tests showed that an average of 93.56% anomaly classification performance was achieved by using models optimized for edge applications.

The contributions of this thesis study can be also summarized as follows.

- A new hybrid approach that utilizes both classification an regression methods was proposed for UAV flight sensor data anomaly detection.
- The proposed flight sensor anomaly detection method was optimized for edge devices, enabling its application on small sized drones.
- A periodic DNN model update function was implemented to facilitate ongoing learning in the system.
- A UAV sensor data set containing various anomaly types was created and utilized in the experiments, in order to examine the anomaly detection performance on real data.
- The anomaly detection algorithm was executed and evaluated on a Raspberry Pi board for an exemplary application, in order to showcase the feasibility of the proposed hybrid DNN-based anomaly detection algorithm on small sized drones and control boards.

Upon examination of the study's findings, it was observed that the suggested classification and regression methods exhibited superior performance in anomaly

detection and sensor data prediction error compared to standard DNN approaches. Moreover, the hybrid approach that has been proposed effectively integrates anomaly detection, analysis of individual instantaneous sensor data, and marking of anomalous data points. Finally, implementing the suggested approach on power-constrained and low-memory devices demonstrates its viability as a practical solution for real-time anomaly detection. Hence, the proposed hybrid anomaly detection approach can be applied to UAV missions that have limited resources, where the ability to identify anomalies onboard and promptly provide corrective control orders is of utmost importance.

In conclusion, this thesis demonstrates that DNN-based solutions solutions can be implemented on control cards with power and memory constraints on the autonomous small drones for detecting anomalies in flight sensor data. These optimized methods will allow for the rapid detection of anomalies in flight sensors while minimizing the computational costs, and will enable their application in small sized drone control boards for swift and efficient control responses in the future.

REFERENCES

[1] G. Wang *et al.*, "Field evaluation of spray drift and environmental impact using an agricultural unmanned aerial vehicle (uav) sprayer," *Science of the Total Environment*, vol. 737, p. 139 793, 2020.

[2] X. Li, D. K. Giles, F. J. Niederholzer, J. T. Andaloro, E. B. Lang, L. J. Watson, "Evaluation of an unmanned aerial vehicle as a new method of pesticide application for almond crop protection," *Pest management science*, vol. 77, no. 1, pp. 527–537, 2021.

[3] Ł. Jełowicki, K. Sosnowicz, W. Ostrowski, K. Osińska-Skotak, K. Bakuła, "Evaluation of rapeseed winter crop damage using uav-based multispectral imagery," *Remote Sensing*, vol. 12, no. 16, p. 2618, 2020.

[4] T. Moranduzzo, F. Melgani, "Monitoring structural damages in big industrial plants with uav images," in *2014 IEEE Geoscience and Remote Sensing Symposium*, IEEE, 2014, pp. 4950–4953.

[5] G. Grenzdörffer, F. Niemeyer, "Uav based brdf-measurements of agricultural surfaces with pfiffikus," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 38, pp. 229–234, 2011.

[6] D. Anthony, S. Elbaum, A. Lorenz, C. Detweiler, "On crop height estimation with uavs," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 4805–4812.

[7] H. Escalante, S. Rodríguez-Sánchez, M. Jiménez-Lizárraga, A. Morales-Reyes, J. De La Calleja, R. Vazquez, "Barley yield and fertilization analysis from uav imagery: A deep learning approach," *International journal of remote sensing*, vol. 40, no. 7, pp. 2493–2516, 2019.

[8] C. Song *et al.*, "Test and comprehensive evaluation for the performance of uav-based fertilizer spreaders," *IEEE Access*, vol. 8, pp. 202 153–202 163, 2020.

[9] J. Li *et al.*, "Distribution of canopy wind field produced by rotor unmanned aerial vehicle pollination operation," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 31, no. 3, pp. 77–86, 2015.

[10] A. Liu *et al.*, "Effects of supplementary pollination by single-rotor agricultural unmanned aerial vehicle in hybrid rice seed production," *Agricultural Science & Technology*, vol. 18, no. 3, pp. 543–552, 2017.

[11] M. Eichleay, E. Evens, K. Stankevitz, C. Parker, "Using the unmanned aerial vehicle delivery decision tool to consider transporting medical supplies via drone," *Global Health: Science and Practice*, vol. 7, no. 4, pp. 500–506, 2019.

[12] C. A. Thiels, J. M. Aho, S. P. Zietlow, D. H. Jenkins, "Use of unmanned aerial vehicles for medical product transport," *Air medical journal*, vol. 34, no. 2, pp. 104–108, 2015.

[13] K. Yakushiji, H. Fujita, M. Murata, N. Hiroi, Y. Hamabe, F. Yakushiji, "Short-range transportation using unmanned aerial vehicles (uavs) during disasters in japan," *Drones*, vol. 4, no. 4, p. 68, 2020.

[14] Y. Huang, H. Han, B. Zhang, X. Su, Z. Gong, "Supply distribution center planning in uav-based logistics networks for post-disaster supply delivery," in *2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM)*, IEEE, 2021, pp. 1–6.

[15] A. Nedjati, B. Vizvari, G. Izbirak, "Post-earthquake response by small uav helicopters," *Natural Hazards*, vol. 80, pp. 1669–1688, 2016.

[16] W. Jin, J. Yang, Y. Fang, W. Feng, "Research on application and deployment of uav in emergency response," in *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, IEEE, 2020, pp. 277–280.

[17] D. Wojtyra, K. Wacławik, K. A. Krenc, M. Długoń, "Concept for the construction and application of a counter-uav defence system," *Problemy Mechatroniki. Uzbrojenie, lotnictwo, inżynieria bezpieczeństwa*, vol. 12, no. 1, 2021.

[18] O. Ozkan, M. Kaya, "Uav routing with genetic algorithm based matheuristic for border security missions," *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, vol. 11, no. 2, pp. 128–138, 2021.

[19] W. Zhang, L. Zhang, B. Yang, H. Gu, D. Wang, K. Yang, "The development of counter-unmanned aerial vehicle technologies," in *Global Intelligence Industry Conference (GIIC 2018)*, SPIE, vol. 10835, 2018, pp. 370–373.

[20] X. Wang, H. Zhu, D. Zhang, D. Zhou, X. Wang, "Vision-based detection and tracking of a mobile ground target using a fixed-wing uav," *International Journal of Advanced Robotic Systems*, vol. 11, no. 9, p. 156, 2014.

[21] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, D. Sarazzi, *et al.*, "Uav photogrammetry for mapping and 3d modeling–current status and future perspectives," *International archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 38, no. 1, p. C22, 2011.

[22] M.-L. Cheng, M. Matsuoka, "Extracting three-dimensional (3d) spatial information from sequential oblique unmanned aerial system (uas) imagery for digital surface modeling," *International Journal of Remote Sensing*, vol. 42, no. 5, pp. 1643–1663, 2021.

[23] M. Elloumi, R. Dhaou, B. Escrig, H. Idoudi, L. A. Saidane, "Monitoring road traffic with a uav-based system," in *2018 IEEE wireless communications and networking conference (WCNC)*, IEEE, 2018, pp. 1–6.

[24] H. Huang, A. V. Savkin, C. Huang, "Decentralized autonomous navigation of a uav network for road traffic monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2558–2564, 2021.

[25] D. K. Villa, A. S. Brandao, M. Sarcinelli-Filho, "A survey on load transportation using multirotor uavs," *Journal of Intelligent & Robotic Systems*, vol. 98, pp. 267–296, 2020.

[26] A. Gupta, T. Afrin, E. Scully, N. Yodo, "Advances of uavs toward future transportation: The state-of-the-art, challenges, and opportunities," *Future transportation*, vol. 1, no. 2, pp. 326–350, 2021.

[27] I. Mademlis *et al.*, "High-level multiple-uav cinematography tools for covering outdoor events," *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 627–635, 2019.

[28] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, A. Messina, "Autonomous uav cinematography: A tutorial and a formalized shot-type taxonomy," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–33, 2019.

[29] O. Zerlenga, V. Cirillo, R. Iaderosa, "Once upon a time there were fireworks. the new nocturnal drones light shows," *img journal*, no. 4, pp. 402–425, 2021.

[30] M. Waibel, B. Keays, F. Augugliaro, "Drone shows: Creative potential and best practices," ETH Zurich, Tech. Rep., 2017.

[31] H. V. Dudukcu, M. Taskiran, N. Kahraman, "Uav sensor data applications with deep neural networks: A comprehensive survey," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106 476, 2023.

[32] R. Chalapathy, S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[33] Y. Liu, W. Ding, "A knns based anomaly detection method applied for uav flight data stream," in *2015 Prognostics and System Health Management Conference (PHM)*, IEEE, 2015, pp. 1–8.

[34] Y. He, Y. Peng, S. Wang, D. Liu, P. H. Leong, "A structured sparse subspace learning algorithm for anomaly detection in uav flight data," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 1, pp. 90–100, 2017.

[35] D. Pan, "Hybrid data-driven anomaly detection method to improve uav operating reliability," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, IEEE, 2017, pp. 1–4.

[36] J. Bu *et al.*, "Integrated method for the uav navigation sensor anomaly detection," *IET Radar, Sonar & Navigation*, vol. 11, no. 5, pp. 847–853, 2017.

[37] A. Manukyan, M. A. Olivares-Mendez, H. Voos, M. Geist, "Real time degradation identification of uav using machine learning techniques," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2017, pp. 1223–1230.

[38] F. Pourpanah, B. Zhang, R. Ma, Q. Hao, "Anomaly detection and condition monitoring of uav motors and propellers," in *2018 IEEE SENSORS*, IEEE, 2018, pp. 1–4.

[39] L. Liu, M. Liu, Q. Guo, D. Liu, Y. Peng, "Mems sensor data anomaly detection for the uav flight control subsystem," in *2018 IEEE SENSORS*, IEEE, 2018, pp. 1–4.

[40] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE internet of things journal*, vol. 5, no. 4, pp. 2315–2322, 2017.

[41] Y. He, Y. Peng, S. Wang, D. Liu, "Admost: Uav flight data anomaly detection and mitigation via online subspace tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 4, pp. 1035–1044, 2018.

[42] B. Wang, D. Liu, X. Peng, Z. Wang, "Data-driven anomaly detection of uav based on multimodal regression model," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, IEEE, 2019, pp. 1–6.

[43] C. Titouna, F. Naıt-Abdesselam, H. Moungla, "An online anomaly detection approach for unmanned aerial vehicles," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2020, pp. 469–474.

[44] D. Pan, L. Nie, W. Kang, Z. Song, "Uav anomaly detection using active learning and improved s3vm model," in *2020 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)*, IEEE, 2020, pp. 253–258.

[45] V. Sindhwani, H. Sidahmed, K. Choromanski, B. Jones, "Unsupervised anomaly detection for self-flying delivery drones," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 186–192.

[46] O. Bektash, A. la Cour-Harbo, "Vibration analysis for anomaly detection in unmanned aircraft," in *ANNUAL CONFERENCE OF THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY 2020*, PHM Society, 2020.

[47] Y.-S. Hsiao *et al.*, "Mavfi: An end-to-end fault analysis framework with anomaly detection and recovery for micro aerial vehicles," *arXiv preprint arXiv:2105.12882*, 2021.

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

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

[50] D. Guo, M. Zhong, H. Ji, Y. Liu, R. Yang, "A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors," *Neurocomputing*, vol. 319, pp. 155–163, 2018.

[51] J. Galvan, A. Raja, Y. Li, J. Yuan, "Sensor data-driven uav anomaly detection using deep learning approach," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, IEEE, 2021, pp. 589–594.

[52] M. W. Ahmad, M. U. Akram, R. Ahmad, K. Hameed, A. Hassan, "Intelligent framework for automated failure prediction, detection, and classification of mission critical autonomous flights," *ISA transactions*, 2022.

[53] H. Shin, J. Lee, P. Kim, "Causality-seq2seq model for battery anomaly detection," *International Journal of Aeronautical and Space Sciences*, pp. 1–10, 2022.

[54] Y. Cheng, Y. Xu, H. Zhong, Y. Liu, "Hs-tcn: A semi-supervised hierarchical stacking temporal convolutional network for anomaly detection in iot," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, IEEE, 2019, pp. 1–7.

[55] Y. He, J. Zhao, "Temporal convolutional networks for anomaly detection in time series," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1213, 2019, p. 042 050.

[56] J. You, J. Liang, D. Liu, "An adaptable uav sensor data anomaly detection method based on tcn model transferring," in *2022 Prognostics and Health Management Conference (PHM-2022 London)*, IEEE, 2022, pp. 73–76.

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

[58] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[59] Y. Cao, J. Cao, Z. Zhou, Z. Liu, "Aircraft track anomaly detection based on mod-bi-lstm," *Electronics*, vol. 10, no. 9, p. 1007, 2021.

[60] H. Lee, G. Li, A. Rai, A. Chattopadhyay, "Real-time anomaly detection framework using a support vector regression for the safety monitoring of commercial aircraft," *Advanced Engineering Informatics*, vol. 44, p. 101 071, 2020.

[61] Y. Chen, B. Wang, W. Liu, D. Liu, "On-line and non-invasive anomaly detection system for unmanned aerial vehicle," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, IEEE, 2017, pp. 1–7.

[62] B. Wang, Y. Chen, D. Liu, X. Peng, "An embedded intelligent system for on-line anomaly detection of unmanned aerial vehicle," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 6, pp. 3535–3545, 2018.

[63] W. Chunhui, J. Zhou, W. Yuanhang, Z. Shi, H. Chuangmian, Y. Yunfan, "An anomaly detecting system for power system of four-rotor uav," in *2020 International Symposium on Autonomous Systems (ISAS)*, IEEE, 2020, pp. 109–114.

[64] W. Pensec, D. Espes, C. Dezan, "Smart anomaly detection and monitoring of industry 4.0 by drones," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2022, pp. 705–713.

[65] V. Bell, D. Rengasamy, B. Rothwell, G. P. Figueredo, "Anomaly detection for unmanned aerial vehicle sensor data using a stacked recurrent autoencoder method with dynamic thresholding," *arXiv preprint arXiv:2203.04734*, 2022.

[66] L. K. SHAR, W. MINN, N. B. D. TA, L. JIANG, D. W. K. LIM, W. K. D. LIM, "Dronlomaly: Runtime detection of anomalous drone behaviors via log analysis and deep learning," 2022.

[67] M. Suresh, S. Chandra Swar, S. Shyam, "Autonomous cooperative guidance strategies for unmanned aerial vehicles during on-board emergency," *Journal of Aerospace Information Systems*, pp. 1–12, 2022.

[68] A. Keipour, M. Mousaei, S. Scherer, "Alfa: A dataset for uav fault and anomaly detection," *The International Journal of Robotics Research*, vol. 0, no. 0, pp. 1–6, Oct. 2020. doi: 10.1177/0278364920966642. eprint: https://doi.org/10.1177/0278364920966642. [Online]. Available: https://doi.org/10.1177/0278364920966642.

[69] T. Rodrigues *et al.*, *Data collected with package delivery quadcopter drone*, 2020.

[70] T. A. Rodrigues *et al.*, "In-flight positional and energy use data set of a dji matrice 100 quadcopter for small package delivery," *Scientific Data*, vol. 8, no. 1, pp. 1–8, 2021.

[71] M. Bonetto, P. Korshunov, G. Ramponi, T. Ebrahimi, "Privacy in mini-drone based video surveillance," in *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, IEEE, vol. 4, 2015, pp. 1–6.

[72] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs)," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4992–4997.

[73] P. Zhu, L. Wen, X. Bian, H. Ling, Q. Hu, "Vision meets drones: A challenge," *arXiv preprint arXiv:1804.07437*, 2018.

[74] P. Zhu *et al.*, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3119563.

[75] R. Krajewski, J. Bock, L. Kloeker, L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125. doi: 10.1109/ITSC.2018.8569552.

[76] D. Avola, L. Cinque, G. L. Foresti, N. Martinel, D. Pannone, C. Piciarelli, "A uav video dataset for mosaicking and change detection from low-altitude flights," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 6, pp. 2139–2149, 2018.

[77] I. Sa *et al.*, "Weedmap: A large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming," *Remote Sensing*, vol. 10, no. 9, p. 1423, 2018.

[78] M. T. Chiu *et al.*, "Agriculture-vision: A large aerial image database for agricultural pattern analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2828–2838.

[79] M. Fonder, M. Van Droogenbroeck, "Mid-air: A multi-modal dataset for extremely low altitude drone flights," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019, pp. 0–0.

[80] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, D. Scaramuzza, "Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

[81] G. Cioffi, T. Cieslewski, D. Scaramuzza, "Continuous-time vs. discrete-time vision-based slam: A comparative study," *IEEE Robotics and Automation Letters, (RA-L)*, 2022.

[82] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, S. Karaman, "The blackbird uav dataset," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1346–1364, 2020.

[83] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, S. Karaman, "The blackbird dataset: A large-scale dataset for uav perception in aggressive flight," in *International Symposium on Experimental Robotics*, Springer, 2020, pp. 130–139.

[84] I. Bozcan, E. Kayacan, "Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 8504–8510.

[85] Y. Chang, Y. Cheng, J. Murray, S. Huang, G. Shi, "The hdin dataset: A real-world indoor uav dataset with multi-task labels for visual-based navigation," *Drones*, vol. 6, no. 8, 2022, issn: 2504-446X. doi: 10.3390/drones6080202. [Online]. Available: https://www.mdpi.com/2504-446X/6/8/202.

[86] C. Brommer *et al.*, "Insane: Cross-domain uav data sets with increased number of sensors for developing advanced and novel estimators," *arXiv preprint arXiv:2210.09114*, 2022.

[87] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

[89] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[90] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[91] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[92] K. Cho *et al.*, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[93] S. Albawi, T. A. Mohammed, S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, Ieee, 2017, pp. 1–6.

[94] C. Lea, R. Vidal, A. Reiter, G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *European conference on computer vision*, Springer, 2016, pp. 47–54.

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

[96] D. Bank, N. Koenigstein, R. Giryes, "Autoencoders," *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*, pp. 353–374, 2023.

[97] X. Wang, C. Wang, "Time series data cleaning: A survey," *Ieee Access*, vol. 8, pp. 1866–1881, 2019.

[98] S. Patro, K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[99] K. Kira, L. A. Rendell, *et al.*, "The feature selection problem: Traditional methods and a new algorithm," in *Aaai*, vol. 2, 1992, pp. 129–134.

[100] B. Schölkopf, C. J. Burges, A. J. Smola, *et al.*, *Advances in kernel methods: support vector learning*. MIT press, 1999.

[101] R. G. Pajares, J. M. Benítez, G. S. Palmero, "Feature selection for time series forecasting: A case study," in *2008 Eighth International Conference on Hybrid Intelligent Systems*, IEEE, 2008, pp. 555–560.

[102] L. Meng *et al.*, "An approach of linear regression-based uav gps spoofing detection," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[103] S. Duangsuwan, M. M. Maw, "Comparison of path loss prediction models for uav and iot air-to-ground communication system in rural precision farming environment.," *J. Commun.*, vol. 16, no. 2, pp. 60–66, 2021.

[104] X. Liu, X. Li, Q. Shi, C. Xu, Y. Tang, "Uav attitude estimation based on marg and optical flow sensors using gated recurrent unit," *International Journal of Distributed Sensor Networks*, vol. 17, no. 4, p. 15 501 477 211 009 814, 2021.

[105] O. Ghorbanzadeh, S. R. Meena, T. Blaschke, J. Aryal, "Uav-based slope failure detection using deep-learning convolutional neural networks," *Remote Sensing*, vol. 11, no. 17, p. 2046, 2019.

[106] E. Basan, M. Lapina, N. Mudruk, E. Abramov, "Intelligent intrusion detection system for a group of uavs," in *Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part II 12*, Springer, 2021, pp. 230–240.

[107] M. Hossin, M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.

[108] Z. Vujović, "Classification model evaluation metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021.

[109] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, V. A. Kamaev, *et al.*, "A survey of forecast error measures," *World applied sciences journal*, vol. 24, no. 24, pp. 171–176, 2013.

[110] R. J. Hyndman, A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[111] T. Chai, R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)," *Geoscientific Model Development Discussions*, vol. 7, no. 1, pp. 1525–1534, 2014.

[112] C. J. Willmott, K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[113] A. De Myttenaere, B. Golden, B. Le Grand, F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016.

[114] D. Chicco, M. J. Warrens, G. Jurman, "The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, e623, 2021.

[115] B. Matthews, *Curated 4 class anomaly detection data set*, 2022. [Online]. Available: `https://c3.ndc.nasa.gov/dashlink/resources/1018`.

[116] T. A. Rodrigues *et al.*, *Data Collected with Package Delivery Quadcopter Drone*, Feb. 2021. doi: `10.1184/R1/12683453.v1`. [Online]. Available: `https://kilthub.cmu.edu/articles/dataset/Data_Collected_with_Package_Delivery_Quadcopter_Drone/12683453`.

[117] A. Choudhry, B. Moon, J. Patrikar, C. Samaras, S. Scherer, "Cvar-based flight energy risk assessment for multirotor uavs using a deep energy model," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 262–268.

[118] Arduino.cc, *Arduino Nano 33 BLE Sense Rev2*, https://docs.arduino.cc/resources/datasheets/ABX00069-datasheet.pdf, [Online; accessed 2023-10-22], Oct. 2023.

[119] H. V. Dudukcu, M. Taskiran, N. Kahraman, "Uav instantaneous power consumption prediction using lr-tcn with simple moving average," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 3, e7913, 2024.

[120] H. V. Dudukcu, M. Taskiran, N. Kahraman, "Unmanned aerial vehicles (uavs) battery power anomaly detection using temporal convolutional network with simple moving average algorithm," in *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, IEEE, 2022, pp. 1–5.

[121] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, Atlanta, Georgia, USA, vol. 30, 2013, p. 3.

[122] L. Lu, Y. Shin, Y. Su, G. E. Karniadakis, "Dying relu and initialization: Theory and numerical examples," *arXiv preprint arXiv:1903.06733*, 2019.

[123] R. J. Hyndman, *Moving averages.* 2011.

[124] H. V. Dudukcu, M. Taskiran, N. Kahraman, "Instantaneous power consumption prediction with modified temporal convolutional network for uavs," in *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2022, pp. 106–109.

[125] E. Systems, *Esp32 series datasheet v. 4.1*, [Online; accessed 2023-10-22], 2022.

[126] *Raspberry pi model b*, https://www.raspberrypi.org/products/raspberry-pi-3-model-b, [Online; accessed 2023-10-22].

PUBLICATIONS FROM THE THESIS

Papers

1. H. V. Dudukcu, M. Taskiran, N. Kahraman, "UAV Instantaneous Power Consumption Prediction Using LR-TCN with Simple Moving Average," *CONCURRENCY COMPUTATION PRACTICE AND EXPERIENCE*, vol.1, no.1, pp.1-12, 2023

2. H. V. Dudukcu, M. Taskiran, N. Kahraman, UAV Sensor Data Applications with Deep Neural Networks: A Comprehensive Survey," *ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE*, vol.123, no.Part C, pp.1-17, 2023

Conference Papers

1. H. V. Dudukcu, M. Taskiran, N. Kahraman, "Unmanned Aerial Vehicles (UAVs) Battery Power Anomaly Detection Using Temporal Convolutional Network with Simple Moving Average Algorithm," *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Biarritz, France, 2022, pp. 1-5, doi: 10.1109/INISTA55318.2022.9894193.

2. H. V. Dudukcu, M. Taskiran, N. Kahraman, "Instantaneous Power Consumption Prediction with Modified Temporal Convolutional Network for UAVs," *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, Prague, Czech Republic, 2022, pp. 106-109, doi: 10.1109/TSP55681.2022.9851230.

Projects

1. "Flight Anomaly Detection in Unmanned Aerial Vehicles with Hybrid Deep Learning Method", supported by Yildiz Technical University Scientific Research Projects Coordination Unit under project number FBA-2021-4671.