

T.R.
ONDOKUZ MAYIS UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF COMPUTATIONAL SCIENCES



**ANOMALY DETECTION SYSTEM FOR AVIONICS ADS-B:
ATTACK VECTORS AND MACHINE LEARNING**

Ph.D. Thesis

Nurşah ÇEVİK

Supervisor
Prof. Dr. Sedat AKLEYLEK

This thesis was supported by YÖK with 100/2000 Ph.D. Scholarship.

SAMSUN
2024

DECLARATION OF COMPLIANCE WITH SCIENTIFIC ETHIC

I hereby declare and undertake that I complied with scientific ethics and academic rules in all stages of my doctoral thesis, that I have referred to each quotation that I used directly or indirectly in the study, and that the works I have used consist of those shown in the references, that it was written in accordance with the institute writing guide and that the stated cases in article 3, section 9 of the Regulation for TÜBİTAK Research and Publication Ethics Board were not violated.

Is an Ethics Committee Necessary?

☐ Yes

☒ No

19/07/2024
Nurşah ÇEVİK

DECLARATION OF THE THESIS STUDY ORIGINALITY REPORT

Thesis Title: ANOMALY DETECTION SYSTEM FOR AVIONICS ADS-B:
ATTACK VECTORS AND MACHINE LEARNING

As a result of the originality report taken by me from the plagiarism detection program on 14/08/2024 for the thesis titled above:

Similarity ratio : %16

Single source rate : %1 has been released.

14/08/2024
Prof. Dr. Sedat AKLEYLEK

ÖZET

AVİYONİK ADS-B SİSTEMİ İÇİN ANOMALİ TESPİT SİSTEMİ: SALDIRI VEKTÖRLERİ VE MAKİNE ÖĞRENİMİ

Nurşah ÇEVİK

Ondokuz Mayıs Üniversitesi

Lisansüstü Eğitim Enstitüsü

Hesaplamalı Bilimler Ana Bilim Dalı

Doktora, Temmuz / 2024

Danışman: Prof. Dr. Sedat AKLEYLEK

Otomatik bağımlı gözetleme-yayın (ADS-B) sistemlerine yönelik güvenlik sorunları ve çözüm yöntemleri üzerine yapılan çalışmalar, artan hava trafik hacmi nedeniyle her geçen gün daha da kritik hale gelmektedir. Uçak ve yer istasyonları, yayınlanan ADS-B verilerini verinin kaynağını ve bütünlüğünü kontrol etmeden almaktadır. Bu nedenle, ADS-B sistemlerine yanlış veri göndererek sistemi kolayca yanıltmak mümkündür. Bu tez kapsamında, ADS-B sistemi için bir güvenlik çözümü olarak anomali/saldırı tespit sistemi geliştirilmiştir. ADS-B verilerindeki anomalileri/saldırıları tespit etmek için çeşitli parametre setleri analiz edilmiş ve kritik parametreler belirlenmiştir. Sahtecilik, mesaj ekleme ve sanal yörünge değişikliği saldırıları gibi sekiz farklı saldırı senaryosu için saldırı vektörleri oluşturulmuştur. Daha sonra gerçek saldırı senaryosuna yakınsamak için farklı saldırı vektörlerini birleştirerek hibrit veri kümeleri oluşturulmuştur. Gelecekteki akademik araştırmaları desteklemek ve diğer araştırmacıların bulgularımızı genişletmesine olanak sağlamak için üretilen veri kümeleri açık olarak paylaşılmaktadır. Literatürde çeşitli saldırı senaryolarını kapsamlı bir şekilde ele alan bildiğimiz ilk saldırı veri kümesidir. Üretilen veri kümeleri, anomali/saldırı tespit sistemlerinin değerlendirilmesi ve sağlamlığını arttırmak amacı ile çeşitli saldırı senaryolarını kapsamaktadır. Farklı makine öğrenmesi ve derin öğrenme modellerinin performansını değerlendirmek için üretilen saldırı veri kümelerini kullanılmıştır. Bu tez kapsamında elde ettiğimiz sonuçlar, mevcut yöntemlere göre doğruluk ve güvenilirliği arttırdığımızı göstermektedir.

Anahtar Sözcükler: ADS-B, Anomali tespit sistemi, Saldırı tespit sistemi, IDS, Makine öğrenmesi, Aviyonik sistemlerin güvenliği, Siber güvenlik

ABSTRACT

ANOMALY DETECTION SYSTEM FOR AVIONICS ADS-B: ATTACK VECTORS AND MACHINE LEARNING

Nurşah ÇEVİK

Ondokuz Mayıs University

Institute of Graduate Studies

Department of Computational Science

Ph.D., July / 2024

Supervisor: Prof. Dr. Sedat AKLEYLEK

The topic of security challenges and solutions for automatic dependent surveillance-broadcast (ADS-B) systems is becoming more critical day-to-day because of the increasing air traffic volume. Since aircraft and ground stations receive broadcast ADS-B data can not check the source and integrity of data, ADS-B systems can be spoofed easily by transmitting false data. In this thesis, we develop an anomaly detection system for ADS-B data as a security solution. To identify anomalies in ADS-B data, various parameter sets are analyzed to identify critical parameters. We define attack vectors for eight different attack scenarios, such as spoofing, message injection, and virtual trajectory change attacks, and create hybrid datasets by combining different attack vectors to increase the detection ability of different attack scenarios. We have made our attack dataset publicly available to support future academic research and enable comparing results. To our knowledge, this dataset is the first dataset in the literature to cover various attack scenarios comprehensively. These datasets have covered a wide range of attack scenarios to increase the robustness of anomaly detection assessments. We use attack datasets to evaluate the performance of different machine learning and deep learning models. Based on the results of our model, we increase the accuracy and reliability compared to existing methods.

Keywords: ADS-B, Anomaly detection system, Intrusion detection system, IDS, Machine learning, Avionics security, Cybersecurity

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Professor Sedat Akleylek, for guiding my thesis with his knowledge and experience, patiently answering my questions, and guiding me. His academic expertise significantly strengthened the scientific foundations of my work. His patient responses to my numerous questions and his continuous support and motivation at every stage enabled me to overcome various challenges with greater ease. His constructive criticisms and insightful suggestions substantially improved the quality of my research and have been an essential guide in my academic journey.

I also want to thank my thesis committee members Prof. Dr. Erdal KILIÇ and Assist. Prof. Dr. İlker ÖZÇELİK, who contributed to my thesis with their valuable suggestions.

I am grateful to my husband, Kadir ÇEVİK, for his patience and understanding. Completing this study without his support would have been much more challenging. I am also deeply grateful to my parents, Cemile KAYA and Fikret KAYA, for their constant presence and belief in me. Special thanks to my sister and brother, Livanur KAYA and Şamil KAYA, for their continuous motivation and support.

I would like to especially thank my son, Erdem ÇEVİK, for his patience and understanding when I was not with him while writing this thesis.

Lastly, I wish to thank all my friends and colleagues who contributed directly or indirectly to this process. This thesis would not have been possible without their support and encouragement.

Nurşah ÇEVİK

CONTENTS

ACCEPTANCE AND APPROVAL OF THE THESIS	i
DECLARATION OF COMPLIANCE WITH SCIENTIFIC ETHIC	ii
DECLARATION OF THE THESIS STUDY ORIGINALITY REPORT	ii
ÖZET	iii
ABSTRACT	iv
ACKNOWLEDGEMENTS	v
CONTENTS.....	vi
SYMBOLS AND ABBREVIATIONS.....	viii
FIGURES LEGENDS	xi
TABLES LEGENDS	xii
1. INTRODUCTION.....	1
1.1. Motivation and Contribution	4
1.2. Organization	7
2. PRELIMINARIES	9
2.1. Components of ADS-B	9
2.2. Technical Specifications.....	10
2.3. Open Source ADS-B Databases.....	11
2.4. Evaluation Methods and Metrics.....	12
2.4.1. Evaluation Methods	13
2.4.2. Evaluation Metrics	14
3. LITERATURE REVIEW	16
3.1. Research Methodology	16
3.2. ADS-B Security Solutions.....	19
3.2.1. Encryption-Based Methods	19
3.2.2. Location-Based Methods	20
3.2.3. Machine Learning-Based Methods.....	20
3.3. Comparison of ADS-B Security Solutions	24
4. ATTACK DATA GENERATION.....	26
4.1. Attack Scenarios on ADS-B.....	26
4.2. Methodology	30
4.3. Attacker Capabilities	34
4.3.1. Eavesdrop.....	34
4.3.2. Message Deletion.....	35
4.3.3. Message Injection	35
4.3.4. Replay Attack.....	36
4.3.5. Summary of Attacker Capabilities	37
4.4. Generation of Attack Data for ADS-B	37
4.5. Details of Attack Datasets	45
5. PROPOSED APPROACH: MACHINE LEARNING BASED ANOMALY/INTRUSION DETECTION SYSTEM FOR ADS-B.....	47
5.1. System Architecture and Design Goals.....	49
5.2. Data Collection and Preprocessing	52
5.3. Feature Engineering and Selection	53
5.3.1. Feature Generation	54
5.3.2. Correlation Analysis	55
5.3.3. Dimensional Reduction	57

5.3.4. Chi-Square Test	59
5.3.5. ANOVA F-test	59
5.3.6. Lasso Regression (L1 Regularization).....	60
5.3.7. Random Forest Classifier	60
5.3.8. Hybrid Approach	60
5.4. Evaluation Methods	61
5.5. Machine Learning for Anomaly Detection	61
5.6. Deep Learning for Anomaly Detection and Hyperparameters	63
6. EXPERIMENTAL RESULTS AND DISCUSSION	65
6.1. The Results of Deep Learning Models	65
6.1.1. Overfitting in Deep Learning	66
6.1.2. Analysis and Performance Evaluation of Deep Learning Algorithms	67
6.2. The Results of Machine Learning Models	69
6.3. Comparison	74
7. CONCLUSIONS AND FUTURE WORK.....	78
REFERENCES.....	81
CURRICULUM VITAE	86

SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS

<i>ACARS</i>	: Aircraft Communications Addressing and Reporting System
<i>ACC</i>	: Accuracy
<i>ADS – B</i>	: Automatic Dependent Surveillance-Broadcast
<i>AE</i>	: Autoencoder
<i>AES</i>	: Advanced Encryption Standard
<i>ANN</i>	: Artificial Neural Network
<i>ANOVA</i>	: Analysis of Variance
<i>API</i>	: Application Programming Interface
<i>ATC</i>	: Air Traffic Control
<i>AUC</i>	: Area Under the Curve
<i>CA</i>	: Certificate Authority
<i>CAE</i>	: Context Autoencoder
<i>HDP</i>	: Hierarchical Dirichlet Process
<i>CDTI</i>	: Cockpit Display of Traffic Information
<i>CNN</i>	: Convolutional Neural Network
<i>DBSCAN</i>	: Density-Based Spatial Clustering of Applications with Noise
<i>DL</i>	: Deep Learning
<i>DNN</i>	: Deep Neural Network
<i>DTC</i>	: Decision Tree Classifier
<i>ECC</i>	: Elliptic Curve Cryptography
<i>EFB</i>	: Electronic Flight Bag
<i>ES</i>	: Extended Squitter
<i>ETC</i>	: Extra Tree Classifier
<i>FAA</i>	: Federal Aviation Administration
<i>FN</i>	: False Negative
<i>FNR</i>	: False Negative Rate
<i>FP</i>	: False Positive
<i>FPR</i>	: False Positive Rate
<i>F1S</i>	: F1-Score

<i>GBT</i>	: Gradient Boosting Trees
<i>GMM</i>	: Gaussian Mixture Model
<i>GNB</i>	: Gaussian Naive Bayes
<i>GNSS</i>	: Global Navigation Satellite System
<i>GPS</i>	: Global Positioning System
<i>HMM</i>	: Hidden Markov Model
<i>ICAO</i>	: International Civil Aviation Organization
<i>IDS</i>	: Intrusion Detection System
<i>IEEE</i>	: Institute of Electrical and Electronics Engineers
<i>KNN</i>	: K-Nearest Neighbors
<i>LOF</i>	: Local Outlier Factor
<i>LR</i>	: Logistic Regression
<i>LSTM</i>	: Long Short-Term Memory
<i>ML</i>	: Machine Learning
<i>MLAT</i>	: Multilateration
<i>MLP</i>	: Multilayer Perceptron
<i>NB</i>	: Naive Bayes
<i>NN</i>	: Neural Network
<i>ORCID</i>	: Open Researcher and Contributor ID
<i>PCA</i>	: Principal Component Analysis
<i>PKI</i>	: Public Key Infrastructure
<i>PRE</i>	: Precision
<i>PSR</i>	: Primary Surveillance Radar
<i>REC</i>	: Recall
<i>RF</i>	: Radio Frequency
<i>RFID</i>	: Radio-Frequency Identification
<i>RNN</i>	: Recurrent Neural Network
<i>ROC</i>	: Receiver Operating Characteristic
<i>SQL</i>	: Structured Query Language
<i>SSR</i>	: Secondary Surveillance Radar
<i>SVM</i>	: Support Vector Machine
<i>TCAS</i>	: Traffic Collision Avoidance System
<i>TDOA</i>	: Time Difference of Arrival
<i>TL</i>	: Transfer Learning

<i>TN</i>	: True Negative
<i>TP</i>	: True Positive
<i>t – SNE</i>	: t-distributed Stochastic Neighbor Embedding
<i>UAT</i>	: Universal Access Transceiver
<i>UAV</i>	: Unmanned Aerial Vehicle
<i>UMAP</i>	: Uniform Manifold Approximation and Projection
<i>VDL</i>	: VHF Data Link
<i>VHF</i>	: Very High Frequency



FIGURES LEGENDS

Figure 2.1.	The general architecture of the ADS-B system	10
Figure 2.2.	ADS-B message format	11
Figure 2.3.	Confusion matrix.....	14
Figure 3.1.	The results for research criteria.....	18
Figure 3.2.	The results for research and filter criteria	19
Figure 4.1.	Threat model	31
Figure 4.2.	Ground station - attack vectors	32
Figure 4.3.	Aircraft - attack vectors.....	33
Figure 4.4.	Attack generation phase	38
Figure 4.5.	Comparison of original and spoofed trajectories	43
Figure 5.1.	System model	47
Figure 5.2.	System model flowchart	48
Figure 5.3.	Designing phase of anomaly detection system.....	50
Figure 5.4.	Correlation matrix - feature to feature	56
Figure 5.5.	Correlation matrix - feature to label.....	56
Figure 5.6.	Dimensional reduction methods.....	58
Figure 6.1.	Model Loss and Accuracy Graphics for CNN	67
Figure 6.2.	Model Loss and Accuracy Graphics for RNN	68
Figure 6.3.	Model Loss and Accuracy Graphics for MLP.....	68

TABLES LEGENDS

Table 2.1.	Comparison of ADS-B databases	12
Table 2.2.	Definitions and formulas for various performance metrics	15
Table 3.1.	Search and filtering criteria	18
Table 3.2.	Publications data from different databases	18
Table 3.3.	Comparison of ADS-B security solutions	25
Table 4.1.	Severity of Impact for Various ADS-B Attacks	29
Table 4.2.	The comparison of ADS-B attack scenarios	30
Table 4.3.	Application in scenarios for eavesdropping	35
Table 4.4.	Application in scenarios for message deletion.....	35
Table 4.5.	Application in scenarios for message injection.....	36
Table 4.6.	Application in scenarios for replay attack.....	36
Table 4.7.	Summary of capabilities in each scenario	37
Table 4.8.	Suitable increment ranges for ADS-B parameters	38
Table 4.9.	Details of hybrid datasets	46
Table 5.1.	OpenSky Network data details	53
Table 5.2.	Normalized dataset	53
Table 5.3.	Feature types and their corresponding features	57
Table 5.4.	Hyperparameters for different deep learning models	64
Table 6.1.	Deep learning model results for dataset 1	66
Table 6.2.	Algorithm performance comparison for dataset 1 (time parameter set without scaling)	69
Table 6.3.	Algorithm performance comparison for dataset 1 (for scaling).....	69
Table 6.4.	Algorithm performance comparison for dataset 1 (distance parameter set with scaling)	70
Table 6.5.	Algorithm performance comparison for dataset 2 (scaling)	71
Table 6.6.	Algorithm performance comparison for dataset 2 (distance parameter set with scaling)	71
Table 6.7.	Algorithm performance comparison for dataset 3 (for scaling).....	72
Table 6.8.	Algorithm performance comparison for dataset 3 (distance parameter set scaling)	72
Table 6.9.	Algorithm performance comparison for dataset 4 (for scaling).....	73
Table 6.10.	Algorithm performance comparison for dataset 4	73
Table 6.11.	Performance evaluation metrics	75

1. INTRODUCTION

Tracking aircraft systems is a significant component of avionic systems. Firstly, aircraft and ground stations used primary surveillance radar (PSR), which emit radio waves and detect the reflections from aircraft. After PSR, secondary surveillance radar (SSR) enhances primary radar capabilities by requiring aircraft to be equipped with a transponder, which responds with information such as identification and altitude. It provides more precise and reliable data. All these technologies are used before the automatic dependent surveillance-broadcast (ADS-B) system. The need for new technology occurred because of their limitation of accuracy and other requirements for ground stations. ADS-B improved real-time tracking systems by increasing the accuracy and precision of tracking data using the satellite navigation system.

Automatic Dependent Surveillance-Broadcast devices are crucial for the aviation industry to enhance safety, efficiency, and situational awareness. Air traffic controllers (ATCs) and aircraft, such as commercial airlines, helicopters, and private planes, mainly use the ADS-B system to track aircraft within the airspace. There are two ADS-B components: ADS-B Out and ADS-B In. ADS-B Out periodically broadcasts its position, velocity, and other essential information to other aircraft and ATCs within the airspace. ADS-B In receives data from other ADS-B Out devices within the airspace. According to the civil aviation authority, ADS-B Out devices have been mandatory for all aircraft since 2020 (Federal Aviation Administration, 2010). The ADS-B system is connected to critical aircraft systems such as the traffic collision avoidance system (TCAS). According to ADS-B data, TCAS decides whether to give an alarm. This example shows the importance of secure ADS-B data transmission.

In the view of cybersecurity, the ADS-B data is broadcast without a cryptographic signature or encryption mechanisms (Schafer et al., 2013). The federal aviation administration (FAA) published a report in 2020 that highlighted possible attack scenarios on ADS-B data, such as eavesdropping, message injection, and deletion (Federal Aviation Administration, 2010). The topic of security challenges and solutions for ADS-B systems is becoming more critical day-to-day because of the increasing air traffic volume. Aircraft and ground stations that received broadcast

ADS-B data could not check the source and integrity of data because of the lack of encryption and signature mechanisms. Therefore, ADS-B data can be spoofed easily by transmitting false data. Attackers can eavesdrop and inject false messages into the ADS-B system, causing confusion and potential hazards. Although it is an easy attack using software-defined radio and open information, its potential damage is also high. ADS-B systems use global positioning system (GPS) to obtain accurate positional data such as latitude, longitude, altitude, and velocity. Therefore, GPS attacks also affect the integrity and availability of the ADS-B systems. The secure transmission of ADS-B messages is an open and exciting research topic. In last decade, many research are conducted to security of ADS-B system thanks to the FAA reports (Federal Aviation Administration, 2010). In the context of attacks against ADS-B systems, there are two main types: air-to-air and ground-to-air attacks. In ground-to-air attacks, attacks come from ground-based sources and target aircraft. The attacker can inject or delete messages in the target aircraft's ADS-B system. For example, injected false messages can create ghost aircraft that seem like real aircraft in the radar system, potentially causing hazardous situations and disrupting normal flight operations. In contrast to ground-based spoofing attacks, possible aircraft attacks are still insufficiently investigated despite being an actual threat. Drones, or unmanned aerial vehicles (UAVs), can be used to carry out these attacks, so more caution has to be taken while considering this kind of possible threat. Besides traditional aviation, ADS-B plays a crucial role in modern UAVs (Strohmeier et al., 2014). With an increase in the number of UAVs and drones that use GPS for navigation, these systems are a potential target for attackers during airspace surveillance. In other words, it is known that it is feasible to cause a UAV's target deviation from its planned trajectory. Therefore, the OpenSky Network research project gathers ADS-B signals and makes them available for security research and analysis (D. Kozovic & Djurdjevic, 2021). The use of the GPS and ADS-B by UAVs for communication and navigation is increasing. However, the security of UAV operations may be threatened by these systems' vulnerability to various cyberattacks, such as ADS-B and GPS spoofing. In addition to academic research, real-world examples demonstrate the vulnerabilities of ADS-B systems. In 2012, researchers successfully showed how to inject false signals into a UAV's navigation system. This research highlighted that taking control of the

UAV and changing its flight path is possible by manipulating ADS-B signals (Humphreys, 2015). Moreover, a recent example occurred at Tartu Airport in Estonia. In 2024, Finnair was forced to suspend flights in Tartu due to GPS jamming attack. Tens of thousands of civilian flights have been affected by GPS jamming in recent months, according to aviation experts. This attack affects all GPS users in the region, including boats in the Baltic Sea. Finnair announced that GPS interference does not affect flight safety if pilots are aware of this, since aircraft have alternative systems in place when the GPS signal is interfered with. These attacks demonstrate the impact of GPS and ADS-B vulnerabilities on flight operations and the importance of awareness of attacks in real-world scenarios.

In this thesis, we concentrate on using machine learning based anomaly/intrusion detection system (IDS) for ADS-B security. The aim of machine learning-based anomaly/intrusion detection techniques is to find anomalies in ADS-B message patterns or deviations from expected behavior. These techniques are capable of identifying abnormal or suspicious behaviors. Attacks, such as message injection, deletion jamming (Wu et al., 2020), and spoofing (Manesh, Kenney, et al., 2019; Manesh, Velashani, et al., 2019; Chen et al., 2019; Ying et al., 2019), can be found using anomaly detection techniques. In recent years, the practical application of machine learning (ML) and deep learning (DL) techniques has grown, and their usability has also improved due to their reliability. Thanks to these advancements, deep learning and machine learning-based techniques can be used as an alternative for anomaly/intrusion detection systems in the aviation industry. In addition, lately, deep learning techniques have come to be favored for anomaly detection systems (Manesh, Kenney, et al., 2019; Wang et al., 2020; Fried & Last, 2021; Habler & Shabtai, 2018; Chevrot, 2022). In this thesis, we develop a novel anomaly/intrusion detection system based on deep learning and machine learning techniques. Our system does not require any protocol changes or additional sensor information. Our primary objective is to detect message injection and modification attacks. The absence of attack/anomaly data is the limitation of these systems. Considering this situation, we generate a hybrid attack dataset consisting of eight scenarios and share this dataset.

This thesis on anomaly detection for ADS-B data is a compelling intersection of aviation systems, cybersecurity, machine learning, and statistics. By deeply

understanding ADS-B technology and the regulatory landscape, these solutions are not only practical but also industry-compliant. We present a comprehensive overview of the current state of the field, and give key concepts, methodologies, and findings related to ML/DL-based intrusion detection for ADS-B devices. The advantages and disadvantages of ML/DL models for ADS-B intrusion detection are evaluated to provide insights into the performance, accuracy, and limitations of different models. Utilizing advanced machine learning models to identify and mitigate attack vectors addresses critical cybersecurity issues in aviation. Our work significantly improves aviation safety by enabling real-time threat detection, thus detecting and preventing potential malicious activities. Beyond aviation, the methodologies and models developed have broader implications for critical infrastructure protection and other fields requiring robust anomaly detection. This thesis advances machine learning techniques and informs policymakers and regulatory bodies, contributing to better standards and practices in aviation security. As a result, this thesis demonstrates the power of cross-disciplinary work to achieve significant advances in both technology and industry applications.

1.1. Motivation and Contribution

ADS-B systems are vulnerable to various types of cyber attacks that can compromise the security and integrity of air traffic management. Despite existing academic researches, several gaps persist in the literature (Strohmeier et al., 2015). Various challenges encountered in anomaly detection in ADS-B data limit the development of practical solutions (Wu et al., 2020). First, large amounts of qualified and labeled data are required to effectively train ML/DL models. However, real-world anomaly/intrusion events for avionics systems are rare. This situation makes obtaining sufficiently diverse and balanced datasets difficult and can lead to bias in model performance. It is necessary to use techniques such as oversampling, undersampling, or creating synthetic data to overcome this problem. In particular, deviations from routes may be caused by weather conditions or air traffic density, which cause deviations from the generally linear flight paths for a given route (Blåberg et al., 2020). Such deviations can limit the effectiveness of anomaly detection systems used for intrusion detection, and these variations must be considered before these systems

can be used practically in the real world. Determining effective thresholds poses a significant real-world challenge, especially when dealing with unknown attack patterns. This situation necessitates the development of automatic adaptation capabilities against emerging threats. Periodic changes in aircraft signatures also complicate the applicability of anomaly detection systems (Cevik & Akleylek, 2024b). Aircraft signatures may change over time due to using different transponders and two ADS-B antennas located in different locations on the aircraft. This situation makes it challenging to detect slow changes. The massive volume of ADS-B data generated annually (approximately 41 TiB according to FAA statistics) requires scalable solutions for the real-world applicability of these systems (Mink et al., 2021). This large volume of data stands out as a significant practical challenge in terms of efficient processing and analysis of data.

On the other hand, implementing such models requires high computational resources during the training and testing phases, which may particularly conflict with real-time data processing requirements (Wang et al., 2020). Real-time processing, which is extremely important for aviation security, stands out as a factor limiting the application of these models to ADS-B devices. In addition, the interpretability of ML/DL models is another major obstacle. The fact that models such as deep neural networks are considered "black boxes" makes it difficult to use these models safely, especially in critical security applications such as aviation (Cevik & Akleylek, 2024b). The complexity of these models makes it difficult to understand classification processes.

Regulatory and certification issues are also significant factors limiting the applicability of ML/DL-based solutions in the aviation industry (Federal Aviation Administration, 2010). New security solutions must comply with existing certification standards, making integrating new technologies into the sector difficult. Furthermore, such solutions' deployment and maintenance costs can pose a significant economic burden, especially for smaller aviation companies and organizations.

Additionally, ML/DL models may be vulnerable to adversarial attacks. Attackers can generate malicious data to mislead these models, leading to potential security breaches. Developing resilient models for such attacks remains an area of ongoing

research (Lei et al., 2023). Finally, the false positive rate of ML/DL models poses a significant problem that can directly affect critical decisions in terms of flight safety. Reducing false positives and improving feature selection methods are essential in this field.

Given the limitations in current methodologies, our study is directed toward generating comprehensive datasets and developing robust detection systems tailored to the unique challenges of ADS-B security. Most of the existing studies focus on a narrow set of attack scenarios. Therefore, they fail to capture complex real-world threats. In addition, there is no comprehensive and well-constructed dataset that include various attack vectors, publicly available. This thesis aims to generate attack data from attack vectors and develop robust anomaly detection systems to solve open problems for ADS-B security.

1. We present a comprehensive overview of ML/DL based IDS for avionic systems and review identifies the challenges and open research problems related to real-time implementations. Recognizing these challenges helps guide the development of new methodologies.
2. We design attack vectors for all attack scenarios in the literature. We create a hybrid dataset for comprehensive coverage of attack scenarios. It provides realistic evaluation and improves detection performance by comparing other research in the literature. We publicly share our dataset, the first comprehensive and detailed dataset we know to be used in ADS-B anomaly detection. Our dataset enables other researchers to extend our findings and includes different attack vectors and parameter sets.
3. We generate additional time, distance, and velocity-based features. Time-based features capture temporal anomalies, distance-based features ensure spatial awareness and velocity-based features highlight unusual speed variations.
4. This thesis provides a deep analysis of the performance of various ML and DL models. The results of the decision tree classifier (DTC) have the lowest false negative (FNR) and false positive rate (FPR), which are 0.004 and 0, respectively. Low FNR and FPR results are essential for critical systems to improve operational

efficiency, safety, and incident response mechanisms.

These contributions advance the state of the art in ADS-B anomaly detection systems.

1.2. Organization

In Chapter 1, we define the tracking system for avionics and discuss the criticality of the ADS-B system from a cybersecurity perspective. This chapter outlines the problem and emphasizes the importance of addressing it. It highlights the motivation for using machine learning and deep learning techniques and outlines the main contributions of the thesis.

In Chapter 2, we provide the technical background of the ADS-B system. This chapter explains and compares the open-source databases used to collect ADS-B data across different aspects. It also describes the evaluation methods and metrics in detail.

In Chapter 3, we offer an overview of the problem of anomaly and intrusion detection in ADS-B data. This chapter presents a comprehensive overview of the field's current state, covering key concepts, methodologies, and findings related to ML/DL-based intrusion detection for ADS-B devices. It evaluates the advantages and disadvantages of ML/DL models for ADS-B intrusion detection to provide insights into different models' performance, accuracy, and limitations.

In Chapter 4, we explain the attack generation process used in this thesis. After defining the attack scenarios for ADS-B, we present the attack generation methodology and details of the attacker's capabilities. As a result, we generate attack data and present hybrid datasets using the defined attack scenarios and attacker capabilities.

In Chapter 5, we define the system architecture. This chapter begins with collecting real ADS-B data from the historical OpenSky Network database. After the data preprocessing steps, we merge the real ADS-B data with the generated attack data to produce the final datasets. We analyze features using methods like chi-square and correlation analysis and select the most critical features to train and test machine learning and deep learning models. We describe the evaluation methods and discuss the machine learning and deep learning algorithms employed for anomaly and intrusion detection.

In Chapter 6, we evaluate the experimental results of the models for the generated datasets. We present detailed results using the evaluation metrics and a thorough analysis and comparison with existing methods.

In Chapter 7, we summarize the key findings of the thesis, discuss the implications of the results, and suggest potential directions for future research.



2. PRELIMINARIES

ADS-B is a cutting-edge technological system that integrates the most recent technology advancements in surveillance, navigation, and telecommunications. It is a crucial component of the Eurocontrol and FAA project NextGen, which aims to enhance the air traffic system.

In this chapter, we explain the components and functionality of ADS-B devices, alongside a comparison of open-source historical ADS-B databases. This chapter is organized as follows: In Section 2.1, we recall ADS-B components which are ADS-B In and ADS-B Out. Section 2.2 reviews the technical details of the ADS-B system, including message format and protocol. The literature presents different data collection methods, such as utilizing open-source datasets and constructing experimental setups to collect ADS-B data. In Section 2.3, we explain our methods and provide detailed comparisons of open-source databases from various perspectives.

This chapter is presented in (Cevik & Akleylek, 2022).

2.1. Components of ADS-B

The ADS-B system depends on the positioning system's precision because the global navigation satellite system (GNSS) is an essential component. ADS-B devices periodically broadcast real-time flight and position information obtained from satellite navigation systems. ADS-B In (receiver) and an ADS-B Out (transmitter) are two functionalities of the ADS-B system. Some aircraft equipped with integrated systems can support both ADS-B In and ADS-B Out functionalities within a single device (Lyons et al., 2016). Others have only ADS-B Out, which has been mandated by the FAA since 2020.

ADS-B Out transmits the location and other flight-related data to other aircraft and ground stations within the range. The ADS-B In enables the aircraft to receive data transmitted by other aircraft positioned within the range and displayed on the cockpit display of traffic information (CDTI) interfaces (typically, multi function display (MFD) and electronic flight bag (EFB) devices). TCAS systems also use the ADS-B data to

avoid collision. The aircraft's status data is transferred within the ADS-B Out system (D. Kozovic & Djurdjevic, 2021).

1. ADS-B Out, situated within the aircraft, transmits exact positional data, including the aircraft's GPS location, altitude, and ground speed, to ground controllers and other aircraft. Its accuracy surpasses that of traditional radar surveillance (D. Kozovic & Djurdjevic, 2021).
2. ADS-B In is positioned on the aircraft or at ground stations, acquiring data from ADS-B Out devices within its range. Initially deployed in ground and air traffic control systems due to its faster position transmission than other radar systems, it is currently being integrated into aircraft for air traffic monitoring purposes.

ADS-B system has components in ground stations and aircraft: 1. Ground station components: stations and antenna system in the ground station, 2. Aircraft components: ADS-B specialized transponder, GPS, receiver, altimeter, CDTI. Besides these components, ADS-B has regulatory procedures for implementation and operational phases (D. V. Kozovic et al., 2023). The general architecture of the ADS-B system is described in Figure 2.1.

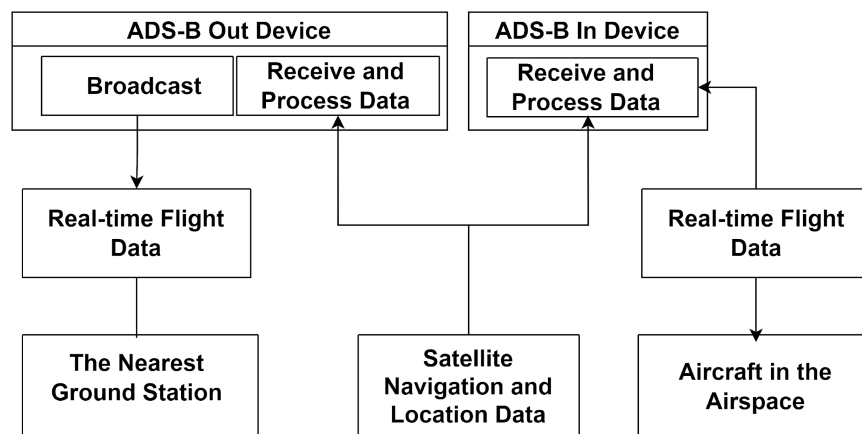


Figure 2.1. The general architecture of the ADS-B system

2.2. Technical Specifications

ADS-B uses the radio system according to standardized communication protocols, such as 1090 MHz extended squitter (1090-ES), 987 MHz universal access

transceiver (UAT), and very high frequency datalink mode 4 (VDL-M4), which will be used depending on the type of aircraft (by the FAA guidelines) (D. V. Kozovic et al., 2023). ADS-B messages include an 8-bit preamble for synchronization and a 56-bit (short) or 112-bit (extended) data block. An extended message contains 112 bits and is transmitted via 1090 MHz ("extended squat") data links (Federal Aviation Administration, 2010). The ADS-B protocol structure, which uses a 112-bit message frame, is given in Figure 2.2.

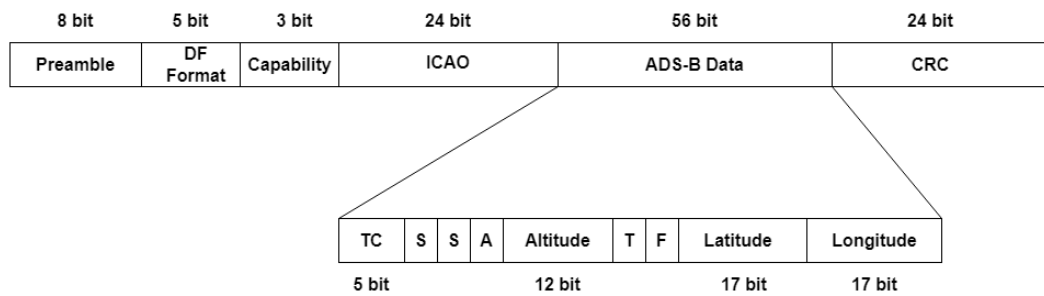


Figure 2.2. ADS-B message format

There is no encryption mechanism in the ADS-B system; therefore, ADS-B messages are broadcast in an unencrypted form. Anyone who receives the messages can understand them. ADS-B messages have no cryptographic integrity mechanism; however, has a parity check against transmission errors in the messages. ADS-B frames are modulated using 1 ms pulses and transmitted at 1 Mbit/s. Hence, the total time of the ADS-B extended message is 120 ms (including the preamble) (D. Kozovic & Djurdjevic, 2021).

2.3. Open Source ADS-B Databases

There are two main methods of collecting ADS-B data: creating a test environment using a hardware-based data collection module and collecting historical data from open source websites such as FlightAware, ADS-B Exchange, FlightRadar 24, OpenSky Network, and the FAA database. When comparing these options in this Section, we decide to use OpenSky Network historical data sources because the scope of the hardware-based data collection module was limited. There are five options for historical databases: FlightAware (FlightAware, 2024), ADS-B Exchange (ADS-B Exchange, 2024), FlightRadar 24 (FlightRadar24, 2024), OpenSky Network

(OpenSky Network, 2024), and the FAA database(Federal Aviation Administration, 2024). In Table 2.1, the comparison of historical databases are given.

Table 2.1. Comparison of ADS-B databases

Feature	FlightAware	ADS-B Exchange	Flight Radar24	OpenSky Network	FAA Database
Type	Commercial	Open-source, community	Commercial	Non-profit, open-access	Government provided
Coverage	Global	Global	Global	Global	Primarily U.S.
Data Types	Real-time, historical	Raw	Real-time, historical	Raw, historical	Historical, air traffic
Free Access	Basic features	Full raw data for personal	Basic features with delay	Free for research	Public access
Paid Plans	Enhanced features, API	Subscription for commercial	Enhanced features, API	Primarily non commercial	N/A
API Access	Yes	Yes	Yes	Yes	No

According to Table 2.1, although the FAA is the most reliable database, its datasets are primarily limited to the United States. FlightAware, ADS-B Exchange, and FlightRadar 24 have subscription costs. The OpenSky Network focuses on research and educational purposes and provides access to real-time and historical data for non-commercial use. That is why we decide to use the OpenSky Network in this thesis. OpenSky Network includes flight data, metadata, and geospatial coverage data with different formats such as comma separated values (CSV), structured query language (SQL) relational database.

2.4. Evaluation Methods and Metrics

Evaluation methods and metrics are critical in machine learning and deep learning. They show predictive models performance in assessing and validating phase. Metrics like accuracy, precision, recall, and the F1 score offer quantitative insights into a model’s effectiveness. These metrics ensure that models generalize for new data, vital for their deployment in real-world applications and practical scenarios. Additionally, evaluation metrics help demonstrate specific weaknesses in a model. For

example, if there are too many false positives or false negatives, precision, and recall metrics will highlight these issues, guiding necessary refinements to improve accuracy and reliability. Standardized evaluation methods also allow for the objective comparison of different models. This is essential for selecting the best model for a particular task or dataset, ensuring it meets performance requirements.

In summary, evaluation methods and metrics are essential in ML and DL for measuring model performance, facilitating model comparison, ensuring generalization, and identifying areas for improvement. They are integral to developing robust, effective, and reliable predictive models. In this section, evaluation methods for machine learning and deep learning methods are described.

2.4.1. Evaluation Methods

In this thesis, we employ two distinct evaluation methods: cross-validation and validation on a different dataset to assess the performance and generalizability of our models. These methods provide a comprehensive understanding of the models' performance on unrecognized data and ensure the robustness of our results.

Cross-Validation: It is commonly used to assess the performance of a model on a limited data sample. In cross-validation, the dataset is divided into various training-test parts. This approach reduces the risk of overfitting and ensures the performance of the model across various data partitions (Kohavi, 1995). Therefore, we use k-fold cross-validation in the model training and testing phases. The general procedure is given in Algorithm 1.

Algorithm 1 K-Fold cross-validation

- 1: **Input:** Dataset D , number of folds k
 - 2: **Output:** The aggregated performance metric
 - 3: Split the dataset D into k subsets (folds): $D_1, D_2, \dots, D_k$
 - 4: **for** each fold $i = 1$ to k **do**
 - 5: Let $D_{train} = D \setminus D_i$ (all folds except D_i)
 - 6: Let $D_{test} = D_i$ (the i -th fold)
 - 7: Train the model on D_{train}
 - 8: Test the model on D_{test}
 - 9: Record the performance metric
 - 10: **end for**
 - 11: Aggregate the performance metrics across all k folds
-

Validation on Different Datasets: Analyzing the model’s performance on completely different datasets provides additional information. Moreover, testing the model on various datasets improves its generalizability and robustness. Therefore, data from different periods, regions, or conditions have been collected for this thesis’s validation process. Applying these practices ensures a more rigorous and comprehensive evaluation of the model’s performance, leading to improved generalization and robustness in real-world applications.

2.4.2. Evaluation Metrics

The performance of an ADS-B anomaly detection system can be influenced by several factors: anomaly definition, data quality, detection algorithms. Therefore, different evaluation metrics are used to compare the performance of an ADS-B anomaly detection models. These metrics provide for identifying strengths and weaknesses of the models. In this thesis, we use accuracy, precision, recall, F1 score, area under the roc curve (ROC AUC), false positive rate, false negative rate, and confusion matrix to get a comprehensive understanding of the model’s performance. This section provides an overview of common performance metrics in the literature.

The confusion matrix, given in Figure 2.3, is used to assess the accuracy and effectiveness of a predictive model, especially in binary classification problems. It provides the calculation of various evaluation metrics such as accuracy, precision, recall, and F1 score. In Table 2.2, we summarize the definition and formula of these metrics (Bishop, 2006).

		Actual Values		
		Positive	Negative	
Predicted Values	Positive	True Positive TP	False Negative FN	
	Negative	False Positive FP	True Negative TN	

TP: Model correctly predicts the positive class.
FN: Model incorrectly predicts the negative class (Type II error).
FP: Model incorrectly predicts the positive class (Type I error).
TN: Model correctly predicts the negative class.

Figure 2.3. Confusion matrix

Table 2.2. Definitions and formulas for various performance metrics

Definition	Formula
Accuracy: It measures the proportion of true results among the total number of cases examined.	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision: It measures the proportion of true positives among the total predicted positives.	$\frac{TP}{TP+FP}$
Recall: It measures the proportion of true positives among the actual positives.	$\frac{TP}{TP+FN}$
F1 Score: It is the harmonic mean of precision and recall.	$2 \times \frac{\text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}}$
FPR: It measures the proportion of false positives among the actual negatives.	$\frac{FP}{FP+TN}$
FNR: It measures the proportion of false negatives among the actual positives.	$\frac{FN}{FN+TP}$

In real-world scenarios for ADS-B anomaly detection systems, FPR and FNR are critical parameters because of the ADS-B's specific implementation and operational environment. While the false positive rate is the ratio of normal ADS-B data points incorrectly identified as intrusions, the false negative rate is the ratio of actual intrusions incorrectly identified as normal ADS-B data. While analyzing the model results, these metrics provide a deep foresight of the performance of models. Recent academic studies show the importance of evaluating model results using metrics like FPR and FNR. When we compare the FPR and FNR metrics for ADS-B anomaly detection systems, each metric serves the system's different goals. FNR is a critical metric in terms of safety and security because missing a real anomaly can have severe consequences for avionic systems. FPR is crucial for operational efficiency and resource management. For example, if FPR is high, too many false alarms can overwhelm the system and distract the pilot. There is a trade-off between false positive and false negative rates. Reducing false positive rates can increase the ratio of false negative rates and vice versa, depending on the threshold and system. Ultimately, these metrics are connected. Therefore, it is necessary to reach an optimal balance by assessing the specific requirements and constraints.

3. LITERATURE REVIEW

This chapter examines security solutions, including traditional and machine learning-based anomaly/intrusion detection methods for ADS-B systems, and shares information about approaches consisting of machine learning and deep learning models. It concentrates on machine learning-based anomaly/intrusion detection systems. Then, the algorithms and evaluation metrics used in anomaly/attack detection systems are explained.

In this literature review, we follow a systematic approach and define selection and elimination criteria before searching academic databases. After filtering within the selection and elimination criteria, 15 studies from academic databases were examined in detail and summarized. The main ideas, advantages, and disadvantages of these studies are analyzed. Then, machine learning, deep learning methods, and datasets used within the scope of the studies are shared.

This chapter is organized as follows: In Section 3.1, we share our research methodology with research questions, selection, and elimination criteria. In Section 3.2, traditional and ML-based ADS-B security solutions are explained. According to our methodology, ML-based security solutions are analyzed and described in detail. In Section 3.3, ADS-B security solutions are compared, and their advantages and disadvantages are shared.

This chapter is presented in (Cevik & Akleylek, 2024b, 2022).

3.1. Research Methodology

This section gives details on research methodology to obtain the machine learning based anomaly detection system for ADS-B data to develop new methods and compare with our results. The research questions are defined below:

1. Which algorithms are commonly used to detect anomalies in ADS-B data? The purpose of the question is analyzing the performance metrics and accuracy rates of algorithms frequently used for intrusion detection in ADS-B systems and to analyze their relationship with the dataset.

2. Which ADS-B databases are used to train models? The purpose of the question is to identify frequently used databases in the literature and to compare the results of studies using the same database.
3. Which features are helpful for training models? The purpose of the question is to determine the parameters used in the literature and to analyze the parameters' effects on the evaluation metrics results. Parameter extraction is the most crucial step in machine learning and deep learning models.
4. What are common metrics used to evaluate models in the anomaly detection system of ADS-B? The purpose of the question is analyzing evaluation metrics used in different fields.
5. For which vulnerabilities of ADS-B can the proposed security solutions be effective? The purpose of the question is identifying which types of attacks the solutions provide security against. The presented intrusion detection systems detect different types of attacks.

Considering these research questions, the keywords "machine learning," "deep learning," and "ADS-B" are searched in academic databases: Scopus, IEEE Xplore, and Web of Science. Studies on the security of ADS-B data and analysis and security measures with machine learning or deep learning methods are examined. Between 2018 and 2024, a total of 34 publications, 25 of which were conference and 9 journal publications, were published in the IEEE database. On Scopus, another important database, a total of 53 publications, 30 of which were conference publications and 23 journal, were published between 2012 and 2024. 2 of the conference publications and one of the journal publications are survey studies. Finally, the Web of Science database is examined, and a total of 41 publications were published between 2018 and 2024, including 19 conference publications, 22 journal publications.

Within the scope of this thesis, studies on the security of ADS-B data and analysis and security measures with machine learning or deep learning methods are examined. During the selection of the studies examined, academic databases are searched using the search terms defined in Table 3.1.

Table 3.1. Search and filtering criteria

Search Criteria (SC)	
SC1	English-language studies published in a journal or conference
SC2	The journal in which the study was published should be at Q1 or Q2.
SC3	Studies focusing on intrusion detection in the ADS-B system
Filtering Criteria (FC)	
FC1	The study is a survey article
FC2	The study does not contain information about intrusion detection.
FC3	The study does not contain information about the dataset used.

According to research and filter criteria, the number of results are given Table 3.2 and Figures 3.1 and 3.2.

Table 3.2. Publications data from different databases

Database	Years	Total	Conference	Journal	Survey
IEEE Xplore	2018-2024	34	25	9	0
Scopus	2012-2024	53	30	23	3
Web of Science	2018-2024	41	19	22	0

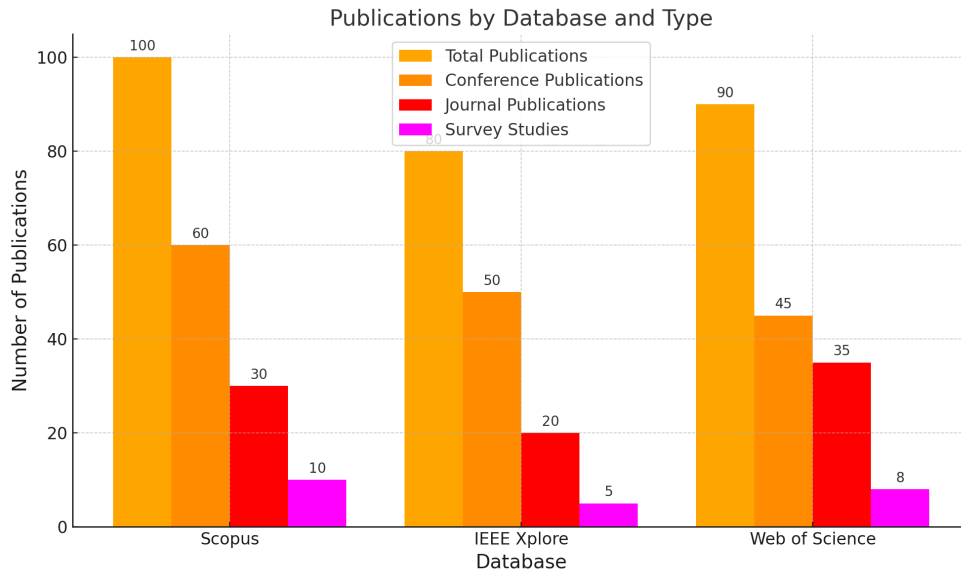


Figure 3.1. The results for research criteria

The summaries of these publications are examined, and their compatibility with the scope of this thesis is reviewed. This way, the number of publications to be examined is reduced from 55 to 14. In this thesis, 14 studies related to the security of the ADS-B

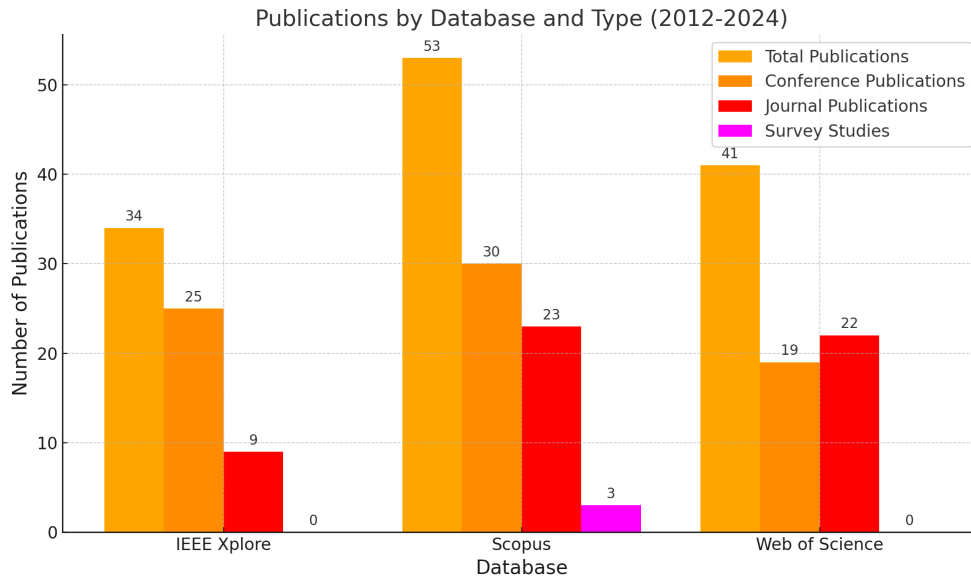


Figure 3.2. The results for research and filter criteria

system in terms of area, title, summary, and scope are examined and compared on issues such as the method used, database, and performance rates.

3.2. ADS-B Security Solutions

This section describes ADS-B security solutions and their advantages and disadvantages. In the literature, ADS-B device security solutions are collected using three main approaches: encryption-based, location-based, and anomaly detection-based techniques.

3.2.1. Encryption-Based Methods

Encryption-based methods use cryptographic encryption techniques to provide confidentiality and integrity of ADS-B messages during transmission. There are two approaches: asymmetric and symmetric encryption mechanisms. For example, elliptic curve cryptography (ECC) based methods are asymmetric encryption mechanisms that provide authentication or signature verification (Cevik & Akleylek, 2022; Wesson et al., 2014). However, encryption techniques require protocol change and computational overhead. Another option is symmetric encryption methods; they are more efficient in encrypting data (Pan et al., 2012). Some researches proposed use of symmetric key encryption techniques (e.g. advanced encryption standard (AES)) and public key infrastructure (PKI) for securing communications. Nevertheless, there is a need for key

exchange mechanisms. When we think about the number of aircraft and ground stations and resource constraints on avionics devices, it is a significant problem. Given these requirements, these techniques may not be feasible for real-time systems like ADS-B. These factors should be carefully considered when implementing encryption solutions for ADS-B security.

3.2.2. Location-Based Methods

Location-based methods use geographical information to verify the legitimacy of ADS-B messages and the transmitted data by comparing the reported location and expected positions. For example, multilateration technology based on the signal arrival time difference (TDOA) is a common technique (Baek et al., 2017; Y. Kim et al., 2017; Kaune et al., 2012). There are numerous techniques, including distance bounds, group authentication, and kalman filtering (Johnson et al., 2012). However, these techniques often suffer from significant errors because of the line-of-sight limitation. Their accuracy and reliability are impacted by a number of issues. Due to environmental factors such as noise and disturbances, location data accuracy cannot be guaranteed (K.-W. Kim et al., 2012; Ma et al., 2021). Therefore, these systems generally have lower accuracy compared to GPS technology (Ma et al., 2021). Moreover, they require additional infrastructure and protocol changes.

3.2.3. Machine Learning-Based Methods

Machine learning-based IDS concentrates on the deviations of flight behavior or pattern to detect anomalies/intrusions in ADS-B data. With advances in machine learning and deep learning technology, models' accuracy rate and efficiency have increased; therefore, an ML-based system is accepted as suitable for critical systems. With these advances, the number of studies in this area has increased in recent years (Cevik & Akleyek, 2024b). The literature on ADS-B data focuses on two basic feature sets: route information and unique properties of electromagnetic waves emitted by ADS-B. These features establish a relationship between the aircraft and the route, and if the aircraft deviates from the route, this move is called an anomaly. Electromagnetic features containing more specific information than route information were extracted and used as radio frequency (RF) fingerprints, and aircraft were classified according to

these features. A database that identifies unique aircraft with international civil aviation organisation (ICAO) numbers must be created to determine whether the aircraft is real or fake.

(Leonardi et al., 2017) and (Manesh, Velashani, et al., 2019) studies explore using neural networks to detect anomaly/intrusion in the ADS-B messages. (Leonardi et al., 2017) proposes a seven-step algorithm for pattern extraction, considering the limits and parameters of ICAO standards, which allow manufacturers to develop devices with system parameters within specific ranges. On the other hand, In (Manesh, Velashani, et al., 2019) used features such as error rate, bad packet ratio, and energy statistic to distinguish received messages. They present comprehensive research that is conducted on machine learning-based classifiers for detecting jamming attacks, using various supervised machine learning algorithms such as support vector machine (SVM), k-nearest neighbor (KNN), artificial neural network (ANN), and decision tree (DT). The best performance was obtained by the neural network with accuracy, detection probability, and false alarm probability of 81%, 90.3%, and 30.9%, respectively. In (Chen et al., 2019), the authors collect signal data from the aircraft communications addressing and reporting system (ACARS) system and create a large dataset using convolutional neural network (CNN), recurrent neural network (RNN), and long short term memory (LSTM). They outperform methods like SVM and logistic regression (LR) in areas such as converting signal data to images and classification using image processing. The dataset size is limited, but the study's results show that ACARS and ADS-B have signal classification accuracy of 98.1% and 96.3%, respectively. When the signal-to-noise ratio exceeds 9 dB, the classification accuracy is more than 92%. The transfer learning experiment shows that models trained on large-scale ADS-B datasets are more suitable for learning and training new tasks than those trained on a small-scale dataset. In (Leonardi & Gerardi, 2020), the authors focused on the features of RF to extract signatures of the aircraft and develop an intrusion detection algorithm. They evaluated the system against different attack scenarios: replacing a part in the in-flight navigation system and adding an incorrect message by an unauthorized transmitter. An aircraft signature database was created based on RF signal characteristics, which allowed for matching signatures with aircraft. The KNN algorithm was evaluated using different frames and

k values, with the best value being 1000 and a 75% accuracy rate and a 10^{-3} false alarm rate. (Manesh, Kenney, et al., 2019), the authors propose a supervised machine learning method based on artificial neural networks to detect GPS spoofing signals. This method does not require modifications to the GPS infrastructure and can be easily applied to any GPS-guided autonomous system using microcontrollers or microcomputers. As a result of the study, the highest accuracy rate of the neural networks is 98.3%.

In route-based anomaly detection systems, Habler et al. utilized an LSTM encoder-decoder algorithm to model flight routes using ADS-B messages, aiming to detect aircraft deviations from the flight path (Habler & Shabtai, 2018). The authors extracted four main geographical location points as a feature by using flight information for a specific route: take-off, the first point of cruise behavior, the endpoint of cruise behavior, and landing. Moreover, they generated different types of anomalies/attacks for the training/test datasets. Five common anomaly/attack detection algorithms, namely gaussian mixture model-hidden markov model (GMM-HMM), DBSTREAM, one class SVM (OC-SVM), local outlier factor (LOF), and isolation forest, were applied to the dataset. The established models detected produced anomalies/attacks with an average false alarm rate of 4.5%. According to the study results, the LSTM encoder-decoder algorithm performed the best. Afterward, in 2022, they presented a new study on a Stack LSTM encoder-decoder architecture, utilizing data from the OpenSky Network (Habler & Shabtai, 2022). The research introduced separate models for each phase of flight, including climb, cruise, and descent, to enhance the robustness of trajectory anomaly detection in air traffic surveillance. Since the results of the LSTM for time series are impressive, researchers are intensely concentrated on the LSTM algorithm. Chevlot et al. developed an LSTM contextual autoencoder (CAE) for trajectory anomaly detection using data from the OpenSky network (Chevrot, 2022). The study analyzed various features such as altitude, consecutive delta, tracking delta, vertical rate, ground speed, and flight phases. Research on anomaly detection systems for aircraft trajectories improves anomaly identification in multivariate time series. Wang et al. introduced an LSTM-based detection approach. Different threshold values were chosen for anomaly detection, and the proposed model achieved a precision of 0.85 for ten different attack types. However, attack-based accuracy rates and attack details were not published.

Teng Yao et al. proposed a system based on the hidden markov model to detect different attack patterns in ADS-B data (Li et al., 2020). The system dynamically obtains parameters via the sticky hierarchical dirichlet process (sHDP) to provide dynamic intrusion detection. The intrusion detection model has three steps: data preparation, sHDP-HMM, and intrusion detection. The authors modeled six distinct attacks: fixed deviation, random deviation, incremental deviation, flight change, replay attack, and denial-of-service (DoS) attack. sHDP-HMM had the best accuracy in random deviation and incremental deviation attacks, while LSTM had the best in fixed deviation, flight change, and DoS attacks. The sHDP-HMM model has the highest accuracy rates in random and incremental deviation attacks. Route-based research additionally include multi-layered filtering methods. For example, in (Ying et al., 2019), the SODA system has two layer classification mechanism: a message classifier to classify incoming messages as normal or attack, and then estimates the ICAO address based on physical layer properties against replay attacks. The system has a minimal false alarm rate (0.43%) in the first step and can detect ground-based spoofing attacks with a probability of 99.34%. The second step classifies 238 aircraft with an accuracy rate of 96.66%. The DNN-based method outperforms other machine learning techniques like XGBoost, logistic regression, and support vector machine. On the other hand, in (Olive & Basora, 2020), the authors focused on unmalicious anomalies during flights and used a density-based spatial clustering of applications with noise (DBSCAN) algorithm was used to classify routes, while automatic encoders were used to detect anomalies. The majority of the anomalies observed in these datasets have been shown to be caused by either the weather or the ATCs' movement tactics.

The capacity of the system to identify threats by analyzing real-time data is the challenge of the IDS of the ADS-B system. Therefore, Mink et al. focused on this topic. Ten thousand false ADS-B messages were generated, and the same amount of real ADS-B messages collected from Opensky were combined to create a dataset (Mink et al., 2021). As a result of this study, the system required 11802.38 seconds to apply the model for prediction and 0.36 seconds for the server to preprocess the messages. Moreover, during the training and testing phases, the SVM classification algorithm produced results with 80% precision, 78% recall, and 79% F1 score values. The system demonstrated real-time intrusion detection capacity by processing about 27

ADS-B messages per millisecond for intrusion detection.

Both route-based and signal-based methods have advantages and disadvantages. In the route-based approach, gathering historical flight data on the relevant route is necessary to create a model. This information can be gathered for several routes from open databases without establishing an infrastructure and can be utilized to create a model. However, finding a dataset that concentrates on the characteristics of electromagnetic waves is challenging. A testbed is required to obtain this data. These characteristics must be recorded in a shared dataset and utilized to identify the aircraft. In order to apply this method, it is essential to avoid changing hardware, software, or aircraft equipment too often because any change will cause a pattern change and require updating the database. Therefore, we concentrate on the route-based anomaly/intrusion detection system (Cevik & Akleyek, 2024b).

Even if machine learning systems are suitable options to use as a solution in ADS-B security, there are gaps: real-time detection capabilities, scalability, and the ability to handle complex attack scenarios. This thesis addresses these gaps by introducing a novel approach that combines multiple machine learning models and handles complex attack scenarios.

3.3. Comparison of ADS-B Security Solutions

ADS-B security solutions enhance the cybersecurity of ADS-B systems, each addressing specific aspects of security concerns in transmitting ADS-B messages. The comparisons of different security approaches are presented in Table 3.3 (Johnson et al., 2012).

Each of these solutions has advantages and disadvantages. Encryption-based methods guarantee confidentiality and integrity; however, they cause latency and computational overhead and require protocol modification. Besides, the ADS-B system has no constraints on the confidentiality of the message. Considering this situation, the computational overhead is not acceptable for real-time systems. Location-based solutions can detect spoofing attacks without changing the message format. However, they require additional information and infrastructure, such as sensors. This additional infrastructure necessitates an update to the ADS-B system.

Table 3.3. Comparison of ADS-B security solutions

Methods	Advantages	Disadvantages
Encryption-based	· Provide confidentiality and integrity	· Computational overhead · Latency · Protocol change
Location-based	· No need for encryption · No need for changing Message format · Robustness to spoofing	· Infrastructure requirements · Detection accuracy
ML-based	· No need for encryption · No need for changing Message format · High detection rates · Adaptability	· Complexity · Resource intensive

Considering that all civil aircraft have the ADS-B Out system, this will be a significant problem. Machine learning-based approaches provide high detection rates and adaptability but are complex. Consequently, among these solutions, machine learning-based methods are the most effective, considering the specific requirements and constraints of the ADS-B system. Thus, we focus on these methods and aimed to improve their complex structures.

4. ATTACK DATA GENERATION

The generation of attack data is a crucial process in understanding and mitigating threats against ADS-B systems. ADS-B is a vital component in modern aviation, providing real-time aircraft position data. However, it is also susceptible to various types of cyberattacks. This chapter covers the key aspects involved in generating attack data for ADS-B, including scenarios, methodology, attacker capabilities, data generation, and dataset details. The methodology for generating attack data is detailed. This includes the step-by-step processes and tools used to simulate attacks on ADS-B systems.

This chapter is organized as follow: The Section 4.1 outlines various attack scenarios targeting ADS-B systems. It defines potential threats such as spoofing, jamming, and data injection attacks, illustrating how adversaries can exploit vulnerabilities in ADS-B communication to disrupt aviation safety and efficiency. Section 4.2 emphasizes the systematic approach taken to ensure that the generated data is representative of real-world attack scenarios. In Section 4.3, we demonstrate attacker capabilities such as eavesdrop, message injection and deletion. In Section 4.4, the focus is on the practical aspects of creating attack data. It describes the techniques employed to generate realistic attack datasets, including the use of simulation environments and real-world data to enhance the authenticity of the attack scenarios. Section 4.5 provides an in-depth look at the attack datasets themselves. It includes descriptions of the data structure, the types of attacks represented, and the key features of the datasets. Additionally, it explains how these datasets can be used for further research and development of mitigation strategies.

4.1. Attack Scenarios on ADS-B

Numerous classification methods have been proposed in the literature to categorize attack scenarios against ADS-B systems. These methods often rely on various criteria, such as the nature of the attack, its impact on system integrity or availability, the attacker's objectives, and the techniques employed. In this thesis, we concentrate on ten distinct attack scenarios defined below:

1. Aircraft reconnaissance is a passive attack scenario known as eavesdropping. It is the basic step for other attack scenarios. The attacker collects actual ADS-B data and other flight-related information to analyze the data and prepare for the next attack step. Considering the open databases described in the previous section, it is an easy step for the attacker to perform. Since the attack is passive, there is no impact on the operation.
2. Aircraft target ghost inject is an active attack scenario. The attacker aims to display injected ghost aircraft on the cockpit display screen. In this scenario, the attacker has to use an ADS-B Out device to broadcast false data within the range of the target's ADS-B In device. This attack requires an ADS-B Out device and logical attack data generation. Since the target is moving, the difficulty of the scenario is medium-high. If the injected ghost aircraft is close to the target aircraft, the pilot of the target aircraft may be required to make unnecessary maneuvers. Moreover, since the ADS-B system is connected to the TCAS, the system can give a false alarm.
3. Ground station target ghost injection is an active attack scenario, like the aircraft target ghost injection attack. This attack requires an ADS-B Out device and logical attack data generation. The attacker aims to broadcast false data to the target ground station. Therefore, it is easier than an aircraft target attack (Pantoja Viveros, 2024).
4. Ground station multiple ghost inject is an active attack scenario. The aim of the scenario can overload the ground station system due to the limits of capacity (Pantoja Viveros, 2024). This attack requires an ADS-B Out device and logical attack data generation. According to (Pantoja Viveros, 2024), with the current surveillance system structure, multiple ghost aircraft injection attacks can be easily done in ADS-B receivers. However, it cannot be done as easily in PSR, SSR, or MLAT receivers (Pantoja Viveros, 2024).
5. A replay attack is an active attack scenario. In this scenario, the attacker can easily collect historical ADS-B data of a specific route, thanks to flight tracking sites, and broadcast them on different days. This attack requires an ADS-B Out device. The attacker's target could be a ground station or an aircraft. Although it

is easier than other attacks because there is no attack data generation phase, the effect of the attack is the same (Pantoja Viveros, 2024).

6. Aircraft spoofing is an active attack scenario that requires a combination of message injection and deletion ability. Target of the attack can be both aircraft and ground stations. This attack requires an ADS-B Out device and logical attack data generation. Since the attackers can combine multiple techniques and inject undetectable messages between normal ADS-B messages, the technical difficulty of the attack is medium (Pantoja Viveros, 2024).
7. Virtual trajectory modification is a combination of other attack scenarios. For example, firstly, the attacker eavesdrops on ADS-B messages and injects messages into the channel using spoofing techniques. Besides this method, an attacker can also change the trajectory by deleting messages with jamming. However, there is no restriction that the messages must be logical and cannot be detected. Target of the attack can be both aircraft and ground stations. According to these scenarios, an attacker can combine multiple techniques and inject or delete messages. Therefore, the technical difficulty of the attack is medium (Pantoja Viveros, 2024).
8. Aircraft disappearance is an active attack scenario. The target of the attack is both the ground station and the aircraft. Attackers delete ADS-B messages of related aircraft using jamming techniques. Since jamming is a signal-based attack, this attack scenario was not focused on during the attack data generation phase.
9. Aircraft flood denial is an active attack targeting aircraft. An attacker aims to cause a denial of service of an aircraft ADS-B system by jamming that disrupts ADS-B channel or GPS signals. Since jamming is a signal-based attack, this attack scenario was not focused on during the attack data generation phase.
10. Ground station flood denial is an active attack targeting ground stations. The attacker aims to cause a denial of service to the ground station's ADS-B system by jamming, which disrupts the ADS-B channel or GPS signals. Since jamming is a signal-based attack, this attack scenario was not focused on during the attack data generation phase.

Table 4.1. Severity of Impact for Various ADS-B Attacks

Attack Type	Average Severity (out of 5)
Aircraft target ghost inject	3.5
Ground station target ghost inject	4.0
Ground station multiple ghost inject	3.0
Replay attack	4.5
Aircraft spoofing	5.0
Virtual trajectory modification	3.5
Aircraft disappearance	4.0
Aircraft flood denial	3.0
Ground station flood denial	3.0

Table 4.1 summarizes the average impact severity of various ADS-B attacks according to aviation experts and clearly outlines potential threats to air traffic management systems. The severity of these attacks varies from moderate to critical. In particular, the aircraft spoofing attack, in which fake aircraft are included in the system, has been determined as the most dangerous type of attack, with an average severity of 5 out of 5. Such attacks pose a significant threat as they can potentially cause serious disruptions in airspace operations. In addition, replay attacks are among the high-risk attacks with a severity value of 4.5.

Other notable attacks include ground station target ghost injection, in which fake targets are introduced to ground stations, and aircraft disappearance attacks, in which real aircraft are removed from the air traffic controller system or other aircraft. Both attacks, with a severity value of 4.0, pose potential hazards that could cause severe disruptions in air traffic control.

The table highlights the importance of robust anomaly detection systems to mitigate these threats. Severity ratings highlight the importance of research and development on ADS-B security. These issues are addressed in (Pantoja Viveros, 2024). This table clearly shows how critical the elimination of vulnerabilities associated with ADS-B systems is in maintaining the security and integrity of air traffic management.

There exist various ways to simulate realistic attack data. For instance, one can modify flight data of previous attacks such that the aircraft positions gradually or randomly change their latitude, longitude, velocity, or altitude. One can also simulate

patterns of previous flights or even merge data from two flights to result in complex attack scenarios. Verification and validation of this generated attack data is very crucial. This may be done by validating synthetic data against known ground accurate data, attack scenario tests, and consulting domain experts to ensure the possibility of the attack scenarios. According to (Pantoja Viveros, 2024), the different attack scenarios based on their impact, likelihood, and overall risk are compared in Table 4.2.

Table 4.2. The comparison of ADS-B attack scenarios

Attack Type	Impact	Likelihood	Risk	Assets
Aircraft Spoofing	High	Med.	High	I, A
Aircraft Target Ghost Inject	High	Low	Med.	C, I, A
Ground Station Target Ghost Inject	Med.	Med.	Med.	I
Ground Station Multiple Ghost Inject	High	Med.	High	I, A
Replay Attack	High	Med.	High	I, A
Virtual Trajectory Modification	High	Med.	High	I
Aircraft Disappearance	High	Low	Med.	I, A
Aircraft Flood Denial	Med.	High	High	A
Ground Station Flood Denial	Med.	High	High	A

4.2. Methodology

The generation of attack vectors and data is crucial for evaluating the effectiveness of the IDS in detecting malicious activities within ADS-B systems. This section outlines the methods used to create realistic attack scenarios that simulate potential threats to ADS-B. Our attack model is given in Figure 4.1.

ADS-B messages are transmitted in an unencrypted form, and the receiver does not authenticate the message source. Therefore, an attacker can easily listen to the channel and inject messages into an ADS-B packet to spoof the receiver.

In the context of attacks against ADS-B systems, there are two main types: air-to-air and ground-to-air attacks. In ground-to-air attacks, attacks come from ground-based sources to target aircraft. The attacker can inject or delete messages in the target aircraft's ADS-B system. For example, injected false messages can create ghost aircraft that seem like real aircraft in the radar system, potentially causing hazardous situations and disrupting normal flight operations. Attack vectors have been created for attacks against the ground station, as shown in Figure 4.2.

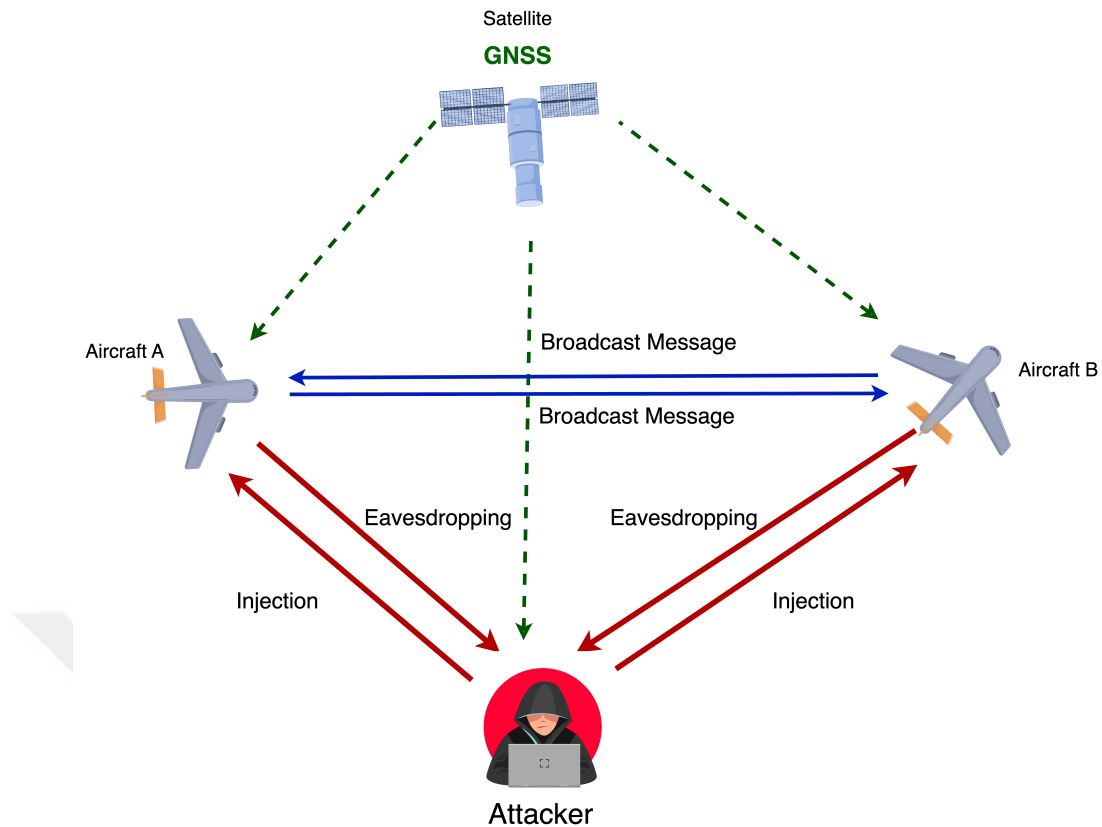


Figure 4.1. Threat model

The Figure 4.2 shows an attack tree related to "ground station attacks" on any aviation or communication system. Each node describes a kind of attack or sub-attack: this depicts the path through which specific malicious goals can be realised.

In flood denial attack, the attacker first eavesdrops on the communication to gather information. After this, they proceed to jam the communication channel, disrupting the service. In the case of virtual trajectory modification, the attacker begins by eavesdropping, then moves on to message injection, where false messages are introduced into the system. This can lead to either a replay attack, where previously captured valid messages are replayed to confuse the system, or the generation of attack data, where the attacker creates new data to further their goals. Multiple ghost injection follows a similar pattern. The attacker starts by eavesdropping and then injects messages. This can result in a replay attack or message modification, both of which lead to the generation of attack data. For target ghost injection, the initial steps are the same: eavesdropping followed by message injection. The subsequent actions can be a replay attack or message modification, both leading to generating attack data. Aircraft disappearance also starts with eavesdropping. The

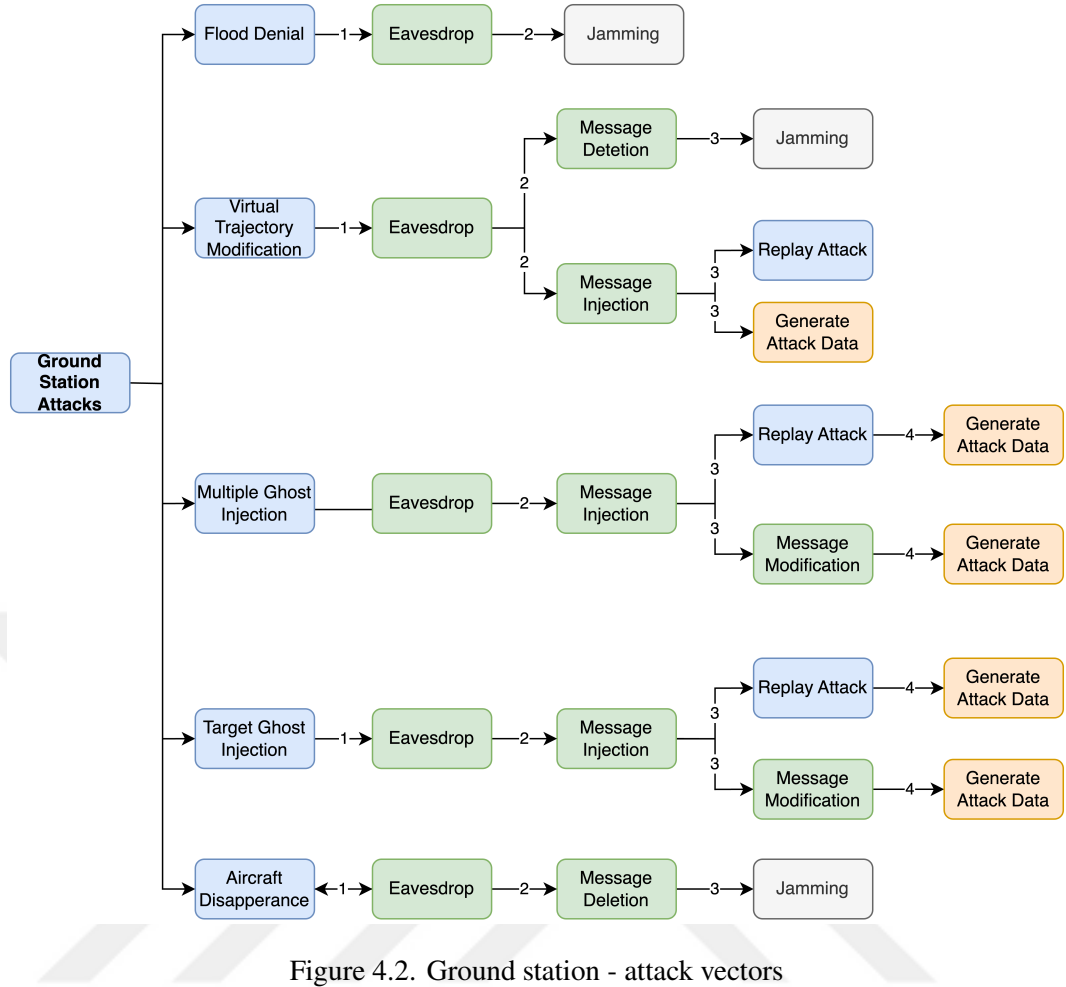


Figure 4.2. Ground station - attack vectors

next step is message deletion, which is followed by jamming, disrupting communication to make the aircraft appear as if it has disappeared.

Ground station attacks and aircraft attacks differ primarily in their origins and the specific scenarios they involve. Ground station attacks originate from ground stations and focus on disrupting or manipulating communications between ground stations and aircraft. On the other hand, aircraft attacks originate from the aircraft, focusing on manipulating the data it transmits, such as in aircraft spoofing, where the aircraft pretends to be another. Attack methods in ground station attacks often involve more complex data manipulation directly affecting the ground station. Aircraft attacks center around deception and spoofing tactics initiated by the aircraft. In contrast to ground-based spoofing attacks, possible aircraft attacks are still insufficiently investigated despite being an actual threat. Drones, or UAVs, can be used to carry out these attacks, so more caution has to be taken while considering this kind of possible threat. Considering the above attack scenarios, attack vectors have been created for

attacks against the aircraft, as shown in Figure 4.3.

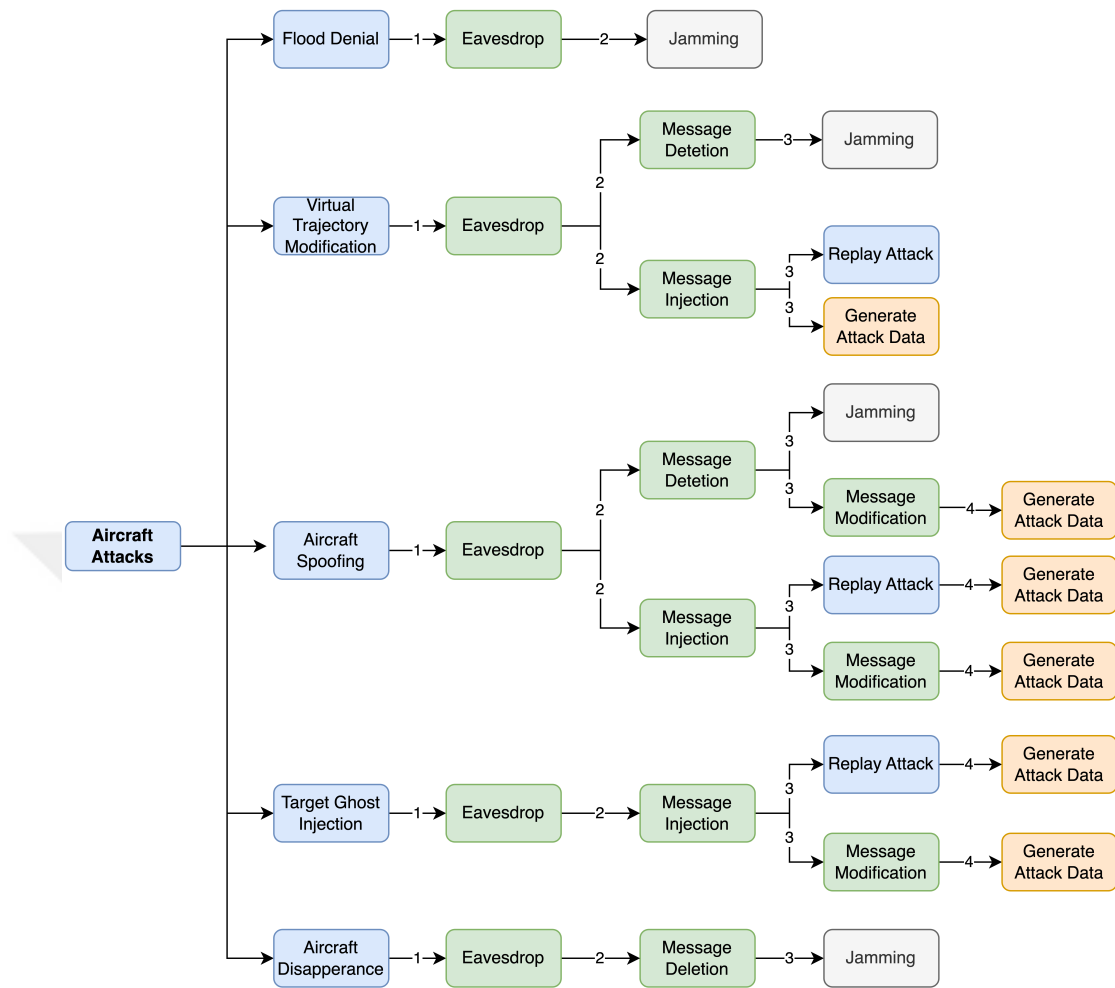


Figure 4.3. Aircraft - attack vectors

Figure 4.3 shows an attack tree related to "aircraft attacks" on any aviation or communication system. Each node describes a kind of attack or sub-attack: this depicts the path through which specific malicious goals can be realized.

In flood denial attacks, virtual trajectory modification, target ghost injection, and aircraft disappearance attacks have the same path as ground station attacks. One difference is that the attack's source is the aircraft system, not the ground system. Only aircraft spoofing attacks are different than other attack types. In this attack scenario, the attacker starts by eavesdropping and then moves to message injection. The subsequent actions can include message deletion, leading to jamming, or message modification, resulting in the generation of attack data. Another potential path is performing a replay attack or further message modification, leading to generating attack data.

4.3. Attacker Capabilities

This section provides a clear understanding of each attack scenario's attacker capabilities, so these can be applied toward compromising ADS-B systems and the importance of systematically generating and validating attack data. Ultimately, the goal is to improve the robustness of anomaly detection systems.

In our threat model, there are two approaches to attack scenarios according to attacker capability: active and passive. Passive attack involves eavesdropping and analysis of ADS-B messages without any alteration. For example, aircraft reconnaissance, where the attacker listens to the ADS-B transmissions for obtaining the knowledge on aircraft position, routes, and other information. This doesn't affect the system directly, but it could give much intelligence for other, more malicious activities. Active attacks, on the other hand, are more direct. These usually have a high degree of interaction with the ADS-B system regarding changing, injecting, or deleting messages. They can be further elaborated based on the capabilities of the attacker. Another method attackers use to delete messages is jamming. They can use jamming devices to block ADS-B signals and prevent other aircraft or ground stations from receiving valid messages. However, since this is a signal-based blocking method, this thesis does not cover it. Within the scope of this thesis, we focus on scenarios where the attacker can inject and modify messages.

4.3.1. Eavesdrop

Eavesdropping is accepted as passive listening to the ADS-B transmissions without interfering with the communication process. Attackers make use of software defined radios to capture these unencrypted broadcasts. In addition, they also use open-access websites, like OpenSky Network and FlightRadar24, to collect ADS-B data. Analyzing these messages gives attackers information on aircraft identification, position, altitude, velocity, and other flight parameters. This capability is the first point of many attacks because it enables attackers to understand regular operations and acquire the data that can be used in more advanced types of attacks. Application scenarios for eavesdropping are given in Table 4.3.

Table 4.3. Application in scenarios for eavesdropping

Attack Scenario	Application
Aircraft Reconnaissance	An attacker collects information without detection.
Virtual Trajectory Modification	An attacker gathers initial data to modify later.
Aircraft Spoofing	An attacker collects legitimate aircraft data to impersonate.
Aircraft Disappearance	An attacker identifies target aircraft.
Flood Denial Attacks	An attacker understands system capacity and typical traffic.
Ghost Injection Attacks	An attacker gathers baseline data to create believable ghost aircraft.
Replay Attack	An attacker captures valid messages for later use.

4.3.2. Message Deletion

Message deletion is a process in which ADS-B messages are blocked or dropped from the communication channel so that the particular message cannot be delivered. This can be done using signal jamming techniques that prevent the ADS-B signals from being transmitted, effectively making an aircraft invisible to the radars of the air traffic control and other aircraft. Application scenarios for message deletion are given in Table 4.4.

Table 4.4. Application in scenarios for message deletion

Attack Scenario	Application
Aircraft Disappearance	An attacker uses jamming to prevent messages from reaching the receiver, making the aircraft vanish from tracking systems.
Ground Station Attacks	An attacker disrupts communications to specific ground stations to remove crucial tracking data.

4.3.3. Message Injection

Message injection includes the following capabilities: The fabrication and transmission of false ADS-B messages that could add wrong data to the system, such as identifying false aircraft, position, altitude, and velocity. This allows the attacker to inject ghost aircraft, change trajectories, and generally deceive ATC and other aircraft

systems. Application scenarios for message injection are given in Table 4.5.

Table 4.5. Application in scenarios for message injection

Attack Scenario	Application
Aircraft Target Ghost Injection	An attacker creates non-existent aircraft at specific locations.
Ground Station Target Ghost Injection	An attacker aims false messages at specific ground stations.
Ground Station Multiple Ghost Injection	An attacker overloads a ground station with numerous ghost aircraft.
Virtual Trajectory Modification	An attacker continuously injects modified position messages to alter the reported path of an aircraft.
Aircraft Spoofing	An attacker injects false messages that appear to come from legitimate aircraft.
Flood Denial Attacks	An attacker overwhelms systems with an excessive number of false messages.

4.3.4. Replay Attack

In the replay attack, an attacker captures valid ADS-B messages and replays them back onto the network at a different time. This reintroduces old information into tracking systems, which would cause confusion and inconsistency for ATC or pilot. The weakness of replay attacks is due to the obviousness and simplicity of capturing ADS-B messages and then retransmitting them without being detected since ADS-B messages are neither encrypted nor authenticated. Application scenarios for replay attack are given in Table 4.6.

Table 4.6. Application in scenarios for replay attack

Attack Scenario	Application
Replay Attack	An attacker directly uses captured messages to cause confusion about aircraft positions.
Virtual Trajectory Modification	An attacker combines replayed messages with modified data to enhance deception.
Ghost Injection Attacks	An attacker reintroduces old messages to create ghost aircraft based on previous real flights.

4.3.5. Summary of Attacker Capabilities

It is essential that the attacker's capabilities, such as eavesdropping, message deletion, message injection, and replay attacks, are understood to produce realistic attack data. These capabilities highlight the vulnerability of the ADS-B system and clearly emphasize the need to develop robust anomaly detection systems to mitigate accompanying risks. By generating consistent and validated attack data, one can provide a relevantly sensitive anomaly detection system with air traffic safety improvement information. The summary of capabilities for each scenarios is given in Table 4.7.

Table 4.7. Summary of capabilities in each scenario

Attack Scenario	Eavesdrop	Message Deletion	Message Injection	Replay Attack
Aircraft Reconnaissance	Yes	No	No	No
Aircraft Target Ghost Injection	Yes	No	Yes	No
Ground Station Target Ghost Injection	Yes	No	Yes	No
Ground Station Multiple Ghost Injection	Yes	No	Yes	No
Replay Attack	Yes	No	No	Yes
Aircraft Spoofing	Yes	No	Yes	No
Virtual Trajectory Modification	Yes	No	Yes	Yes
Aircraft Disappearance	Yes	Yes	No	No
Aircraft Flood Denial	Yes	No	Yes	No
Ground Station Flood Denial	Yes	No	Yes	No

4.4. Generation of Attack Data for ADS-B

In this thesis, four different attack generation strategy were simulated: 1. Constant change attack, 2. Random change attack, 3. Merge attack, 4. Replay attack. In the context of the constant change attack, new route information was generated by adjusting the latitude, longitude, altitude, and velocity parameters to predefined constant values. In the case of the random change attack scenario, alterations in latitude, longitude, altitude, and velocity occurred randomly at specified intervals as outlined in Table 4.8.

Table 4.8 enables the exercise of a wide range of attack data that looks very much like real ADS-B data. Through careful parameter adjustments and combinations, different approaches can be used to make varied and sophisticated attack scenarios. To make our attack data less easily detectable and closer to real ADS-B data, we should

Table 4.8. Suitable increment ranges for ADS-B parameters

Parameter	Suitable Increment Range
Velocity (knots)	1 to 5
Latitude (degrees)	0.0001 to 0.001
Longitude (degrees)	0.0001 to 0.001
Baroaltitude (feet)	10 to 100
Geoaltitude (feet)	10 to 100

choose increment/decrement values that mimic realistic changes in velocity, latitude, longitude, and altitude. This method will ensure that the constructed synthetic data is as believable as possible, thus reducing the chances that anomaly detection systems can easily distinguish it from genuine data. To generate attack data, we mainly focus on two strategies: modified previous real flights and replay attacks. The procedures are outlined by the attack tree in the Figure 4.4, which makes it intuitive to understand the different approaches and techniques applied to create realistic attack data.

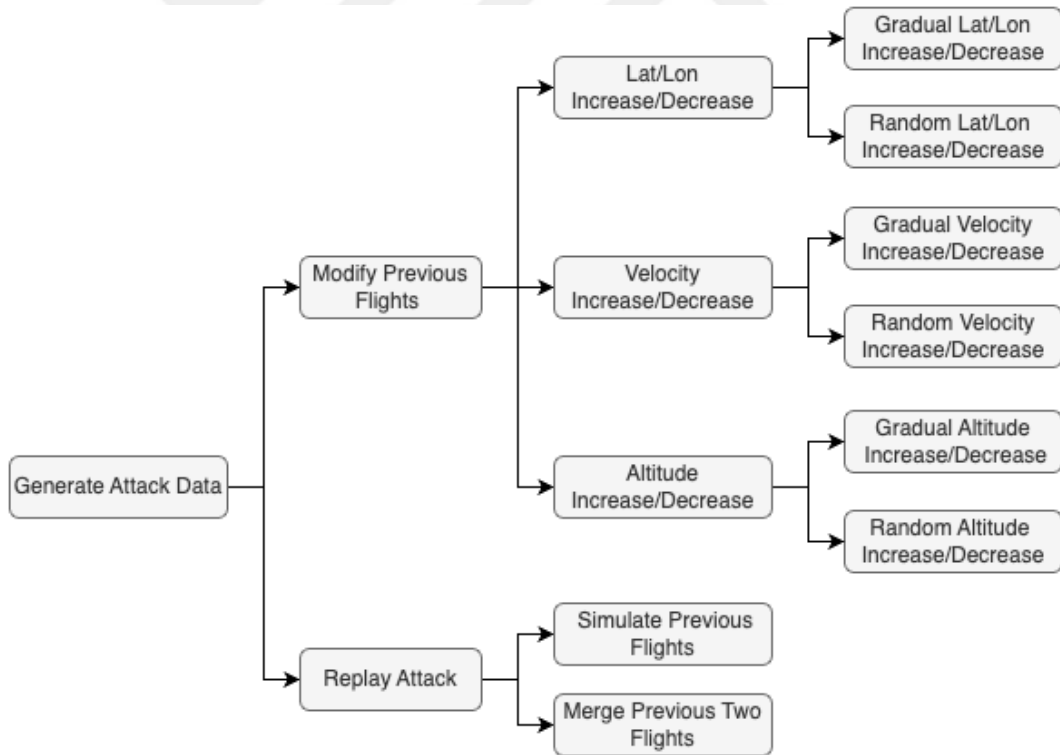


Figure 4.4. Attack generation phase

The first technique for adaption is based on past flights. It means making some changes to previous flights of the data to have simulations that indicate different attack scenarios. The main areas in which changes have been made are latitude, longitude,

velocity, and altitude. Defining appropriate range for random and gradual change attack scenarios against ADS-B systems requires a detailed understanding of various operational and technical factors. One critical factor to consider is the typical margins of error and tolerances of ADS-B receivers and associated flight management systems. Changes to ADS-B data must initially be small enough to fall within these tolerances so that they may not be detected immediately. The accuracy of ADS-B data generally falls within a few meters for position and a few feet for altitude. For gradual change attacks, changing latitude and longitude by about 0.001 degrees (about 100 meters) per update cycle, or making increments of about 10 feet in altitude, can be effective in slowly moving the aircraft's reported position, which may not trigger an alarm.

Additionally, operational context plays an important role in defining these ranges. For example, larger differences in the cruise phase of flight may not be immediately detected compared to the approach phase. It is also important to consider existing detection mechanisms, such as ground-based radar verification and data fusion systems, because these systems can detect inconsistencies by cross-validating ADS-B data with radar data. This information helps adjust for gradual changes that go undetected over a long period of time.

For random change attacks, it is essential to understand the range of changes normally seen in ADS-B parameters due to operational changes (e.g., turbulence or course corrections). To ensure that random variations remain within a realistic range but are sufficiently irregular, it may be sufficient to make changes within ± 0.01 degrees (about 1 km) in latitude and longitude, within ± 100 feet in altitude, and within ± 10 knots in speed. These adjustments can introduce sufficiently unpredictable changes that can disrupt normal operations and cause misinterpretations.

In the case of latitude and longitude, changes can be done either gradually or randomly. A gradual shift is making numerous minor adjustments with time, simulating slow movements similar to an aircraft. Random changes are more detectable in nature and mimic the unpredictable movements of an aircraft. The algorithms for these attacks are provided in Algorithms 2 and 3.

Velocity changes may also change gradually and randomly. Gradual velocity changes present smooth acceleration or deceleration while random velocity changes

Algorithm 2 Random latitude/longitude attack

- 1: **Input:** ADS-B data D (a DataFrame), maximum latitude/longitude change L_{max}
 - 2: **Output:** $D_{spoofed}$ with manipulated latitude and longitude data
 - 3: Create a copy of D and store it in $D_{spoofed}$
 - 4: **for** each row r in $D_{spoofed}$ **do**
 - 5: Generate a random latitude change Δlat such that $-\Delta L_{max} \leq \Delta lat \leq \Delta L_{max}$
 - 6: Generate a random longitude change Δlon such that $-\Delta L_{max} \leq \Delta lon \leq \Delta L_{max}$
 - 7: Update the latitude in row r by adding Δlat
 - 8: Update the longitude in row r by adding Δlon
 - 9: **end for**
-

Algorithm 3 Gradual latitude/longitude attack

- 1: **Input:** ADS-B data D (a DataFrame), latitude/longitude increment L_{inc}
 - 2: **Output:** $D_{spoofed}$ with manipulated latitude and longitude data
 - 3: Create a copy of D and store it in $D_{spoofed}$
 - 4: **for** each row r in $D_{spoofed}$ **do**
 - 5: Increment the latitude in row r by L_{inc}
 - 6: Increment the longitude in row r by L_{inc}
 - 7: **end for**
-

result in a more erratic flight path. The algorithms for these attacks are provided in Algorithms 4 and 5.

Algorithm 4 Random velocity attack

- 1: **Input:** ADS-B data D (a DataFrame), maximum velocity change C_{max}
 - 2: **Output:** $D_{spoofed}$ with manipulated velocity data
 - 3: Create a copy of D and store it in $D_{spoofed}$
 - 4: **for** each row r in $D_{spoofed}$ **do**
 - 5: Generate a random velocity change Δv such that $-\Delta C_{max} \leq \Delta v \leq \Delta C_{max}$
 - 6: Update the velocity in row r by adding Δv
 - 7: **end for**
-

Algorithm 5 Gradual velocity attack

- 1: **Input:** ADS-B data D (a DataFrame), velocity increment V_{inc}
 - 2: **Output:** $D_{spoofed}$ with manipulated velocity data
 - 3: Create a copy of D and store it in $D_{spoofed}$
 - 4: **for** each row r in $D_{spoofed}$ **do**
 - 5: Generate a random velocity change Δv such that $\Delta v = \pm V_{inc}$
 - 6: Update the velocity in row r by adding Δv
 - 7: **end for**
-

Altitude may also change gradually and randomly. Gradual altitude changes are modeled to obtain realistic climb or descent patterns, while random altitude changes

model an abrupt change in the flight level. The algorithms for these attacks are provided in Algorithms 6 and 7.

Algorithm 6 Random altitude attack

- 1: **Input:** ADS-B data D (a DataFrame), maximum altitude change A_{max}
 - 2: **Output:** $D_{spoofed}$ with manipulated altitude data
 - 3: Create a copy of D and store it in $D_{spoofed}$
 - 4: **for** each row r in $D_{spoofed}$ **do**
 - 5: Generate a random altitude change Δa such that $-\Delta A_{max} \leq \Delta a \leq \Delta A_{max}$
 - 6: Update the barometric altitude in row r by adding Δa
 - 7: Update the geometric altitude in row r by adding Δa
 - 8: **end for**
-

Algorithm 7 Gradual altitude attack

- 1: **Input:** ADS-B data D (a DataFrame), altitude increment A_{inc}
 - 2: **Output:** $D_{spoofed}$ with manipulated altitude data
 - 3: Create a copy of D and store it in $D_{spoofed}$
 - 4: **for** each row r in $D_{spoofed}$ **do**
 - 5: Generate a random altitude change Δa such that $\Delta a = \pm A_{inc}$
 - 6: Update the barometric altitude in row r by adding Δa
 - 7: Update the geometric altitude in row r by adding Δa
 - 8: **end for**
-

The second one is replay attacks, where attackers can easily use previously recorded ADS-B messages and retransmit them to simulate the attack. It is given in Algorithm 8. This can be the total simulation of previous flights' data or a complex merging of two flights to create a new, highly complex flight scenario. One way is to do the playback of entire past flights so there are realistic attack scenarios.

Algorithm 8 Generate replay attack data

- 1: **Input:** DataFrame R (replay attack data)
 - 2: **Output:** Updated replay attack data R
 - 3: $R['time'] \leftarrow \text{pd.to_datetime}(R['time'], \text{unit} = 's')$
 - 4: $R['time'] \leftarrow R['time'] + \text{pd.DateOffset}(\text{years} = 1)$
 - 5: $R['time'] \leftarrow R['time'].astype('int64') // 10^9$
 - 6: $R \leftarrow R.head(1200000)$
-

In a merge attack, two different previous flight paths and behaviors are combined to create a new and complex flight pattern. In this process, a flight frame is determined based on the length of both flights, and whether the coordinate points in this frame

intersect with the other flight is checked. Flights passing through the same or near coordinates are combined using this intersection point, thus creating a new flight path with the characteristics of two different flight paths. The merge attack algorithm is given in Algorithm 9.

Algorithm 9 Merge attack data

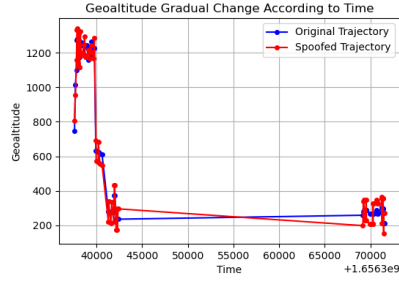
```

1: Input: DataFrames  $D_1$  and  $D_2$ , number of frames  $n$ 
2: Output: Attack data  $A$ 
3: Create an empty DataFrame  $A$  with the same columns as  $D_1$ 
4: for  $i = 1$  to  $n$  do
5:   Chosen frame length  $L_1$  from  $[0, 30]$ 
6:   Chosen start index  $s_1$  from  $[0, \text{len}(D_1) - L_1]$ 
7:   Compute end index  $e_1 = s_1 + L_1$ 
8:   Extract frame  $S_1$  from  $D_1[s_1 : e_1]$ 
9:   Chosen frame length  $L_2$  from  $[0, 30]$ 
10:  Chosen start index  $s_2$  from  $[0, \text{len}(D_2) - L_2]$ 
11:  Compute end index  $e_2 = s_2 + L_2$ 
12:  Extract frame  $S_2$  from  $D_2[s_2 : e_2]$ 
13:  Concatenate  $S_1$  and  $S_2$  to form  $C$ 
14:  if  $A$  is empty or all-NA then
15:     $A \leftarrow C$ 
16:  else if  $C$  is empty or all-NA then
17:     $A \leftarrow C$ 
18:  else
19:    Concatenate  $A$  and  $C$  to form  $A$ 
20:  end if
21: end for

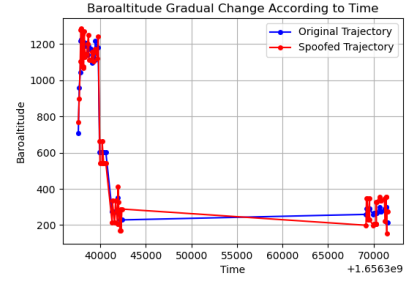
```

In summary, according to Figure 4.4, we generate attack data with eight different way: random increase/decrease altitude, gradual increase/decrease altitude, random increase/decrease velocity, gradual increase/decrease velocity, random increase/decrease latitude/longitude, gradual increase/decrease latitude/longitude, merge previous two flights, simulate previous flight. In Figure 4.5, a comparison of attack and normal flights is given.

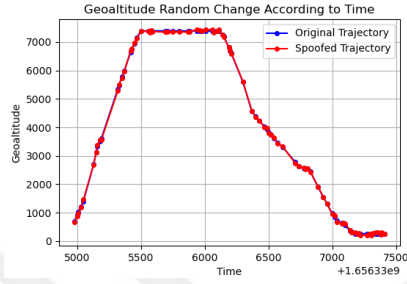
The Figure 4.5 includes ten sub-figures indicated from a to j, which describe the effect of various kinds of changes in the flight data. These graphs compare the original trajectory (in blue) against the spoofed trajectory (in red) to depict the shift that various methods of attack generation lead to in-flight parameters. Each sub-figure allows a particular type of change.



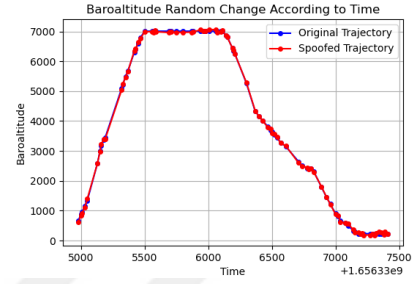
(a) Geoaltitude gradual change



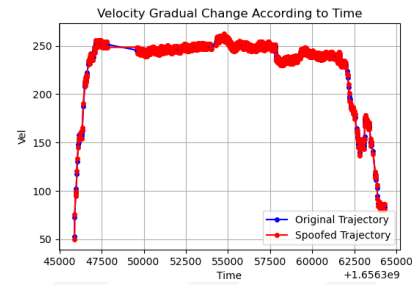
(b) Geoaltitude gradual change



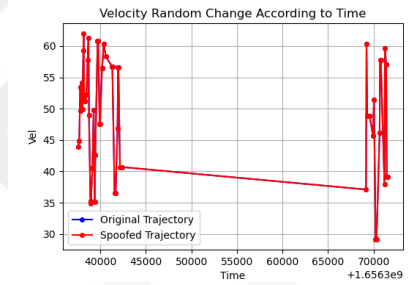
(c) Baroaltitude random change



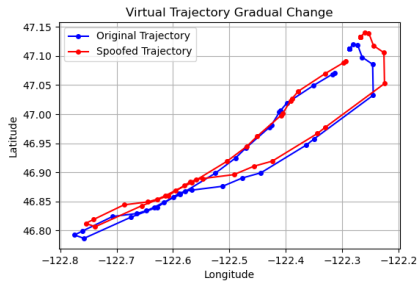
(d) Baroaltitude gradual change



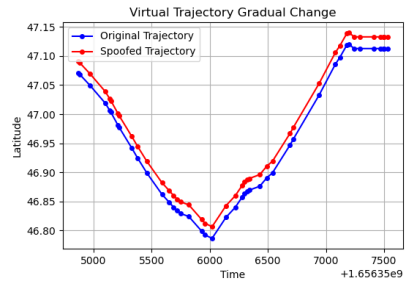
(e) Velocity gradual change



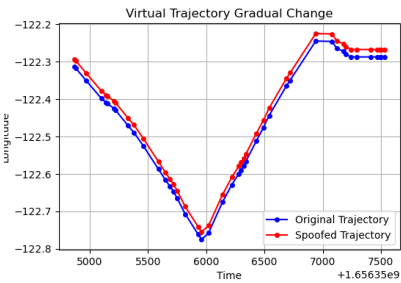
(f) Velocity random change



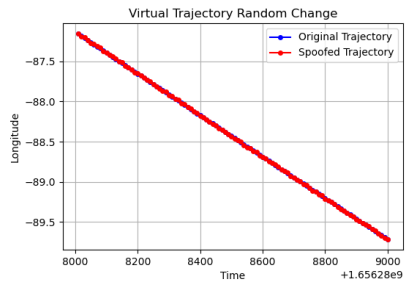
(g) Virtual trajectory gradual change



(h) Virtual trajectory gradual change



(i) Virtual trajectory gradual change



(j) Virtual trajectory random change

Figure 4.5. Comparison of original and spoofed trajectories

1. According to the Figure 4.5-(a), the original trajectory follows a specific profile of the altitudes, while the spoofed trajectory exhibits a smooth continuous adjustment mimicking realistic patterns of climb and descent.
2. In the Figure 4.5-(b), attack data has gradual changes in barometric altitude. Spoofed trajectory will follow a smooth adjustment, giving an appearance of realistic changes in altitude, as an aircraft ascends or descends.
3. In the Figure 4.5-(c), there are random changes in gealtitude values. In the genuine trajectory, the altitude profile is stable, whereas the altitude in the spoofed one exhibits sudden changes.
4. In the Figure 4.5-(d), there are random changes in baroaltitude. Spoofed trajectory shall introduce unexpected changes into the profile, entirely deviating from the original profile.
5. In the Figure 4.5-(e), there is a slow change of velocity over time. Original path moves at a constant speed whereas, in the spoofed path, it gradually speeds up or slows down over time.
6. In the Figure 4.5-(f), there are random and unexpected changes in velocity. Real trajectory has a smooth velocity, while the spoofed trajectory has sharp increase and decrease.
7. In the Figure 4.5-(g), there are gradual changes of the virtual trajectory with latitude and longitude, spoofed data within the intervals—smooth and continuous adjustment.
8. In the Figure 4.5-(h), another example of gradual changes in the virtual trajectory. Spoofed trajectory follows the original path closely with smooth adjustments.
9. In the Figure 4.5-(i), there are slow continuous changes in the virtual trajectory with more significant deviation. A spoofed trajectory presents a smooth but large angular change in direction.
10. In the Figure 4.5-(j), the virtual trajectory is changed randomly. The original trajectory is nearly linear, whereas the spoofed trajectory randomly varies in latitude and longitude.

These visualizations outline the diversity of changes applied to the flight data through the attack data generation phase. The original and spoofed trajectory compares the change in gealtitude, baroaltitude, velocity, and virtual trajectory, with both gradual and random change. The understanding of such modifications lies at the heart of devising realistic attack scenarios for testing and improving anomaly detection robustness in ADS-B data.

4.5. Details of Attack Datasets

Hybrid attack datasets are essential for ADS-B attack data generation because they provide a comprehensive and realistic framework when evaluating the resilience of ADS-B systems against various threats. This comprehensive combination of data helps create robust machine learning models and intrusion detection systems that recognize various attack vectors. Additionally, hybrid datasets increase the overall resilience and security of ADS-B communications in operational environments by ensuring that the attack scenarios generated are both realistic and diverse. This approach, which better reflects the diversity and complexity of security threats, is critical to securing ADS-B systems.

In this thesis, we generate the hybrid attack datasets that merge different attack types. We share our datasets in (Cevik, 2024). This section provides a general description of various attributes together with their respective values under four datasets or circumstances, labeled as 1, 2, 3, and 4. Data is measured in millions (m) or thousands (k) number of data points in Table 4.9.

Table 4.9 identifies what constitutes and characterizes each dataset and elaborates on the types of data modifications and attack simulations applied. Insights gained from such datasets will allow a derivation of the effects of the different changes and attack scenarios on the ADS-B data, which is greatly important for developing an effective anomaly detection system.

Table 4.9. Details of hybrid datasets

Attribute	1	2	3	4
Normal Data	30m	10m	6m	6m
Random Vel	2m	-	-	-
Gradual Vel	4m	2.5m	-	-
Random Alt	2m	-	-	-
Gradual Alt	4m	2.5m	-	-
Random Lat/Lon	2m	-	-	-
Gradual Lat/Lon	4m	2.5m	4m	-
Merge Attack	85.7k	-	85.7k	85.7k
Replay Attack	400k	-	-	400k
Labels	0: 30m 1: 18.5m	0: 10m 1: 7.5m	0: 6m 1: 4.1m	0: 2m 1: 485.7k

5. PROPOSED APPROACH: MACHINE LEARNING BASED ANOMALY/INTRUSION DETECTION SYSTEM FOR ADS-B

In this chapter, we propose machine learning based anomaly/intrusion detection system. Our system model consists of the general ADS-B components ADS-B In and ADS-B Out, as shown in Figure 5.1. Aircraft broadcast their position using ADS-B Out, and nearby aircraft and ground stations can receive the messages using ADS-B In. Our approach presents anomaly detection algorithms for aircraft and ground stations that process ADS-B data and search for anomalies/intrusions in air traffic.

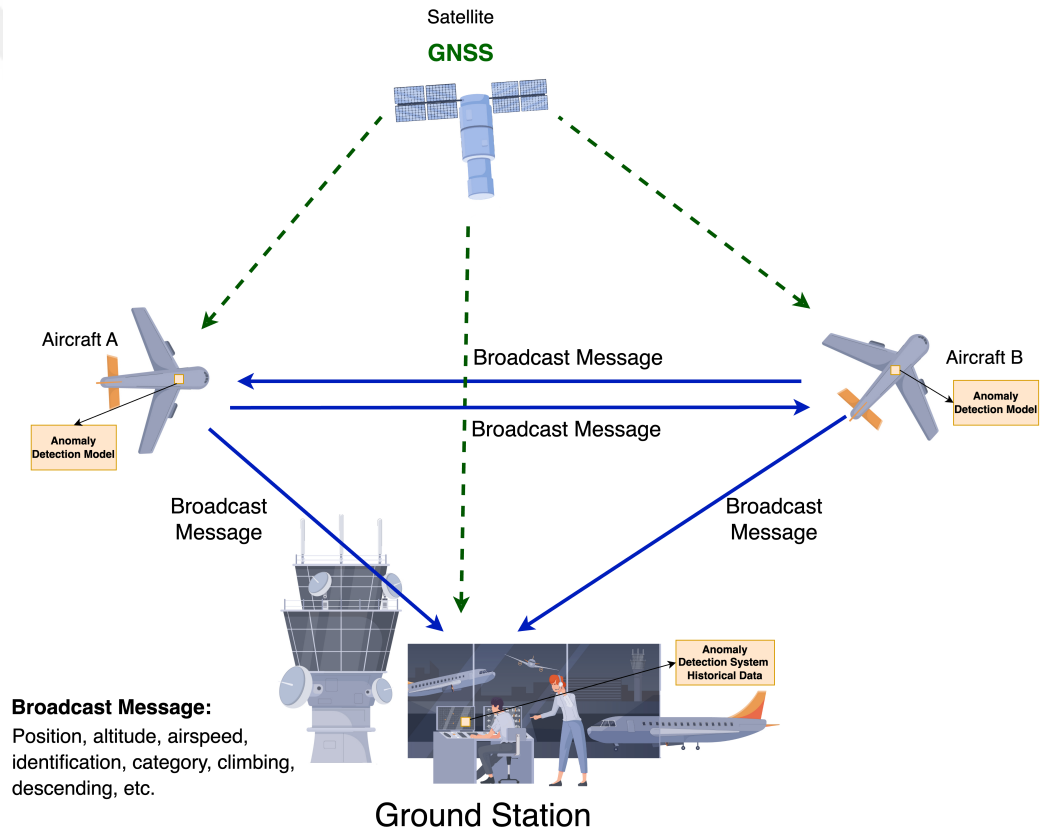


Figure 5.1. System model

Anomaly/intrusion detection systems can be located in ground stations and aircraft. There is a difference between the training phases of anomaly/intrusion detection systems according to their location. Training data gains importance according to the system location. Since flight characteristics are different, training data for ground stations should be collected from the range of 200 km to the nearest

ground stations.

The anomaly detection system at the ground station analyzes messages received from the aircraft, looking at it against historical data to identify any anomalies. The messages are monitored by air traffic controllers (ATC), and actions are taken if an anomaly is occurred. On the other hand, the broadcast messages from the other aircraft are analyzed in real-time by the aircraft anomaly detection system. The model is capable of identifying potential threats to the aircraft's navigation or communication systems. The detailed process is illustrated in Figure 5.2.

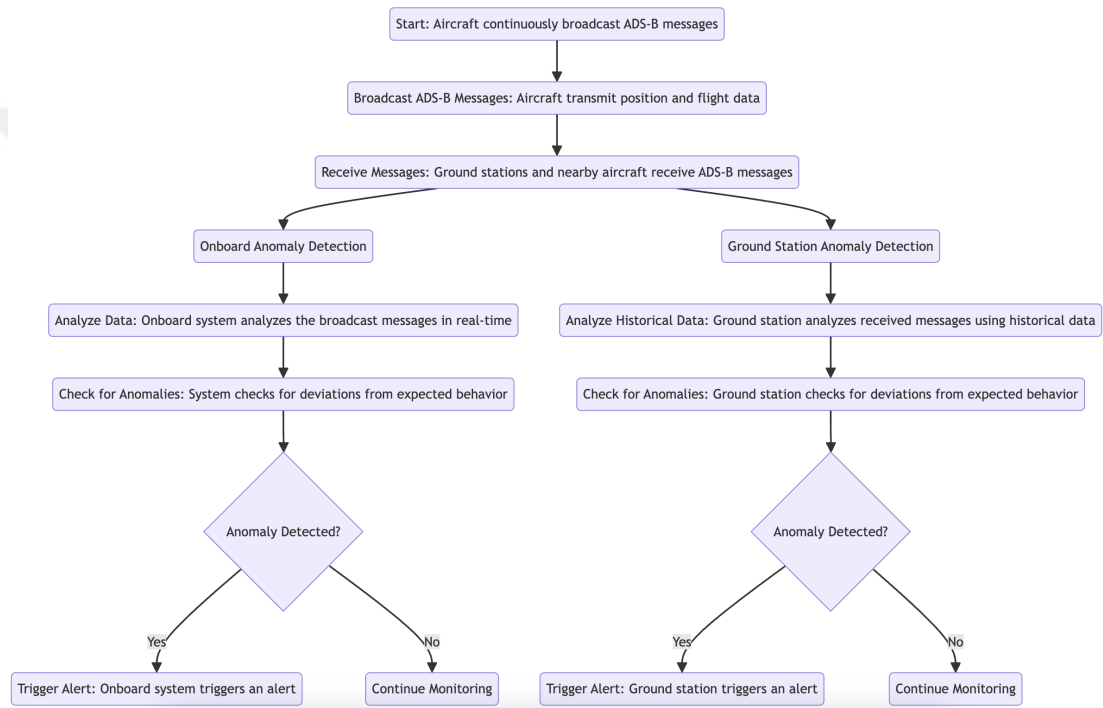


Figure 5.2. System model flowchart

By continuously monitoring and analyzing broadcast messages to allow for quick responses to potential safety threats. Detecting anomalies can also help in identifying potential security threats. The system provides situational awareness for both pilots and ground station personnel. This system model demonstrates an integrated approach to anomaly detection in ADS-B communication for onboard and ground-based systems in real-time.

This chapter is organized as follow: Section 5.1 outlines the overall architecture of the proposed anomaly and intrusion detection system for ADS-B. It highlights the design goals such as ensuring real-time detection, high accuracy, scalability, and

robustness. The architecture includes various components like data acquisition, preprocessing modules, and the integration of machine learning models. In Section 5.2, we focus on data collection phase. This includes gathering normal data and generating malicious data to train and test the models. We discuss preprocessing steps like data cleaning, normalization, and handling missing values to ensure the data is suitable for machine learning algorithms. Section 5.3 details the process of identifying and extracting relevant features from the ADS-B data that are crucial for detecting anomalies and intrusions. Feature engineering involves creating new features that can help improve the model's performance, while feature selection focuses on choosing the most important features to reduce complexity and enhance accuracy. In Section 5.4, we describe evaluation methods for assessing the performance of the IDS. This section discusses the importance of cross-validation and testing on unseen data to ensure the model's generalizability. Section 5.5 explores various machine learning models that can be applied to detect anomalies and intrusions in ADS-B data. It includes a discussion on different algorithms like decision trees, random forests, and logistic regression methods. The structure and workflow of integrating these models into the IDS are also covered. Section 5.6 concentrates the use of deep learning models for intrusion detection. It discusses the architecture of neural networks, including CNNs and RNNs, and how they can be used to identify complex patterns in ADS-B data. Additionally, the section covers the selection and tuning of hyperparameters to optimize model performance.

This chapter is presented in (Cevik & Akleyek, 2024a).

5.1. System Architecture and Design Goals

This thesis aims to develop an anomaly-based attack detection system against some security vulnerabilities of the ADS-B device. The system design is presented in Figure 5.3.

Within the scope of this thesis, practical constraints such as real-time data transfer, encryption/decryption time, and memory consumption have been considered, and it has been decided that data encryption should not be included in our design goals. It is expected that the developed anomaly detection system has the capacity to detect changes in ADS-B data or the source transmitting the data. Our design goals:

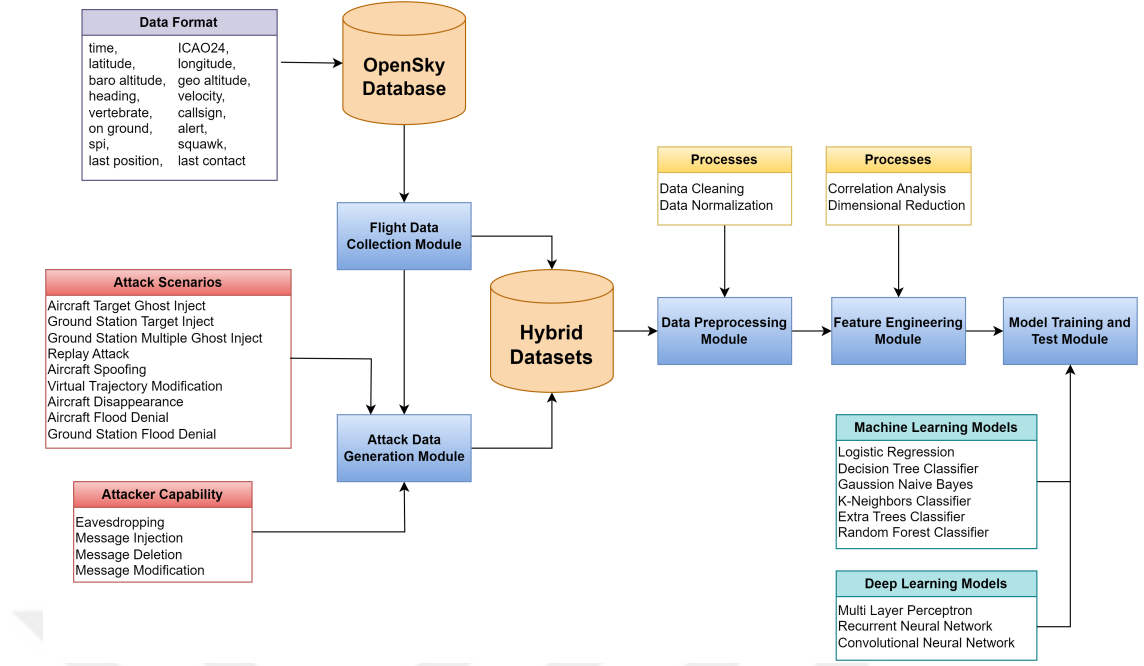


Figure 5.3. Designing phase of anomaly detection system

1. Ensuring reliable data communication in the ADS-B system, 2. Verifying that messages are transmitted from the correct source, 3. The objective is to detect intrusion, message injection, and deletion attacks. The architectural framework given above are systematically analyzed:

1. ADS-B messages, which include time, ICAO24, latitude, longitude, baroaltitude, geoaltitude, heading, velocity, vertebrate, callsign, on ground, alert, spi, squawk, last position update, and last contact parameters, are obtained via the OpenSky Network. The OpenSky Network data has an average missing data rate of approximately 5%, making it one of the most popular datasets in the literature. Details are given in Section 5.2
2. The collected ADS-B messages cover flights all over the world. There are difficulties in training models across such a wide range. Researchers frequently favor flights inside specific ranges in the literature (e.g., routes between two airports, airport landing or takeoff routes) (Pan et al., 2012), (Baek et al., 2017). Nevertheless, this method can result in over-fitting. In order to mitigate over-fitting, our thesis is not limited to a specific route. Details are given in Section 5.2

3. An attack dataset is needed to train ML models. Therefore, attack scenarios are simulated using the real flight data to create the attack dataset.
4. After attack generation steps, the generated attack datasets are merged with the real ADS-B dataset collected from the OpenSky Network.
5. Subsequently, commonly used parameters in the literature are examined, and for the training and testing of models, the following parameters are selected: time, ICAO24, latitude, longitude, altitude, heading, velocity, phase, and acceleration.

The designed system architecture includes classic machine learning steps. In addition to these classic steps, this thesis extensively investigate potential attack scenarios. Since there was no data labeled as anomaly/attack within ADS-B data, attack scenarios are simulated using real ADS-B data (Wesson et al., 2014). According to the Figure 5.3, it is a process that starts with data collection, continues with the preparation and transformation of the data, and ends with detecting anomalies in-flight data. Each part is critical in ensuring the system's ability to accurately detect unusual flight data patterns, thereby improving the overall security and reliability of aviation operations. Definitions of the modules are given below:

1. The flight data collection module collects flight information from the OpenSky database. It collects flight time, latitude, longitude, altitude, speed, and other relevant flight parameters. This module ensures that the system has a comprehensive and continuous flow of necessary data for subsequent use.
2. The attack data generation module simulates various attack scenarios based on predefined attacker capabilities. Examples of malicious activities include modifying, eavesdropping, and deleting messages. Combined with normal flight data, this generated data is vital to prepare the system to recognize different types of anomalies.
3. The data preprocessing module prepares datasets containing both real flight and simulated attack data. It involves cleaning data to eliminate errors and inconsistencies and normalizing it to ensure all data points are comparable. These steps are crucial to improve the data's quality and prepare it for further analysis.

4. The feature engineering module transforms the prepared data to make it more helpful in identifying patterns. It involves analyzing the relationships between different data points and reducing the number of data points to focus on the most important ones. These steps are necessary to optimize the performance and efficiency of the system.
5. The model training and testing module focuses on training and testing various anomaly detection methods. It uses the transformed data to train different ML methods, such as logistic regression, decision trees, and random forests. This section also tests these methods to evaluate their effectiveness and determine the most suitable methods for detecting anomalies.

5.2. Data Collection and Preprocessing

This section provides a detailed description of our research's data preprocessing steps. There are two main methods of collecting ADS-B data: creating a test environment using a hardware-based data collection module and collecting historical data from open source websites such as FlightAware, ADS-B Exchange, FlightRadar 24, OpenSky Network, and the FAA database. When comparing these options in Section 2.3, we decide to use OpenSky Network historical data sources because the scope of the hardware-based data collection module was limited.

At the beginning of the data collection phase, flight data are downloaded from the OpenSky Network. Details of the data row and an example of normal flight graphics are shown in Table 5.1.

While critical data from the ADS-B device is broadcast every two seconds, less critical data is broadcast every five seconds to reduce the transmission load. Consequently, There are missing parts and noise in the raw data. Additionally, missing values may arise in the data due to transmission. Considering these situations, it was observed that there is a significant amount of missing data within the messages. When examining studies in the literature, it has been observed that filling in these missing data negatively affects the model's accuracy. Although it might work for some datasets, adding the mean to rows containing NaN components is not appropriate for features that determine flight paths. Therefore, the dataset's rows with

Table 5.1. OpenSky Network data details

time	icao24	lat	lon	vel	heading	vertrate	callsign	baroalt	geoalt
1656367200	471f75	51.911	3.554	254.692	112.317	0.0	WZZ4400	10668.0	10881.3
1656367200	48507f	51.921	2.881	197.150	258.103	6.5024	KLM1057	9258.3	9471.6
1656367200	4ca90f	52.504	2.568	178.923	236.492	-4.8768	RYR2MT	8100.0	8275.3
1656367200	4ca702	54.380	-4.134	173.470	233.198	0.0	RYR717R	7315.2	7459.98
1656367200	4ca63b	53.666	-3.877	194.113	281.309	-11.7043	EIN33W	8892.5	9121.1
1656367200	4ca24f	53.858	-1.729	219.901	324.041	-5.2019	RYR59KA	11948.1	12169.14
1656367200	4ca294	52.222	-3.372	213.281	297.121	-8.7782	EIN4VM	10591.8	10888.9
1656367200	45ce48	52.783	-1.462	108.112	222.878	11.0541	SRR6977	1424.9	1501.1
1656367200	3c6483	52.776	-1.281	171.588	319.864	-8.7782	DLH1RV	5905.5	6065.5
1656367200	407caf	53.516	-2.769	151.669	85.720	-1.9507	RUK2UU	2247.9	2316.4
1656367200	405637	53.666	-1.578	224.445	326.783	-0.3251	EZY13NU	11597.6	11803.3
1656367200	407535	53.389	-1.579	220.586	335.766	0.0	SHT8G	9753.6	9959.3
1656367210	ab417a	42.779	-73.80	81.288	0.725	7.4778	UAL239	533.4	518.1
1656367210	c8195a	-37.214	174.8	97.839	245.456	5.8522	ANZ043L	2110.7	2202.1
1656367210	a2d359	44.043	-117.0	186.339	136.789	-10.0787	SKW132K	4381.5	4686.3

NaN values were eliminated. Then, all float and integer values are normalized with a min max scaler. It is shown in Table 5.2.

Table 5.2. Normalized dataset

time	icao24	lat	lon	vel	heading	vertrate	callsign	baroalt	geoalt
1656367200	471f75	0.853	0.509	0.085	0.312	0.5	WZZ4400	0.293	0.295
1656367200	48507f	0.853	0.508	0.066	0.717	0.5196	KLM1057	0.258	0.260
1656367200	4ca90f	0.857	0.507	0.060	0.657	0.4853	RYR2MT	0.229	0.230
1656367200	4ca702	0.872	0.488	0.058	0.647	0.5	RYR717R	0.209	0.210
1656367200	4ca63b	0.866	0.489	0.065	0.781	0.4647	EIN33W	0.249	0.251
1656367200	4ca24f	0.868	0.495	0.073	0.900	0.4843	RYR59KA	0.326	0.327
1656367200	4ca294	0.855	0.490	0.071	0.825	0.4735	EIN4VM	0.291	0.295
1656367200	45ce48	0.859	0.495	0.036	0.619	0.5333	SRR6977	0.060	0.062
1656367200	3c6483	0.859	0.496	0.057	0.888	0.4735	DLH1RV	0.173	0.175
1656367200	407caf	0.856	0.492	0.051	0.238	0.4941	RUK2UU	0.081	0.082
1656367200	405637	0.866	0.495	0.075	0.908	0.499	EZY13NU	0.317	0.318
1656367200	407535	0.864	0.495	0.074	0.932	0.5	SHT8G	0.270	0.272
1656367210	ab417a	0.781	0.295	0.027	0.002	0.5225	UAL239	0.038	0.037
1656367210	c8195a	0.157	0.985	0.032	0.682	0.5176	ANZ043L	0.077	0.079
1656367210	a2d359	0.791	0.174	0.062	0.380	0.4696	SKW132K	0.135	0.141

The phase of data collecting is the most critical step in this thesis. After analyzing the literature findings, it is clear that the dataset, especially the attack dataset, directly impacts how accurate machine learning models are.

5.3. Feature Engineering and Selection

This section provides a detailed description of feature extraction and feature selection steps. One of the crucial phases in the machine learning pipeline is the feature selection phase, which finds the most relevant attributes to increase the

model's performance. The appropriate features should be chosen to effectively identify anomalies and ensure the reliability of the IDS. Selecting relevant features not only improves model performance but also reduces computational complexity. Including contextual features, such as the type of aircraft and sensors used, in ADS-B data for anomaly detection models will be investigated. The models can be trained for airspace states in specific geolocations besides other features to provide more contextual information and improve detection ability. Moreover, postprocessing of IQ samples will be performed to extract more stable features, and training models for different flight phases may enhance overall accuracy. Domain-specific knowledge and contextual information minimize false positive rates are decreased. Additionally, minimizing data preprocessing time and feature extraction processes is crucial for the real-time application of ADS-B anomaly detection systems.

In this thesis, the initial features for the proposed IDS included ICAO24, latitude, longitude, velocity, heading, vertical rate, callsign, onground, alert, spi, squawk, baroaltitude, geoaltitude, lastposupdate, and lastcontact. These features form the basis of our analysis and model development. Features relevant to this thesis purposes are given below:

- **Latitude, Longitude, and Altitude:** Crucial for detecting anomalies in aircraft positioning.
- **Velocity, Heading, and Vertical Rate:** Essential for understanding flight dynamics and identifying suspicious maneuvers.
- **ICAO24 and Callsign:** Help in tracking and identifying individual aircraft.

5.3.1. Feature Generation

This section explains how we extract new features. After main feature selection step, new features were extracted from the raw data to improve the model's performance. Distance metrics have been calculated to measure the distance traveled between consecutive locations. We calculate the distance between two points on the Earth using the Haversine formula. In raw data, time is represented in UNIX time form. To extract time-based features like time of day and day of the week, we change the format of the time to timestamps, which included milliseconds. In addition, acceleration was derived from velocity to identify sudden speed changes, which may

indicate potential velocity attacks. It is given in Algorithm 10.

Algorithm 10 Calculate acceleration and distance

```

1: Input: Dataset  $D$  with parameters ICAO24, time, latitude ( $\phi$ ), and longitude ( $\lambda$ )
2: Output: Acceleration and Distance
3: Sort  $D$  according to ICAO24 and time parameters
4: for each aircraft  $a$  in  $D$  do
5:   for each consecutive pair of data points  $(p_1, p_2)$  for aircraft  $a$  do
6:     Compute time difference  $\Delta t = t_2 - t_1$ 
7:     Compute velocity difference  $\Delta v = v_2 - v_1$ 
8:     Compute acceleration  $a = \frac{\Delta v}{\Delta t}$ 
9:     Convert latitudes and longitudes from degrees to radians:
10:     $\phi_1 = \text{radians}(\phi_1)$ 
11:     $\phi_2 = \text{radians}(\phi_2)$ 
12:     $\lambda_1 = \text{radians}(\lambda_1)$ 
13:     $\lambda_2 = \text{radians}(\lambda_2)$ 
14:    Compute differences:
15:     $\Delta \phi = \phi_2 - \phi_1$ 
16:     $\Delta \lambda = \lambda_2 - \lambda_1$ 
17:    Compute  $d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\Delta \phi}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\Delta \lambda}{2} \right)} \right)$ 
18:   end for
19: end for

```

5.3.2. Correlation Analysis

In this section, we try to understand the relationship between features using correlation analysis. Firstly, we calculate and visualize the correlation matrix. It is given in Figures 5.4 and 5.5.

According to feature to feature correlation matrix, velocity, heading, and altitude (baroaltitude and geoaltitude) show moderate correlations among themselves. They indicate significant changes in flight dynamics, which is critical for anomaly detection. The geoaltitude and baroaltitude features are highly correlated. These two features are measuring similar aspects of altitude. Therefore, baroaltitude could be considered an unnecessary feature for ADS-B IDS. Latitude and longitude have a moderate negative correlation with each other, their relatively low correlation with other features. They provide unique information about positional anomalies. Acceleration and distance features show very low correlation with others. They are crucial for identifying anomalies related to sudden changes in movement dynamics.

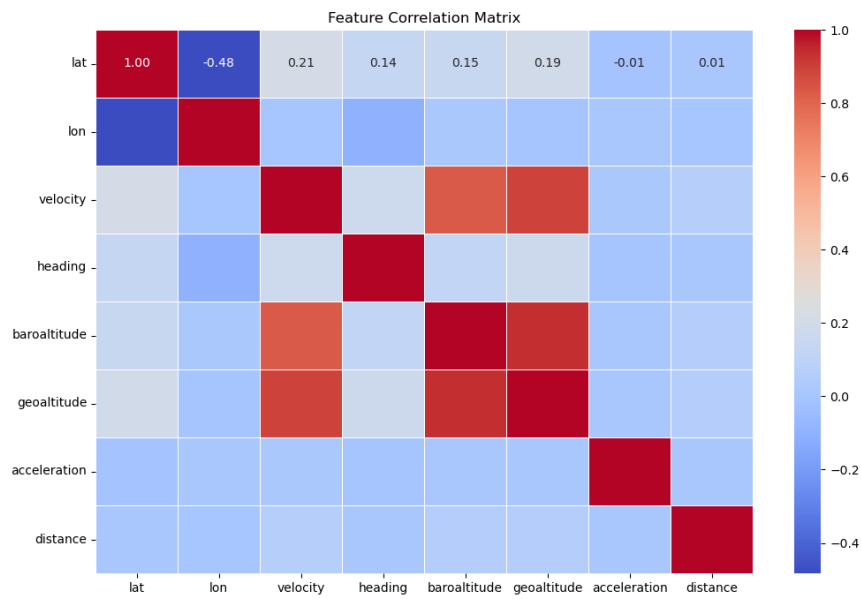


Figure 5.4. Correlation matrix - feature to feature

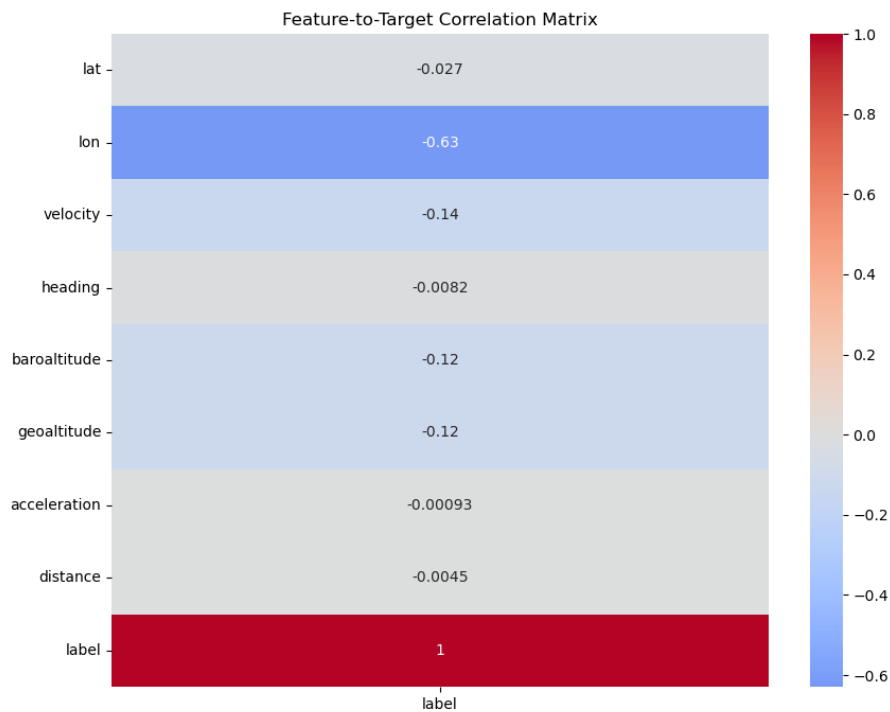


Figure 5.5. Correlation matrix - feature to label

According to feature to label correlation matrix, longitude has the strongest negative correlation with the target among all features, meaning as it increases, the target value tends to decrease. Longitude is the most influential feature on the target. All of these features were evaluated in the final feature selection phase. As a result, using domain knowledge in aviation, the final feature set selected: latitude, longitude, altitude, velocity, heading, ICAO24, callsign, distance-based, and time-based features. The final set of selected features given in Table 5.3, including both raw and derived features, plays a crucial role in our ADS-B anomaly detection model.

Table 5.3. Feature types and their corresponding features

Feature Type	Features
Positional features	Latitude, longitude, altitude
Kinematic features	Velocity, heading, vertical rate
Identification features	Icao24, callsign
Derived features	Distance and time-based metrics

5.3.3. Dimensional Reduction

In this section, we use dimensional reduction methods, such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), uniform manifold approximation and projection (UMAP), and autoencoder, to extract and create new features that may be more informative for the model. Visualizations of the dimensional reduction methods' results are given in Figure 5.6.

According to PCA results, the points of attack and normal data are mixed. However, the attack data tends to form distinct clusters. Although it captures the largest variance, PCA, a linear dimensionality reduction technique, could not be as effective in clearly separating classes as nonlinear approaches. The figure of t-SNE shows multiple distinct clusters for normal data (blue) and attack data (orange). Local structures can be effectively preserved via t-SNE. It can be concluded that t-SNE outperforms PCA in capturing complex patterns and local neighborhood relationships in the data. UMAP is known for preserving both local and global structures in data. The graph shows a more precise separation between normal and attack data, forming several clusters. Therefore, it is clear that UMAP effectively captures the manifold

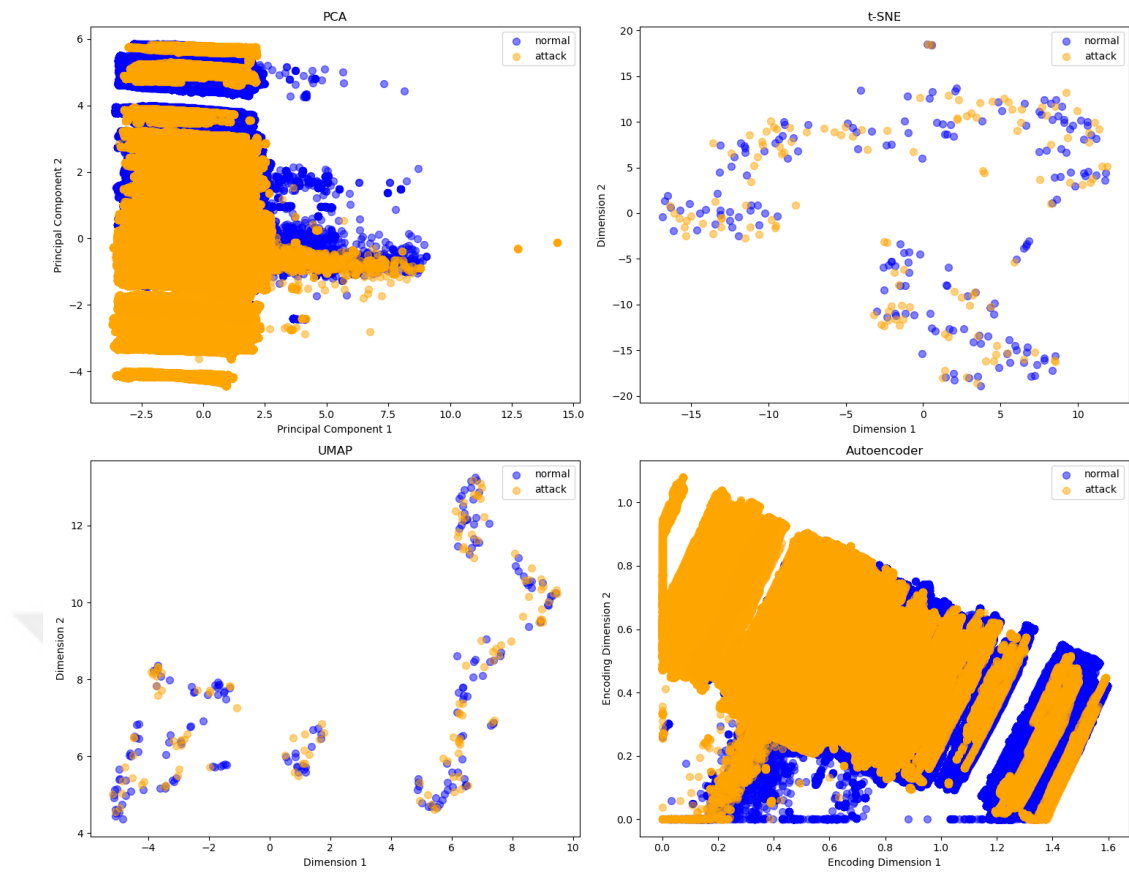


Figure 5.6. Dimensional reduction methods

structure of the data and allows better visualization of class distinctions. The autoencoder shows separation between normal and attack data, but there is significant overlap, especially towards the bottom of the graph. This indicates that the autoencoder can capture nonlinear interactions. Nevertheless, additional fine-tuning of the design and training parameters may be required to achieve better class separability. These results are used to identify patterns, clusters, or anomalies that would not be apparent in higher dimensions.

5.3.4. Chi-Square Test

The chi-Square test is a statistical method used to measure the dependence between categorical variables. It calculates the difference between observed and expected frequencies to determine if there is a significant association between the variables. This test is often used in feature selection for categorical data to identify which features have the strongest relationship with the target variable. For instance, in a dataset where the target variable is categorical, such as classifying whether an email is spam or not, the chi-square test can help identify the most relevant features, like the presence of certain words or phrases (Agresti, 2018). According to chi-Square test, selected features are latitude, longitude, velocity, heading, baroaltitude, geoaltitude, acceleration, distance.

5.3.5. ANOVA F-test

Analysis of variance (ANOVA) is used to compare the means of three or more groups to see if at least one of them is statistically different from the others. It helps in determining whether different groups have different impacts on the dependent variable. ANOVA is commonly applied in experimental data where multiple groups or treatments are compared. For example, in agricultural research, ANOVA can be used to compare crop yields under different treatment conditions to determine which treatment produces the best yield (Montgomery, 2017). According to ANOVA, selected features are latitude, longitude, velocity, heading, baroaltitude, geoaltitude, acceleration, distance.

5.3.6. Lasso Regression (L1 Regularization)

Lasso regression is a type of linear regression that includes a penalty term equivalent to the absolute value of the coefficients. This penalty causes some coefficients to shrink to zero, effectively performing feature selection. This is particularly useful when dealing with datasets with many features, as lasso helps in identifying the most important ones while reducing overfitting. For example, in genomics, lasso regression can be used to select the most relevant genes associated with a particular trait from a large set of gene expression data (Tibshirani, 1996). According to lasso regression, selected features are latitude, longitude, velocity, heading, baroaltitude.

5.3.7. Random Forest Classifier

Decision trees are a type of supervised learning algorithm that can be used for both classification and regression tasks. They work by splitting the data into subsets based on the value of input features, creating a tree-like structure. Feature importance can be derived from tree-based methods, such as random forests (RF), by measuring the decrease in impurity or error when a feature is used to split the data. This helps in identifying which features contribute the most to the prediction. For instance, in customer churn prediction, decision trees can identify key factors like customer service calls or contract length that are most indicative of churn (Breiman, 2001). According to random forrest, selected features are latitude, longitude, velocity, heading, baroaltitude, geoaltitude, acceleration, distance.

5.3.8. Hybrid Approach

A hybrid method combines filter and wrapper methods for more effective feature selection. First, a filter method like SelectKBest with the ANOVA F-test is used to select the top features based on their statistical significance. Then, a wrapper method such as recursive feature elimination (RFE) with a logistic regression model is applied to further refine the selection. This approach leverages the strengths of both methods: the efficiency and scalability of filter methods, and the accuracy and consideration of feature interactions of wrapper methods. For example, in medical research, a hybrid

method can be used to first identify the most relevant biomarkers for a disease and then fine-tune the selection to ensure the best predictive accuracy of the diagnostic model (Guyon & Elisseeff, 2003). According to the hybrid approach, selected features are latitude, longitude, velocity, heading, baroaltitude, geoaltitude, acceleration, distance.

5.4. Evaluation Methods

In evaluation phase, we use cross validation technique. In cross-validation, the dataset is divided into various training-test parts. This approach reduces the risk of overfitting and ensures the performance of the model across various data partitions (Kohavi, 1995). Therefore, we use k-fold cross-validation in the model training and testing phases. Moreover, analyzing the model's performance on completely different datasets provides additional information. Moreover, testing the model on various datasets improves its generalizability and robustness. Therefore, data from different periods, regions, or conditions have been collected for this thesis's validation process. The aim of applying these techniques is to ensure a more reliable and comprehensive evaluation of the model's performance and to lead to better generalization and robustness in real-world applications.

5.5. Machine Learning for Anomaly Detection

In the context of an anomaly detection system, each of these machine learning models plays a crucial role in identifying patterns that deviate from the normal data. This section briefly overviews each model and highlights its significance inside an anomaly detection system.

1. Logistic regression distinguishes between normal and abnormal instances because it is a binary classification procedure that aims to classify every example into one of two classes. Its simplicity and effectiveness make it a fitting method for anomaly detection. Logistic regression models the probability that a given instance belongs to a particular class. This would be more specifically useful, for example, in identifying if an instance is typical or an anomaly. In other words, it could separate the two classes using establishing a decision boundary (Weaver & Shannon, 1948).

2. Since a decision tree classifier can handle numerical and categorical data, it shall be more versatile in working with various data formats. It helps one to identify anomalies and learning decision rules out of input data, which are in the form of a tree structure where each node represents a feature and branches represent decision rules. At the end of the trees are the leaves with the possible outcomes; thus, outliers could be classified as those who do not end in any of the outcomes from established patterns or rules (Breiman et al., 1984).
3. Gaussian naive bayes (GNB) helps to detect anomalies by following the basics of Bayes' theorem principles. It estimates the probabilities of each class and classifies every instance according to the maximum likelihood. The gaussian assumption makes this model suitable for high-dimensional data, or rather, it would be hard to detect anomalies as probability distributions will vastly differ from typical cases (John & Langley, 1995).
4. In anomaly detection, simplicity and efficiency are helpful in the k-neighbors classifier. A KNN method finds the k closest instances (neighbors) to a given data point and classifies the end based on the majority class among its neighbors. All these help one to notice the potential outliers, as under most of the typical conditions, they are usually very far from other instances and do not have any close neighbors or the neighborhood belongs to some different class (Aha et al., 1991).
5. Extra trees classifier (ETC) is another ensemble method that builds multiple decision trees based on bootstrapped input, with the final prediction being made by aggregating the results. The model is robust and has good accuracy because the features and data points are randomly selected for each of these trees, which reduces the chance of overfitting. This is most important in anomaly detection because it ensures that the model generalizes well to new, unseen data to be reliable for anomaly detection (Geurts et al., 2006).
6. Another robust classifier of the ensemble method is the random forest classifier (RFC). Many decision trees are created and combined to make a final decision. It is very effective for anomaly detection because it is robust and can manage massive datasets. The random nature followed by the Random Forests naturally

helps reduce overfitting because of the averaging in multiple decision trees, which would improve the generalization capability of a model. As well it can handle sizeable amounts of data with an ability for parallel processing, thus suitable for detecting anomalies in vast datasets (Breiman, 2001).

5.6. Deep Learning for Anomaly Detection and Hyperparameters

Deep learning models play a crucial role in identifying patterns that deviate from the normal data for an anomaly detection system. This section briefly overviews each model and highlights its significance inside an anomaly detection system.

1. Multilayer Perceptron (MP) is an artificial neural network model and it is a supervised learning algorithm. The most crucial factor in the Perceptron Model is determining the threshold value (Rumelhart et al., 1986).
2. Recurrent Neural Networks (RNN) is defined as a class of artificial neural networks in which the connections between nodes form a directed loop. This allows the algorithm to exhibit dynamic temporal behavior (Gers et al., 2000).
3. Convolutional Neural Network (CNN) is a sub-branch of deep learning and is generally used to analyze visual information. Typical uses include image and video recognition, suggestive systems, image classification, medical image analysis, and natural language processing (LeCun et al., 1998).

Deep learning models contain hyperparameters that specify these models' structure and learning behavior. At the same time, essential classes of hyperparameters are the learning rate, batch size, number of layers, and number of neurons in each layer that need to be optimally selected so that the model learns effectively. A well-tuned model achieves high accuracy and generalizes to new data, meaning it can make sound predictions over data points it has never seen before. If the hyperparameters are not set right, the model may learn too slowly, fail to learn, or become unstable and produce sub-optimal results. Hyperparameters of our deep learning model is given in Table 5.4.

A multi-layer perceptron (MLP) is a type of feedforward artificial neural network consisting of an input layer, one or more hidden layers, and an output layer, commonly

Table 5.4. Hyperparameters for different deep learning models

	MLP	CNN	RNN
Architecture	Sequential	Sequential	Sequential
Input Layer	Dense (64 units, relu)	Conv1D (64 filters, 2 kernel, relu)	LSTM (50 units)
Hidden Layer	Dense (32 units, relu)	MaxPooling1D (2 pool size)	-
Output Layer	Dense (1 unit, sigmoid)	Dense (1 unit, sigmoid)	Dense (1 unit, sigmoid)
Optimizer	Adam	Adam	Adam
Loss Function	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy

used for tasks with structured data like tabular data. In the provided MLP model, the architecture is sequential. The input layer is a dense layer with 64 units using the rectified linear unit (ReLU) activation function, which transforms the input data to extract relevant features. Following this, there is a hidden layer with 32 units, also using ReLU activation, to capture more complex patterns in the data. The final output layer has 1 unit with sigmoid activation for binary classification, outputting a probability value between 0 and 1 to indicate the likelihood of a particular class. The model uses the Adam optimizer, an adaptive learning rate optimization algorithm that combines the benefits of AdaGrad and RMSProp. The loss function used is binary crossentropy, which is suitable for binary classification tasks and measures the performance of the model by comparing the predicted probability to the actual class label. A RNN is designed to handle sequential data, making it suitable for tasks like time series prediction and natural language processing. The RNN model described here utilizes an LSTM layer, which is capable of learning long-term dependencies in data. The architecture of model includes an LSTM layer with 50 units and a dense output layer with 1 unit and sigmoid activation for binary classification. The Adam optimizer and binary crossentropy loss function are used to compile the model, with accuracy as the performance metric. The model is trained over 10 epochs with a batch size of 32 and a validation split of 0.2. A CNN is characterized by its use of convolutional layers, which apply a set of filters to the input data to extract features.

6. EXPERIMENTAL RESULTS AND DISCUSSION

This chapter defines the experimental results which contains the performance metrics of various machine learning algorithms used for anomaly/intrusion detection on ADS-B data. The attack types include spoofed random velocity, spoofed gradual velocity, spoofed random altitude, spoofed gradual altitude, spoofed random latitude/longitude, spoofed gradual latitude/longitude, spoofed merge attacks, and replay attacks. The labels 0 and 1 represent normal data and attack data, respectively. Additionally, three different parameter sets affect the performance of the algorithms. The key performance metrics include accuracy, precision, recall, F1 score, ROC AUC.

This chapter is organized as follow: In Section 6.1, we present the outcomes of our deep learning models, evaluated using various metrics across different hybrid datasets. The focus is on demonstrating the effectiveness and performance of these models under different conditions. Section 6.2 mirrors the structure of the previous one but shifts the focus to machine learning models. We discuss the results obtained, highlighting the evaluation metrics used and the performance across various hybrid datasets. Section 6.3 compare our results with existing methods in the literature. This comparison aims to contextualize our findings within the broader field, showcasing how our models stack up against established approaches.

6.1. The Results of Deep Learning Models

The performance of deep learning models in overall evaluation provides a comprehensive understanding of how well different deep learning algorithms perform across datasets. The results of the deep learning algorithms are given in Table 6.1.

The MLP model demonstrates excellent performance for all evaluation metrics. In addition, the MLP model is very good at both detecting true positives and minimizing false positives. The FNR of MLP is 0.025160, and the FPR is 0.014005, indicating high detection reliability and robustness. While performing well, the RNN model has lower results than the MLP. The lower recall (0.866185) compared to the MLP model indicates that the RNN misses more true positives. The ROC AUC of 0.912180 shows

Table 6.1. Deep learning model results for dataset 1

Metric	MLP	RNN	CNN
Accuracy	0.982	0.923	0.950
Precision	0.977	0.927	0.958
Recall	0.975	0.866	0.909
F1 Score	0.976	0.895	0.933
ROC AUC	0.980	0.912	0.942
False Negative Rate	0.025	0.133	0.090
False Positive Rate	0.014	0.041	0.024

good discrimination ability but is lower than the MLP's. The CNN model also performs very well for all evaluation metrics. These results are slightly lower than those of the MLP but better than the RNN. The ROC AUC of 0.942878 shows strong discrimination capability. The FNR of 0.090119 and FPR of 0.024125 are relatively low, making the CNN model a reliable option, though less robust than the MLP.

6.1.1. Overfitting in Deep Learning

Overfitting is a critical problem in deep learning, where the model performs very well on training data, but it needs to be generalized to new and unseen data. The model learns the noise and details in the training data set. Therefore the generalization ability of the model reduces (Caruana et al., 2001). It causes the model to perform poorly on various inputs it will encounter in real-world applications. Factors of the overfitting are high complexity of the model, insufficient training data and excessive training time. For example, since a MLP may contain thousands or even millions of parameters, model fails on new data by fully adapting to the training data.

Various techniques can be used to prevent overfitting. Early stopping stops the training process at a certain point by monitoring the validation error. The dropout technique prevents overfitting of the model and increases its generalization ability by randomly turning off certain neurons of the neural network (Srivastava et al., 2014). Data augmentation enriches the training data with various transformations, allowing the model to be exposed to different data samples. Additionally, regularization techniques such as L1 and L2 regularization methods reduce overfitting by constraining the weights of the model (Ng, 2004). These methods prevent the model from overfitting the training data and increase its generalization capacity, allowing it to perform better on new

datasets.

6.1.2. Analysis and Performance Evaluation of Deep Learning Algorithms

In this section, we will examine the memorization problem on our deep learning architectures: Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). By analyzing the loss curves of these models, we will better understand how memorization occurs and its effects on model performance. The information obtained from these analyses is vital for developing strategies to reduce memorization and increase the robustness of deep learning models.

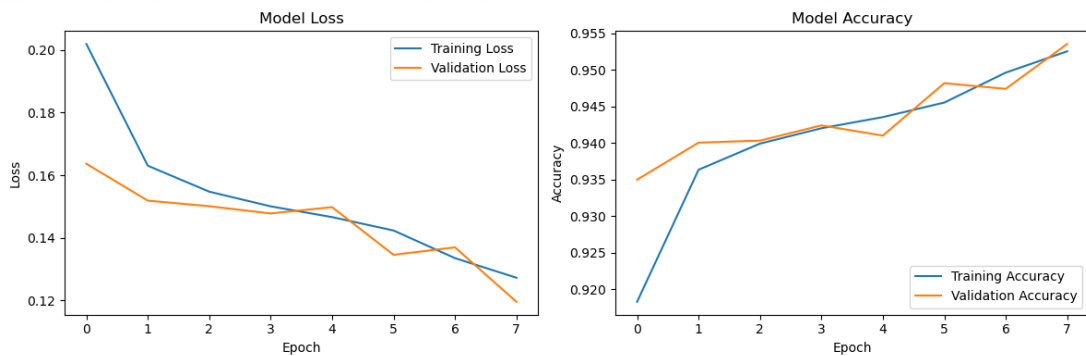


Figure 6.1. Model Loss and Accuracy Graphics for CNN

According to Figure 6.1, the CNN model's training and validation performance looks relatively high. The continuous decrease in training loss shows that the model adapts better to the training data in the model loss graphic. Validation loss decreases similarly, indicating that the model performs better on validation data. The fact that there is no significant difference between training and validation losses indicates that the model is balanced and its generalization ability is good. In the model accuracy line, the continuous increase in training accuracy indicates that the model performs better on the training data. Likewise, the general increase in validation accuracy reveals that the model performs well on validation data. The slight difference between training and validation accuracy indicates that the model does not overfit and has good generalization ability on validation data. These graphs prove that the model performs well on training and validation data and is balanced. The fact that both the training loss and validation loss are constantly decreasing, and the training accuracy and validation accuracy are

constantly increasing, indicates that the model fits nicely on the training data and can maintain this fit on the validation data.

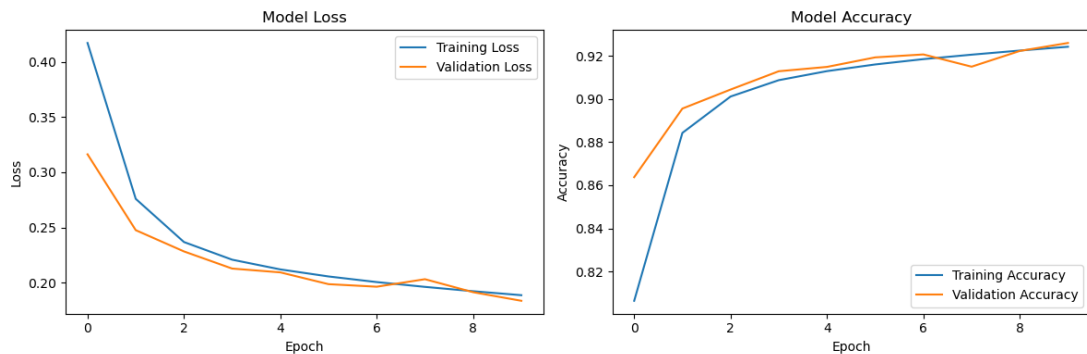


Figure 6.2. Model Loss and Accuracy Graphics for RNN

According to Figure 6.2, it can be seen that the RNN performs well on both training and validation data and does not overfit. In the model loss plot, both training loss and validation loss continuously decrease, indicating that the model performs increasingly better on both datasets. The slight difference between training and validation losses indicates that the model has good generalization ability. In the model accuracy chart, the continuous increase in training accuracy and the corresponding increase in validation accuracy indicates that the model has learned well from the training data and maintains this performance on the validation data. The close training and validation accuracies show that the model has achieved balanced and successful learning, and its generalization ability is strong.

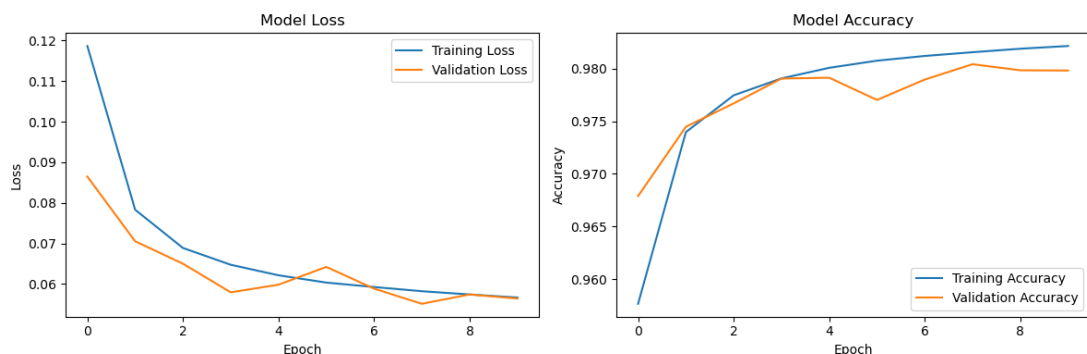


Figure 6.3. Model Loss and Accuracy Graphics for MLP

According to Figure 6.3, the model accuracy graph shows that training accuracy is constantly increasing and reaching high levels. Validation accuracy is also

increasing similarly and is very close to training accuracy. This reveals the model's robust generalization ability and performs well on new data. It is understood that the hyperparameters and architecture of the model were chosen correctly, and the training process was managed effectively.

6.2. The Results of Machine Learning Models

In this section, experimental results for ML algorithms are given for different datasets. In Tables 6.2, 6.3, and 6.4, the performance of different algorithms is evaluated for dataset 1. This dataset includes all attack vectors. For this dataset, the parameter set (standard, time phase, distance) and scaling affect the results of evaluation metrics differently.

Table 6.2. Algorithm performance comparison for dataset 1 (time parameter set without scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
DTC	No	0.955	0.998	0.896	0.944	0.948	0.104	0.001	Phase
DTC	No	0.783	0.641	0.979	0.775	0.821	0.021	0.338	Std.
GNB	No	0.571	0.000	0.000	0.000	0.500	1.000	0.000	Phase
GNB	No	0.628	0.671	0.261	0.376	0.583	0.739	0.096	Std.
LR	No	0.571	0.000	0.000	0.000	0.500	1.000	0.000	Phase
LR	No	0.687	0.717	0.295	0.418	0.612	0.705	0.072	Std.

Table 6.3. Algorithm performance comparison for dataset 1 (for scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
LR	No	0.687	0.717	0.295	0.418	0.612	0.705	0.072	Std.
LR	Yes	0.619	0.000	0.000	0.000	0.500	1.000	0.000	Std.
RFC	No	0.783	0.641	0.979	0.775	0.821	0.021	0.338	Std.
RFC	Yes	0.966	1.000	0.911	0.953	0.955	0.089	0.000	Std.
KNN	No	0.621	0.711	0.012	0.023	0.504	0.988	0.003	Std.
KNN	Yes	0.961	0.998	0.899	0.946	0.949	0.101	0.001	Std.
ETC	No	0.791	0.652	0.968	0.779	0.825	0.032	0.318	Std.
ETC	Yes	0.960	1.000	0.896	0.945	0.948	0.104	0.000	Std.
GNB	No	0.381	0.376	0.947	0.539	0.490	0.053	0.967	Std.
GNB	Yes	0.619	0.000	0.000	0.000	0.500	1.000	0.000	Std.
DTC	No	0.783	0.641	0.979	0.775	0.821	0.021	0.338	Std.
DTC	Yes	0.958	0.989	0.900	0.943	0.947	0.100	0.006	Std.

According to tables above, scaling significantly improves the performance of

Table 6.4. Algorithm performance comparison for dataset 1 (distance parameter set with scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
LR	Yes	0.619	0.000	0.000	0.000	0.500	1.000	0.000	Std.
LR	Yes	0.841	0.716	0.969	0.823	0.866	0.031	0.237	Dist.
ETC	Yes	0.960	1.000	0.896	0.945	0.948	0.104	0.000	Std.
ETC	Yes	0.841	0.721	0.949	0.820	0.862	0.051	0.226	Dist.
DTC	Yes	0.958	0.989	0.900	0.943	0.947	0.100	0.006	Std.
DTC	Yes	0.804	0.678	0.927	0.783	0.828	0.073	0.271	Dist.
GNB	Yes	0.619	0.000	0.000	0.000	0.500	1.000	0.000	Std.
GNB	Yes	0.835	0.708	0.967	0.817	0.861	0.033	0.246	Dist.

several algorithms. For instance, the accuracy of the random forest classifier increases from 0.783 to 0.966 with scaling. Similarly, the decision tree classifier's accuracy improves from 0.955 to 0.958. Scaling also positively affects precision and recall, making the models more balanced in their predictions.

Moreover, additional features such as distance and acceleration have increased the performance of some algorithms. For example, the accuracy and F1 score of gaussian naive bayes increase to 0.835 and 0.817, respectively. Even if the results of logistic regression also improved by achieving 0.841 accuracy and 0.823 F1 score, with additional parameters, it generally performs poorly. random forest classifier and extra tree classifier consistently show strong performance across different configurations for all metrics. K-neighbors classifier has mixed results, performing well with scaling but poorly in recall without it.

FNR and FPR are critical metrics. For example, the random forest classifier with scaling achieves 0.089 FNR and 0.0 FPR, indicating a shallow rate of false negatives and no false positives. Without scaling, the random forest's FNR increases to 0.021, and its FPR increases to 0.338. Gaussian naive bayes shows high FNR and FPR values without scaling (0.739 and 0.096, respectively), which improve significantly with additional features (0.033 and 0.246, respectively). Logistic regression without additional features has 1.0 FNR and 0.0 FPR, indicating it fails to detect any anomalies, but these metrics improve when additional features are included. The k-Neighbors classifier without scaling has 0.988 FNR, which shows poor performance in detecting anomalies.

In Tables 6.5 and 6.6, the performance of different algorithms is evaluated for dataset 2. It includes normal data, spoofed gradual velocity, spoofed gradual altitude, and spoofed gradual latitude and longitude.

Table 6.5. Algorithm performance comparison for dataset 2 (scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
ETC	No	0.853	0.758	0.966	0.849	0.867	0.034	0.232	Std.
ETC	Yes	0.954	1.000	0.893	0.944	0.947	0.107	0.000	Std.
DTC	No	0.838	0.734	0.977	0.838	0.855	0.023	0.266	Std.
DTC	Yes	0.951	0.988	0.897	0.940	0.944	0.103	0.008	Std.
GNB	No	0.417	0.420	0.947	0.582	0.483	0.053	0.981	Std.
GNB	Yes	0.571	0.000	0.000	0.000	0.500	1.000	0.000	Std.
LR	No	0.683	0.924	0.283	0.434	0.633	0.717	0.017	Std.
LR	Yes	0.571	0.000	0.000	0.000	0.500	1.000	0.000	Std.
KNN	No	0.576	0.931	0.012	0.024	0.506	0.988	0.001	Std.
KNN	Yes	0.955	0.998	0.896	0.944	0.948	0.104	0.001	Std.
RFC	No	0.838	0.733	0.977	0.838	0.855	0.023	0.266	Std.
RFC	Yes	0.960	1.000	0.908	0.952	0.954	0.092	0.000	Std.

Table 6.6. Algorithm performance comparison for dataset 2 (distance parameter set with scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
LR	Yes	0.571	0.000	0.000	0.000	0.500	1.000	0.000	Std.
LR	Yes	0.938	0.895	0.968	0.930	0.941	0.032	0.085	Dist.
ETC	Yes	0.954	1.000	0.893	0.944	0.947	0.107	0.000	Std.
Etc	Yes	0.970	0.957	0.974	0.965	0.971	0.026	0.033	Dist.
RFC	Yes	0.960	1.000	0.908	0.952	0.954	0.092	0.000	Std.
RTC	Yes	0.959	0.930	0.977	0.953	0.961	0.023	0.055	Dist.
GNB	Yes	0.571	0.000	0.000	0.000	0.500	1.000	0.000	Std.
GNB	Yes	0.934	0.890	0.965	0.926	0.938	0.035	0.089	Dist.
DTC	Yes	0.951	0.988	0.897	0.940	0.944	0.103	0.008	Std.
DTC	Yes	0.907	0.842	0.963	0.898	0.914	0.037	0.136	Dist.

According to the results, the parameter set (standard, time phase, distance) and scaling affect models' performance rates. Scaling improves the performance of many algorithms. For example, the accuracy of the random forest classifier increases to 0.960. The accuracy of the decision tree classifier increased from 0.838 to 0.951. Scaling also positively affects precision and recall. For example, the extra tree classifier with scaling achieves 0.954 accuracy, 1.000 precision, and 0.893 recall. Moreover,

additional features such as distance and acceleration further increase performance. With these parameters, logistic regression achieves 0.938 accuracy. Similarly, the extra tree classifier achieves 0.970 accuracy, while the random forest classifier achieves 0.959 accuracy. These improvements reduced FNR and FPR. For example, the extra tree classifier with distance and acceleration features has 0.026 FNR and 0.033 FPR.

In Tables 6.7 and 6.8, the performance of different algorithms is evaluated for dataset 3. It includes normal data, spoofed gradual lat/lon, and merge attack.

Table 6.7. Algorithm performance comparison for dataset 3 (for scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
ETC	No	0.855	0.746	0.974	0.844	0.874	0.026	0.226	Std.
ETC	Yes	0.942	1.000	0.857	0.923	0.928	0.143	0.000	Std.
DTC	No	0.843	0.726	0.983	0.835	0.865	0.017	0.252	Std.
DTC	Yes	0.946	0.988	0.876	0.929	0.934	0.124	0.007	Std.
GNB	No	0.395	0.397	0.951	0.560	0.484	0.049	0.983	Std.
GNB	Yes	0.621	0.574	0.253	0.351	0.563	0.747	0.128	Std.
LR	No	0.693	0.920	0.266	0.412	0.625	0.734	0.016	Std.
LR	Yes	0.618	0.571	0.228	0.325	0.556	0.772	0.117	Std.
KNN	No	0.599	0.892	0.011	0.022	0.505	0.989	0.001	Std.
KNN	Yes	0.628	0.579	0.297	0.393	0.575	0.703	0.147	Std.
RFC	No	0.843	0.726	0.983	0.835	0.865	0.017	0.252	Std.
RFC	Yes	0.635	0.649	0.218	0.326	0.569	0.782	0.080	Std.

Table 6.8. Algorithm performance comparison for dataset 3 (distance parameter set scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
ETC	Yes	0.946	0.902	0.973	0.936	0.951	0.027	0.072	Dist.
ETC	Yes	0.942	1.000	0.857	0.923	0.928	0.143	0.000	Std.
DTC	Yes	0.918	0.854	0.962	0.905	0.925	0.038	0.112	Dist.
DTC	Yes	0.946	0.988	0.876	0.929	0.934	0.124	0.007	Std.
RFC	Yes	0.940	0.886	0.977	0.929	0.946	0.023	0.085	Dist.
RFC	Yes	0.950	1.000	0.878	0.935	0.939	0.122	0.000	Std.
GNB	Yes	0.909	0.829	0.977	0.897	0.920	0.023	0.137	Dist.
GNB	Yes	0.595	0.000	0.000	0.000	0.500	1.000	0.000	Std.
LR	Yes	0.922	0.851	0.978	0.910	0.931	0.022	0.117	Dist.
LR	Yes	0.595	0.000	0.000	0.000	0.500	1.000	0.000	Std.

According to the results, the parameter set (standard, time phase, distance) and scaling affect models' performance rates. Scaling improves the performance of many

algorithms. For example, accuracy of the random forest classifier increases to 0.950 with scaling. Scaling generally improves the precision and recall, making models more balanced same with our dataset results. Additional features such as distance and acceleration further enhances performance of logistic regression, random forest classifier, and extra tree classifier. These features also reduced FNR and FPR. For example, the extra tree classifier with distance and acceleration features has 0.027 FNR and 0.072 FPR.

In Tables 6.9 and 6.10, the performance of different algorithms is evaluated for dataset 4. It includes normal data, spoofed merge attacks, and spoofed replay attacks.

Table 6.9. Algorithm performance comparison for dataset 4 (for scaling)

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
GNB	No	0.198	0.190	0.952	0.317	0.484	0.048	0.985	Std.
GNB	Yes	0.805	0.000	0.000	0.000	0.500	1.000	0.000	Std.
LR	No	0.837	0.709	0.283	0.404	0.627	0.717	0.028	Std.
LR	Yes	0.805	0.000	0.000	0.000	0.500	1.000	0.000	Std.
KNN	No	0.805	0.669	0.008	0.015	0.503	0.992	0.001	Std.
KNN	Yes	0.705	0.278	0.320	0.298	0.559	0.680	0.202	Std.
RFC	No	0.734	0.423	0.997	0.594	0.833	0.003	0.330	Std.
RFC	Yes	0.999	1.000	0.997	0.998	0.999	0.003	0.000	Std.
ETC	No	0.745	0.432	0.982	0.600	0.834	0.018	0.313	Std.
ETC	Yes	0.999	1.000	0.997	0.998	0.998	0.003	0.000	Std.
DTC	No	0.733	0.423	0.997	0.594	0.833	0.003	0.331	Std.
DTC	Yes	0.992	0.967	0.993	0.979	0.992	0.007	0.008	Std.

Table 6.10. Algorithm performance comparison for dataset 4

Algorithm	Sca.	Acc.	Pre.	Rec.	F1S.	ROC	FNR	FPR	Set
LR	Yes	0.786	0.478	0.993	0.645	0.864	0.006	0.263	Dist.
LR	Yes	0.805	0.000	0.000	0.000	0.500	1.000	0.000	Std.
ETC	Yes	0.999	0.999	0.996	0.998	0.998	0.003	0	Dist.
ETC	Yes	0.999	1.000	0.997	0.998	0.998	0.003	0.000	Std.
DTC	Yes	0.766	0.454	0.973	0.619	0.844	0.026	0.283	Dist.
DTC	Yes	0.992	0.967	0.993	0.979	0.992	0.007	0.008	Std.
RFC	Yes	0.790	0.482	0.992	0.649	0.866	0.007	0.258	Dist.
RFC	Yes	0.999	1.000	0.997	0.998	0.999	0.003	0.000	Std.
GNB	Yes	0.766	0.455	0.993	0.624	0.852	0.006	0.287	Dist.
GNB	Yes	0.805	0.000	0.000	0.000	0.500	1.000	0.000	Std.

According to the results, the parameter set (standard, time phase, distance) and scaling affect models' performance rates. Scaling improves the performance of many algorithms. For instance, the random forest classifier has 0.734 accuracy, dramatically increasing to 0.999 with scaling. The same applies to the decision tree classifier's accuracy, which improves from 0.733 to 0.992 with scaling. Additional features such as distance and acceleration further improve performance. Logistic regression, extra tree classifier, and random forest classifier increase their accuracy and F1 score. In addition, these improvements also reduced FNR and FPR.

Comparing these results with those from previous datasets, it is evident that the random forest classifier consistently performs well, especially in terms of recall, making it a reliable choice across different datasets. However, its lower precision suggests a need for further tuning to reduce FPR. Logistic regression also shows consistent, average performance but struggles with recall, indicating a need for improvement in capturing true positive results. Gaussian naive bayes consistently underperforms, highlighting its limitations on a variety of datasets. Due to its poor performance, k-neighbour classifier should only be used with extensive tuning or other methods. As a result, extra tree classifier and decision tree classifier emerged as solid options with consistently high-performing metrics.

6.3. Comparison

The results of this thesis demonstrate a significant improvement in performance metrics compared to many of the existing academic works listed in Table 6.11.

For our dataset, the random forest classifier with scaling achieved an accuracy of 0.999, precision of 1.000, recall of 0.997, F1 score of 0.998, an FPR of 0.000, and an FNR of 0.003. This is superior to many other methods, including LSTM CAE by Chevrot et al. (Chevrot, 2022), which reported an accuracy of 85.7% and a recall of 54.9%. Similarly, the stack LSTM AE by Habler et al. had a recall of 95% and an FPR of 0.4%, which are lower compared to our model's performance (Habler & Shabtai, 2022). In terms of recall, Wang et al. reported a recall of 89% with an LSTM, whereas our best models achieved recall values as high as 0.997, showing a significant improvement (Wang et al., 2020). The sHDP HMM method by Tengyaoa et al. reported a high accuracy of 99.3%, which is slightly lower than our 0.999 achieved by the random

Table 6.11. Performance evaluation metrics

Number	Method	Acc.	Pre.	Rec.	F1S.	FPR	FNR
Our Result	RFC	96 %	100 %	91 %	95 %	8 %	0 %
Our Result	MLP	98 %	97 %	97 %	98 %	2 %	1 %
(Chevrot, 2022)	LSTM	85.7%	-	54.9%	73.8%	1.0%	-
	CAE						
(Habler & Shabtai, 2022)	LSTM AE	-	-	95%	-	0.4%	5%
(Wang et al., 2020)	LSTM	-	91.3%	89%	89.3%	-	-
(Li et al., 2020)	sHDP	99.3%	-	-	-	-	-
	HMM						
(Olive & Basora, 2020)	LSTM AE	-	-	61.5%	-	44.6%	-
(Fried & Last, 2021)	LSTM AE	-	-	-	-	-	-
(Mink et al., 2021)	SVM	-	80.2%	78.2%	79.2%	-	-
(Chen et al., 2019)	CNN TL	96.3%	-	-	-	-	-
(Ying et al., 2019)	DNN	96.6%	96.6%	96.6%	96.6%	-	0.43%
(Manesh, Kenney, et al., 2019)	ANN	98.3%	-	-	-	-	2.6%
(Manesh, Velashani, et al., 2019)	ANN	81%	-	-	-	-	30.9%
(Leonardi et al., 2017)	ANN	-	-	-	-	-	3.8%
(Habler & Shabtai, 2018)	GMM-HMM-	-	-	98%	-	-	4.5%
(Leonardi & Gerardi, 2020)	& KNN	-	-	-	-	-	0.1%

forest classifier (Habler & Shabtai, 2018). Our models also show an improvement over traditional methods like SVM used by Mink et al., which had a precision of 80.2% and a recall of 78.2% (Mink et al., 2021). The CNN transfer learning method by Chen et al. reported an accuracy of 96.3%, which is lower than our best models' performance (Chen et al., 2019). Comparing with the deep neural network (DNN) method by Ying et al., which reported an accuracy and F1 score of 96.6% (Ying et al., 2019), our models still show superior performance. Our models also outperformed the ANN method by Manesh et al., which reported an accuracy of 98.3%, and other ANN models in terms of accuracy and recall (Manesh, Kenney, et al., 2019). The KNN method by Leonardi reported an FNR of 0.1%, which is higher than the FNR of 0.003 achieved by our best models (Leonardi & Gerardi, 2020). Overall, our research results indicate that the random forest and extra tree classifiers, particularly with scaling and additional features,

provide robust and highly accurate anomaly detection in ADS-B data, outperforming many existing methods in the literature.

Upon examining the performance across all four datasets and adding the MLP model, several patterns and insights emerge. The extra tree classifier consistently demonstrates high performance, particularly in dataset 2 and dataset 3, where it maintains high accuracy, precision, recall, and F1 scores. This consistency makes it a reliable choice for diverse datasets. Similarly, the decision tree classifier exhibits strong performance, especially notable in dataset 1 and dataset 3. It achieves high metrics across all key performance indicators, showing its versatility under different conditions, including min max scaling and time phase parameters.

The random forest classifier stands out in dataset 1, achieving exceptional results with min max scaling and standard parameters. It also performs reasonably well in dataset 4, although with some issues related to precision. Overall, the random forest classifier is dependable but may require tuning to reduce false positives.

Gaussian naive bayes consistently underperforms across all datasets. It suffers from low accuracy, indicating a significant disagreement between predicted and actual values. Despite its high recall, which suggests it captures most true positives, this comes at the expense of precision, resulting in a high number of false positives. Logistic regression shows moderate performance across all datasets. It achieves high precision but struggles with low recall, indicating it avoids false positives effectively but fails to capture all true positives. This pattern is consistent, suggesting a need for improvements in recall to enhance its overall effectiveness. The k-neighbors classifier exhibits poor performance throughout, particularly struggling with recall and precision. Its metrics across all datasets suggest it is unsuitable for these scenarios without significant tuning or alternative approaches.

The MLP model, however, demonstrates outstanding performance with an accuracy of 0.982, precision of 0.977, recall of 0.975, F1 score of 0.976, and a ROC AUC score of 0.980. The false negative rate is low at 0.025, and the false positive rate is also low at 0.014. These metrics indicate that the MLP model is highly effective, showing a balance of high precision and recall, and thus excelling in capturing true positives while minimizing false positives.

Overall, the MLP model emerges as the top performer, surpassing other algorithms in terms of accuracy and balanced performance metrics. The extra tree classifier and decision tree classifier also remain strong contenders, with the extra tree classifier slightly leading. The random forest classifier demonstrates strong potential, especially with preprocessing steps like min max scaling. In contrast, gaussian naive bayes and k-neighbors classifier show significant limitations, making them less effective choices. Logistic regression, while precise, requires better recall for more balanced performance. This comprehensive comparison underscores the importance of selecting and tuning algorithms based on the specific characteristics of the dataset and the critical performance metrics for the given application.



7. CONCLUSIONS AND FUTURE WORK

Nowadays, with the ADS-B system becoming mandatory in civil aviation and the developments in the field of cybersecurity, the security of unencrypted and unsigned messages transmitted by ADS-B systems has become an important issue. Attack scenarios that can be made against the ADS-B system are increasing day by day. It seems that it will become a bigger problem with the recent events at Tartu airport and the increased use of unmanned aerial vehicles. For this reason, studies on the security of the ADS-B system have become even more critical. Considering these developments, this thesis focuses on detecting possible attacks against AD-B using machine learning-based systems. We evaluate ML and DL based models using these datasets and explored different parameter sets to optimize model performance.

Chapter 4 examines how attacks can target ADS-B systems, showing how threats such as spoofing, jamming and data injection can disrupt aviation security. We describe the methods used to accurately generate data that represents real-world attack scenarios. We also cover attackers' ability to eavesdrop, add, and delete messages. In this section, creating realistic attack data using simulations and real-world information is discussed. Finally, we share our attack dataset, which, to our knowledge, is the first comprehensive attack dataset. This dataset includes detailed structures, attack types, and key features. This information is crucial for future research and developing strategies to protect ADS-B systems.

Chapter 5 provides a comprehensive overview of our proposed anomaly and intrusion detection system for ADS-B. The overall architecture of the system is defined, highlighting design goals such as real-time detection, high accuracy, scalability and robustness. Its key components are described, including data collection, preprocessing modules, and integration of machine learning models.

The results of Chapter 6 reveal a clear distinction in the performance of different machine learning and deep learning models for detecting anomalies in ADS-B data. The random forest classifier and the extra tree classifier are the standout performers, both achieving an impressive accuracy of 0.999. This indicates that these models

are exceptionally reliable, correctly identifying almost all anomalies with minimal errors. The high accuracy also suggests robustness in handling diverse data patterns and potential attacks. The decision tree classifier, with an accuracy of 0.992, also demonstrates strong performance, though slightly below that of the random forest and extra tree models. This model's effectiveness shows that it remains a viable option, especially when computational simplicity and interpretability are valued.

Among the deep learning models, the MLP model achieves notable success with an accuracy of 0.981744. This result indicates that the MLP is highly capable in complex data environments, benefiting from its deep learning architecture to identify intricate patterns. The CNN and RNN models, with accuracy of 0.950728 and 0.923123 respectively, perform well but fall behind the MLP. These results suggest that while CNNs and RNNs are effective, they may require further optimization or larger datasets to match the leading models.

In conclusion, ML-based models, particularly random forest and extra tree classifiers, provide superior performance in detecting anomalies in ADS-B data, showcasing high accuracy, precision, recall, and low error rates. While DL-based models like MLP and CNN perform well. The decision tree classifier (no scaling) and random forest classifier emerge as top performers due to their high accuracy, precision, recall, and low error rates, effectively handling various attack types and large datasets, with the added benefits of interpretability and feature importance.

As a result, while ML/DL based anomaly detection methods promise for enhancing the security of ADS-B devices, it is essential to recognize and address these shortcomings and challenges. Data quality and quantity are two of the critical problems in anomaly detection systems for ADS-B. Research should focus on developing techniques for augmenting labeled anomaly data, such as active learning and transfer learning. While ADS-B datasets are imbalanced, future research can focus on robust anomaly detection algorithms for class imbalance. On the other hand, efficient training algorithms and optimization techniques should be developed to minimize computational complexity and memory requirements without sacrificing detection accuracy. Lightweight ML/DL architectures optimized for resource-constrained environments, such as edge devices or embedded systems, are

investigated.

Future research and development efforts aim to mitigate these issues and improve the effectiveness and robustness of intrusion detection systems in the context of aviation security. Future research in ADS-B anomaly detection systems involves comparing model results after tuning hyper-parameters and incorporating additional features. We will design hybrid approaches that combine statistical methods, rule-based systems, and machine-learning models to improve detection accuracy. We aim to develop adaptable systems that learn and evolve from new attack models and changing operational conditions. We will also improve the real-time detection capabilities of our models. We plan to leverage generative adversarial networks (GANs) to create realistic attack scenarios and increase model robustness and flexibility.

REFERENCES

- ADS-B Exchange. (2024). *Ads-b exchange*. Retrieved from <https://www.adsbexchange.com> (Accessed: 2024-05-20)
- Agresti, A. (2018). *Statistical methods for the social sciences*. Pearson.
- Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, 6(1), 37–66.
- Baek, J., Hableel, E., Byon, Y.-J., Wong, D. S., Jang, K., & Yeo, H. (2017). How to protect ads-b: Confidentiality framework and efficient realization based on staged identity-based encryption. *IEEE Transactions on Intelligent Transportation Systems*, 18(3), 690–700. doi: 10.1109/TITS.2016.2586301
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.
- Blåberg, A., Lindahl, G., Gurtov, A., & Josefsson, B. (2020). Simulating ads-b attacks in air traffic management. In *2020 aiaa/ieee 39th digital avionics systems conference (dasc)* (p. 1-10).
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Wadsworth International Group, New York.
- Caruana, R., Lawrence, S., & Giles, C. L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems* (Vol. 13, pp. 402–408).
- Cevik, N. (2024). *Adsb flight data*. Retrieved from https://github.com/nkcevik/ADSB_flight_data (Accessed: 2024-08-13)
- Cevik, N., & Akleyek, S. (2022). *Cyber security in aviation systems*. Nobel Academic Publishing Education Consultancy, Ankara, Türkiye. Retrieved from <https://dergipark.org.tr/en/pub/ijiss/page/13335>
- Cevik, N., & Akleyek, S. (2024a). Comparison of machine learning based anomaly detection methods for ads-b system. In K. Aida-zade (Ed.), *2nd international conference on information technologies and their applications (itta 2024): Abstracts*. Baku, Azerbaijan. doi: 10.54381/itta2024.abstracts
- Cevik, N., & Akleyek, S. (2024b). Sok of machine learning and deep learning based anomaly detection methods for automatic dependent surveillance- broadcast. *IEEE Access*, 12, 35643–35662.
- Chen, S., Zheng, S., Yang, L., & Yang, X. (2019). Deep learning for large-scale real-world acars and ads-b radio signal classification. *IEEE Access*. doi: 10.1109/ACCESS.2019.2925569
- Chevrot, T. (2022). Cae: Contextual auto-encoder for multivariate time-series anomaly detection in air transportation. *Computers & Security*, 116, 102652. doi: <https://doi.org/10.1016/j.cose.2022.102652>

- Federal Aviation Administration. (2010). *Automatic dependent surveillance-broadcast (ads-b) out performance requirements to support air traffic control (atc) service - final rule*.
- Federal Aviation Administration. (2024). *Faa database*. Retrieved from <https://www.faa.gov> (Accessed: 2024-05-20)
- FlightAware. (2024). *Flightaware*. Retrieved from <https://www.flightaware.com> (Accessed: 2024-05-20)
- FlightRadar24. (2024). *Flightradar24*. Retrieved from <https://www.flightradar24.com> (Accessed: 2024-05-20)
- Fried, A., & Last, M. (2021). Facing airborne attacks on ads-b data with autoencoders. *Computers & Security*, 109. doi: 10.1016/j.cose.2021.102405
- Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10), 2451-2471.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1), 3-42.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157-1182.
- Habler, E., & Shabtai, A. (2018). Using lstm encoder-decoder algorithm for detecting anomalous ads-b messages. *Computers & Security*, 78. doi: 10.1016/j.cose.2018.07.004
- Habler, E., & Shabtai, A. (2022). Analyzing sequences of airspace states to detect anomalous traffic conditions. *IEEE Transactions on Aerospace and Electronic Systems*, 58(3), 1843-1857. doi: 10.1109/TAES.2021.3124199
- Humphreys, T. (2015). *Statement on the vulnerability of civil unmanned aerial vehicles (uavs) and other systems to civil gps spoofing* (Technical Report). The University of Texas at Austin. (Accessed: 2024-05-20)
- John, G. H., & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 338-345.
- Johnson, J., Neufeldt, H., & Beyer, J. (2012). Wide area multilateration and ads-b proves resilient in afghanistan. In *Ieee integrated communications, navigation, and surveillance conference (icns)* (p. 1-8). doi: 10.1109/ICNSURV.2012.6218377
- Kaune, R., Steffes, C., Rau, S., Konle, W., & Pagel, J. (2012). Wide area multilateration using ads-b transponder signals. In *2012 15th international conference on information fusion* (p. 727-734).
- Kim, K.-W., Park, M.-G., Hishikawa, T., Han, J., & Lee, C.-G. (2012). Exploiting ultrasonic reflections for improving accuracy of indoor location tracking. *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, 87-95.
- Kim, Y., Jo, J.-Y., & Lee, S. (2017). Ads-b vulnerabilities and a security solution with a timestamp. *IEEE Aerospace and Electronic Systems Magazine*, 32, 52-61.

- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on artificial intelligence - volume 2* (pp. 1137–1143). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Kozovic, D., & Djurdjevic, D. (2021, 04). Spoofing in aviation: Security threats on gps and ads-b systems. *Vojnotehnicki glasnik*, 69, 461-485. doi: 10.5937/vojtehg69-30119
- Kozovic, D. V., Durdevic, D. Z., Dinulovic, M. R., Milic, S., & Rasuo, B. P. (2023). Air traffic modernization and control: Ads-b system implementation update 2022 - a review. *FME Transactions*, 51(1), 117-130. doi: 10.5937/fme2301117K
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lei, J., Jiang, R., & Wu, Z. (2023). Malicious ads-b data generation based on improved gan. In *Proceedings of the 2023 ieee 9th international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (ieee hpsc), and ieee international conference on intelligent data and security (ieee ids)* (p. 72-77). doi: 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00023
- Leonardi, M., & Gerardi, F. (2020). Aircraft mode s transponder fingerprinting for intrusion detection. *Aerospace*, 7, 30. doi: 10.3390/aerospace7030030
- Leonardi, M., Gregorio, L., & Fausto, D. (2017). Air traffic security: Aircraft classification using ads-b message's phase-pattern. *Aerospace*, 4(4), 51. doi: 10.3390/aerospace4040051
- Li, T., Wang, B., Shang, F., Tian, J., & Cao, K. (2020). Dynamic temporal ads-b data attack detection based on shdp-hmm. *Computers & Security*, 93. doi: https://doi.org/10.1016/j.cose.2020.101789
- Lyons, J. B., Ho, N., Fergusson, W. E., Sadler, G., Cals, S. D., Richardson, C., & Wilkins, M. (2016). Trust of an automatic ground collision avoidance technology: A fighter pilot perspective. *Military Psychology*, 28(4), 271-277.
- Ma, B., Tong, C., Zou, L., & Tian, M. (2021, 08). A tdoa localization method for complex environment localization. *Journal of Physics: Conference Series*, 2004, 012003. doi: 10.1088/1742-6596/2004/1/012003
- Manesh, M. R., Kenney, J., Hu, W. C., Devabhaktuni, V. K., & Kaabouch, N. (2019). Detection of gps spoofing attacks on unmanned aerial systems. In *2019 16th ieee annual consumer communications & networking conference (ccnc)* (p. 1-6). doi: 10.1109/CCNC.2019.8651804
- Manesh, M. R., Velashani, M. S., Ghribi, E., & Kaabouch, N. (2019). Performance comparison of machine learning algorithms in detecting jamming attacks on ads-b devices. In *2019 ieee international conference on electro information technology (eit)* (p. 200-206). doi: 10.1109/EIT.2019.8833789
- Mink, D., McDonald, J., Bagui, S., Glisson, W., Shropshire, J., Benton, R., & Russ, S. (2021). Near-real-time ids for the u.s. faa's nextgen ads-b. *Big Data and Cognitive Computing*, 5. doi: 10.3390/bdcc5020027

- Montgomery, D. C. (2017). *Design and analysis of experiments*. Wiley.
- Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on machine learning* (p. 78).
- Olive, X., & Basora, L. (2020). Detection and identification of significant events in historical aircraft trajectory data. *Transportation Research Part C: Emerging Technologies*, 119. doi: 10.1016/j.trc.2020.102737
- OpenSky Network. (2024). *Opensky network database*. Retrieved from <https://opensky-network.org> (Accessed: 2024-05-20)
- Pan, W., Feng, Z., & Wang, Y. (2012). Ads-b data authentication based on ecc and x.509 certificate. *Journal of Electronic Science and Technology*, 10, 51–55.
- Pantoja Viveros, C. A. (2024). *Analysis of the cyber attacks against ads-b perspective of aviation experts* (Master's Thesis). University of Tartu, Institute of Computer Science.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533-536.
- Schafer, M., Lenders, V., & Martinovic, I. (2013). Experimental analysis of attacks on next generation air traffic communication. In *Applied cryptography and network security* (Vol. 7954, p. 253-271). Springer Berlin Heidelberg.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1), 1929–1958.
- Strohmeier, M., Lenders, V., & Martinovic, I. (2015). On the security of the automatic dependent surveillance-broadcast protocol. *IEEE Communications Surveys & Tutorials*, 17(2), 1066–1087. doi: 10.1109/comst.2014.2365951
- Strohmeier, M., Schafer, M., Lenders, V., & Martinovic, I. (2014). Realities and challenges of nextgen air traffic management: the case of ads-b. *IEEE Communications Magazine*, 52(5), 111-118. doi: 10.1109/MCOM.2014.6815901
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288.
- Wang, J., Zou, Y., & Ding, J. (2020). Ads-b spoofing attack detection method based on lstm. *EURASIP J. Wirel. Commun. Netw.*. doi: 10.1186/s13638-020-01756-8
- Weaver, W., & Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379-423, 623-656.
- Wesson, K. D., Humphreys, T. E., & Evans, B. L. (2014, Mar). *Can cryptography secure next generation air traffic surveillance?* (Tech. Rep.) Austin, TX, USA: Radionavigation Secur. Res. UT Austin, Univ. Texas Austin.
- Wu, Z., Shang, T., & Guo, A. (2020). Security issues in automatic dependent surveillance - broadcast (ads-b): A survey. *IEEE Access*, 8, 122147–122167. doi: 10.1109/ACCESS.2020.3007182

Ying, X., Mazer, J., Bernieri, G., Conti, M., Bushnell, L., & Poovendran, R. (2019). Detecting ads-b spoofing attacks using deep neural networks. In *Ieee conference on communications and network security (cns)* (p. 1-9).



CURRICULUM VITAE

Nurşah Çevik received the B.S. degree in computer engineering from the Gazi University, Ankara, in 2015 and Ms. degree in computer engineering from Ondokuz Mayıs University, Samsun, 2018. From 2017 to 2019, she studied as a Project Researcher in national and international TÜBİTAK projects. From 2019 to 2021, she worked as an Research and Development Engineer with the IBSS. Until 2022, she has been an Platform Security Engineer with the HAVELSAN. Her research interests include post-quantum secure cryptographic protocols, cyber threats for avionic systems and security applications on avionic systems.

Contact Information

ORCID NO : 0000-0001-7066-3633

Publications

a- Journal Articles

1. Çevik, N., & Akleylek, S. (2024). SoK of Machine Learning and Deep Learning Based Anomaly Detection Methods for Automatic Dependent Surveillance-Broadcast. IEEE Access, 12, 35643-35662.
2. Akleylek, S., & Çevik, N. (2020). MaTRU-KE Revisited: CCA2 Secure Key Exchange Protocol Based on MaTRU. International Journal of Communication Systems (IJCS), Wiley.
3. Akleylek, S., & Kaya, N. (2017). Üç Terimli Polinomlar için Karatsuba Benzeri Çarpma Yöntemlerinin Araştırılması. Uluslararası Bilgi Güvenliği Mühendisliği Dergisi, 3(2), 22-32.

b- Book Chapters

1. Çevik, N., & Akleylek, S. (2022). Cyber security in aviation systems. Nobel Academic Publishing Education Consultancy, Ankara, Türkiye. Retrieved from <https://dergipark.org.tr/en/pub/ijiss/page/13335>.
2. Çevik, N., & Akleylek, S. (2022). Authentication with Keyboard Pattern. Nobel Academic Publishing Education Consultancy, Ankara, Türkiye. Retrieved from <https://dergipark.org.tr/en/pub/ijiss/page/13335>.

c- Conference Papers

1. Çevik, N., Akleylek, S., & Koç, K. Y. (2021). Keystroke Dynamics Based Authentication System. 6th International Conference on Computer Science and Engineering, UBMK 2021, Ankara, Turkey, September 16-17, 2021.

2. Çevik, N., & Akleylek, S. (2021). Modified Quantum Secure Identification Scheme. 1st International Conference on Electrical-Electronics and Computer Engineering, ICEECE 2021, Trabzon, Turkey, July 24-25, 2021.
3. Çevik, N., & Çelenli, H. İ. (2020). Authentication and Anomaly Detection System based on Behavioral Biometrics. 4th International Conference on Information Mathematics, ICOM 2020, İstanbul, Turkey, July 1-3, 2020.
4. Akleylek, S., & Çevik, N. (2019). Formal Analysis of MaTRU Cryptosystem. 4th International Conference on Computer Science and Engineering, UBMK 2019, Samsun, Turkey, September 11-15.
5. Akleylek, S., & Kaya, N. (2018). New Quantum Secure Key Exchange Protocols Based on MaTRU. 6th International Symposium on Digital Forensic and Security ISDFS'18, Antalya, Turkey, March 22-25.
6. Akleylek, S., & Kaya, N. (2017). Searching New Karatsuba-Like Polynomial Multiplication Algorithms For 3-Term Polynomials. 10th International Conference on Information Security and Cryptology, ISCTurkey 2017, pp. 123-127, Ankara, Turkey, September 20-21, 2017.