AIRCRAFT TRAJECTORY PREDICTION USING DEEP LEARNING

Master's Thesis

Enes ÖZÇELİK

Eskişehir 2025

AIRCRAFT TRAJECTORY PREDICTION USING DEEP LEARNING

Enes ÖZÇELİK

MASTER'S THESIS

Department of Electrical and Electronics Engineering

Programme in Theory of Circuits and Systems

Supervisor: Assoc. Prof. Dr. Emin GERMEN

(Co-Supervisor: Asst. Prof. Dr. Fulya AYBEK ÇETEK)

Eskişehir

Eskişehir Technical University

Institute of Graduate Programs

July 2025

This thesis has been supported by the project 24LÖT224, which is approved by the SRP Committee.

FINAL APPROVAL FOR THESIS

This thesis titled AIRCRAFT TRAJECTORY PREDICTION USING DEEP LEARNING has been prepared and submitted by Enes ÖZÇELİK in partial fulfillment of the requirements in "Eskişehir Technical University Directive on Graduate Education and Examination" for the Degree of Master's in Electrical and Electronics Engineering Department has been examined and approved on 31/07/2025.

Committee Members		**Title, Name and Surname**	**Signature**
Member	:	Assoc. Prof. Dr. Emin GERMEN	
Member	:	Prof. Dr. Daniel DELAHAYE	
Member	:	Assoc. Prof. Dr. Tomislav RADIŠIĆ	
Member	:	Asst. Prof. Dr. Fulya AYBEK ÇETEK	
Member	:	Asst. Prof. Dr. Emre AYDOĞAN	

Prof. Dr. Harun BÖCÜK
Director of the Institute of Graduate Programs

31/07/2025

SUPERVISOR APPROVAL

Master's student Enes ÖZÇELİK, whom I supervise, has completed this thesis titled AIRCRAFT TRAJECTORY PREDICTION USING DEEP LEARNING. According to my inspections, the work is scientifically and ethically appropriate for the student to the thesis defense exam.

Supervisor
Assoc. Prof. Dr. Emin GERMEN

ABSTRACT

AIRCRAFT TRAJECTORY PREDICTION USING DEEP LEARNING

Enes ÖZÇELİK

Department of Electrical and Electronics Engineering

Programme in Theory of Circuits and Systems

Eskişehir Technical University, Institute of Graduate Programs, July 2025

Supervisor: Assoc. Prof. Dr. Emin GERMEN

(Co-Supervisor: Asst. Prof. Dr. Fulya AYBEK ÇETEK)

With the continued growth of global air traffic, accurate trajectory prediction has become increasingly important for ensuring safety and improving efficiency, particularly within congested terminal airspaces. This thesis presents a data-driven, neural network-based approach for predicting the trajectories of arrival aircraft in the Istanbul Terminal Maneuvering Area, focusing on the spatiotemporal dynamics of this complex environment.

The model is trained using 2023 Automatic Dependent Surveillance–Broadcast data, which includes flight parameters such as position, altitude, and speed. A comprehensive preprocessing pipeline is applied, including noise filtering, interpolation of missing values, temporal segmentation, and grouping by callsign to generate time-series samples. Each sample is formatted in JSONL, with input vectors containing both aircraft-specific features and slot-based attributes of surrounding aircraft. This structure allows the model to incorporate local traffic complexity and proximity interactions.

A Multilayer Perceptron architecture is adopted to process the structured inputs. The model is tested on a scenario entirely excluded from training: 23 June 2024, the busiest traffic day of the year at Istanbul Airport, with 1,612 recorded flights. In this scenario, trajectories of six aircraft present in the TMA are predicted.

Performance is assessed using Dynamic Time Warping and Mean Absolute Percentage Error. The model yields an average horizontal deviation of 2.34 NM and a vertical error of 369 feet per step. These results suggest that the model is capable of generating reasonably accurate trajectory predictions under high-density traffic conditions, while considering the presence and dynamics of nearby aircraft.

Keywords: Air traffic management, Deep learning, Multilayer perceptron, Terminal maneuvering area, Trajectory prediction.

ÖZET

DERIN ÖĞRENME YÖNTEMI ILE UÇAK YÖRÜNGESI TAHMINI

Enes ÖZÇELİK

Elektrik Elektronik Mühendisliği Bölümü Anabilim Dalı

Devreler ve Sistemler Teorisi Bilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Temmuz 2025

Danışman: Doç. Dr. Emin GERMEN

İkinci Danışman: Dr. Öğr. Üyesi Fulya AYBEK ÇETEK

Küresel hava trafiğinin artmasıyla birlikte, uçuş yörüngelerinin doğru şekilde tahmin edilmesi, özellikle yoğun terminal hava sahalarında emniyetin sağlanması ve operasyonel verimliliğin artırılması açısından giderek daha önemli hâle gelmiştir. Bu tez çalışması, İstanbul Terminal Kontrol Sahası içerisinde varış yapan uçakların yörüngelerini tahmin etmek amacıyla, veri odaklı ve sinir ağı tabanlı bir modelleme yaklaşımı sunmaktadır. Model, bu karmaşık hava sahasının zamansal ve mekânsal dinamiklerine odaklanmaktadır.

Modelin eğitimi için, 2023 yılına ait Automatic Dependent Surveillance–Broadcast verileri kullanılmıştır. Bu veri seti; konum, irtifa ve hız gibi uçuş parametrelerini içermektedir. Veriler gürültü giderme, eksik değerlerin enterpolasyonu, zaman dilimlerine ayrılması ve çağrı işaretine göre gruplanması gibi adımları içeren kapsamlı bir ön işleme sürecinden geçirilmiştir. Her örnek JSONL formatında yapılandırılmış; giriş vektörleri, hem uçağa özgü öznitelikleri hem de çevredeki diğer uçaklara ait slot tabanlı bilgileri içermektedir. Bu yapı, modelin yerel trafik yoğunluğunu ve yakın hava araçlarıyla etkileşimleri dikkate almasını sağlamaktadır.

Modelleme için Çok Katmanlı Algılayıcı mimarisi tercih edilmiştir. Model, eğitim verilerinden tamamen bağımsız bir senaryo olan 23 Haziran 2024 tarihinde, İstanbul Havalimanı’nda yılın en yoğun gününde (1612 uçuş) test edilmiştir. Bu senaryoda Terminal Kontrol Sahası içinde bulunan altı uçağın yörüngeleri tahmin edilmiştir.

Modelin performansı, Dynamic Time Warping ve Ortalama Mutlak Yüzde Hata metrikleriyle değerlendirilmiştir. Ortalama adım başına yatay sapma 2.34 deniz mili, dikey sapma ise 369 feet olarak bulunmuştur. Bu sonuçlar, modelin yüksek yoğunluk koşullarında, çevredeki hava araçlarını da göz önünde bulundurarak, makul doğrulukta tahminler üretebildiğini göstermektedir.

Anahtar Sözcükler: Hava trafik yönetimi, Derin öğrenme, Çok katmanlı algılayıcı, Terminal kontrol sahası, Yörünge tahmini.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Dr. Emin GERMEN, for his invaluable guidance, constructive feedback, and continuous support throughout the entire research process.

I am also thankful to my co-supervisor, Asst. Prof. Dr. Fulya AYBEK ÇETEK, for her kind support and encouragement, especially during moments when motivation was most needed. Her thoughtful advice and sincere help during the experimental phases made a significant difference.

My heartfelt appreciation goes to the one for being a quiet strength behind it all. Your patience, understanding, and constant faith in me have helped me overcome many challenges. Thank you for walking beside me through every step of this journey.

Enes ÖZÇELİK

31/07/2025

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskişehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Enes ÖZÇELİK

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

4D	:	Four-Dimensional
ACC	:	Area Control Centre
Adam	:	Adaptive Moment Estimation
ADS-B	:	Automatic Dependent Surveillance–Broadcast
ANN	:	Artificial Neural Network
APP	:	Approach Control
ASM	:	Airspace Management
ATC	:	Air Traffic Control
ATCO	:	Air Traffic Controller
ATFM	:	Air Traffic Flow Management
ATM	:	Air Traffic Management
ATS	:	Air Traffic Services
BADA	:	Base of Aircraft Data
CNN	:	Convolutional Neural Network
DSS	:	Decision Support System
DTW	:	Dynamic Time Warping
ECAC	:	European Civil Aviation Conference
ETA	:	Estimated Time of Arrival
EUROCONTROL	:	European Organisation for the Safety of Air Navigation
FAA	:	Federal Aviation Administration
FIS	:	Flight Information Service
HMM	:	Hidden Markov Model
ICAO	:	International Civil Aviation Organization
IFR	:	Instrument Flight Rules
LSTM		Long Short-Term Memory
MAE	:	Mean Absolute Error
MAPE	:	Mean Absolute Percentage Error
MLP	:	Multilayer Perceptron
MSE	:	Mean Squared Error
NextGEN	:	Next Generation Air Transportation Systems
NM	:	Nautical Mile

ReLU	:	Rectified Linear Unit
RMSE	:	Root Mean Square Error
SESAR3JU	:	Single European Sky ATM Research 3 Joint Undertaking
TBO	:	Trajectory Based Operations
TMA	:	Terminal Maneuvering Area

1. INTRODUCTION

Air transportation plays an important role in connecting the world. For business, tourism, education, and other purposes, people use air transportation because of its speed and capability to reach other continents, among other reasons. Besides these utilizations, air transportation is also used for shipment service, military, or political purposes. The capability of air transportation systems makes it preferred, so the air transportation industry is growing, and the sector plays an important role in the global economy. According to the European aviation overview report of the European Organisation for the Safety of Air Navigation (EUROCONTROL), 10.2 million flights and 1.19 billion passengers were recorded in 2023, and these total flight operations released 87 million tons of CO_2, which is 12% than the previous year. The number of flights in 2019, 2020, 2022, and 2023 is shown in Figure 1, with comparisons (http-1). The COVID-19 outbreak caused a sharp decline in passenger and flight numbers. But following the outbreak, the number of flights has shown a continued recovery since the end of the COVID-19 pandemic. In Figure 1.1, the monthly recovery rate is the flight numbers rate between 2023 and 2019.

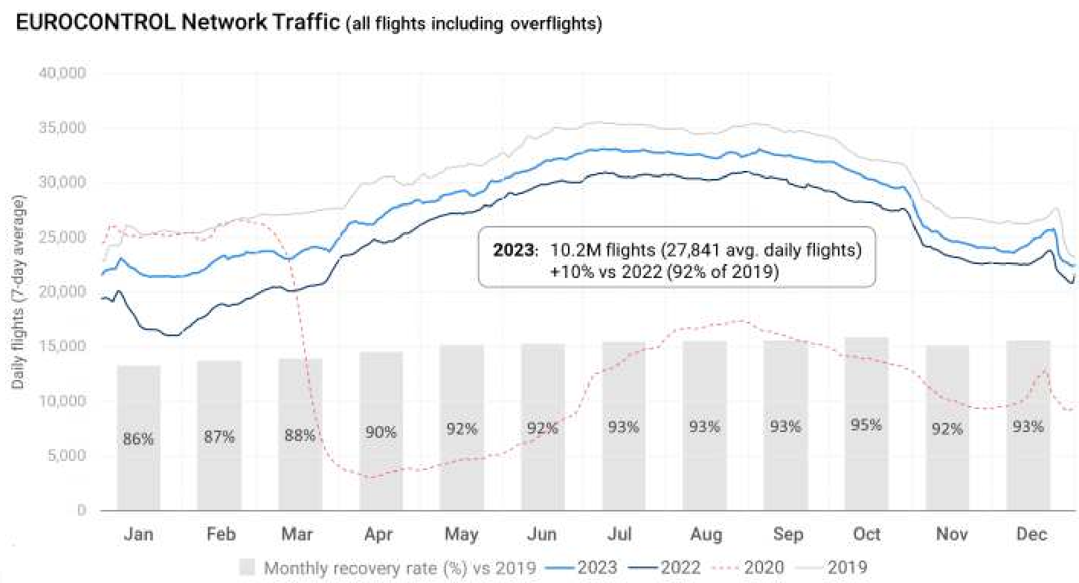

Figure 1.1. *Daily flight numbers and recovery rates in European air traffic by year (http-1)*

According to the seven-year forecast of EUROCONTROL, air traffic growth is expected to be limited by airport capacity in the major European Civil Aviation

Conference (ECAC) airports (http-2). Considering this prediction, Air Traffic Management (ATM) has played a crucial role in meeting the increasing demand. As expanding airport capacity is limited and costly, improving the efficiency of air traffic operations is crucial to meeting growing demand. With high demand and limited capacity, operational efficiency, flight safety, cost-effectiveness, and environmental impact concepts have played an important role. Many countries and organizations created projects to research future ATM concepts and current ATM solutions, such as in Europe, Single European Sky ATM Research 3 Joint Undertaking (SESAR3JU), and in the USA, Next Generation Air Transportation Systems (NextGEN). This growth in industry and demand for air transportation led to an increase in air traffic and fossil fuel consumption. Using fossil fuels releases polluting gases, and it contributes to global warming. To prevent air pollution and global warming, in Europe, EUROCONTROL and, Federal Aviation Administration (FAA) in the USA have sustainability regulations for achieving zero CO2 emissions.

2. LITERATURE REVIEW

ATM is an integrated system that aims to ensure the safe, efficient, and continuous flow of air traffic. According to ICAO (International Civil Aviation Organization), ATM is defined as "dynamic and integrated management of air traffic and airspace; provision of facilities to cover air traffic services, airspace management and air traffic flow management in a safe, economical and efficient manner, and realization of this management with the cooperation of all stakeholders, including air and ground-based functions."(ICAO, 2016, p. 1-4). ATM consists of 3 main parts: Air Traffic Services (ATS), Air Traffic Flow Management (ATFM), and Airspace Management (ASM). The structure of the ATM is shown in Figure 2. 1.

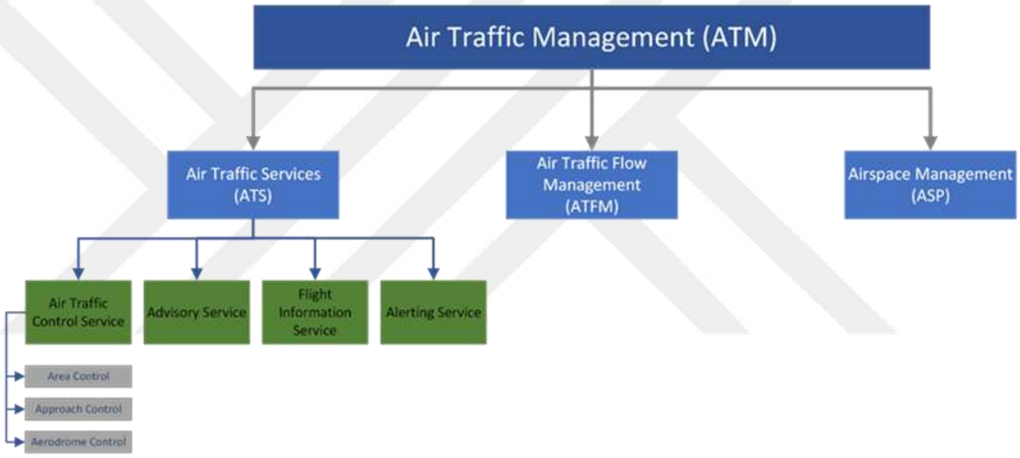

Figure 2.1. *Air traffic management structure*

2.1. Air Traffic Services

The main task of ATS is to prevent collisions between aircraft, to prevent aircraft from colliding with obstacles, to maintain and accelerate the orderly flow of air traffic, to provide information and useful advice for the safe and efficient operation of flights, to inform and assist relevant organizations for aircraft in need of search and rescue (ICAO, 2018, p. 2-2). Air Traffic Controllers (ATCOs) generally perform ATS in the Air Traffic Service Unit (ATSU). ATCOs ensure that traffic flows safely and quickly in the airspace, which is their responsibility.

2.1.1. Air traffic control services

Air Traffic Control (ATC) is a fundamental service that prevents collisions between aircraft and ensures the orderly flow of traffic. According to the ICAO, ATC services are classified into three categories: Aerodrome Control, Approach Control (APP), and Area Control (ICAO, 2018). Ensuring separation, sorting, and guidance at all stages of the flight is one of the services' main duties, from ground movements to navigation and landing (ICAO, 2016).

2.1.1.1. Aerodrome control service

The Aerodrome Control Service is responsible for ensuring the safety of aircraft during landing, takeoff, and ground movements within the control zone of an airport. The service is carried out by the control tower and includes the regulation of traffic on the runway, taxiways, and aprons. The ATCOs coordinate air and ground traffic through visual observation and radio communication.

2.1.1.2. Approach control service

The Approach Control Service is responsible for ensuring the safe management of flights during critical phases, such as climbing after take-off and descending before landing in approach airspace. In radar-controlled areas, air traffic controllers are responsible for the meticulous task of sorting and separating aircraft. This is achieved by issuing precise tactical commands to aircraft, such as instructions regarding speed, altitude, and heading. According to ICAO Annex 11, this service is essential for ensuring the orderly flow of Instrument Flight Rules (IFR) traffic (ICAO, 2018). In addition, according to ICAO Doc 4444, the Approach Control Service is responsible for organizing the entire process, from integrating departing traffic into the airway as quickly as possible to ensuring that landing traffic descends as efficiently as possible, all according to radar separation minima (ICAO, 2016).

2.1.1.3. Area control service

The Area Control Service is responsible for the safe management of high-altitude (en-route) flights. The execution of this service is usually the responsibility of Area Control Centers (ACC) or en-route centers. Controllers are responsible for the coordination of traffic flow across large airspace segments, ensuring separation both horizontally and vertically (ICAO, 2018).

2.1.2. Flight information service

The Flight Information Service (FIS) provides information, including meteorological data, temporary airspace restrictions, air traffic status updates, and changes to navigation aids to support flight safety. All flights, whether in controlled or uncontrolled airspace, can receive this service (ICAO, 2018). FIS does not provide guidance or discrimination, but more informed decisions are made by the pilot and ATCOs. According to ICAO Doc 9854, FIS is central to informed decision-making processes (ICAO, 2005).

2.1.3. Alerting service

The Alerting Service aims to notify the relevant search and rescue units in the event of a suspected aircraft emergency, loss of radio contact, or imminent accident for all controlled flights, IFR flights, and Visual Flight Rules (VFR) flights with a flight plan (ICAO, 2018). Air traffic units are responsible for providing this service and can initiate emergency procedures (ICAO, 2016).

2.1.4. Advisory service

The Advisory Service provides advisory information to IFR flights operating in uncontrolled airspace, helping them to avoid conflicts. It is only available in specific advisory airspaces and enables pilots to keep each other informed according to their flight plans (ICAO, 2018, p. §2.4).

2.2. Air Traffic Flow Management

The purpose of ATFM is to prevent air traffic congestion by considering air traffic demand and available capacity. ATFM regulates traffic density to prevent it from exceeding capacity in airspace or at airports. For this purpose, measures are taken to ensure that demand is spread over time and space, and traffic flow is planned in accordance with controller capacity. For example, under ATFM, flights may be assigned ground take-off slot restrictions (CTOT – Coordinated Time for Take-Off), or flights that meet specific criteria may be required to use certain routes (http-3). ATFM, Traffic demand-capacity balancing activities carried out in the pre-tactical phase (pre-flight planning process) provide the opportunity to intervene by foreseeing potential imbalances; thus, they contribute to the smooth and rapid traffic flow as much as possible while maintaining safety.

2.3. Airspace Management

According to ICAO Doc 9854 (2005), ASM is 'the process of identifying and implementing airspace options to meet the needs of air traffic management users. This process aims to enable flexible time sharing and the efficient utilization of airspace, while considering the requirements of all civil and military users.

The demand for air transportation is increasing rapidly worldwide. This growth trend is widening the gap between airspace capacity and demand, creating significant challenges for the civil aviation sector. Traffic growth is expected to continue in Europe, with the number of flights projected to exceed 16 million in the most likely scenario, Regulation and Growth, or nearly 20 million in Global Growth (http-4). This growth in air traffic demand creates congestion in air traffic management, leading to challenges in the field.

The ATM system is a complex system where aircraft operate for a common purpose, e.g., departure, landing, and cruise. These common purposes involve shared resources, such as the same runway, departure route, or landing path. The ATM system aims to optimize travel distance and fuel consumption of the aircraft while keeping system safety (Prandini et al., 2011). The ATM system seeks to ensure efficient and safe management of air traffic in all phases of flight. However, the rapid increase in air traffic demand and congestion presents ATM systems with various challenges. Heavy air traffic has negative effects, including increased controller workload, safety risks, delays, and reduced productivity.

The simultaneous presence of a significant number of aircraft within a sector or airport amplifies the quantity and intricacy of responsibilities that ATCOs are required to manage. ATCOs monitor the location of each aircraft, maintain separation between aircraft, organize landing/take-off sequences, assess weather conditions, and implement alternative plans when necessary. As traffic density increases, the cognitive load and stress level of controllers increase. A workload that is too high for an extended period can negatively affect controllers' performance; it can lead to a decrease in attention and situational awareness, and an increase in the probability of making errors. Controller workload is considered the primary constraint determining sector capacity; the key factor for capacity expansion is the amount of traffic that controllers can safely handle (Hah, Willems, & Phillips, 2006, p.50). The significant demand and continuous growth in aviation cause complex air traffic management. ATM, in increasing demand, is challenging for congested airports or airspaces. Increasing demand causes complex air traffic management, and due to this complexity, the efficiency and safety of airspace are decreased, and the workload of ATCOs increases.

Safety is the number one priority in air traffic management. ATCOs provide safety in the airspace by separating aircraft horizontally and vertically to avoid conflict. They also ensure safety and traffic flow by adjusting aircraft's heading, altitude, and speed and issuing directives to pilots (Nolan, 2004, p. 213). However, in congested airspace, ATCOs face increased workload, leading to stress and distractibility and making operations more prone to errors. Under high stress, ATCOs are more susceptible to small errors, which can have serious consequences in heavy traffic conditions. The need to find a balance between safety and capacity is a critical issue in ATM decisions. When taking steps to increase capacity, it is necessary to "carefully manage the safety-capacity boundary" to ensure that safety is not compromised (Fron, 2001, p. 25). For this reason, aviation authorities implement every capacity-increasing measure by supporting it with safety analyses.

High air traffic leads to delays and inefficiencies in airline operations. When an airport or sector experiences demand that exceeds capacity, aircraft are held on the ground before take-off or are forced to circle in holding areas in the air. These delays result in longer flight times and increased fuel consumption. Delays not only cost passengers and airlines time and money but also increase the environmental impact, as carbon emissions rise from burning fuel during waiting times. Since congestion-related inefficiencies negatively impact the economic and environmental performance of air transportation, air traffic management authorities are working on solutions to address the issue of delays. In air traffic, problems caused by density are interrelated. For example, in a system that has reached capacity limits, measures are taken to ensure safe operation, which slows down flights, resulting in delays and reducing efficiency. When efficiency decreases, economic costs increase. It also directly or indirectly affects the system's future operations.

Therefore, both short-term operational measures and long-term technological and structural improvements are being implemented to meet the growing traffic demand and optimize efficiency.

2.4. Air Traffic Management Solutions

Various strategies are being implemented to alleviate the problems brought about by air traffic congestion. These solutions are both tactical (short-term, within the flow of daily operations) and strategic medium- to long-term planning and investments).

ATFM involves planning the timing of flights to avoid congestion. For example, it prevents excessive traffic accumulation at airports or in sectors where capacity is limited by delaying the departure of planes. Network management units, such as the Eurocontrol Network Manager, balance demand and capacity across the entire European flight network, applying departure slot restrictions and route changes when necessary. For a sector that is about to reach its capacity, some flights are advised to plan alternative routes, or flights that will pass through that sector are made to wait on the ground. With ATFM, traffic demand is spread to a manageable level, delays are distributed throughout the system, and chaotic blockages are prevented, while maintaining safety (http-3). However, ATFM practices do not eliminate delays; they only make them manageable; delay costs may occur for passengers and airlines.

Airspace is divided into controllable subunits called sectors. Each sector is designed to enable a single controller (or team) to handle a particular volume of traffic safely. One solution when traffic increases is to split an existing sector into two or more, thus assigning a separate ATCO to each new sector and allowing more flights to be controlled simultaneously in the same geographical area. For example, a sector with expected high traffic at certain times of the day is divided into two as those hours approach, with two controller teams on duty in each sector. During quiet periods, such as at night, adjacent sectors can be merged, allowing large areas to be managed by a small number of controllers. This flexible sector management ensures that human resources are used effectively while maintaining safety during busy periods.

Developments in Communication, Navigation, and Surveillance technologies used in air traffic management have made significant contributions to capacity increase. For example, ground-based radar and satellite-based surveillance systems (i.e., Automatic Dependent Surveillance–Broadcast (ADS-B)) provide air traffic controllers with more precise and widespread traffic information. This allows them to reduce the separation minima between aircraft. Furthermore, using automatic flight plan processing systems and advanced decision support tools reduces the cognitive load on ATCOs, enabling them to manage more aircraft simultaneously (Hah, Willems, & Phillips, 2006). The NextGen modernization program in the USA and the SESAR projects in Europe are integrating more automation into the ATM system. Developed under the leadership of EUROCONTROL in collaboration with key aviation stakeholders, the European ATM Master Plan serves as the strategic roadmap for modernizing ATM across Europe. It outlines the operational, technological, and institutional changes needed to deliver an efficient, interoperable, and environmentally sustainable ATM system. As part of this transformation, the new Master Plan is a key enabler in guiding the European ATM ecosystem's transition into a digital, dynamic, performance-based environment (Figure 2.2). One of its core objectives is to inform strategic, data-driven investment decisions, for which robust modelling, predictive analytics, and advanced Decision Support Systems (DSSs) are essential. These tools play a critical role in identifying the most efficient and sustainable ways to implement emerging technologies, such as four-dimensional (4D) trajectory management, virtual centers, and system-wide information management, while ensuring the network is scalable, safe, and operationally resilient (http-4).

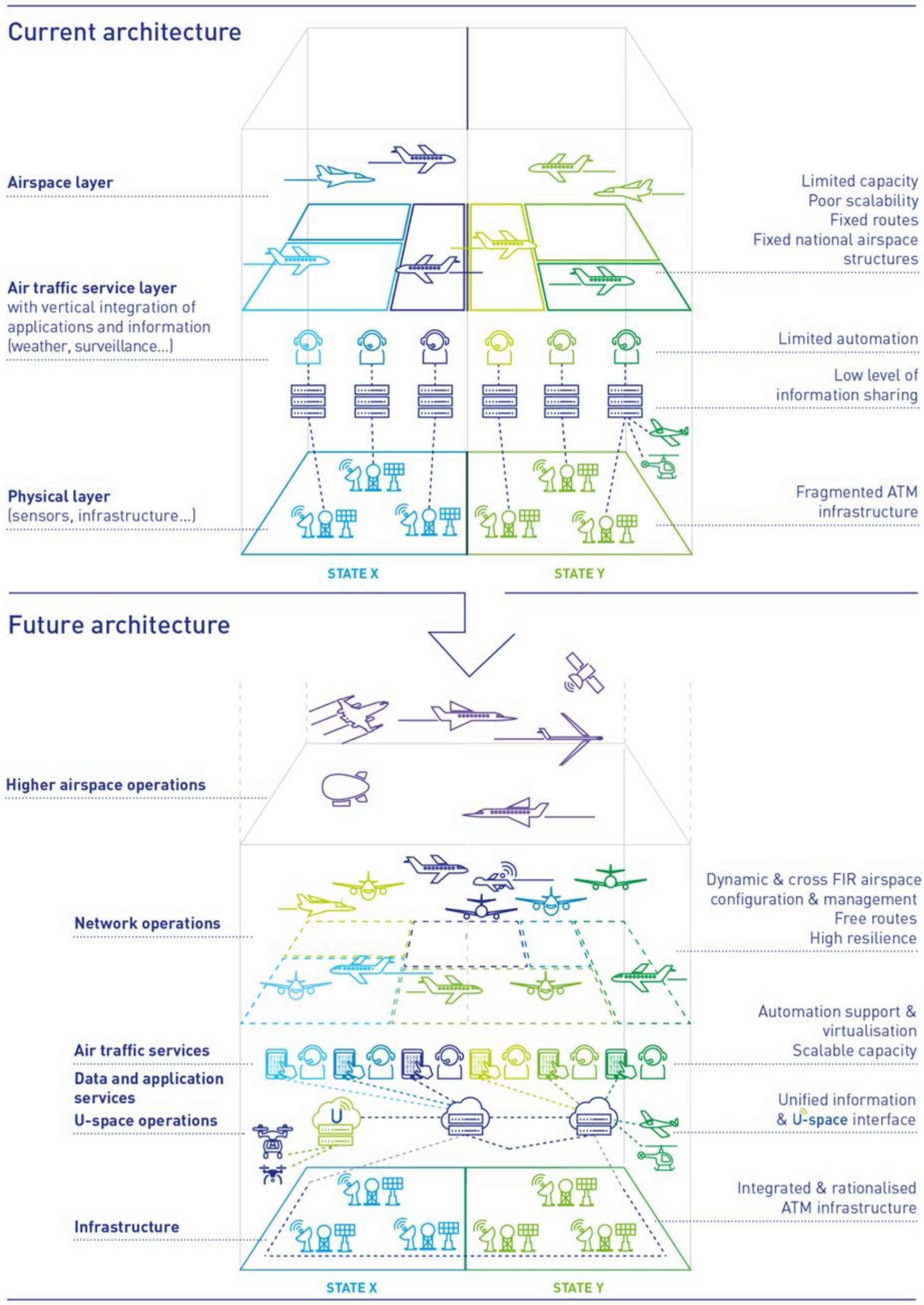

Figure 2.2. *Current and Future ATM system architecture concept (https://www.eurocontrol.int/article/european-atm-master-plan-and-eurocontrols-role-wider-atm-transformation)*

EUROCONTROL's mission is to facilitate this evolution by providing stakeholders with comprehensive guidance, system performance assessments, and simulation-based insights, enabling them to make informed choices regarding infrastructure and procedural development. The “Trajectory Based Operations (TBO)” concept is also being developed. This concept is based on 4D flight profile management, satellite-aided navigation, and information sharing among all stakeholders. TBO is an ATM concept that is developed to facilitate strategic planning by enabling the continuous sharing of a flight's four-dimensional (4D: latitude, longitude, altitude, and time) trajectory with all operational stakeholders. In TBO, the 4D trajectory of each flight is defined before take-off and updated according to conditions and operator inputs throughout the flight, then transmitted to ATC and other systems (FAA, 2021). In TBO, the estimated four-dimensional trajectory of each flight is continuously transmitted to ATCOs, allowing controllers to predict future aircraft positions and reduce the need for aircraft to wait, thus optimizing both capacity utilization and contributing to reduced emissions (http-6).

A DSS is a computerized system designed to help users make complex decisions. It has been developed in various fields, including meteorology, transportation, and defense (http-7). Decision support tools are essential for overcoming ATM challenges. DSSs play a crucial role in reducing cognitive load by facilitating planning, monitoring, and real-time decision-making. By providing ATCOs with timely and accurate information and predictive insights, these tools help them to maintain situational awareness, make informed decisions more quickly, and manage air traffic more safely and efficiently, even in high-stress situations.

Within the scope of TBO, the FAA provides decision support for strategic planning and capacity management through automation platforms (Traffic Flow Management System, Time-based Flow Management, Terminal Flow Data Manager) (FAA, 2022). National Aeronautics and Space Administration has also conducted DSS studies, including conflict detection and landing arrangements, using flight trajectory prediction algorithms (Gong & McNally, 2004). The Traffic Management Advisor system is a decision support tool that aims to regulate traffic flow and minimize holding requirements due to congestion. It also seeks to optimize the landing sequence and timing of flights approaching the terminal airspace within approximately 200 nautical miles, thereby regulating controller workload. This is achieved using a greedy optimization approach (Erzberger, 1990, p. 1).

The Enhanced Tactical Flow Management System, developed by EUROCONTROL, is a decision support system that manages air traffic in European airspace. It identifies potential overflow risks by monitoring aircraft movements throughout the airspace using flight plans, radar data, and real-time position information. It balances air traffic volume through slot planning, alternative route proposals (rerouting), and collaborative decision-making mechanisms. The system performs sector occupancy analyses using 4D trajectory forecasts updated at one-minute intervals, and coordinates traffic management between airports and air traffic service providers. With its data-driven approach, Enhanced Tactical Flow Management System increases operational efficiency and reduces delays in modern ATM systems (http-8).

DSSs aim is to assist ATCOs in their decision-making processes and increase safety and operational efficiency by reducing their workload in situations where air traffic management becomes complex, such as during periods of high demand, emergencies, and adverse weather conditions. In addition, DSSs provide recommendations to prevent overcapacity before operations begin, based on estimated metrics (such as trajectory and estimated time of arrival) and planned flights, to ensure a balance of demand and capacity.

Terminal Maneuvering Area (TMA) is a controlled airspace that is designed for big airports that have high-density traffic or areas that include more than one major airport. Due to the airspace structure of the TMA, the air traffic density is high, so it is difficult to use the airspace efficiently, and the management of traffic increases the workload of ATCOs. High-density air traffic in the TMA may occur, resulting in delays in the arrival and departure of aircraft, increased fuel consumption, noise pollution, and reduced airspace safety. Many approaches have been proposed: Aircraft Route Optimisation (Liang, 2018; Chida, Zuniga and Delahaye, 2016), Airspace Sector Optimisation (Granberg et al., 2019; Sidiropoulos, Majumdar and Han, 2018), etc. Solutions such as aircraft trajectory prediction, aircraft route planning, airspace capacity prediction, and optimization have been proposed to solve congestion in the TMA. In this thesis, a trajectory prediction model is proposed, which is the basis of DSSs, to increase ATM safety and efficiency and reduce the workload of ATCOs during and before operations in intensive and complex TMAs.

2.5. Aircraft Trajectory Prediction

Aircraft trajectory is a sequential time series of data. It is defined as 4D data by EUROCONTROL, comprising latitude, longitude, altitude, and time (EUROCONTROL, 2010, p. 10). ICAO defines aircraft trajectory as the motions, position, and other features of aircraft; latitude, longitude, altitude, time, speed, heading, and other attributes (ICAO, 2005, p. 4). Aircraft flights are arranged on predefined routes from the departure airport to the arrival airport. For some reason, it may deviate from the pre-defined route on the instructions of the ATCOs. Conflict resolution, aircraft separation, metrological events, emergencies, etc., are the reasons that cause the change of aircraft routes (Figure 2.3). ATCOs manage these operations by issuing instructions to pilots. In congested airspace, especially within TMAs, handling high traffic volumes under time pressure significantly increases their stress and workload. This can lead to reduced focus, decreased efficiency, and an increased risk of operational errors, compromising overall safety.

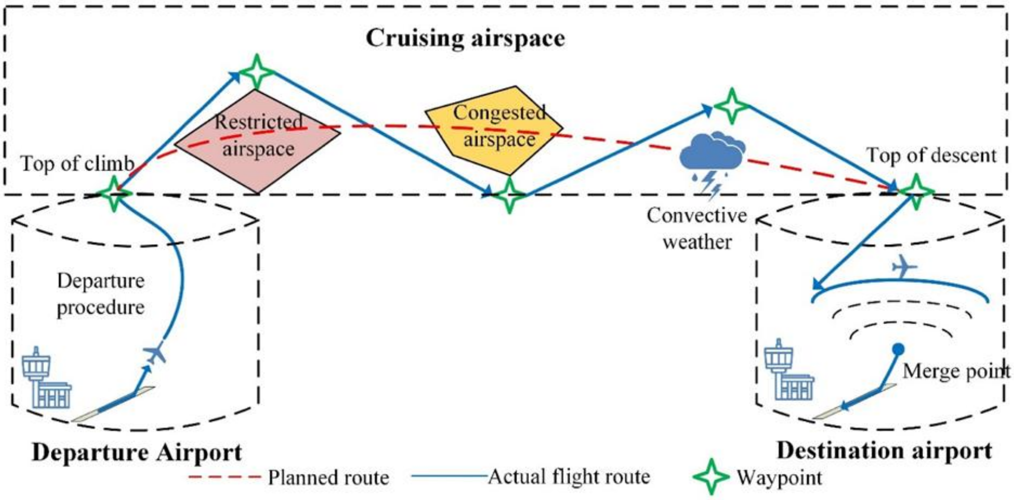

Figure 2.3. *The variation between the planned and actual flight phases of an aircraft (Zeng, W., et. al., 2022)*

Accurately predicting an aircraft's future trajectory is a key technology that enables more efficient air traffic management and overall operational performance (Song, Cheng, & Mu, 2012, p. 1). Accurate prediction of aircraft trajectories forms the basis of decision-making systems, such as sequencing of arrival and departure aircraft, aircraft flight flow management, conflict detection, and situational awareness of airspace (Zeng et al, 2022, p. 1). Trajectory prediction is essential for the proper functioning of DSSs, for forecasting complex traffic, and for pre- and post-operational analysis. It also provides operational predictions. It is also important for predicting lethal threats, such as conflict detection. Accurate trajectory prediction is crucial for ensuring the safety and efficiency of operations.

Trajectory prediction methods can be divided into three broad categories: state estimation-based models, kinematic models, and deep learning models.

2.5.1. State estimation-based model

Information such as the aircraft's current position, speed, and acceleration is used to predict the next state using state estimation-based methods. These approaches usually employ sequential state estimation techniques, such as the Kalman filter and the hidden Markov model. For instance, the Kalman filter and Markov models calculate the aircraft's position at the next instant based on the current state vector (Zeng, W., et. al., 2022).

Ayhan, S., & Samet, H. (2016) define airspace as a three-dimensional mesh network and attempt to predict aircraft routes by integrating weather information. The Hidden Markov Model (HMM) is used to predict the probability of aircraft passing on this network.

Lymperopoulos & Lygeros (2010) use sequential Monte Carlo (particle filtering) to predict the future positions of multiple aircraft. The motion of each aircraft is modelled stochastically; 4D estimation is performed on particles representing uncertainties, and the collision potentials of the system are analyzed.

Lin et al. (2019) used a state transition model based on historical data and similarity with a HMM to manage the 4D trajectories of flights in airspace. This method aims to estimate flight phases and make more accurate predictions using structures learned from similar trajectories.

The simplicity of the mathematical structure of state predictive models makes them effective for short-term predictions. However, in the long term, errors can accumulate over time because the system cannot detect unexpected aircraft maneuvers. Short-term predictions and real-time tracking in air traffic control are the typical applications for these methods (Yang, Z., et. al., 2023, p.3). Trajectory estimates produced by estimated model-based forecasting approaches may deviate from operational reality because they exclude human and tactical variables. These variables include control behavior, pilot preference, and airspace-specific traffic management strategies.

2.5.2. Kinematic model

Trajectory prediction based on kinematic and dynamic models considers aircraft-type-specific performance parameters, meteorological conditions, and flight intentions. Flight profiles are generated by models that are kinematic or dynamic, and these are based on the aerodynamic performance characteristics of the aircraft and the forces that are applied. Newton's laws of motion form the basis of the differential equations used to describe forces such as thrust, drag, lift, and gravity, as well as environmental influences such as wind, in these models (Zeng, W., et. al., 2022). The flight intention comprises operational preferences, such as the planned route of the aircraft, target altitudes, the speed profile, and the climb/descent strategies. Based on this information, the aircraft's future four-dimensional trajectory is estimated using physics-based calculations (Puranik, T. G., Rohani, A. S., & Kalyanam, K. M., 2022). Performance parameters are typically obtained from databases such as EUROCONTROL’s Base of Aircraft Data (BADA) and include characteristics of the aircraft, such as maximum climb rate, cruising speed, and fuel consumption (Nuic, A., Poles, D., & Mouillet, V., 2010).

Zhang, J., et. al. (2018) propose a real-time 4D trajectory estimation method. The method enables the model to adapt to changes in the aircraft's intention during flight, ensuring that it can respond accordingly to any alterations in the aircraft's trajectory or purpose. It develops an adaptive model that can update the estimation by detecting such changes. Aircraft motion is estimated using physical quantities such as speed, acceleration, climb rate, and turn radius, based on the classical kinematic model. The model automatically detects changes in flight intention and updates the model parameters accordingly. The flight intention is updated by looking at components such as target speed profile, target altitude, route deviations, and planned path, with data received from ADS-B or radar.

Trajectory prediction methods based on kinematic models provide short-term trajectory forecasts using physical parameters such as the aircraft's current location, speed, and acceleration. However, models do not consider operational factors such as airspace traffic density, air traffic control interventions, pilot decisions, or interactions with other aircraft. Kinematic models can generate quick and efficient results in the short term. However, they might not accurately represent real-world operations and can lead to significant prediction biases, especially in complex or congested airspaces.

2.5.3. Deep learning model

Machine Learning is a method that enables classification or prediction by identifying patterns within data. As computer technologies have advanced, Machine Learning techniques have become popular in various fields, such as finance, business, and genetics, due to their ability to recognize complex structures in data. However, the growing volume, variety, and complexity of modern data, ranging from audio and text to video, has exposed the limitations of traditional Machine Learning models and necessitated the development of more sophisticated architectures. The growing demand to handle high-dimensional, unstructured data has paved the way for Deep Learning to emerge as a subfield of Machine Learning. Deep Learning uses artificial neural networks with multiple layers to learn hierarchical representations (LeCun, Y., Bengio, Y., & Hinton, G., 2015, p. 1).

The aviation industry is facing increasing air traffic density and complexity. Deep Learning has emerged as a powerful tool for enhancing various aspects of ATM. Deep Neural Networks offer scalable, data-driven solutions to long-standing operational challenges, including trajectory prediction, conflict detection, delay forecasting, and controller workload estimation. Deep learning-based trajectory prediction studies performed at different airspace levels (terminal area, approach, and en-route) and structures in ATM are presented.

LeFablec and Alliot (1997, p. 524) designed a single hidden-layer, neural network that predicts changes in aircraft altitude using ADS-B/radar data sampled at 10-second intervals in the vertical plane (height–time). The data preprocessing stage involved modelling several inputs, including the current altitude of the flight, the altitude difference with the target flight level (Request Flight Level), and the vertical speeds from the previous step. The output is the expected vertical speed for the next time step. The model has three usage strategies: Standard, Sliding Window, and Two Networks. While the Sliding Window method provides real-time updates, the Two Networks method allows pre-flight plan prediction in two stages by estimating previous speeds for n steps. The performance of the model was assessed by comparing the average errors resulting from the Sliding Window method (ranging from 10 to 448 ft for 1–5 min forecasts) with those of the Standard and Two Networks methods (582–617 ft).

Wang, Liang and Delahaye, (2017) presented a model for short-term (tactical-level) 4D trajectory prediction using ADS-B flight data in the TMA of Beijing Capital International Airport. Each observation point of the flight contains three-dimensional location information (latitude, longitude, altitude), horizontal and vertical speeds, heading, temporal information, and descriptive metadata. During data preprocessing, missing or out-of-range observations are filtered, and each trajectory vector is resampled to a fixed length of 450 components. Principal component analysis is then used to reduce the dimensionality, after which similar trajectories are separated using the Density-Based

Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm. Only regular traffic is focused on; noisy data (such as holding patterns and dense vectoring) is eliminated. Multi-cells Neural Network (MCNN) models are developed for each cluster. Each cell acts as an individual predictor, learning the patterns of its cluster. The model's input is normalized position (X, Y, Z), direction, distances to the reference point, and heading angles (with sin/cos components). The estimated time of arrival (ETA) is the output. The training process was optimized using nested cross-validation, with Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) used as performance measures. According to the results, the proposed model produced more accurate ETA estimates than the classical multiple linear regression model when tested with the same preprocessing steps.

Wu, Tian, and Ma (2020, p. 1545) present an Multilayer Perceptron (MLP) neural network model for four-dimensional trajectory prediction (latitude, longitude, altitude, and time) using ADS-B data obtained from the Qingdao–Beijing route. Flight time variability is analysed using hierarchical and k-means clustering techniques. The data is then transformed into equal time steps using cubic spline interpolation. The network structure consists of a 15-neuron input layer, a 14-neuron single hidden layer, and a 4-neuron output layer. The input layer uses information obtained in the last three time steps, which includes time, latitude, longitude, altitude, and speed. The expected position (time, latitude, longitude, and altitude) at the next time step is predicted in the output. The model is trained using a dataset comprising approximately 91,000 training points and 13,000 test points. The performance of the model is evaluated using the specified metrics (maximum absolute error, MAE, and RMSE) and compared with the SVM model. The model achieves high accuracy with errors not exceeding one minute in terms of time and 50 meters in terms of altitude.

Li et. al. (2025, p. 1097) develop a sine chaotic mapping that is employed to enhance the sparrow search algorithm, optimizing the threshold parameters of the Back Propagation model (Sine-SSA-BP) for estimating four-dimensional flight trajectories (latitude, longitude, altitude, and time). The model uses four-dimensional flight information from ADS-B. During the data pre-processing process, the data are cleaned, and any missing points are completed using interpolation. During the training process, the Back Propagation model is optimized and improved using the Sparrow Search Algorithm (SSA). The SSA algorithm uses sine-based chaotic mapping to vary the initial weights and then globally optimize the weight and threshold parameters, with the best parameters obtained being passed to the MLP. The model was compared with three different versions: basic Back Propagation, SSA-Back Propagation, and optimized Sine-SSA-Back Propagation. Tests are performed according to the phases of the flight (descent, climb, and cruise), with the model's performance measured separately for each phase. The Sine-SSA-BP model demonstrated superior performance to both SSA-BP and basic BP in all flight phases. In the departure and cruise climbing phases, the prediction accuracy was 91.2% for Back Propagation, 99.4% for SSA-Back Propagation, and 99.6% for Sine-SSA-Back Propagation. During the cruise phase, the accuracy of the Back Propagation model reached 95.1%, that of the SSA-Back Propagation model reached 99.6%, and that of the Sine-SSA-Back Propagation model reached 99.7%. During the landing approach phase, the accuracy of the Back Propagation model stood out at 82.6%, compared to 97.8% for the SSA-Back Propagation model and 99.7% for the Sine-SSA-Back Propagation model. Sine-SSA-Back Propagation showed more stable and higher prediction accuracy in all flight phases.

Ma and Tian (2020) present a hybrid model (CNN-LSTM) for predicting the four-dimensional trajectories of civil flights navigating the Qingdao–Beijing route at the en-route stage using ADS-B data. The input data of the model includes time, latitude, longitude, altitude, velocity, and heading information for each flight observation. These features are normalized and completed with spline interpolation. The hybrid deep learning model combines 1D Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) structures. The aim is to use CNN layers to learn the spatial correlations in the flight path and LSTM layers to learn the temporal dependencies between successive time steps. The model uses input data, each of which contains six features, to predict the four-dimensional trajectory point (time, latitude, longitude, and altitude) at the next time step, over six time steps. The experiments show that the proposed CNN-LSTM model achieves significantly lower error rates than the classical LSTM and Back Propagation models and is highly successful in single-step and multi-step predictions. The model's overall prediction accuracy ranged from 91–95%.

The Social LSTM model was adapted by Xu, Z., et. al. (2021) to predict aircraft trajectories. The model was trained using ADS-B-based flight data from the Northern California TMA. Data is converted into an equidistant, cleaned, four-dimensional format (latitude, longitude, altitude, and speed) and is prepared using a pre-processing method that takes social interactions into account. Each aircraft has its own LSTM network, and a "social pooling" layer combines information from neighboring aircraft by sharing their hidden states. This structure enables each aircraft in a multi-aircraft environment to make more accurate predictions. These predictions are influenced by the aircraft's past data, as well as the dynamics of nearby aircraft. The performance of the architecture is compared with both single-aircraft traditional LSTM models and other baseline approaches. A significant reduction in horizontal and vertical deviation errors is demonstrated by S-LSTM, with a substantial improvement being provided, especially in scenarios with intense interactions. A collaborative trajectory estimation method that includes social interactions is presented by the study, which can be functional in both conflict avoidance and approach/departure sequencing processes in air traffic management.

An innovative boosted spatiotemporal deep learning ensemble approach is developed by Zhu, X., et. al. (2024) using real flight and radar data from Beihang University to model the impact of convective weather conditions on flight trajectory. During the pre-processing phase, a 'Relative Coordinate System' is designed instead of the traditional geographic coordinate system. The Relative Coordinate System is updated with each time step according to the planned route of the flight and the convective weather conditions encountered at that moment. Thus, both horizontal deviations and weather effects are included in the model's input in a time-adaptive manner. "REM blocks" extracted from radar echo maps along the flight path are detected as spatial inputs representing the density of the surrounding convective structure at each point on the aircraft and used for spatial feature extraction via CoordCNN. The spatial information from CoordCNN is combined with a stacked LSTM that processes historical flight data, enabling the effective learning of spatial and temporal correlations. The experiments used flights on the Beijing–Shanghai route, and the model produced significantly lower RMSE and deviation density errors than other methods (LSTM, GCRNN, CNNLSTM, Seq2Seq, and CoCLSTM_RCS), for both the entire test set and the convective weather subset. It demonstrated clear superiority in deviation, particularly in convective weather scenarios.

Bastas, A., Kravaris, T., & Vouros, G. A. (2020) present an approach to aircraft trajectory prediction from an imitation learning perspective, modelling real pilot and controller behaviours. The "expert" representations are made using the enriched 4D flight data from the Barcelona-Madrid route, which included additional features such as latitude, longitude, altitude, speed, and weather. Data is first separated into different trajectory patterns using a clustering method. Then, a Random Forest classification method is used to determine the new flight class. Finally, a separate Generative Adversarial Imitation Learning based model is trained for each class. While Generative Adversarial Imitation Learning learns a policy that mimics agent behavior by the expert distribution, it has become possible to predict the entire route with high accuracy in both pre-tactical (start of take-off) and tactical (mid-flight) stages. The model's performance is significantly superior to traditional machine learning and physics-based approaches, especially in long-term predictions, as evaluated using RMSE and track deviation measures (e.g., Dynamic Time Warping, Along-Track Error). Successful results covering the general flight cycle have also been obtained.

Zhang and Liu (2024) used real ADS-B data from December 2016 in Hong Kong airspace to predict the flight arrival trajectories of aircraft arriving in Hong Kong using an alternative Generative Adversarial Network model. A Conditional Tabular Generative Adversarial Network consisting of a generator and a discriminator is used. During sampling, a real frequency distribution is used instead of a log-frequency distribution, and a leave-one-out encoding is used instead of a one-hot encoding. Experiments show that Conditional Tabular Generative Adversarial Network has a lower MAE than LSTM, particularly for medium- to long-term predictions, and reduces the Mean Euclidean Distance by 81.5%.

3. NEURAL NETWORKS

Artificial Neural Networks (ANNs) are a model inspired by the human brain cell that makes predictions based on the relationship between data. An ANN is a computer model of biological neurons and their synaptic connections. Neural networks are formed by the connection of many interconnected neurons. The connection between nodes is represented by the weights that depend on the relationship between each node. Each neuron aggregates input signals by multiplying them by weights (a bias constant is generally added to the aggregation). The output of the node is obtained by passing the aggregation through a nonlinear activation function (Kumar and Garg, 2018, p.700).

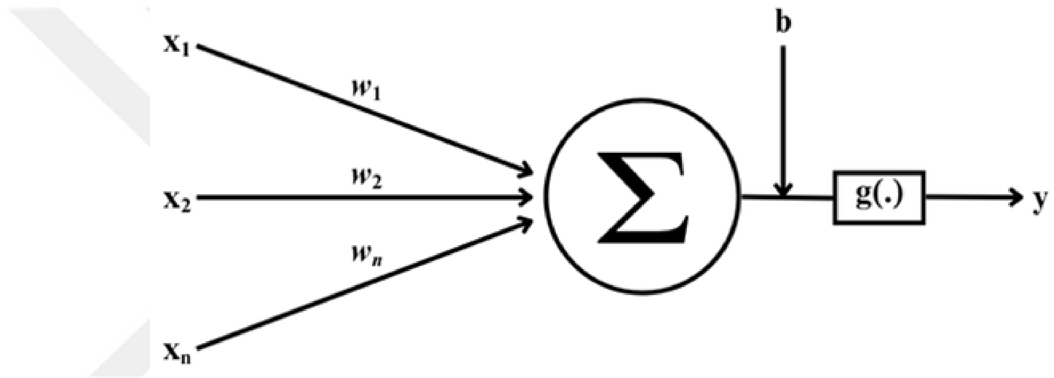

Figure 3.1. *Computational structure of a single artificial neuron*

Figure 3.1 shows the computational structure of a single artificial neuron. A neuron can have multiple input values, and the input is expressed as a vector:

$$\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T \tag{3.1}$$

The $x_n$values are numerical values coming from the external environment (input values) for the first layer, or from neurons in the previous layer for the next layers. Each input value is multiplied by a learnable weight coefficient w_n:

$$\boldsymbol{w} = [w_1, w_2, \cdots, w_n]^T \tag{3.2}$$

The net input value calculated by the neuron, z is expressed as follows:

$$z = \sum_{i=1}^{n} w_i x_i + b = \boldsymbol{w}^T \boldsymbol{x} + b \tag{3.3}$$

The bias term (b) is a scalar value. Although the bias value is a scalar, it does not seem to contain as much information as the weight vector. It plays a crucial role in the learning process of ANNs, as it is indirectly effective in each layer, neuron, and activation function of the network (Wang, Zhou, and Bilmes, 2019, p.1). The biases help the network shift decision boundaries and increase learning capacity. After calculating the net input, z is passed through a nonlinear activation function. The activation function determines the activation conditions of the neuron's output (y). Figure 3.2 illustrates the sigmoid function, hyperbolic tangent function (Tanh), rectified linear unit **(**ReLU) activation functions, mathematical equations, and graphs. The choice of activation function affects the network's ability to learn nonlinear patterns due to the differences in the activation function that activate the neuron.

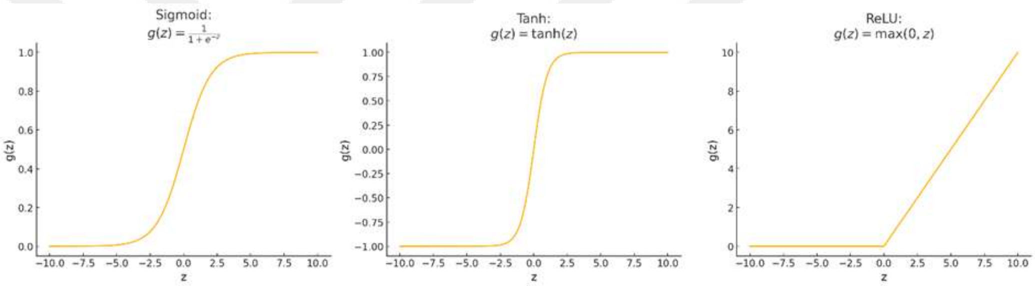

Figure 3.2. *Activation function examples*

A single-layer ANN (i.e., a single perceptron in the output layer) is a linear classifier. A single-layer network can separate two classes by dividing the input space by a single plane (line or hyperplane). If the data in the problem space can be divided by a single linear boundary, a single-layer ANN can learn this scenario successfully (Du et. al, 2022, p.5). Since single-layer ANNs are not sufficient for today's complex problems, structures consisting of multiple layers and using more neurons have started to be developed to solve such problems.

ANNs can comprise multiple artificial neurons organized in layers and may include one or more hidden layers as needed. Neurons in each layer take the outputs of the previous layer as inputs and send signals to the next layer by calculating their own weighted sums and applying activation functions. A hierarchical learning and representation mechanism is developed within the network. For simple ANNs to solve complex problems, using multiple neurons in a layer can split the input space into several linear pieces by learning different weights. Using multiple layers can generate much more complex and non-linear decision boundaries by learning the weighted combination of these intermediate outputs. If an ANN contains at least one hidden layer between the input and output layers, it is referred to as a Multi-Layer Perceptron model in the literature (Du et. al, 2022, p.22). Multi-layered structures can hierarchically divide the function to be learned or the decision boundary into parts through layers and capture different levels of features in each layer. Multi-layer ANN structures can hierarchically divide the decision boundary into parts through layers, capturing different levels of features in each layer. Increasing the number of layers in ANNs enhances the model's expressive power, allowing it to learn complex patterns that a single perceptron cannot resolve. However, adding many layers, depending on the problem, reduces the overall generalization ability of the model, creating a risk of memorizing the training data, a situation known as overfitting.

3.1. Training Process of Neural Network

MLP is a type of feedforward ANN that contains at least one hidden layer between the input layer and the output layer. MLP consists of an input layer, an output layer, and one or more hidden layers between the input layer and the output layer; neurons in each layer are connected to all neurons in the next layer with full connections (weights) (Figure 3.3). Each neuron calculates the weighted sum of the inputs it receives and produces an output by passing this sum through an activation function, and the calculation continues so that these outputs become inputs for the next layers. Due to the limitations of the single-layer perceptron (for example, the XOR problem cannot be solved), researchers aimed to increase the computational power of the neural network by adding a hidden layer. The multilayer networks developed for this purpose have become capable of solving nonlinear problems such as XOR, which the perceptron cannot solve in theory (Sejnowski, 2018, p.99).

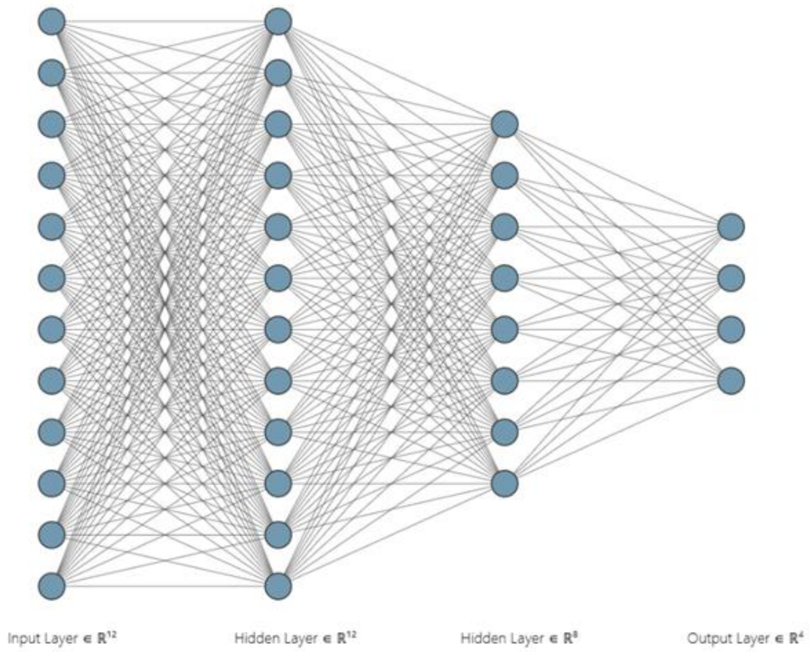

Figure 3.3. *Example of 2-hidden layer MLP*

The learning (training) processes of MLP networks are more complex compared to the perceptron. An MLP is generally trained using supervised learning, which involves data with known true expected outputs for inputs. With the discovery of the backpropagation algorithm, Rumelhart, Hinton, and Williams (1986, p.533) described a learning algorithm that adjusts the weights of each connection by propagating the output errors of a multilayer network back through the layers. The training process of MLP is divided into two stages: forward propagation and back propagation. In the forward propagation stage, the network produces an output in response to the given input. In the backpropagation stage, the error between the predicted output and the true output is calculated, and the weights of the network are updated to reduce this error. For MLP models, the most common training algorithm is the backpropagation gradient descent method, which efficiently implements the prediction-error calculation and weight update cycle for each training example. When MLP models are trained on the data for enough epochs (full training cycles), they can understand the complex relationships in the training data by updating their parameters (weights and biases). At the end of each epoch, the network's performance is measured using a loss function (e.g., Mean Squared Error); the goal is to reduce the loss to a lower value by updating the model's parameters.

3.1.1. Forward propagation

In the forward propagation, all weight and bias parameters of the network are initialized with random or specific values, and the input vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$ of a training example is applied to the input layer of the network. Each node in the input layer directly transmits the corresponding input value to the next layer. The weighted sums of the values coming from the input layer are calculated in the first hidden layer. The net input calculation for the jth neuron in the first hidden layer is shown in equation 3.4.

$$net_j^{(1)} = \sum_{i=1}^{n} w_{ji}^{(1)} x_i + b_j^{(1)} \tag{3.4}$$

Where:

- $w_{ji}^{(1)} \in \mathbb{R}$ is the weight from the i-th input neuron to the j-th neuron in the first hidden layer.
- $b_j^{(1)} \in \mathbb{R}$ is the bias term for the j-th hidden neuron.
- $net_j^{(1)} \in \mathbb{R}$ is the scalar pre-activation input for neuron j

The calculated net input value is passed through activation function of the neuron $f(\cdot)$ to produce the output $\mathrm{y_j^{(1)}} = \mathrm{f}\left(\mathrm{net_j^{(1)}}\right)$. This process is performed for all neurons ($H^{(1)}$) in the relevant layer , and the output vector of the first hidden layer, $\boldsymbol{y}^{(1)} = \left[y_1^{(1)}, y_2^{(1)}, \cdots, y_{H^{(1)}}^{(1)}\right]^T \in \mathbb{R}^{H^{(1)}}$ is obtained. The obtained output vector is given as input to the next layer, and the output value of each subsequent hidden layer is calculated similarly:

$$net_k^{(l)} = \sum_{j=1}^{H^{(l-1)}} w_{kj}^{(l)} y_j^{(l-1)} + b_k^{(l)}$$

$$y_k^l = f\left(net_k^{(l)}\right) \tag{3.5}$$

The forward propagation flow of a neuron in an ANN is shown in the equations 3.5. The data in the input layer is multiplied by the weight values and collected, and the bias coefficient is added to the sum of the products and fed into the first layer of the ANN model. This calculated net sum is passed through a nonlinear function, and the output values for the first hidden layer are calculated. The output of the first hidden layer is multiplied by the weight values and summed to be used as the input of the next hidden layer. The bias coefficient of the layer is added to this sum obtained for the hidden layer. The net sum obtained in the hidden layer is passed through the activation function of that layer, and the output for the hidden layer is obtained. The obtained output is used as input for the next layer. This process is ANN forward propagation and is applied layer by layer according to the ANN architecture. The general forward propagation formula for the lth layer is shown vectorially in equation 3.6.

$$\boldsymbol{y}^{(l)} = f\left(\boldsymbol{W}^{(l)}\boldsymbol{y}^{(l-1)} + \boldsymbol{b}^{(l)}\right) \tag{3.6}$$

Where:

- $\boldsymbol{y}^{(l)}$: Output of the layer
- $\boldsymbol{W}^{(l)}$: Weight matrix of the relevant layer
- $\boldsymbol{y}^{(l-1)}$: Output of the previous layer or input layer
- $\boldsymbol{b}^{(l)}$: The bias vector

Forward propagation is the basic operation in MLP models that processes input data across layers to produce output. In this process, neurons in each layer multiply the inputs they receive by certain weights, add a bias coefficient, and pass the result through a nonlinear activation function to the next layer. Forward propagation produces the model output for the loss function calculation during training and forms the basis for backpropagation.

3.1.2. Backpropagation

The backpropagation algorithm is an optimization technique that forms the basis of the learning process in ANNs, allowing for the systematic updating of weights and bias values to minimize the difference between the model's output and the true value. The backpropagation algorithm is based on the use of the chain rule to propagate the error signal back from the output layer of the network to the input layer, calculating the contribution of each parameter to this error (Rumelhart, Hinton & Williams, 1986).

In backpropagation, the difference between the model output (prediction) and the true (real) value is used to calculate the gradients of the weights at each layer of the network. During this process, a loss function such as Mean Squared Error (MSE) or Cross-Entropy is applied, and its choice depends on the type of model output and the problem being solved. Each weight parameter is updated to minimize the loss function according to the selected optimization algorithm (e.g., Stochastic Gradient Descent). As a result, the model can adapt to better represent patterns and make more accurate predictions.

The error calculation is calculated by the difference between the output values obtained by the neurons in the output layer of the ANN and the real values. Different loss functions can be used in regression estimations to measure the performance of the model, depending on the structure of the problem in error calculations. The choice of loss function varies depending on the sensitivity of the model to errors and its interpretability. The formulas of MSE, MAE, and RMSE loss functions are shown in Equation 3.7. MSE and RMSE square the errors, so they penalize large errors more. MAE is less sensitive to large deviations than MSE and RMSE, but is easier to interpret.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - t_i)^2 ,$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - t_i| ,$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - t_i)^2} \tag{3.7}$$

The input values $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T$ is applied to the neural network model and processed layer by layer with weights and the output $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_m]^{\,T}$ is obtained in the output layer. The difference between the value y predicted by the neural network model and the actual value $\boldsymbol{t} = [t_1, t_2, \cdots, t_m]^T$ is compared. The loss function is defined as a function of the difference between the value predicted by the model and the real value and is tried to be optimized by learning the model weights and bias value during the learning process.

During the training process of Neural Networks, the prediction error of each neuron in the output layer is calculated, and this calculated error is propagated backward through the network and used to update the model weights. Gradient calculations are calculated with the chain rule method to know in what direction and how much the model parameters (weights and bias) affect the loss to reduce the model estimation error function (equation 3.8).

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial net} \cdot \frac{\partial net}{\partial w} \tag{3.8}$$

- E is the loss function (e.g., MSE or MAE).
- $\hat{y}$ is the output of a neuron from the network.
- net represents the weighted total input of relevant neuron.
- w is the weight parameter to be learned.

Equation 3.8 expresses the effect of each weight parameter on the loss function ($\frac{\partial E}{\partial w}$) with a three-part chain derivative. The first part of the equation ($\frac{\partial E}{\partial \hat{y}}$) is the effect of network output on error. For the MSE function (equation 3.7), the influence of the loss function on the model output is determined by differentiating the loss function with respect to the network's output (equation 3.9). This derivative indicates how the weights should be adjusted: decreased (negative gradient) when the prediction exceeds the true value, and increased (positive gradient) when the prediction falls short.

$$\frac{\partial E}{\partial \hat{y}} = (\hat{y} - t) \tag{3.9}$$

The second part of the equation represents the change in output from the net input to the neuron, which means the sensitivity of the activation function. For the ReLU activation function, the change of the net input to the neuron according to the output is shown in equation 3.10.

$$\begin{gathered} \hat{y} = f(net), \\ f(net) = max(0, net), \\ \frac{\partial \hat{y}}{\partial net} = f'(net) = \begin{cases} 1, & net > 0 \\ 0, & net \leq 0 \end{cases} \end{gathered} \tag{3.10}$$

The derivative of the activation function in backpropagation determines how much neurons can learn, directly affecting the direction and magnitude of each weight update. The ReLU function provides efficient learning because it provides a constant derivative (one) in positive values, but there is no learning in the negative region.

The third and last part of the equation ($\frac{\partial net}{\partial w}$) indicates the effect of the weight on the net input of the neuron, and it is shown by w_j weight in equation 3.11. The contribution of the weight is equal to the value of the input to which it is attached.

$$\frac{\partial net}{\partial w_j} = \frac{\partial}{\partial w_j}\left(\sum_{i=1}^{n} w_i x_i + b\right) = x_i \tag{3.11}$$

This derivative is used in the backpropagation algorithm to calculate the effect of the error function on the weights. The input value of the neuron (x_i) on which the weight depends is the factor that determines the gradient. The impact of a weight on the error is determined by the size of the input to which that weight is connected.

Error signal (δ) for each neuron in the output layer:

$$\delta_k^{(L)} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial net} = (\hat{y}_k - t_k) \cdot f'\left(net_k^{(L)}\right) \quad (3.12)$$

The neurons in the hidden layer are not directly compared to the target values; the neurons in the hidden layer are backpropagated using the error signals of the neurons from the next layer. The calculation of the error signal of neuron j in the hidden layer l is given in equation 3.12.

$$\delta_j^{(L-1)} = f'\left(net_j^{(L-1)}\right) \cdot \sum_k w_{jk}^{(L)} \delta_k^{(L)} \quad (3.13)$$

- $\delta_j^{(L-1)}$ is the error signal of j neuron in the $L - 1$th hidden layer.
- $f'\left(net_j^{(L-1)}\right)$ is the derivative of the activation function.
- $w_{jk}^{(L)}$is the weight between jth neuron from $L - 1$th layer and kth neuron from Lth layer.
- $\delta_k^{(L)}$ is the error signal of k neuron in the Lth hidden layer.

The error signal of a neuron in the hidden layer is calculated by summing the error signals of all neurons in the next layer that are connected to it, multiplying them by the relevant weights, and multiplying by the activation derivative (equation 3.13). This process is applied sequentially backward for each hidden layer, so that all δ values up to the first hidden layer are calculated.

For the backpropagation of the bias term, the last part of equation 11 is taken as the derivative to calculate the change in the bias term (equation 3.14). Since the bias term is a constant, its derivative is one. Bias is used to enhance the learning capacity of the model and prevent the weighted sum from being zero-centered.

$$\frac{\partial net}{\partial b} = \frac{\partial}{\partial b}\left(\sum_{i=1}^{n} w_i x_i + b\right) = 1 \quad (3.14)$$

The error signal is a measure that shows the effect of the output of each neuron on the error in ANNs. The backpropagation algorithm uses error signals to calculate the effects of weights and biases on the total error function. The error signal serves as the main determinant in the process of optimizing the weight and bias parameters. While this process is calculated directly in the output layer in the form of $\frac{\partial E}{\partial net}$, the error signal in the hidden layers is associated with the error signals and weights in the forward layer. With this structure, all layers of the network become adjustable according to the error function.

3.1.3. Gradient descent

Gradient calculations show how and at what rate each weight and bias affects the error. This information is used in the parameter update process. The gradient descent algorithm updates the weights and bias to reduce the error. This update is accomplished by multiplying by a generally small constant called the learning rate (η).

$$\begin{gathered} w_i \leftarrow w_i - \eta \cdot \delta \cdot x_i \\ b \leftarrow b - \eta \cdot \delta \end{gathered} \tag{3.15}$$

The learning rate is a hyperparameter that controls the update size of the model parameters. If the learning rate is selected too small, the learning rate of the model slows down. The training time of the model increases, and there is a risk of getting stuck in a local minimum. If a large value of learning rate is selected, the model may overshoot the minimum of the error function and fail to converge.

The Adam (Adaptive Moment Estimation) algorithm is an optimization method that uses both momentum and learning rate to make the updating of parameters (weights and biases) more efficient during the training of ANNs (Kingma and Ba, 2014, p.1). In the Adam algorithm, gradients are calculated for each observation or mini-batch and used to obtain two different moments, that is, mean values. The first moment m indicates the direction and average of the gradients, and the second moment v indicates the magnitude of the gradients.

$$\begin{gathered} m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \frac{\partial E}{\partial w_t} \\ v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot \left(\frac{\partial E}{\partial w_t}\right)^2 \end{gathered} \tag{3.16}$$

Moments are updated depending on the moment values from the previous step and the β_1, β_2 hyperparameters. The hyperparameters β_1 and β_2 control how closely the moments remain dependent on past values. In the first step, the momentum values are taken as zero. This causes the momentum values to be smaller than the true average in the first steps. To prevent this deviation in the first steps, momentum values are magnified according to the deviation rate (equation 3.17).

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{3.17}$$

After the magnified moment values are calculated, they are used to update the model parameters:

$$w_{t+1} = w_t - \eta \cdot \frac{\widehat{m}_t}{\sqrt{\hat{v}_t - \in}} \tag{3.18}$$

Moment calculations for the bias term are also performed, and the moment values are magnified. The magnified moment values are used to update the bias term:

$$b_{t+1} = b_t - \eta \cdot \frac{\widehat{m}_{b,t}}{\sqrt{\hat{v}_{b,t} - \in}} \tag{3.19}$$

$\in$, used in bias and weight updates, is a very small constant used to prevent division by zero. The Adam algorithm provides a more stable learning process by preventing sudden changes using previous gradient information.

The forward propagation and backpropagation processes are repeated for each observation or mini-batch in the training dataset. A complete transition is called an epoch. If the loss function of the model is not at the desired level at the end of an epoch, the epoch number is increased. Selecting a low epoch number can lead to poor performance on both training and test data. However, training the model for too many epochs risks overfitting, causing the model to memorize the training data. In this scenario, while the model achieves high accuracy on the training data, its performance declines on the test data.

3.1.4. Regularization

In deep learning models, it is common for the model to overfit to the training data. Overfitting causes the model to perform well on training data, but poorly on test data used to test the model and not used during training. Some regularization techniques are employed to enhance the model's generalizability and boost its performance on data beyond the training set. Regularization methods aim to balance the learning process of the model by placing some constraints on the training process of the model.

3.1.4.1. Early stopping

Early stopping technique is a method used to prevent overfitting during the model training process. In this method, while the model is being trained, the losses in the training and test data are examined simultaneously in each epoch. In the early stopping technique, if the loss in the test data does not decrease or increase for a certain time, the model training is terminated. This particular time is called patience. If the test error does not decrease during the patience epoch, the training ends.

3.1.4.2. Dropout

The dropout method is introduced by Hinton et al., (2012, p.1) as a regularization method in which neurons in each layer of the model are temporarily disabled at a certain random rate to prevent over-learning. When a certain percentage dropout is applied, this percentage of neurons is randomly omitted in each iteration. The dropout method prevents the network from becoming dependent on specific neurons during training (Figure 3.4).

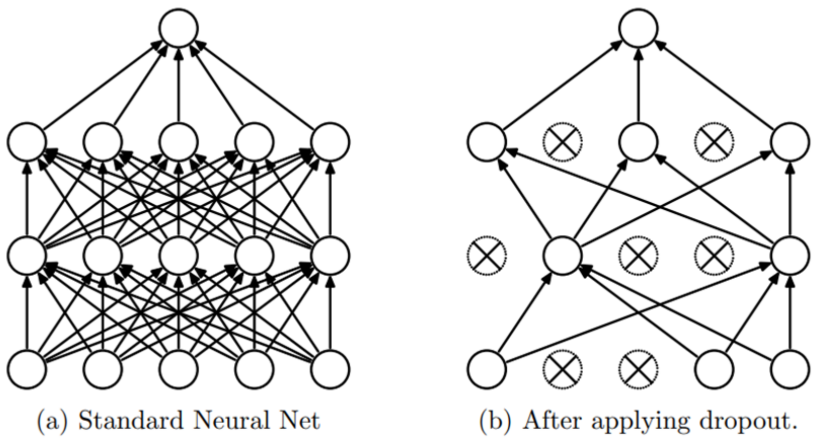

Figure 3.4. Example of a network with dropout applied (Srivastava et al., 2014)

3.1.4.3. L1 and L2 regularization

L1 and L2 regularization aim to control model complexity by penalizing the model's loss function depending on the load values. With these methods, overfitting is prevented by reducing the complexity of the model.

L1 regularization (Lasso Regularization), the absolute value sum of the weights is added to the loss function of the model as a penalty. Thus, some weight values are set to zero, and some values are not included in the learning process (equation 3.20).

$$E_{total} = E_{loss} + \lambda \sum_{i} |w_i| \tag{3.20}$$

Since some weights of L1 regularization are zero, the feature selection effect is created during the training process of the model. λ term is a hyperparameter used to adjust the effect of the regularization term.

L2 regularization (Ridge Regularization or Weight Decay), like L1 regularization, aims to reduce model complexity by penalizing the loss function. L2 regularization penalizes the loss function using the sum of the squares of the model weights (Equation 24). Since it uses a squared penalty, it makes the weights approach zero, but usually not exactly zero. L2 regularization increases the generalization ability of the model because it keeps the model weights at small values.

$$E_{total} = E_{loss} + \lambda \sum_{i} w_i^2 \tag{3.21}$$

In training an ANN model, regularization, early stopping, and dropout are frequently used to prevent overfitting and improve generalization ability.

4. AIRCRAFT TRAJECTORY PREDICTION MODEL AND PREPROCESS

With the rise of studies and applications in artificial intelligence, data-driven research has become increasingly important. Directly analyzing raw data can seriously affect the reliability of the analysis and the performance of the models because the data is incomplete, inaccurate, inconsistent, or irregular. Therefore, data preprocessing has a critical role in improving analysis and model performance.

Data preprocessing is the cleaning, transformation, and structuring of raw data to make it suitable for modelling. While data sets are being created, missing observations or missing values occur in data sets due to situations such as data transfer errors, permission and access restrictions, and shifts in time series during data collection.

Missing data may mislead the analysis results and cause the model to make biased or incomplete predictions. Two different strategies are proposed to handle the missing data: Deletion and imputation. Deletion is the removal of the missing data from the dataset. Removing the observation that has missing data is called listwise deletion. Removing only the observation with a missing value in the relevant analysis or modelling is called pairwise deletion.

The dataset may contain inconsistent and incorrect data. This data may have been entered incorrectly or in an incorrect format due to errors in the data collection process. This flawed data compromises the modelling and analysis processes, rendering the outputs unreliable. For inaccurate or inconsistent data cleaning, datasets are checked to detect erroneous data types (e.g., entering text into data with numeric values) and duplicate observations. Since abnormal values in the dataset can manipulate data analysis and models, outliers in the dataset are detected with Z-score and Interquartile Range. While z-score detects outliers based on the mean and standard deviation of the values in the data set, Interquartile Range performs outlier analysis by dividing the data set into quartiles.

The variables in the dataset usually have different value ranges. Bringing the variables to a standard scale makes it easier for models, such as deep learning models, to learn better and interpret the relationship between the variables. Especially in models that require parameter optimization, unscaled data can unbalance the learning process and negatively affect model performance. Variables are transformed to a standard range by various scaling and normalization methods. With the Min-Max Scaling method, variables are rescaled between zero and one. The Z-score normalization method transforms each value into a distribution with a mean of zero and a standard deviation of one. In cases where the distribution needs to be normalized, techniques such as logarithmic transformation, square root transformation, or Box-Cox transformation are used to reduce skewness.

Since deep learning algorithms generally work with numerical inputs, textual (categorical) data is converted into numerical form. If the categories in the variable have a particular order (for example, "low", "medium", "high"), the Label Encoding method is applied by assigning numbers such as 0, 1, 2 to these categorical values. However, suppose there is a variable with nominal values that do not specify an order (e.g., name, color, gender). In that case, the One-Hot Encoding method is applied by converting each category into a binary variable.

During the model creation process, the features in the dataset are selected appropriately according to the problem. Feeding models with unnecessary features increases computational costs and leads to model overfitting. For feature selection, significant features are chosen using statistical methods such as correlation analysis, Information Gain, Chi-square, and Recursive Feature Elimination. Training deep learning models using too many features introduces the issue of high dimensionality. Dimensionality reduction methods are employed to simplify the model's complexity. For example, Principal Component Analysis analyses the linear relationships between variables in multidimensional data and creates axes that represent these relationships. Dimensionality reduction is a tool used to make high-dimensional data more efficient, understandable, and manageable.

4.1. Istanbul Terminal Maneuvering Area

Istanbul Airport (ICAO code: LTFM) is located North-West Turkey, around 40 km north-northwest of the center of Istanbul. Situated within the borders of the Arnavutköy district, the airport is constructed near the Black Sea coast. Istanbul Airport has five runways (16R/34L, 16L/34R, 17R/35L, 17L/35R, 18/36), which are arranged in a combination of parallel and cross structures. Double parallel runways enable independent take-off and landing operations simultaneously. Istanbul Airport is the first in Europe to implement triple parallel runway operations as of 2025 (Devlet Hava Meydanları İşletmesi (DHMİ), 2025-a; DHMİ, 2025-b).

Istanbul Airport became operational in 2018. For the third consecutive year, Istanbul Airport finished 2024 at the top, with an average of 1,401 daily flights (http-9). Its location at the crossroads between Europe and Asia makes Istanbul Airport a popular destination for flights to Europe, the Middle East, Asia, and Africa.

Take-off operations are initiated by the Tower (TWR), while the aircraft's climb and exit from the Istanbul TMA are managed by the Approach Control (APP) unit. TMA is a controlled airspace that is designed for big airports that have high-density traffic or areas that include more than one major airport. Yeşilköy APP provides approach services and manages an airspace structure-oriented East and West. This structure comprises a total of eight arrival and five departure sectors. These sectors form a multi-layered air traffic control architecture that is separated according to direction and altitude. This architecture is designed to manage the high traffic volume safely, continuously, and efficiently. Istanbul TMA is the busiest terminal airspace region in Turkey, serving primarily Istanbul Airport and the major airports around it (e.g., Sabiha Gökçen Airport), and the flight operations taking place in this region. TMA ensures that aircraft are guided safely and regularly during critical phases such as climb after take-off and descent before landing (International Civil Aviation Organization, 2016).

The Istanbul TMA has a complex structure, including two major airports, active military airfields, and drone activity, as well as many restricted and prohibited airspaces. The high traffic demand in the region also significantly increases the workload of ATCOs. To support ATCOs in discriminating and predicting in this dense and variable traffic environment, we propose an aircraft trajectory prediction model, which forms the basis of DSSs. This model aims to predict the flight trajectories of aircraft arriving at Istanbul Airport within the Istanbul TMA airspace. To estimate the trajectory of aircraft, ADS-B data of aircraft arriving at Istanbul Airport in 2023 is obtained from OpenSky Network (Schäfer, M., et. al., 2014) using the traffic library (Olivie, X., 2019). Each flight consists of a time series of observations, including latitude, longitude, altitude, callsign, and groundspeed at each timestamp.

4.2. Data Preparation

A Filtering is applied according to the Istanbul TMA's geographical coordinates to detect trajectory data within the Istanbul TMA airspace. Data are filtered according to latitude, longitude, and altitude according to the flight level of TMA airspace. To analyze each flight within TMA separately, a unique identity is assigned to the data for each flight. A unique ID is assigned to each observation by looking at the time difference between the callsigns. If the callsigns of the observations are the same and the time difference is less than 2 hours, the same unique identification is assigned. From 2023 ADS-B historical data, 161,051 flights landing at Istanbul Airport in Istanbul TMA are detected. Time data for each detected flight is analyzed, and the latitude and longitude features of the flights are interpolated linearly as binary, and the altitude feature is interpolated as a single value to fill in the missing flight data. After filtering the flight data, the daily flight distribution by month is given as a heat map in Figure 4.1. The daily number of inbound flights at Istanbul Airport throughout 2023 is visualized by the heat map, with data presented by month and day. The horizontal axis shows the days of the month (1–31), and the vertical axis shows the months (January–December, i.e., 1–12). Each cell represents the intensity of flights for a specific day and month combination. The color scale shows high flight intensity in dark red tones, medium levels in light colors, and low intensity days in blue tones.

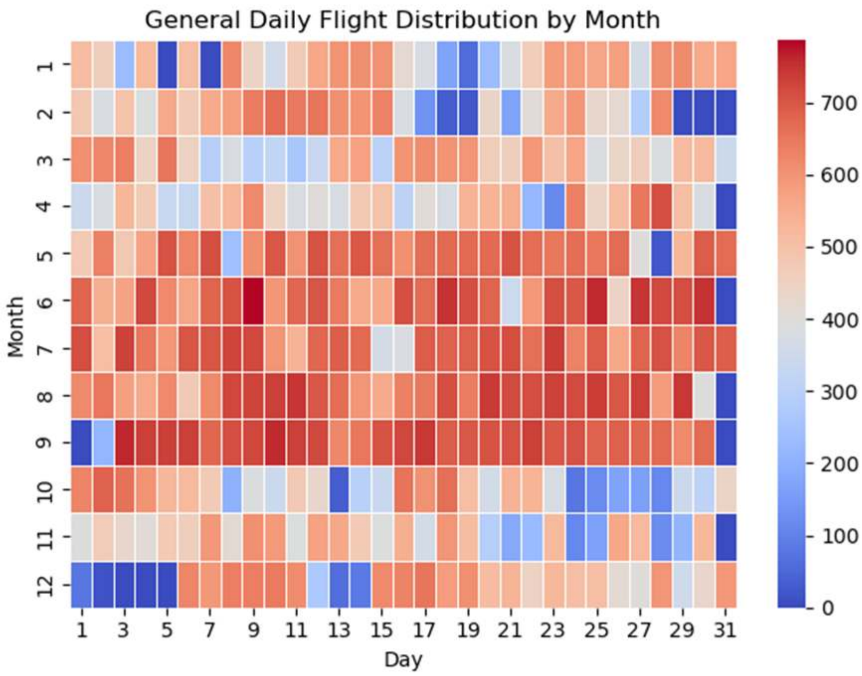

Figure 4.1. *Daily arrival flight distribution of Istanbul Airport in 2023 by month*

High flight density was observed almost every day during the summer months of June, July, and August. This period coincided with the tourist season, when air traffic increases seasonally, indicating increased pressure on TMA. Conversely, a significant decrease in flight numbers was observed, particularly towards the end of December. This may be attributed to the end-of-year and holiday period. The heat map clearly shows changes in the number of flights throughout the year and is an important tool for checking the effects of the time of year on the information used to train and test the model. Moreover, the analysis provides a basis for comparison of the model's response to days with high and low traffic.

Istanbul Airport has three runways and two approach configurations for take-off and landing, depending on the wind direction. For the arrival aircraft, North and South configurations are detected and added to the observations as a new feature by looking at the flight paths for each flight. After determining the runway configuration that the aircraft arrives at Istanbul Airport. The monthly distribution of flights in 2023 is presented in Figure 4.2 according to the runway configuration. Table 4.1. shows the flight numbers by month for the North and South configurations.

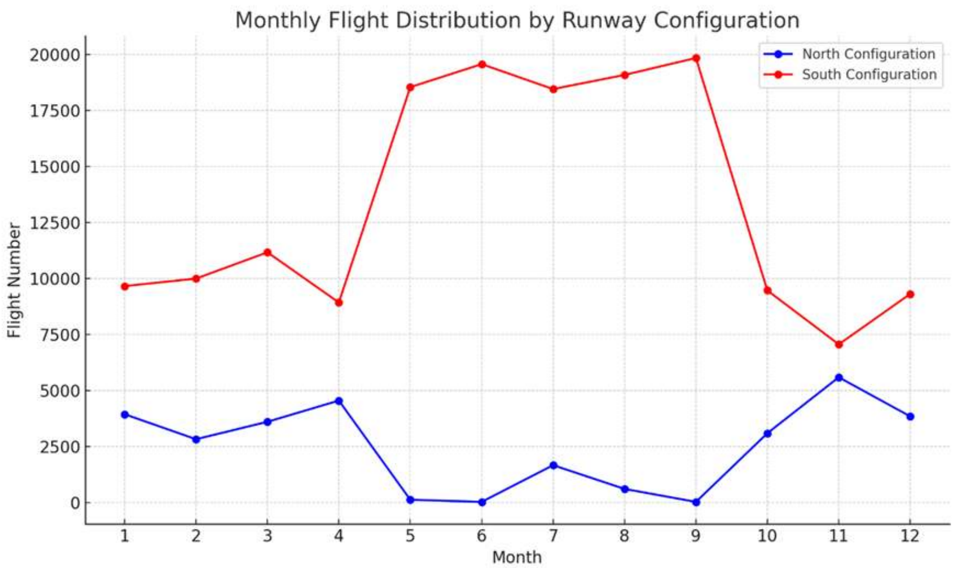

Figure 4.2. *Monthly flight distribution by runway configuration (North and South)*

The southern runways are the dominant landing site throughout the year. Approximately 91.4% of flights are in this configuration. The northern runway is used only 8.6% of the time, indicating that the operational preference is largely for the southern runway. The southern runways are used the most, especially in the months of May-September (months 5-9), with over 18,000 flights each. The busiest month is September with 19,833 flights. The northern runways have relatively higher utilization for aircraft arrivals in January, April, November, and December. The lowest utilization is evident in September (only 40 flights), indicating that the southern runway is almost exclusively used during the summer season.

Table 4.1. *Monthly flight numbers divided by runway configuration*

Month	North Configuration Flight Number	South Configuration Flight Number
1	3,949	9,663
2	2,835	9,996
3	3,612	11,168
4	4,558	8,931
5	137	18,533
6	41	19,559
7	1,677	18,447
8	619	19,078
9	40	19,833
10	3,100	9,468
11	5,597	7,064
12	3,855	9,311

ADS-B data is a data source that provides information about flights, including location, speed, and altitude. However, since this data is collected automatically directly from sensors, it may contain various errors and omissions. Inadequate signals for some flights, incorrect or incomplete transmission of call codes, and occasional loss or skipping of coordinate data may cause observation deficiencies or deviations in the analyzed data sets.

Due to the high variability of geographical coordinates, the point where the landing vehicles leave the TMA airspace has different geographical values (latitude, longitude, and altitude). To standardize the point at which the aircraft leave and for the ANN model to understand better the point at which the TMA airspace is left, a fixed value is assigned to the last coordinate values of the flights. The fixed values, latitude and longitude values, are determined by examining the average coordinates of the aircraft according to which approach configuration they are in. The altitude value is defined as 3000 feet for each flight's last observation.

ADS-B data contains the changing features (latitude, longitude, altitude, etc.) of the aircraft every second. Since the change in seconds is stochastic and the prediction of the time after one second is useless, a specific time interval is determined for the aircraft's trajectory prediction. A one-minute time interval is determined for each flight, and the observations in the ADS-B data are adjusted accordingly.

4.2.1. Scaling

Before training and testing the model, the dataset is scaled and encoded to enable the model's learning process to progress more quickly, effectively, and steadily by bringing different feature values of varying sizes into a common scale range. The Min-Max Scaler is used to transform feature values to a range of zero to one. Minimum and maximum values of each feature are determined, and the scaled values are transformed according to equation 4.1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

4.2.2. Encoding

Encoding is the conversion of categorical variables, strings, or textural features into numerical representations. One-hot encoding is used to convert the categorical runway configuration feature into numerical representations. The new features are generated for each category (North and South) instead of the runway configuration feature (Figure 4.3).

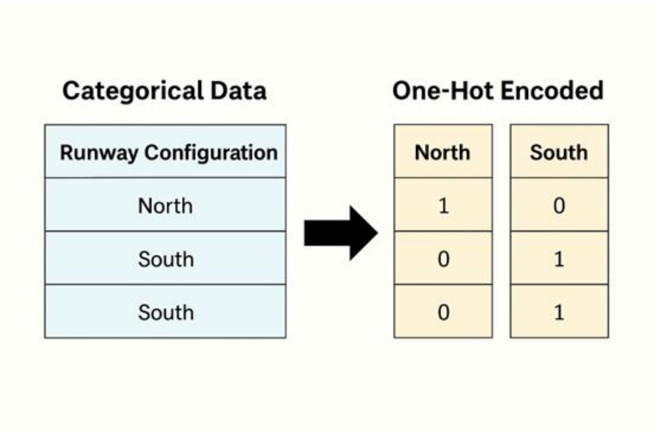

Figure 4.3. *Applying One-hot encoding to the runway configuration feature*

4.3. JSON Lines Format

In this thesis, data obtained from air traffic observations are structured in row-based JavaScript Object Notation Lines (JSONL) format. Each row contains a momentary observation of an aircraft, along with contextual information about the corresponding traffic status observation. The JSONL Dataset presents three principal components for each sample: input vector, output vector, and observation callsign. The observation callsign is also the target aircraft callsign and is used for analysis and matching input and output values. The observation callsign is not used in the learning process.

The input vector used by the model in the learning process consists of three basic components:

1. Slot-based aircraft location information
2. Temporal attributes
3. Runway configuration information

The input vector is denoted as x^i, and is defined in equation 4.2:

$$x^i = \begin{bmatrix} s_1^i \\ s_2^i \\ s_3^i \\ \vdots \\ s_K^i \\ t^i \\ r^i \end{bmatrix} \in R^{4K+4+2} \tag{4.2}$$

The $s_k^i \in R^4$, is the normalized slot vector of the kth aircraft. The vector is constructed as:

$$s_k^i = \begin{bmatrix} lat_k \\ lon_k \\ alt_k \\ gsd_k \end{bmatrix} \tag{4.3}$$

These values represent normalized latitude, longitude, altitude, and ground speed information, respectively. The first slot ($k = 1$) belongs to the target (output) aircraft to be predicted. The remaining slots represent other aircraft in the TMA airspace at the same time. The slot number (K) is constant, and the number is calculated according to the maximum number of aircraft in Istanbul TMA airspace at the same time in 2023. If there are fewer than K aircraft at the relevant time, the missing slots are filled with fixed values with coordinates depending on the runway configuration (dummy slot). The dummy slots are populated with synthetic values that represent the coordinates of the TMA airspace exit points, determined based on the runway configuration of the observation. These dummy entries are assigned a fixed altitude of 3000 feet and the ground speed of zero.

t^i is the four-dimensional vector that includes normalized month, day, hour, and minute time information (equation 4.4).

$$t^i = \begin{bmatrix} month \\ day \\ hour \\ minute \end{bmatrix} \tag{4.4}$$

The approach direction used by aircraft to land at Istanbul airport is represented as r^i (Equation 4.5):

$$r^i = \begin{cases} [1, 0] \ if \ North \\ [0,1] \ if \ South \end{cases} \tag{4.5}$$

The output vector y^i is a three-dimensional vector that includes normalized location features of the target aircraft in the next time step (equation 4.6).

$$y^i = \begin{bmatrix} latitude_{t+1} \\ longitude_{t+1} \\ altitude_{t+1} \end{bmatrix} \tag{4.6}$$

Each observation is converted to the specified JSONL format and prepared for training the model. The defined data structure integrates the spatial (s_k), temporal (t), and operational (r) dimensions into a single fixed-length tensor. With the data structure, the model is intended to learn not only its positional information but also the effects of other aircraft around it, operational conditions over time, and runway configuration changes while estimating the trajectory of the target aircraft. The data structure enables modeling of dynamic, multi-variable, and contextual air traffic information.

4.4. Trajectory Prediction Model

In this thesis, a MLP model is trained to predict the trajectories of aircraft landing at Istanbul Airport within the Istanbul TMA.

The MLP model's input layer neuron number adjusts according to the maximum aircraft number, at the same time multiplied by the number of aircraft features plus common features (date features and runway configuration). The input vector for the model's training and testing data is of fixed size. It contains components representing each observation's spatial, temporal, and operational context, in the JSONL format described in section 4.3. The first slot in this vector always represents the current instantaneous state of the target aircraft. This slot contains the following four normalized features: latitude, longitude, altitude, and groundspeed. The task that the model is expected to learn is to predict the position of this target aircraft at the next specified time step. Therefore, the first slot data in the input vector represents the situation at time t, while the model's output reflects the predicted positional value for $t+1$. In this study, we determined the time difference between t and $t+1$ as 1 minute.

The JSONL data structure, its design plays a crucial role in how the model learns the trajectory behavior of aircraft. Specifically, the JSONL format is structured to provide not only the features of the target aircraft but also contextual information about surrounding traffic, including the states and attributes of other aircraft in the TMA, temporal dynamics, and additional operational features. This rich representation allows the model to capture the complex interactions within the airspace and learn trajectory behavior in a more context-aware and temporally informed manner (equation 4.7).

$$\boldsymbol{x}^{(t)} = \left[\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)}, \cdots, \boldsymbol{s}_K^{(t)}, \boldsymbol{t}^{(t)}\boldsymbol{r}^{(t)}\right] \Rightarrow \boldsymbol{y}^{(t+1)} \tag{4.7}$$

Where:

- $\boldsymbol{s}_1^{(t)}$ is the target aircraft, represents the positional (latitude, longitude, altitude) and groundspeed information of the target aircraft at time t.
- $\boldsymbol{t}^{(t)}$ is the vector containing month, day, hour, minute information at time t.
- $\boldsymbol{r}^{(t)}$ is the vector containing runway configuration information at time t.
- $\boldsymbol{y}^{(t+1)}$is the vector containing the positional information (latitude, longitude, altitude) of the target aircraft at time $t+1$.

This time-shifted structure allows direct future predictions on the time series. The number of aircraft K is determined as 41, the maximum number of aircraft in TMA at time t, based on the data. A detailed analysis is conducted to determine the number of aircraft (K) parameter in the model's input structure, involving scanning the concurrent aircraft density in the TMA area throughout the time series. To achieve this, aircraft in the TMA airspace are identified at each time step across the entire dataset, with the number of concurrent aircraft at each time point being recorded. Scanning this resulting time series revealed that, at peak times, a maximum of 41 aircraft were present within the TMA simultaneously. Based on this analysis, the K value was set to 41 to ensure that the model received a fixed-length input vector at each time step. This number was then used as an upper bound on the model's spatial-temporal inputs. Consequently, the model was configured to process information from the maximum number of aircraft that could be present in the TMA at any given time.

The input layer of the model is fed with fixed-length vectors converted to JSONL format. The fixed length of the vectors is also the input layer of the model, and it calculates as the product of the maximum number of aircraft and the number of features per aircraft, added to the number of common features. This design allows the model to handle variable traffic scenarios in a fixed-size input format, enabling it to process both individual aircraft characteristics (such as position, ground speed) and shared contextual features (such as time of day or runway configuration) simultaneously.

The maximum number of aircraft at the same time is 41, and each aircraft has four features: latitude, longitude, altitude, and ground speed. The common features are date features and runway configuration of aircraft: month, day of month, hour, minute, and runway configuration (North or South). The model is defined as a MLP architecture consisting of four layers. The number of neurons in the input layer neuron number is calculated as 170. The training of the model is done using supervised learning, and the three-dimensional position information (latitude, longitude, and altitude) of the target aircraft at time $t + 1$ is used for the output layer. Due to the three-dimensional position information, three neurons are used in the output layer of the MLP model.

A structure consisting of three hidden layers is preferred, and the first two hidden layers consist of 128 neurons, and the last hidden layer consists of 64 neurons. The ReLU activation function is used in each hidden layer (Figure 4.4). The MLP model is trained with normalized data in JSONL format as 64 mini-batches.

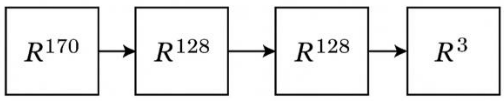

Figure 4.4. *MLP Architecture for Trajectory Prediction*

To evaluate the learning success of the model, a time-based discrete training-testing splitting strategy is applied. Considering the temporal dependency of the flight data, month-based splitting is preferred instead of random splitting of the data. This strategy provides a more realistic framework for evaluating the overall performance of the model over unobserved periods in the past. The training set includes data from January, February, March, May, June, September, October, and November of 2023. The test set is created from data obtained from April, August, and December of 2023. Separating the datasets by month aimed to cover the temporal variation in the dataset sufficiently and to ensure that the test set is independent enough to measure the generalization ability of the model. The train set consists of 2,438,591 observations of 124,539 different target aircraft, and the test set consists of 907,316 observations of 46,329 different target aircraft. When we look at the overall amount of data, training data accounts for approximately 73% of the overall amount of data, while test data accounts for approximately 27%.

The model is trained using the MSE loss function, which minimizes the difference between the true and predicted positions, ensuring accurate trajectory estimation. The Adam optimization algorithm is employed to efficiently update the model's parameters during training. The initial learning rate is set to $\eta = 10^{-3}$, and the learning rate is decreased by a factor of 0.5 every 10 epochs during the training period. The training process continued for 100 epochs, and the performance of the model on the test data is evaluated at the end of each epoch. During evaluation, MAE and MSE metrics are calculated and recorded separately for both training and testing datasets. In this way, the change in model performance within each epoch could be systematically monitored. At the end of each epoch, the model's weights are recorded. In addition, the differences between the predicted and actual values obtained during the evaluation process are visualized for each target variable.

4.4.1. Model results

The performance of the developed MLP model is evaluated comprehensively using both numerical error metrics and visual analysis. To reveal the model's overall performance, the values for the training and test sets are analyzed for each epoch, and the model's ability to generalize is examined. The estimated and actual values of the output attributes (latitude, longitude, and altitude) in the test data are visualized and compared to observe changes throughout the epochs. This allowed for a detailed examination of the model's success in capturing the target values during the learning process. To evaluate the model's performance, various error metrics obtained during the training and testing processes are examined. Measures such as MAE, MSE, RMSE, and R^2 score are used in this context (Figure 4.5). The general tendency of the model in the learning process and its generalization capacity are analyzed through these metrics. Separate comparisons are made of training and testing performances, and situations such as overfitting or underfitting are evaluated.

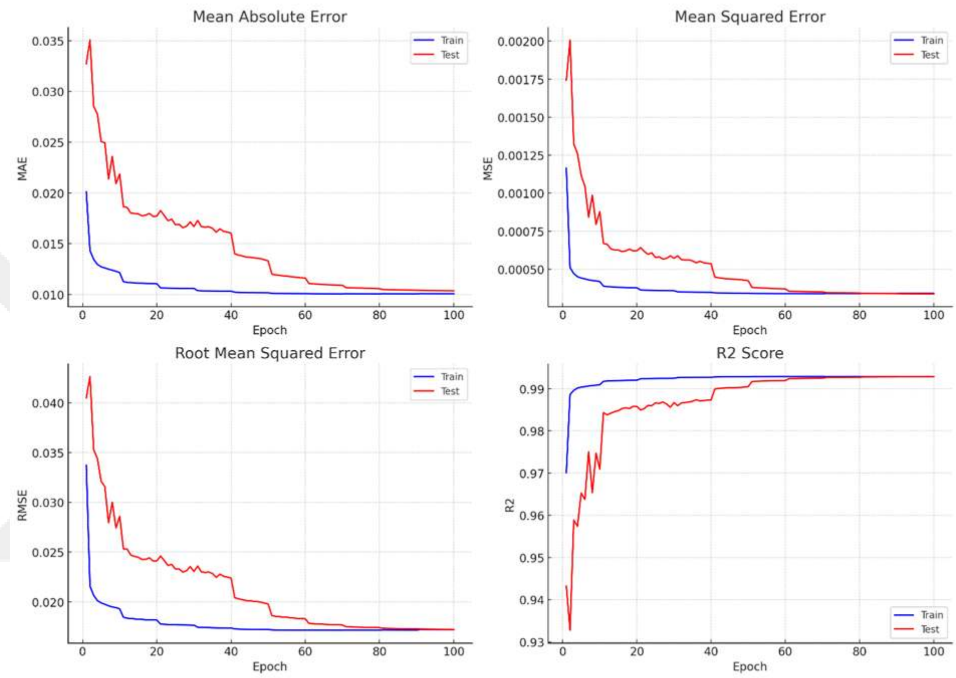

Figure 4.5. *MLP model train and test performance metrics (MAE, MSE, RMSE, and R^2)*

Analyses of the metrics used to evaluate the model reveal that its learning process progresses in a balanced and effective manner. The MAE shows, on average, how far the model's predictions at the end of each epoch are from the actual values. In the initial epochs, the training and test errors are both relatively high (approximately 0.020 and 0.033, respectively). However, a rapid decrease is observed in the first ten epochs, after which the trend becomes more horizontal. The parallelism between the training and test curves indicates that the model has a high generalization capacity and is not showing signs of overfitting. The test error did not increase again in subsequent epochs, indicating stable learning.

Although the MSE is initially high, it decreases sharply, especially during the first 5–10 epochs. Thereafter, the training and test loss values continue to decline slowly but steadily, with no increase in test loss being observed, which would indicate overfitting. This demonstrates that the model remains stable throughout the learning process and continues to generalize effectively.

The RMSE value provides information about the error distribution, since it shows the standard deviation of the prediction errors. In the training data, the RMSE value starts at approximately 0.034 and decreases to around 0.014. Similarly, the test RMSE value starts at around 0.040 and decreases to approximately 0.025. The similarity of the training and test curves indicates that the model fits the dataset well and that the risk of deviation is low. Additionally, the absence of sudden jumps in the RMSE curve suggests that the model is learning in a balanced manner.

The R^2 score is a measure of how much the model's predictions explain the data variance. In the training data, the R^2 value begins at 0.97 and increases to 0.995. In the test data, it begins at 0.94 and stabilizes at approximately 0.97. These high scores suggest that the model effectively explains the variance in both the training and test data. According to the curve profile, the R^2 scores remain consistently high and stable, confirming that the model is balanced and high performing.

In general, all evaluation metrics show significant improvement, particularly between epochs 1 and 30. The error metrics (MAE, MSE, and RMSE) decrease fastest in this early period. A slower but steady improvement continues between epochs 30 and 85, indicating that the model is still learning. After approximately epoch 90, all metrics, especially the R^2 score, start to follow a stable trend, indicating that the model has reached a saturation point in terms of performance. This suggests that the model has successfully learned the underlying patterns in the data, and that further training would not lead to significant improvement — in other words, the model had reached an optimal level of generalization without overfitting. The model achieves the Mean Absolute Percentage Error (MAPE) value of 1.47% on the test data, to which the inverse transform is applied at the 100th epoch. This is highly significant when evaluating prediction results on a real-value scale. In this case, the error rate is calculated based not only on normalized values, but also on the actual location (latitude, longitude, and altitude) units. This makes it possible to give the model outputs direct physical meaning and interpret the model performance more accurately in the context of the application. The model's accuracy in spatial predictions is indicated by its low error rate of 1.47%, suggesting that it can produce outputs that are consistent with real-world data. Such performance is a strong indicator of the model's potential for integration into field applications, such as air traffic management, where operational decisions are directly related to spatial accuracy.

For a better evaluation of the model's prediction performance, scatter plots of epochs 1, 50, and 100 are examined on the test data of the output features. These graphs show the relationship between the model's outputs and the actual values, highlighting the difference between the true and predicted values. Ideally, the relationship between the actual and predicted values should accumulate along the y = x line. The closeness of the distribution to this line provides important visual clues about the model's accuracy. In this context, analysis is performed on the three output attributes of altitude (alt), latitude (lat), and longitude (lon) (Figure 4.6).

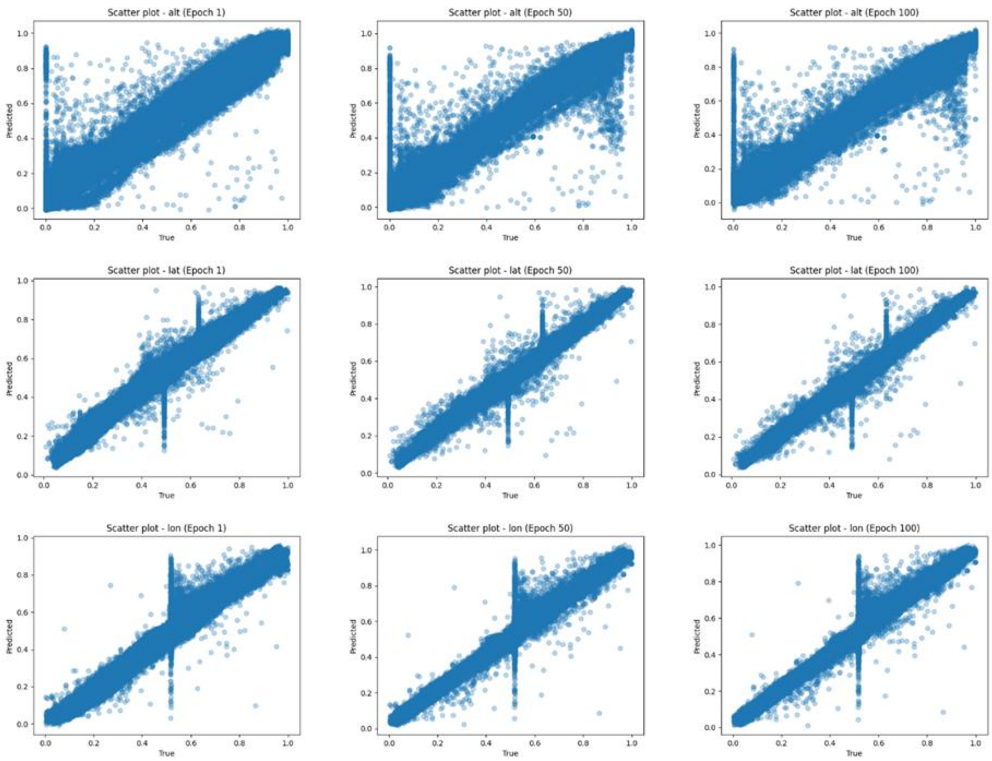

Figure 4.6. *Scatter plot comparison of true and predicted values based on epoch (altitude (alt), latitude(lat), and longitude(lon))*

During the first epoch of the model's training process, it was observed that the predictions were rather inaccurate. This is particularly evident in altitude values, where vertical densities occurred close to zero, indicating that the model tended to produce fixed values. For latitude and longitude attributes, the distribution is widespread and far from the y = x line. This situation demonstrates that the model has not yet learned the structural patterns in the data, resulting in high error rates.

There is a noticeable increase in the model's prediction capacity in the 50th epoch. Scatter plots show that the altitude, latitude, and longitude predictions are closer to the linear structure, with reduced scattering. While the latitude and longitude predictions generally converge on the y = x line, there are still some deviations and limited generalization issues with the altitude predictions. This period shows that, while the model has started to understand the general structure, it is not yet fully optimized.

By the end of epoch 100, the scatter plots demonstrate that the model is capable of making highly accurate predictions. The predictions for the latitude component are almost aligned with the y = x line. While there are region-specific deviations in longitude predictions, the overall distribution remains consistent with a linear structure. While some clusters of constant predictions close to zero persist in altitude values, the general trend has improved significantly. This epoch demonstrates that the model's generalization ability has reached its highest level.

Scatter plot analyses visually confirm a steady increase in the model's prediction performance with each epoch. At the beginning of the training process, predictions become dispersed and weak. However, as the training continues, they begin to take on a more linear form, particularly from the 50th epoch onwards. By the 100th epoch, they exhibit a high degree of alignment with the y = x line, demonstrating significant improvement in their predictability. These visuals demonstrate that the model exhibits successful overall performance, both numerically and visually, and undergoes balanced learning without risking overfitting.

4.4.2. Scenario-based performance analysis of prediction model

To evaluate the accuracy of the model's prediction, separate traffic scenarios are created for the South and North runway configurations. For both configurations, six different days are determined: the day with the most flights, the day with the fewest flights, and the day closest to the average number of flights. The daily average flight number for North Configuration is approximately 109. The day with the most flights is the Max Day scenario, that is 26 July 2023 (474 flights), the day closest to the average number of flights is Mean Day scenario, that 24 November 2023 (108 flights), and the day with the fewest flights is Min Day scenario that 12 January 2023 (one flight). The daily average flight number for the South Configuration is approximately 467; 9 June 2023 has the most flights (785), 20 March 2023 has the closest number of flights to the average (465), and 10 January 2023 has the fewest flights (one flight). The effects of high, medium, and low-density conditions on the accuracy of model predictions are examined on these selected days to understand how the model performs under different traffic levels, comparing the real and the predicted values.

Scatter plots reflecting the relationships between the predictions and the actual latitude, longitude, and altitude values are created for three days to evaluate the model's spatial prediction accuracy visually. For the North configuration, the latitude and longitude scatter plots show that the forecasts are extremely close to the true values on the maximum and mean days. There is a clear alignment of the points to the $y = x$ line. By contrast, the estimates are more scattered and biased in the Min Day scenario, and the model's positional accuracy is significantly impaired in low-density traffic conditions (Figure 4.7).

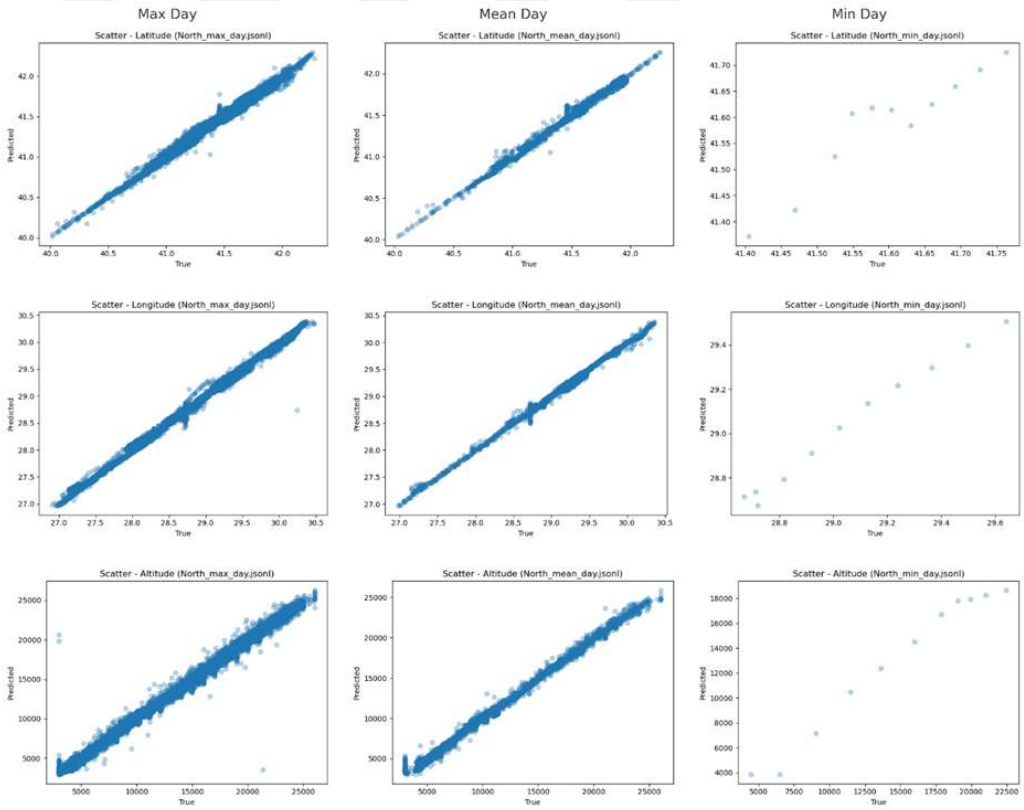

Figure 4.7. *Spatial prediction performance of the MLP model for flights in the North configuration*

Figure 4.8 shows scatter plots of the spatial prediction performance of the MLP model for flights arriving in the South configuration. The rows represent the latitude, longitude, and altitude attributes, respectively, and the columns represent the scenarios concerning flight density for the busiest day (Max Day), mean day (Mean Day), and least busy day (Min Day).

In the Max and Mean Days scenario, the model predictions generally show a high degree of overlap with the actual values, with the scatter points concentrated around the y = x line. This demonstrates the model's stable predictive capacity under high-density traffic conditions. The linear relationship is particularly strong in the horizontal position components (latitude and longitude). The prediction performance is significantly reduced in the Min Day scenario. The points are scattered, and the deviations increase. This suggests that the model struggles to predict less frequent flight patterns under low traffic density. Altitude predictions show a slightly larger variance on all days. This indicates that positional learning in the altitude axis is relatively more difficult.

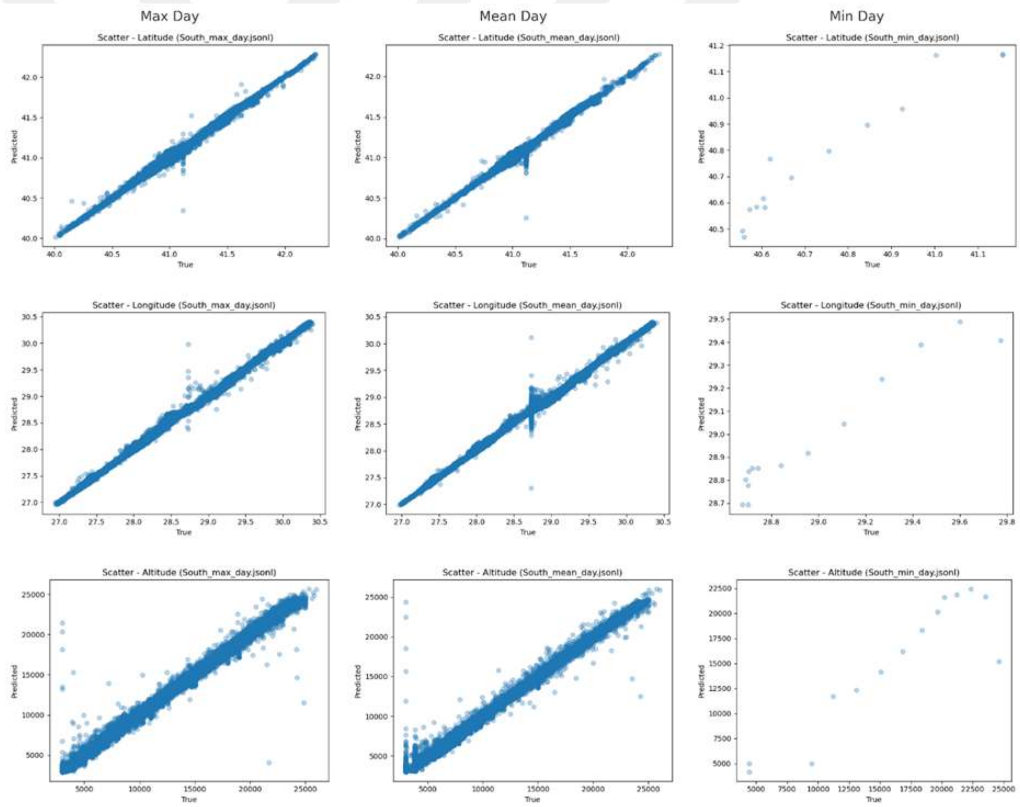

Figure 4.8. *Spatial prediction performance of the MLP model for flights in the South configuration*

The Dynamic Time Warping (DTW) metric is used to evaluate positional accuracy in detail. DTW is a method for accurately measuring pattern similarity in time series data, regardless of differences in speed, length, or variance. It is an algorithm developed to compare two asynchronous time series and measure their similarity. DTW attempts to align the two series in the most appropriate way possible by warping them and minimizing the distance between them. For each flight, the model's predicted and actual values are compared using the Haversine distance for horizontal position (latitude-longitude) and the Euclidean distance for vertical position (altitude). The Haversine distance formula is a trigonometry-based method used to calculate the shortest distance between two points on the Earth's surface. It assumes a global model and uses geographic coordinates (latitude and longitude). Table 4.2 presents the values of DTW metrics by scenario and configuration.

Table 4.2. *DTW metrics by scenario and configuration*

Scenario	Runway Configuration	Average Latitude-Longitude DTW (NM)	Average Altitude DTW (feet)	Number of Flights
Min Day	North	30.58	14,223	1
Min Day	South	74.36	19,305	1
Mean Day	North	47.26	7,119	108
Mean Day	South	35.9	6,290	465
Max Day	North	52.8	7,240	474
Max Day	South	33.75	5,783	785

According to the DTW analysis of the North configuration, the model demonstrates stable performance, particularly in terms of latitude-longitude position estimation. The average latitude-longitude DTW value is approximately 51.73 nautical mile (NM), while the average altitude DTW value is 9,527 feet. Since the flight numbers in the Mean and Max Day scenarios are 108 and 474, respectively, these scenarios more accurately reflect the model's generalization ability. While the positional deviations in these scenarios remain reasonable, the altitude deviation is 14,223 ft, especially in the Min Day scenario, due to the presence of only one flight. The model produces more stable results as the number of flights increases, but makes more errors in altitude estimates at low density in the North configuration.

The overall performance of the model is more variable in the Southern configuration than in the Northern configuration. The average latitude-longitude DTW value is slightly higher at around 34.5 NM. However, on busy flight days, especially in the Max Day scenario with 785 flights, this value drops to around 33.75 NM. This demonstrates that the model is more successful at making predictions at high density in the South configuration. The altitude DTW values exhibit a significant variance. While a high deviation of 19,305 ft is observed in the Min Day scenario, this decreases to 5,783 ft in the Max Day scenario. The latitude-longitude (35.9 NM) and altitude (6,290 ft) DTW values obtained using 465 flight data in the Mean Day scenario demonstrate that the model operates with realistic and acceptable accuracy in the South configuration.

The model's prediction performance is more stable and has lower variance as flight density increases when both configurations are considered together. On days with high traffic, the latitude-longitude DTW values in the South configuration are reduced, while altitude deviations demonstrate a more marked variation. In the North configuration, altitude estimates are more stable, and positional deviations are generally moderate. The model's performance varies significantly depending on the data density, indicating different levels of generalization to different configuration dynamics. Figure 4.9 shows a comparative analysis of the actual and model-predicted routes of randomly selected sample flights from each scenario.

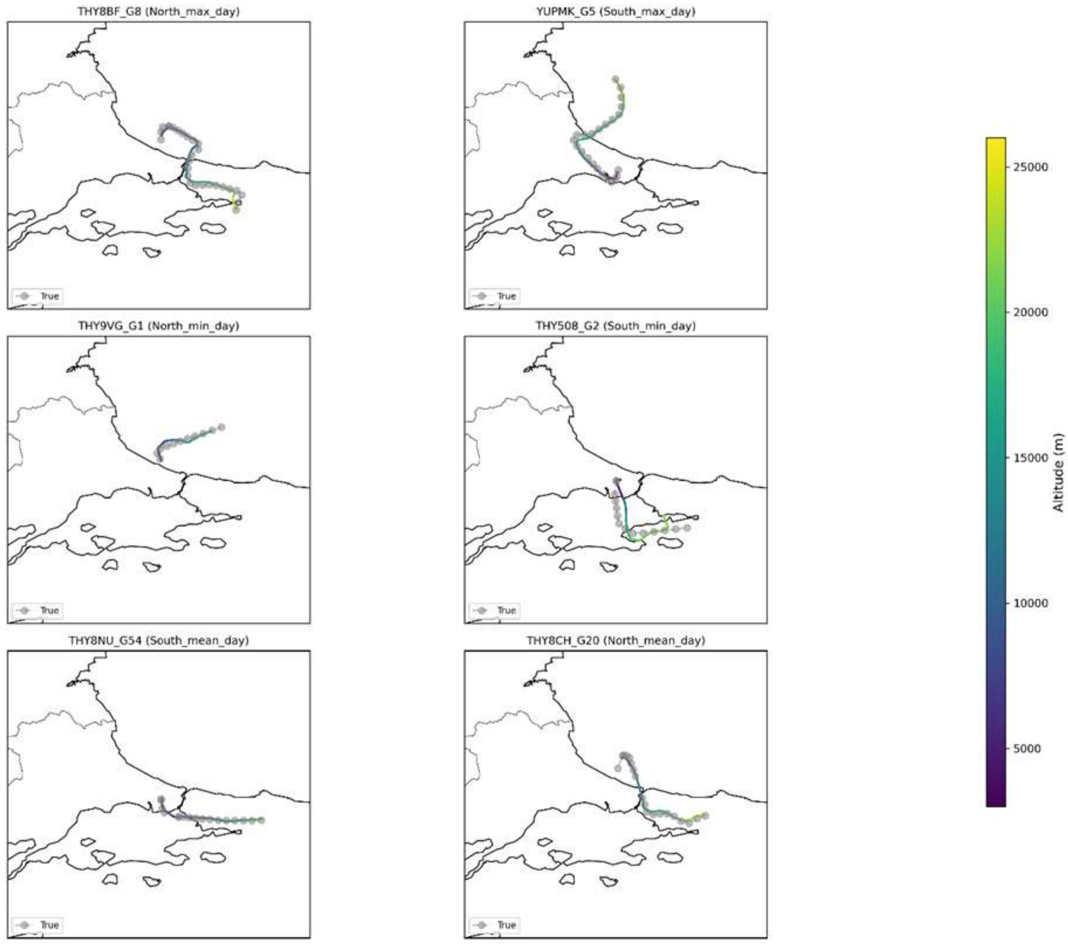

Figure 4.9. *The actual and predicted routes of randomly selected flights based on the scenario (Min/Mean/Max) and runway configuration (North/South)*

An independent dataset from 2024 is used to evaluate the model's accuracy, which has not been used in the training or testing phases. Including this dataset in the evaluation process allowed us to measure the model's ability to generalize and observe its performance under real-world conditions. June 23, 2024, recorded 1,612 flights at Istanbul Airport, making it the busiest air traffic day of the year (http-10). Analysis of the number of aircraft in ADS-B data, operating in the Istanbul TMA on that date, revealed a maximum of 26 aircraft arrivals simultaneously. This density coincided with the period between 1:22 and 2:15 on June 23, 2024. The routes of the selected 26 aircraft operating in the TMA during this period are predicted using the developed model. Figure 4.10 presents a comparative visualization of the predicted routes and actual flight paths.

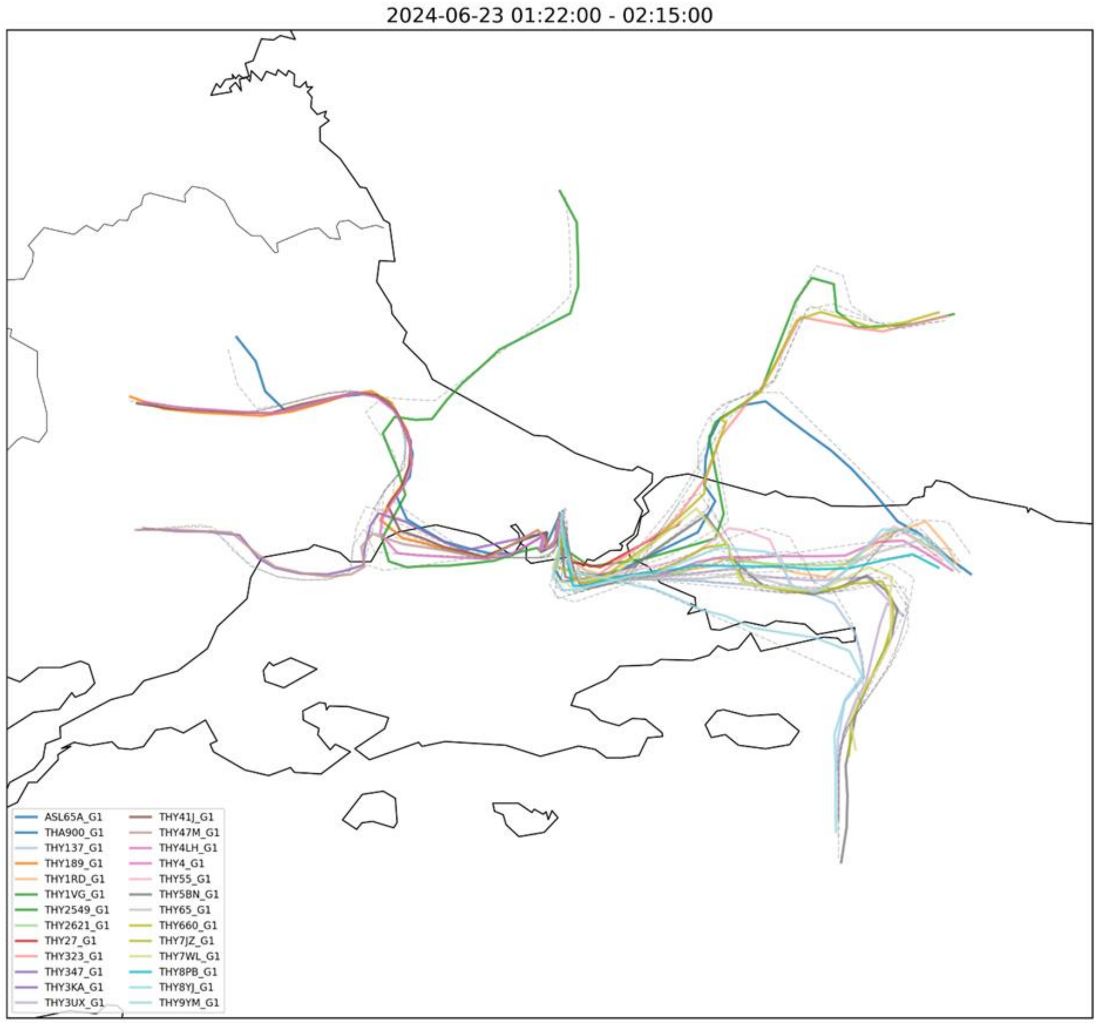

Figure 4.10. *Comparison of aircraft trajectory prediction with actual trajectory during heavy traffic (23.06.2024, 01:22–02:15)*

The scatter plots created to evaluate the model's prediction performance show a strong linear relationship between the predicted and actual values (Figure 4.11). For latitude and longitude variables, the model's predictions largely agree with the actual values. The data are densely distributed along the linear axis. This demonstrates the model's high accuracy in spatial predictions. A general trend is shown by altitude predictions, but more pronounced deviations are observed, particularly at higher altitudes. In general, it can be said that the model is quite successful, particularly in terms of horizontal (latitude/longitude) plane trajectory predictions, and reasonably successful in vertical (altitude) plane predictions.

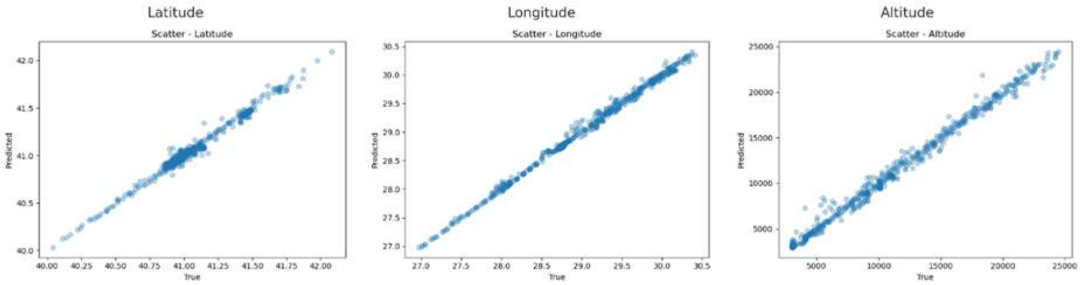

Figure 4.11. *Spatial prediction performance of the MLP model for 23.06.2024, 01:22–02:15 flights*

Figure 4.11 in the first row of scatter plots illustrates the relationship between the true and predicted latitude, longitude, and altitude values for 26 aircraft. Each aircraft is represented by a different color, enabling predictions to be visually distinguished by flight. The densely distributed dots in the plots, which are parallel to the slope, demonstrate the model's high level of accuracy in its predictions. In particular, the dots in the latitude and longitude scatter plots, which are almost aligned along the 45-degree line, demonstrate the accuracy of the positional predictions. While the model's accuracy is maintained overall in the altitude axis, small deviations in altitude predictions are noticeable in some flights.

The second row of Figure 4.12 is created using six randomly selected aircraft to make these analyses easier to understand. With fewer colors and dots, these graphs allow for detailed monitoring of each aircraft's forecast performance. Systematic shifts or inconsistencies in altitude estimates for certain aircraft are more clearly visible, which provides a significant advantage when assessing the model's accuracy on a per-aircraft basis. Together, these graphs effectively demonstrate the model's generalizability across a large sample and its predictive performance on an individual aircraft basis.

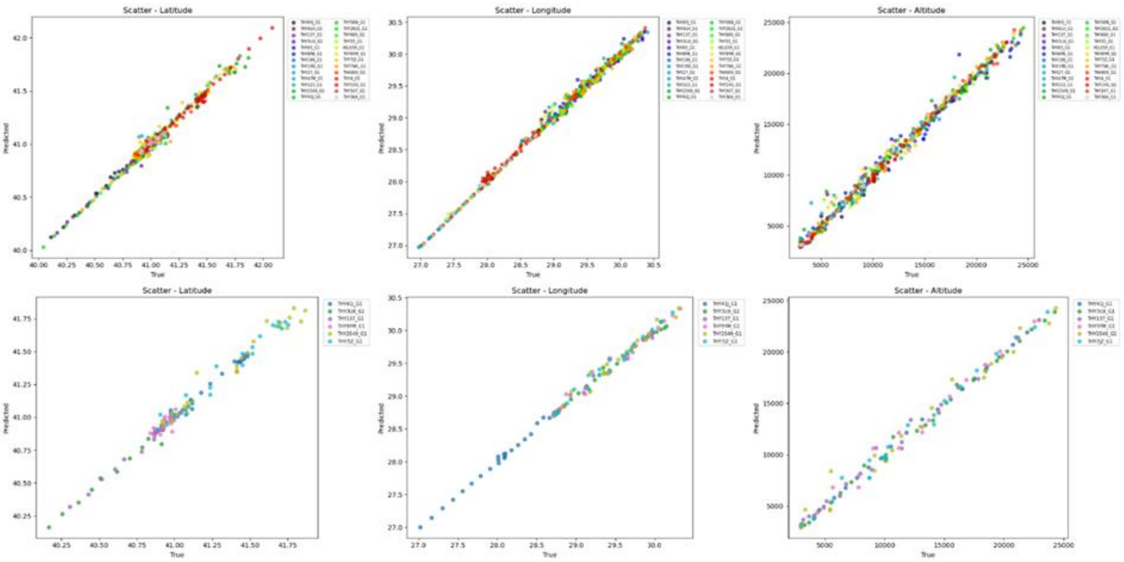

Figure 4.12. *Spatial prediction performance of the MLP model for 23.06.2024, 01:22–02:15 flights with different colors. (The first row shows the latitude/longitude/altitude of all 26 aircraft, the second row shows six random aircraft)*

Figure 4.13 shows the actual altitude values (blue line) and the values predicted by the model (orange line) for six randomly selected flights (callsigns: THY41J_G1, THY3UX_G1, THY137_G1, THY9YM_G1, THY2549_G1, and THY7JZ_G1. Each subgraph shows the altitude change for a specific flight at a given time. Graphs demonstrate that the model accurately tracks the general descent trend for almost all flights. Altitudes typically start at 22,000–25,000 feet and gradually decrease to 3,000 feet. The model's predictions show very close agreement with the actual values, particularly during the initial and final stages of the flight. However, some flights (e.g., THY2549_G1) exhibit minor deviations or temporary fluctuations during the middle stages. These differences may be due to abrupt changes in flight dynamics or the model's tendency to generalize short-term altitude variations.

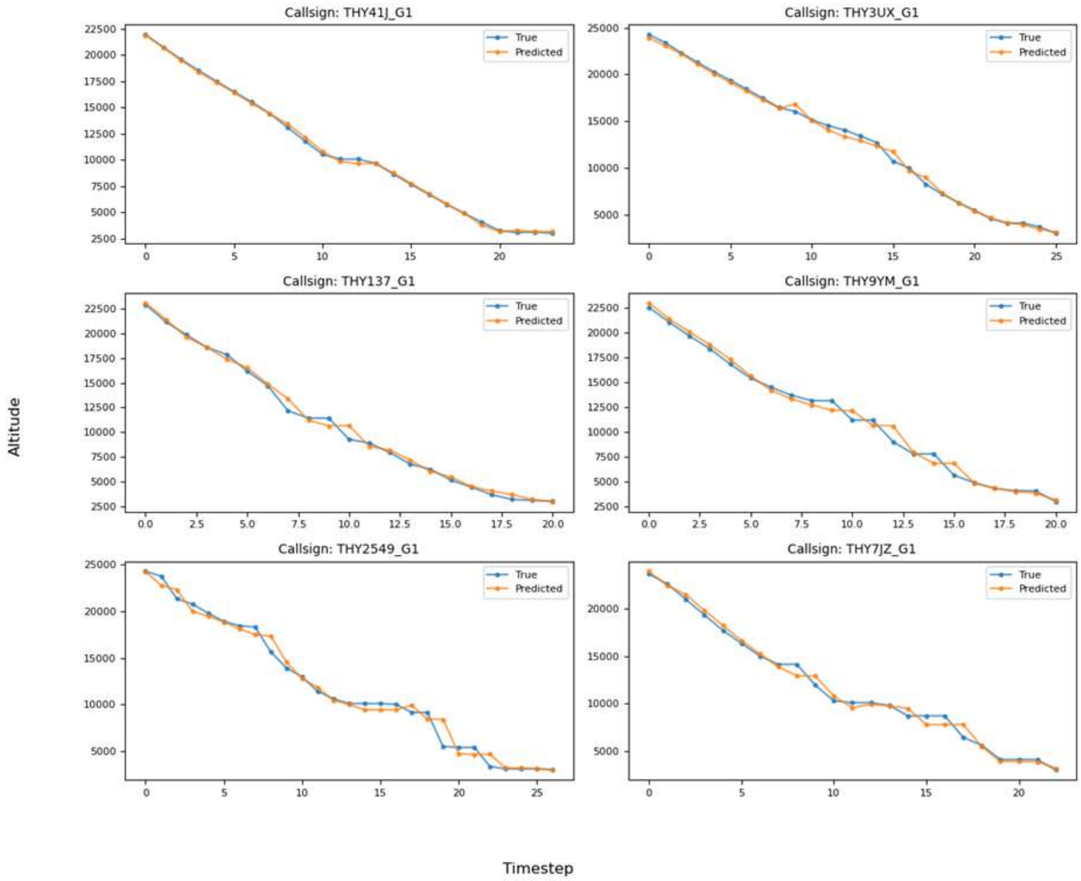

Figure 4.13. *Comparison of true and predicted altitude profiles for selected flights*

Figure 4.14 shows a comparison of the latitude-longitude flight path predictions of the model for six different aircraft, alongside their actual paths. In each subplot, the 'true path', shown in blue and dashed, represents the actual flight path, while the 'predicted path', shown in orange, represents the model's prediction. An examination of the horizontal flight paths of six different aircraft reveals that the model's predictions are generally similar to the actual routes. However, deviations are noticeable for some flights, particularly during the turning points and cruise phase. For example, while the model successfully followed the general route for flights like THY41J_G1 and THY2549_G1, there are errors in the timing and orientation of some sharp turns for THY137_G1 and THY3UX_G1. This suggests that the model may be limited in its ability to learn certain dynamic patterns. The visualization offers a meaningful and detailed illustration with which to evaluate the model's ability to predict spatial data in the horizontal plane.

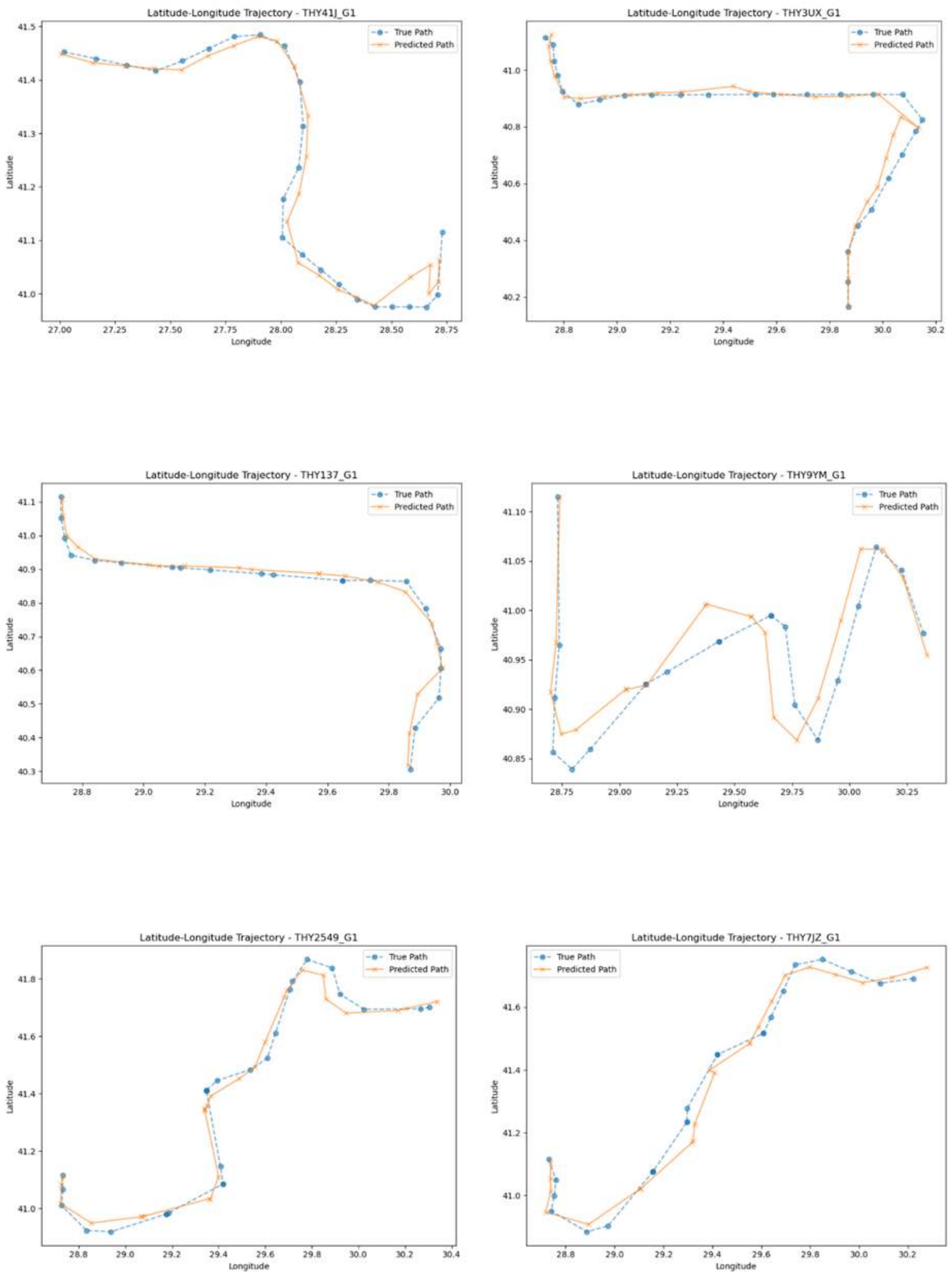

Figure 4.14. *Comparison of predicted and actual horizontal flight trajectories for six aircraft*

The model's performance is evaluated multidimensionally using DTW and MAPE metrics. The curvilinear similarity between the predicted and actual latitude, longitude, and altitude values is measured. DTW values are normalized by considering the total error (using distance or altitude as the metric) and the length of each flight (using time as the metric), as shown in Table 4.3.

Table 4.3. *Overall summary of DTW values (total and normalized)*

Metric	Latitude-Longitude DTW (NM)	Altitude DTW (feet)	Latitude-Longitude DTW / Step (NM)	Altitude DTW / Step (feet)
Mean	55.48	8,689	2.34	368.73
Median	50.81	8,768	2.35	415.09
Minimum	39	3,152	1.62	131.34
Maximum	83.91	13,496	3.10	499.85
Standard Deviation	18.2	3,360	0.69	149.35

This analysis shows that, on average, horizontal and vertical deviations of approximately 2.34 NM and 369 feet, respectively, occur at each time step across the flights. While the overall error levels, particularly in latitude and longitude predictions, are not significantly lower than those reported in comparable studies, they remain stable and within acceptable operational bounds. However, in certain cases, deviations increase substantially, reaching up to 83.91 NM in the most extreme instance.

The percentage error rates are particularly low for latitude and longitude estimations. The average MAPE for latitude is 0.057%, and for longitude, 0.118%. This indicates that the model achieved almost linear accuracy. The error rate in altitude estimation is higher than in the horizontal direction (4.19%), which is due to aircraft maneuvers involving sudden changes, and is also affected by the wide range of values of altitude. (Table 4.4).

Table 4.4. *MAPE values of output variables*

Output	Average MAPE (%)
Latitude	0.057
Longitude	0.118
Altitude	4.190

The model's ability to produce high-accuracy predictions, particularly along the horizontal axis (latitude/longitude), is evident when the DTW and MAPE metrics are considered collectively. On the other hand, its performance on the vertical axis (altitude) exhibits a comparatively lower but still acceptable level of accuracy.

5. CONCLUSION

This thesis presents a comprehensive and data-driven modelling process designed to understand and predict the temporal and spatial dynamics of air traffic, with a particular emphasis on inbound aircraft approaching Istanbul Airport within the highly complex and congested Istanbul Terminal Maneuvering Area (TMA). Recognizing the critical role of accurate trajectory prediction in ensuring airspace safety and efficiency, the study focuses on modelling arrival flows based on real-world flight behavior observed in historical data. The primary objective is to develop a robust neural network model that can predict the future trajectories of aircraft by learning from past patterns of movement, especially during periods of high traffic density when controller workload and airspace complexity are significantly elevated.

To achieve this, the model is trained on flight tracks derived from Automatic Dependent Surveillance–Broadcast (ADS-B) data, which provides high-resolution positional information for each aircraft. The use of ADS-B data enables the model to capture fine-grained spatiotemporal patterns—such as altitude changes, speed variations, holding maneuvers, and vectoring instructions—that are characteristic of arrival procedures into Istanbul Airport. By grounding the model in historical traffic dynamics, the approach aims to reflect not only the physical movement of aircraft but also the operational and procedural realities that shape those trajectories within the Istanbul TMA.

To this end, a thorough data engineering process is undertaken, involving cleaning the raw ADS-B data, interpolating missing data, feature extraction, dividing into time intervals, and grouping flights by aircraft to create callsign-based time series. Subsequently, the data points corresponding to each time step are vectorized to include attributes such as next coordinates, direction, and altitude of the flight, and sequential data structures are prepared on a time basis. To ensure that the neural network model can process these structures appropriately, the data is organized in JSONL format. In this format, each line is created according to the callsigns, time step, and the corresponding input-output vectors. The input vector includes slot-based attributes of the aircraft itself and other aircraft to enable the network to understand and learn about the aircraft's interactions with other aircraft in the TMA. The records are kept in sequential order by time. The categorical feature is converted to numerical values, and numerical features such as latitude, longitude coordinates, altitude, speed, and time are normalized and made suitable for the model.

A MLP is chosen as the model, and this network was tested on a specific date, 23 June 2024, that is independent of the dataset on which it was previously trained. This date is identified as the busiest day of the year 2024 at Istanbul Airport, with 1,612 flights. Between 01:22 and 02:15, 26 aircraft are simultaneously in the TMA area. Even under these high-density conditions, the model predicts the trajectory of each aircraft individually. The results are then evaluated using metrics such as DTW and MAPE. Furthermore, all errors are normalized on a time-step basis to compensate for the effect of flight length on DTW scores.

The analysis results show that the model provides particularly high levels of accuracy in estimating horizontal positions (latitude and longitude). The normalized DTW value averaged at 2.34 NM per step, and the MAPE values are 0.057% for latitude and 0.118% for longitude. For altitude estimations, the DTW value averaged at 369 feet per step, and the MAPE value is 4.19%. This difference is attributed to the difficulty of predicting vertical maneuvers such as descent, as well as to the wide range of altitude values.

The findings demonstrate that the proposed model has the potential to be adapted not only to the Istanbul TMA but also to other terminal airspaces where sufficient ADS-B data density is available. The model's data-driven and flexible structure allows it to be retrained with historical flight data specific to different geographic and operational contexts, transforming it into a trajectory prediction system specific to that airspace. In this respect, the study enables the development of highly generalizable, reusable, and data-driven automation systems in air traffic management and provides the foundation for a modular and scalable architecture that can be integrated into operational decision support processes.

Overall, these results demonstrate that the model can accurately predict flight trajectory in the horizontal plane, even under high traffic conditions. Meanwhile, it operates with a measurable and controllable error rate in the vertical plane. This comprehensive analysis not only demonstrates the model's instantaneous prediction performance but also its generalizability.

Fine-tuning the model using data specific to a particular airspace or operational context can significantly enhance its predictive capabilities. This process should involve re-optimizing both the training data and the model's hyperparameters. Systematically adjusting hyperparameters such as the learning rate, number of layers, number of neurons, and activation functions using grid search or similar strategies can strengthen the model's ability to generalize. This fine-tuning approach can enhance the accuracy of the model's spatial and altitude predictions, producing more reliable and practical outputs.

Future studies may seek to enhance prediction accuracy further by incorporating additional attributes, such as aircraft type, heading, departure airport, flight phase characteristics, and meteorological conditions, into the model. Furthermore, including aircraft arriving at and departing from other airports within the TMA could provide a more comprehensive representation of regional traffic complexity. Furthermore, to more deeply model both traffic interaction and time-series dynamics in the future, it is recommended that structural models based on Graph Neural Networks (GNNs) are integrated with models that robustly capture temporal dependencies, such as LSTM. GNN-based approaches can provide a more comprehensive representation of the airspace by simultaneously accounting for spatial and operational interactions between aircraft within the TMA. Meanwhile, LSTM models can improve prediction accuracy by learning the temporal patterns of individual flight paths more precisely.

REFERENCES

Ayhan, S., and Samet, H. (2016). Aircraft trajectory prediction made easy with predictive analytics. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 21-30).

Bastas, A., Kravaris, T., and Vouros, G. A. (2020). Data driven aircraft trajectory prediction with deep imitation learning. *arXiv preprint arXiv:2005.07960.*

Chida, H., Zuniga, C., and Delahaye, D. (2016). Topology design for integrating and sequencing flows in terminal maneuvering area. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *230*(9), 1705–1720. https://doi.org/10.1177/0954410016636155

DHMİ. (2025a). *İstanbul Airport Triple Independent Parallel Approach Operations (TRO) Awareness Leaflet*. Devlet Hava Meydanları İşletmesi. https://www.dhmi.gov.tr/Lists/AnaSayfaSlider_SSD/Attachments/55/TRO%20Leaflet.pdf/ (Accessed on: 23.06.2025)

DHMİ. (2025b). *AIP Türkiye – AD 2 LTFM (İstanbul Havalimanı)*. Devlet Hava Meydanları İşletmesi. https://www.dhmi.gov.tr/AIPDocuments/LT_AD_2_LTFM_en.pdf/ (Accessed on: 23.06.2025)

Du, K.-L., Leung, C.-S., Mow, W. H., and Swamy, M. N. S. (2022). Perceptron: Learning, Generalization, Model Selection, Fault Tolerance, and Role in the Deep Learning Era. *Mathematics*, *10*(24), 4730. https://doi.org/10.3390/math10244730

Erzberger, H. (1990). *Traffic Management Advisor (TMA)* (NASA Technical Memorandum 102229). NASA Ames Research Center. https://ntrs.nasa.gov/citations/19910045440 (Accessed on: 02.06.2025)

EUROCONTROL. (2010). *EUROPEAN ORGANISATION FOR THE SAFETY OF AIR NAVIGATION Introduction to the Mission Trajectory DOCUMENT APPROVAL DOCUMENT CHANGE RECORD*. Brussels.

Federal Aviation Administration. (2021). *Trajectory Based Operations (TBO)*. U.S. Department of Transportation. https://www.faa.gov/air_traffic/technology/tbo (Accessed on: 24.06.2025)

Federal Aviation Administration. (2022). *Trajectory Based Operations (TBO)*. U.S. Department of Transportation, FAA. https://www.faa.gov/air_traffic/technology/tbo (Accessed on: 23.06.2025)

Fron, X. (2001). *Dealing with airport and airspace congestion in Europe.* In *ATM Performance, the paper presentation, Brussels, EUROCONTROL meeting, March* (Vol. 25). https://bpb-use1.wpmucdn.com/blog.umd.edu/dist/9/604/files/2019/02/2001_AAC_Fron-1xt5wpt.pdf (Accessed on: 23.06.2025)

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Gong, C., and McNally, D. (2004). A methodology for automated trajectory prediction analysis. *In AIAA Guidance, Navigation, and Control Conference and Exhibit,* Providence, Rhode Island, Aug. 16–19, 2004. p. 4788. https://doi.org/10.2514/6.2004-4788

Granberg, T. A., Polishchuk, T., Polishchuk, V., and Schmidt, C. (2019). Integer programming–based airspace sectorization for terminal maneuvering areas with convex sectors. *Journal of Air Transportation*, *27*(4), 169–180. https://doi.org/10.2514/1.D0148

Hah, S., Willems, B., and Phillips, R. (2006). The effect of air traffic increase on controller workload. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 50, No. 1, pp. 50-54). Sage CA: Los Angeles, CA: SAGE Publications.

International Civil Aviation Organization**.** (2005). *Global air traffic management operational concept (Doc 9854, 1st edition).* Montreal: ICAO.

International Civil Aviation Organization. (2016). *Doc 4444 – Procedures for Air Navigation Services – Air Traffic Management (PANS-ATM)*. 16th Edition. Montreal: ICAO.

International Civil Aviation Organization. (2018). *Annex 11 to the Convention on International Civil Aviation: Air traffic services* (15th ed.). Montreal, Canada: ICAO. https://ffac.ch/wp-content/uploads/2020/10/ICAO-Annex-11-Air-Traffic-Services.pdf (Accessed on: 20.06.2025)

Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, V., and Garg, M. L. (2018). Deep learning techniques and their applications: A short review. *Bioscience Biotechnology Research Communications*, *11*(4), 699-709.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539.

Le Fablec, Y., and Alliot, J. M. (1999). Using Neural Networks to Predict Aircraft Trajectories. In *IC-AI* (pp. 524-529).

Li, H., Si, Y., Zhang, Q., and Yan, F. (2025). 4D Track Prediction Based on BP Neural Network Optimized by Improved Sparrow Algorithm. *Electronics*, *14*(6), 1097.

Liang, M. (2018). *Aircraft Route Network Optimization in Terminal Maneuvering Area*. Doctoral Dissertation, Université de Toulouse. https://theses.hal.science/tel-01703861v1 (Accessed on: 18.06.2025)

Lin, Y., Yang, B., Zhang, J., and Liu, H. (2019). Approach for 4-d trajectory management based on HMM and trajectory similarity. *Journal of Marine Science and Technology*, *27*(3), 7.

Lymperopoulos, I., and Lygeros, J. (2010). Sequential Monte Carlo methods for multi-aircraft trajectory prediction in air traffic management. *International Journal of Adaptive Control and Signal Processing*, *24*(10), 830-849.

Ma, L., and Tian, S. (2020). A hybrid CNN-LSTM model for aircraft 4D trajectory prediction. *IEEE access*, *8*, 134668-134680.

Nolan, M. S. (2004). *Fundamentals of Air Traffic Control*. United State of America: Brooks/Cole

Nuic, A., Poles, D., and Mouillet, V. (2010). BADA: An advanced aircraft performance model for present and future ATM systems. *International journal of adaptive control and signal processing*, *24*(10), 850-866.

Olive, X. (2019). *traffic, a toolbox for processing and analysing air traffic data*. *Journal of Open Source Software, 4*(39), 1518. https://doi.org/10.21105/joss.01518.

Prandini, M., Piroddi, L., Puechmorel, S., and Brázdilová, S. L. (2011). Toward air traffic complexity assessment in new generation air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, *12*(3), 809–818. https://doi.org/10.1109/TITS.2011.2113175

Puranik, T. G., Rohani, A. S., and Kalyanam, K. M. (2022). An ODE-fitting approach to estimate critical aircraft performance parameters for trajectory prediction. In *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)* (pp. 1-9). IEEE

Rumelhart, D., Hinton, G., and Williams, R. Learning representations by back-propagating errors. *Nature* 323, 533–536 (1986). https://doi.org/10.1038/323533a0

Sejnowski, T. J. (2018). *The deep learning revolution*. MIT press.

Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., and Wilhelm, M. (2014, April). Bringing up OpenSky: A large-scale ADS-B sensor network for research.

In *IPSN-14 proceedings of the 13th international symposium on information processing in sensor networks* (pp. 83-94). IEEE.

Song, Y., Cheng, P., and Mu, C. (2012). *Information and Automation (ICIA), 2012 International Conference on: Shenyang, China: June 6-8, 2012*. IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

Wang, S., Zhou, T., and Bilmes, J. (2019, May). Bias also matters: Bias attribution for deep neural network explanation. In *International Conference on Machine Learning* (pp. 6659-6667). PMLR.

Wang, Z., Liang, M., and Delahaye, D. (2017, November). Short-term 4d trajectory prediction using machine learning methods. In *SID 2017, 7th SESAR Innovation Days*. Belgrade, Serbia. https://hal.science/hal-01652041 (Accessed on: 23.06.2025)

Wu, Z. J., Tian, S., and Ma, L. (2019). A 4D trajectory prediction model based on the BP neural network. *Journal of Intelligent Systems*, *29*(1), 1545-1557

Xu, Z., Zeng, W., Chu, X., and Cao, P. (2021). Multi-aircraft trajectory collaborative prediction based on social long short-term memory network. *Aerospace*, *8*(4), 115.

Zeng, W., Chu, X., Xu, Z., Liu, Y., and Quan, Z. (2022). Aircraft 4D Trajectory Prediction in Civil Aviation: A Review. In *Aerospace* (Vol. 9, Issue 2). MDPI. https://doi.org/10.3390/aerospace9020091

Zhang, H., and Liu, Z. (2024). Four-dimensional aircraft trajectory prediction based on generative deep learning. *Journal of Aerospace Information Systems*, 21(7), 554–567. https://doi.org/10.2514/1.I011333

Zhang, J., Liu, J., Hu, R., and Zhu, H. (2018). Online four dimensional trajectory prediction method based on aircraft intent updating. *Aerospace Science and Technology*, *77*, 774-787.

Zhu, X., Zhang, K., Zhang, Z., and Tan, L. (2024). Predicting Flight Trajectory in Convective Weather through Boosted Spatiotemporal Deep Learning Ensemble. Journal of Advanced Transportation, 2024(1), 6400839.

http-1: *https://www.eurocontrol.int/sites/default/files/2024-01/eurocontrol-european-aviation-overview-20240118-2023-review.pdf* (Accessed on: 09.05.2025)

http-2: *https://www.eurocontrol.int/sites/default/files/2024-02/eurocontrol-seven-year-forecast-2024-2030-february-2024.pdf* (Accessed on: 12.05.2025)

http-3: https://skybrary.aero/articles/air-traffic-management-atm (Accessed on: 23.06.2025)

http-4: https://www.eurocontrol.int/sites/default/files/2019-07/challenges-of-growth-2018-annex1_0.pdf (Accessed on: 22.06.2025)

http-5: *https://www.eurocontrol.int/article/european-atm-master-plan-and-eurocontrols-role-wider-atm-transformation* (Accessed on: 20.06.2025)

http-6: https://www.eurocontrol.int/project/atc-tbo (Accessed on: 25.05.2025)

http-7: https://ral.ucar.edu/technologies/decision-support-systems (Accessed on: 23.06.2025)

http-8: https://www.eurocontrol.int/system/enhanced-tactical-flow-management-system (Accessed on: 23.06.2025)

http-9: *https://www.eurocontrol.int/sites/default/files/2025-01/eurocontrol-european-aviation-overview-20250123-2024-review.pdf* (Accessed on: 18.06.2025)

http-10: *https://www.eurocontrol.int/Economics/2024-DailyTrafficVariation-Airports.html* (Accessed on: 22.06.2025)

CURRICULUM VITAE

ORCID ID: 0009-0002-8330-460X

Name Surname : **Enes Özçelik**

Foreign Language : **English**

Education and Professional Background:

- 2021, Kütahya Dumlupınar University, Faculty of Engineering, Electrical and Electronics Engineering (Undergraduate)
- 2023, Research Assistant, Faculty of Aeronautics and Astronautics, Air Traffic Control Department

Publications and/or Scientific/Artistic Activities:

- E. Ozcelik, E. Germen and F. A. Cetek. (2025). Prediction of Arrival Aircraft Time Spent in TMA with Graph Neural Network. Integrated Communications, Navigation and Surveillance Conference (ICNS), 1-7.
- Researcher in ATMACA – "Air Traffic Management and Communication over ATN/IPS" project (Grant ID: 101167070, SESAR 3 JU, Digital European Sky).
- Researcher – "Derin Öğrenme Yöntemleriyle Havaalanı İniş Yörünge Tahmini" 303-Yüksek Lisans Tez Projesi (LÖT), BAP Project No: 24LÖT224, Eskişehir Technical University.