



APJESS

Journal of Engineering
and **Smart Systems**

Volume : 12

Issue : 1

Year : 2024

Volume 12 / Issue 1

Academic Platform Journal of Engineering and Smart Systems

Editor in Chief (Owned By Academic Perspective)

Dr. Mehmet SARIBIYIK, Sakarya University of Applied Sciences, Türkiye

Editors

Dr. Caner ERDEN, Sakarya University of Applied Sciences, Türkiye

Dr. John YOO, Bradley University, USA

Editorial Board

Dr. Abdullah Hulusi KÖKÇAM, Sakarya University, Türkiye

Dr. Ali Tahir KARAŞAHİN, Karabuk University, Türkiye

Dr. Aydın MÜHÜRÇÜ, Kırklareli University, Türkiye

Dr. Ayşe Nur AY, Sakarya University of Applied Sciences, Türkiye

Dr. Cengiz KAHRAMAN, Istanbul Technical University, Türkiye

Dr. Elif Elçin GÜNAY, Sakarya University, Türkiye

Dr. Erkan ÇELİK, Istanbul University, Türkiye

Dr. Fatih VARÇIN, Sakarya University of Applied Sciences, Türkiye

Dr. Gürcan YILDIRIM, Abant İzzet Baysal University, Türkiye

Dr. Hacı Mehmet ALAKAŞ, Kirikkale University, Türkiye

Dr. Huseyin SEKER, Birmingham City University, Birmingham, United Kingdom

Dr. Kerem KÜÇÜK, Kocaeli University, Türkiye

Dr. Marco A. ACEVES-FERNÁNDEZ, Universidad Autónoma De Querétla, Mexico

Dr. Mazin MOHAMMED, University Of Anbar, Iraq

Dr. Mehmet Emin AYDIN, University of The West Of England, United Kingdom

Dr. Muhammet KURULAY, Yıldız Technical University, Türkiye

Dr. Muhammed Maruf ÖZTÜRK, Suleyman Demirel University, Türkiye

Dr. Ömer AYDIN, Celal Bayar University, Türkiye

Dr. Rakesh PHANDEN, Amity University Uttar Pradesh, India

Dr. Uğur Erkin KOCAMAZ, Bursa Uludağ University, Türkiye

Dr. Tuğba TUNACAN, Abant İzzet Baysal University, Türkiye

Dr. Turgay Tugay BİLGİN, Bursa Technical University, Türkiye

Dr. Tülay YILDIRIM, Yıldız Technical University, Türkiye

Dr. Valentina E. BALAS, Aurel Vlaicu University of Arad, Romania

Language Editor

Dr. Hakan ASLAN, Sakarya University, Türkiye

Editorial Assistants

Selim İLHAN, Sakarya University, Türkiye

İbrahim MUCUK, Sakarya University, Türkiye

Correspondence Address

Academic Platform Journal of Engineering and Smart Systems

Akademik Perspektif Derneği, Tığcılar Mahallesi Kadir Sokak No:12

Kat:1 Adapazarı SAKARYA

+90 551 628 9477 (WhatsApp only)

<https://dergipark.org.tr/tr/pub/apjess>

Issue Link: <https://dergipark.org.tr/tr/pub/apjess/issue/>

Aim and Scope

Academic Platform Journal of Engineering and Smart Systems (APJESS) is a peer reviewed open-access journal which focuses on the research and applications related to smart systems and artificial intelligence. APJESS accepts both **original research papers** and **review articles** written in **English**. It is essential that the information created in scientific study needs to be new, suggest new method or give a new dimension to an existing information. Articles submitted for publication are evaluated by at least two referees in case the editor finds potential scientific merit, and final acceptance and rejection decision are taken by editorial board. The authors are not informed about the name of referees who evaluate the papers. In similar way, the referees are not allowed to see the names of authors. The papers which do not satisfy the scientific level of the journal can be refused with unexplained reason.

There are two key principles that APJESS was founded on: Firstly, to publish the most exciting, novel, technically sound, and clearly presented researches with respect to the subjects of smart systems and artificial intelligence. Secondly, to provide a rapid turn-around time possible for reviewing and publishing, and to disseminate the articles freely for research, teaching and reference purposes.

Any information about a submitted manuscript cannot be disclosed by the editor and any other editorial staff to anyone other than the corresponding author, reviewers, potential reviewers, other editorial advisers, and the publisher. No confidential information or ideas obtained through peer review can be used for personal advantage.

Journal History

The journal was published between 2013-2021 with the title of "Academic Platform - Journal of Engineering and Science". It will be published under its new title "Academic Platform Journal of Engineering and Smart Systems" after 2022.

Former Title: Academic Platform - Journal of Engineering and Science

Years: 2013-2021

Scope

APJESS aims to publish research and review papers dealing with, but not limited to, the following research fields:

- Knowledge Representation and Reasoning,
- Data Mining & Data Science,
- Supervised, Semi-Supervised and Unsupervised Learning,
- Machine Learning (ML) and Neural Computing,
- Evolutionary Computation,
- Natural Language Processing, Internet of Things, Big Data
- Fuzzy Systems,
- Intelligent Information Processing,
- AI Powered Robotic Systems,
- Multi-agent Systems and Programming for Smart Systems

Author Guidelines

Article Types

Manuscripts submitted to APJESS should neither be published previously nor be under consideration for publication in another journal.

The main article types are as follows:

Research Articles: Original research manuscripts. The journal considers all original research manuscripts provided that the work reports scientifically sound experiments and provides a substantial amount of new information.

Review Articles: These provide concise and precise updates on the latest progress made in a given area of research.

Checklist for Submissions

Please,

- read the [Aims & Scope](#) to see if your manuscript is suitable for the journal,
- use the [Microsoft Word template](#) to prepare your manuscript;
- Download [Copyright Transfer Form](#) and signed by all authors.
- make sure that issues about [Ethical Principles and Publication Policy](#), [Copyright and Licensing](#), [Archiving Policy](#), [Repository Policy](#) have been appropriately considered;
- Ensure that all authors have approved the content of the submitted manuscript.

The main text should be formed in the following order:

Manuscript: The article should start with an introduction written in scientific language, putting thoughts together from diverse disciplines combining evidence-based knowledge and logical arguments, conveying views about the aim and purpose of the article. It must address all readers in general. The technical terms, symbols, abbreviations must be defined at the first time when they are used in the article. The manuscript should be formed in the following order:

Introduction,

Material and Method,

Findings,

Discussion and Conclusion.

References: At the end of the paper provide full details of all references cited in-text. The reference list should be arranged in the order of appearance of the in-text citations, not in an alphabetical order, beginning with [1], and continuing in an ascending numerical order, from the lowest number to the highest. In the reference list, only one resource per reference number is acceptable.

References must be numbered in order of appearance in the text (including citations in tables and legends) and listed individually at the end of the manuscript. We recommend preparing the references with a bibliography software package, such as EndNote, Reference Manager or Zotero to avoid typing mistakes and duplicated references. Include the digital object identifier (DOI) for all references where available. Please use IEEE style.

IEEE Sample Reference List

- [1] R. E. Ziemer and W. H. Tranter, Principles of Communications: Systems, Modulation, and Noise, 7th ed. Hoboken, NJ: Wiley, 2015.
- [2] J. D. Bellamy et al., Computer Telephony Integration, New York: Wiley, 2010.

- [3] C. Jacks, High Rupturing Capacity (HRC) Fuses, New York: Penguin Random House, 2013, pp. 175–225.
- [4] N. B. Vargafik, J. A. Wiebelt, and J. F. Malloy, "Radiative transfer," in Convective Heat. Melbourne: Engineering Education Australia, 2011, ch. 9, pp. 379–398.
- [5] H. C. Hottel and R. Siegel, "Film condensation," in Handbook of Heat Transfer, 2nd ed. W. C. McAdams, Ed. New York: McGraw-Hill, 2011, ch. 9, pp. 78–99.
- [6] H. H. Gaynor, Leading and Managing Engineering and Technology, Book 2: Developing Managers and Leaders. IEEE-USA, 2011. Accessed on: Oct. 15, 2016. [Online]. Available: <http://www.ieeeusa.org/communications/ebooks/files/sep14/n2n802/Leading-and-Managing-Engineering-and-Technology-Book-2.pdf>
- [7] G. H. Gaynor, "Dealing with the manager leader dichotomy," in Leading and Managing Engineering and Technology, Book 2, Developing Leaders and Mangers. IEEE-USA, 2011, pp. 27–28. Accessed on: Jan. 23, 2017. [Online]. Available: <http://www.ieeeusa.org/communications/ebooks/files/sep14/n2n802/Leading-and-Managing-Engineering-and-Technology-Book-2.pdf>
- [8] M. Cvijetic, "Optical transport system engineering," in Wiley Encyclopedia of Telecommunications, vol. 4, J. G. Proakis, Ed. New York: John Wiley & Sons, 2003, pp. 1840–1849. Accessed on: Feb. 5, 2017. [Online]. Available: <http://ebscohost.com>
- [9] T. Kaczorek, "Minimum energy control of fractional positive electrical circuits", Archives of Electrical Engineering, vol. 65, no. 2, pp.191–201, 2016.
- [10] P. Harsha and M. Dahleh, "Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy", IEEE Trans. Power Sys., vol. 30, no. 3, pp. 1164–1181, May 2015.
- [11] A. Vaskuri, H. Baumgartner, P. Kärhä, G. Andor, and E. Ikonen, "Modeling the spectral shape of InGaAlP-based red light-emitting diodes," Journal of Applied Physics, vol. 118, no. 20, pp. 203103–203103-7, Jul. 2015. Accessed on: Feb. 9, 2017. [Online]. Available: doi: 10.1063/1.4936322
- [12] K. J. Krishnan, "Implementation of renewable energy to reduce carbon consumption and fuel cell as a back-up power for national broadband network (NBN) in Australia," Ph.D dissertation, College of Eng. and Sc., Victoria Univ., Melbourne, 2013.
- [13] C. R. Ozansoy, "Design and implementation of a Universal Communications Processor for substation integration, automation and protection," Ph.D. dissertation, College of Eng. and Sc., Victoria Univ., Melbourne, 2006. [Online]. Accessed on: June 22, 2017. [Online]. Available: <http://vuir.vu.edu.au/527/>
- [14] M. T. Long, "On the statistical correlation between the heave, pitch and roll motion of road transport vehicles," Research Master thesis, College of Eng. and Sc., Victoria Univ., Melb., Vic., 2016.
- [15] Safe Working on or Near Low-voltage Electrical Installations and Equipment, AS/NZS 4836:2011, 2011.

Ethical Principles and Publication Policy

Peer Review Policy

Academic Platform Journal of Engineering and Smart Systems (APJESS) applies double blind peer-review process in which both the reviewer and the author are anonymous. Reviewer selection for each submitted article is up to area editors, and reviewers are selected based on the reviewer's expertise, competence, and previous experience in reviewing papers for APJES.

Every submitted article is evaluated by area editor, at least, for an initial review. If the paper reaches minimum quality criteria, fulfills the aims, scope and policies of APJES, it is sent to at least two reviewers for evaluation.

The reviewers evaluate the paper according to the Review guidelines set by editorial board members and return it to the area editor, who conveys the reviewers' anonymous comments back to the author. Anonymity is strictly maintained.

The double-blind peer-review process is managed using “ULAKBİM Dergi Sistemleri”, namely Dergipark platform.

Open Access Policy

APJESS provides immediate open access for all users to its content on the principle that making research freely available to the public, supporting a greater global exchange of knowledge.

Archiving Policy

APJESS is accessed by Dergipark platform which utilizes the LOCKSS system to create a distributed archiving system among participating libraries and permits those libraries to create permanent archives of the journal for purposes of preservation and restoration.

Originality and Plagiarism Policy

Authors by submitting their manuscript to APJESS declare that their work is original and authored by them; has not been previously published nor submitted for evaluation; original ideas, data, findings and materials taken from other sources (including their own) are properly documented and cited; their work does not violate any rights of others, including privacy rights and intellectual property rights; provided data is their own data, true and not manipulated. Plagiarism in whole or in part without proper citation is not tolerated by APJESS. Manuscripts submitted to the journal will be checked for originality using anti-plagiarism software.

Journal Ethics and Malpractice Statement

For all parties involved in the publishing process (the author(s), the journal editor(s), the peer reviewers, the society, and the publisher) it is necessary to agree upon standards of expected ethical behavior. The ethics statements for APJESS are based on the Committee on Publication Ethics (COPE) Code of Conduct guidelines available at www.publicationethics.org.

1. Editor Responsibilities

Publication Decisions & Accountability

The editor of APJESS is responsible for deciding which articles submitted to the journal should be published, and, moreover, is accountable for everything published in the journal. In making these decisions, the editor may be guided by the journal's editorial board and/or area editors, and considers the policies of the journal. The editor should maintain the integrity of the academic record, preclude business needs from compromising intellectual and ethical standards, and always be willing to publish corrections, clarifications, retractions, and apologies when needed.

Fair play

The editor should evaluate manuscripts for their intellectual content without regard to race, gender, sexual orientation, religious belief, ethnic origin, citizenship, or political philosophy of the author(s).

Confidentiality

The editor and any editorial staff must not disclose any information about a submitted manuscript to anyone other than the corresponding author, reviewers, potential reviewers, other editorial advisers, and the publisher, as appropriate.

Disclosure, conflicts of interest, and other issues

The editor will be guided by COPE's Guidelines for Retracting Articles when considering retracting, issuing expressions of concern about, and issuing corrections pertaining to articles that have been published in APJES.

Unpublished materials disclosed in a submitted manuscript must not be used in an editor's own research without the explicit written consent of the author(s). Privileged information or ideas obtained through peer review must be kept confidential and not used for personal advantage.

The editor should seek so ensure a fair and appropriate peer-review process. The editor should recuse himself/herself from handling manuscripts (i.e. should ask a co-editor, associate editor, or other member of the editorial board instead to review and consider) in which they have conflicts of interest resulting from competitive, collaborative, or other relationships or connections with any of the authors, companies, or (possibly) institutions connected to the papers. The editor should require all contributors to disclose relevant competing interests and publish corrections if competing interests are revealed after publication. If needed, other appropriate action should be taken, such as the publication of a retraction or expression of concern.

2. Reviewer Responsibilities

Contribution to editorial decisions

Peer review assists the editor in making editorial decisions and, through the editorial communication with the author, may also assist the author in improving the manuscript.

Promptness

Any invited referee who feels unqualified to review the research reported in a manuscript or knows that its timely review will be impossible should immediately notify the editor so that alternative reviewers can be contacted.

Confidentiality

Any manuscripts received for review must be treated as confidential documents. They must not be shown to or discussed with others except if authorized by the editor.

Standards of objectivity

Reviews should be conducted objectively. Personal criticism of the author(s) is unacceptable. Referees should express their views clearly with appropriate supporting arguments.

Acknowledgement of sources

Reviewers should identify relevant published work that has not been cited by the author(s). Any statement that an observation, derivation, or argument had been previously reported should be accompanied by the relevant citation. Reviewers should also call to the editor's attention any substantial similarity or overlap between the manuscript under consideration and any other published data of which they have personal knowledge.

Disclosure and conflict of interest

Privileged information or ideas obtained through peer review must be kept confidential and not used for personal advantage. Reviewers should not consider evaluating manuscripts in which they have

conflicts of interest resulting from competitive, collaborative, or other relationships or connections with any of the authors, companies, or institutions connected to the submission.

3. Author Responsibilities

Reporting standards

Authors reporting results of original research should present an accurate account of the work performed as well as an objective discussion of its significance. Underlying data should be represented accurately in the manuscript. A paper should contain sufficient detail and references to permit others to replicate the work. Fraudulent or knowingly inaccurate statements constitute unethical behavior and are unacceptable.

Originality and plagiarism

The authors should ensure that they have written entirely original works, and if the authors have used the work and/or words of others that this has been appropriately cited or quoted.

Multiple, redundant, or concurrent publication

An author should not in general publish manuscripts describing essentially the same research in more than one journal or primary publication. Parallel submission of the same manuscript to more than one journal constitutes unethical publishing behavior and is unacceptable.

Acknowledgement of sources

Proper acknowledgment of the work of others must always be given. Authors should also cite publications that have been influential in determining the nature of the reported work.

Authorship of a manuscript

Authorship should be limited to those who have made a significant contribution to the conception, design, execution, or interpretation of the reported study. All those who have made significant contributions should be listed as co-authors. Where there are others who have participated in certain substantive aspects of the research project, they should be named in an Acknowledgement section. The corresponding author should ensure that all appropriate co-authors are included in the author list of the manuscript, and that all co-authors have seen and approved the final version of the paper and have agreed to its submission for publication. All co-authors must be clearly indicated at the time of manuscript submission. Request to add co-authors, after a manuscript has been accepted will require approval of the editor.

Hazards and human or animal subjects

If the work involves chemicals, procedures, or equipment that has any unusual hazards inherent in their use, the authors must clearly identify these in the manuscript. Additionally, manuscripts should adhere to the principles of the World Medical Association (WMA) Declaration of Helsinki regarding research study involving human or animal subjects.

Disclosure and conflicts of interest

All authors should disclose in their manuscript any financial or other substantive conflict of interest that might be construed to influence the results or their interpretation in the manuscript. All sources of financial support for the project should be disclosed.

Fundamental errors in published works

In case an author discovers a significant error or inaccuracy in his/her own published work, it is the author's obligation to promptly notify the journal's editor to either retract the paper or to publish an appropriate correction statement or erratum.

4. Publisher Responsibilities

Editorial autonomy

Academic Perspective Foundation is committed to working with editors to define clearly the respective roles of publisher and of editors in order to ensure the autonomy of editorial decisions, without influence from advertisers or other commercial partners.

Intellectual property and copyright

We protect the intellectual property and copyright of Academic Perspective Foundation, its imprints, authors and publishing partners by promoting and maintaining each article's published version of record. Academic Perspective Foundation ensures the integrity and transparency of each published article with respect to: conflicts of interest, publication and research funding, publication and research ethics, cases of publication and research misconduct, confidentiality, authorship, article corrections, clarifications and retractions, and timely publication of content.

Scientific Misconduct


In cases of alleged or proven scientific misconduct, fraudulent publication, or plagiarism the publisher, in close collaboration with the editors, will take all appropriate measures to clarify the situation and to amend the article in question. This includes the prompt publication of a correction statement or erratum or, in the most severe cases, the retraction of the affected work.

Contents

Research Articles		
Title	Authors	Pages
AI-Embedded UAV System for Detecting and Pursuing Unwanted UAVs	Ali Furkan KAMANLI	1 - 13
Determination of Electricity Production by Fuzzy Logic Method	Beyza ÖZDEM, Muharrem DÜĞENÇİ, Mümtaz İPEK	14-20
Statistical and Artificial Intelligence Based Forecasting Approaches for Cash Demand Problem of Automated Teller Machines	Michele CEDOLIN, Deniz ORHAN, Müjde GENEVOIS	21-27
IoT-based Smart Home Security System with Machine Learning Models	Selman HIZAL, Ünal ÇAVUŞOĞLU, Devrim AKGÜN	28-36
Genetically Tuned Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor	Ali Tahir KARAŞAHİN	37-46

AI-Embedded UAV System for Detecting and Pursuing Unwanted UAVs

*¹Ali Furkan KAMANLI

¹Department of Electrical and Electronics Engineering, Sakarya University of Applied Sciences, Sakarya, Türkiye, fkamanli@sakarya.edu.tr 

Abstract

In recent years, the use of unmanned aerial vehicle (UAV) platforms in civil and military applications has surged, highlighting the critical role of artificial intelligence (AI) embedded UAV systems in the future. This study introduces the Autonomous Drone (Vechür-SIHA), a novel AI-embedded UAV system designed for real-time detection and tracking of other UAVs during flight sequences. Leveraging advanced object detection algorithms and an LSTM-based tracking mechanism, our system achieves an impressive 80% accuracy in drone detection, even in challenging conditions like varying backgrounds and adverse weather.

Our system boasts the capability to simultaneously track multiple drones within its field of view, maintaining flight for up to 35 minutes, making it ideal for extended missions that require continuous UAV tracking. Moreover, it can lock onto and track other UAVs in mid-air for durations of 4-10 seconds without losing contact, a feature with significant potential for security applications.

This research marks a substantial contribution to the development of AI-embedded UAV systems, with broad implications across diverse domains such as search and rescue operations, border security, and forest fire prevention. These results provide a solid foundation for future research, fostering the creation of similar systems tailored to different applications, ultimately enhancing the efficiency and safety of UAV operations. The novel approach to real-time UAV detection and tracking presented here holds promise for driving innovations in UAV technology and its diverse applications.

Keywords: Deep Learning; Object Detection; ROS; Aerial Vehicle; LSTM; UAV

1. INTRODUCTION

In recent years, unmanned aerial vehicle (UAV) platforms in civil and military fields have increased daily and become a critical technology. The artificial intelligence (AI) embedded UAV system has great importance and potential in the future [1-3]. Research and development, search and rescue operations, aerial photography, videography, border security, traffic control, forest fire prevention, working in toxic chemical gas environments, preventing poaching, natural resource exploration, agriculture, extraction, and UAV detection while in the air are all applications for AI embedded UAVs. UAV usage is becoming more common due to the nature of aircraft. The increased usage has given resulted in unwanted UAVs in restricted areas. Tracking trespassing UAVs is also important and detecting the UAVs is an emerging field [4-6]. Especially drone detection in the air is challenging due to the fast-changing background. Therefore drones can be detected using AI with other vehicles.

Specialized systems must be created for tracker UAVs. Rotary wing UAVs(drones) can contain hardware units such

as brushless DC motors, electronic speed controllers, various sensors (pressure, gyro, compass, GPS, ultrasonic), propellers, power systems, cameras, and communication systems. The rotary-wing unmanned aerial vehicle takes off with the lift force obtained by rotating the rotors of the propellers. Rotation of the rotor at different speeds, the rotation movements performed on the fuselage axis set in the center of the aircraft enable it to move in the horizontal or vertical axis. When the rotors' angular velocity changes, a signal is sent to them to allow the flight controller to compile the directives and provide the rotors with the correct orientation [1].

UAVs are aircraft that can move in a predetermined trajectory utilizing the sensors on board or can be controlled by the user using a remote control. Advances in sensor technology and embedded electronic device manufacturing have allowed these tools to increase in capabilities and reduce size. In line with these developments, the UAV has become widely used in defense and civil applications. UAV border security is ensured in the defense industry and at crime scenes. In addition to being employed for applications like research, surveillance, adversary, and target

identification, it may also be used in civil applications for things like locating fire zones, search and rescue, and assessing geographic changes brought on by natural catastrophes [4-6]. With the aid of cameras, Drones can detect UAVs with embedded deep-learning algorithms.

With object tracking, the aim is to keep the information of the relevant object from entering the scene to its exit by matching the detected objects between consecutive video frames. As a result of the object tracking, activities given as input to the optimization phase of the video synopsis are created [3]. The tracking algorithm and deep learning embedded system can detect and track other UAVs [7-9]. LSTM (long short-term memory) can be tracked with a deep learning algorithm [10-12].

In this project, Autonomous Drone (Vechür-SIHA) was developed for detecting and tracking other UAVs while in flight sequence. The system was simulated in Robot Operating System (ROS). Inside the simulated environment drone model was used for tracking UAV. After simulations and calculations, a Fighter drone was developed and built. The real-time UAV detection was made with object detection algorithms inside the simulation and in real-time. The system was embedded with a tracking algorithm based on LSTM (Long-short term memory). The system achieves high-accuracy drone detection in changing backgrounds and weather conditions (%80 accuracy), LSTM drone tracking can be accomplished even with multiple drones inside the field of view, and the system is capable of flying for 35 minutes. The system also can be locked other UAVs and track them while flying. The system can track other UAV's 4-10 seconds without losing contact.

In recent years, the usage of unmanned aerial vehicle (UAV) platforms has increased dramatically in both civil and military fields, leading to the development of critical new technologies. One area that holds particular promise for the future is the use of artificial intelligence (AI) embedded UAV systems, which have the potential to revolutionize a wide range of applications [1-3]. These applications include research and development, search and rescue operations, aerial photography and videography, border security, traffic control, forest fire prevention, working in toxic chemical gas environments, preventing poaching, natural resource exploration, agriculture and extraction, and UAV detection while in the air. UAV usage is becoming increasingly common, thanks to its unique capabilities and versatility, but this has also led to the emergence of unwanted UAVs in restricted areas. Tracking and detecting these trespassing UAVs has become an urgent need in many contexts, making UAV detection and tracking an emerging field of research [4-6]. One of the main challenges in this area is detecting UAVs in the air, which is made difficult by the fast-changing background. However, AI can be used to detect drones in combination with other vehicles, and specialized systems can be developed for tracking UAVs.

Rotary wing UAVs (also known as drones) are equipped with hardware units such as brushless DC motors, electronic speed controllers, various sensors (including pressure, gyro, compass, GPS, and ultrasonic sensors), propellers, power systems, cameras, and communication systems [1]. The

rotary-wing unmanned aerial vehicle takes off with the lift force obtained by rotating the rotors of the propellers. Rotation of the rotor at different speeds, and the rotation movements performed on the fuselage axis set in the center of the aircraft, enable it to move in the horizontal or vertical axis. When the rotors' angular velocity changes, a signal is sent to them to allow the flight controller to compile the directives and provide the rotors with the correct orientation. Advances in sensor technology and embedded electronic device manufacturing have allowed UAVs to increase in capabilities and reduce in size, making them widely used in defense and civil applications.

In the defense industry and at crime scenes, UAV border security is ensured by deploying UAVs for applications like research, surveillance, adversary, and target identification. UAVs may also be used in civil applications, such as locating fire zones, search and rescue operations, and assessing geographic changes brought on by natural catastrophes [4-6]. With the aid of cameras, drones can detect other UAVs using embedded deep-learning algorithms.

Object tracking aims to keep the information of the relevant object from entering the scene to its exit by matching the detected objects between consecutive video frames. As a result of object tracking, activities given as input to the optimization phase of the video synopsis are created [3]. The tracking algorithm and deep learning embedded system can detect and track other UAVs with high accuracy [7-9]. One popular method of doing this is to use a long short-term memory (LSTM) algorithm, which has shown promising results in previous research [10-12].

The developed autonomous drone system, called Vechür-SIHA, was able to detect and track other UAVs in real time during flight sequences. The system was first simulated in the Robot Operating System (ROS) environment using a drone model to track the UAVs. Based on the simulation results and calculations, a small fighter drone was then designed and built. The system was equipped with an object detection algorithm that allowed for real-time UAV detection, even in changing backgrounds and weather conditions, achieving an accuracy of up to 80%. Furthermore, the system utilized a tracking algorithm based on long-short-term memory (LSTM) to track multiple drones simultaneously within its field of view, with a tracking duration of 4-10 seconds before losing contact. The developed system was also capable of flying for up to 35 minutes and locking onto and tracking other UAVs while in flight. These results demonstrate the feasibility of using AI-embedded UAVs for detecting and tracking other UAVs, which could be useful in a wide range of applications such as border security, surveillance, and search and rescue operations.

2. MATERIALS AND METHODS

In this section, the system development from simulation to real-life application. The study was organized: ROS simulation, Deep learning-based drone detection algorithm development, mechanical and electronic system design, and real-time flight application.

Vechür-SIHA is designed for fighter UAV competitions in Teknofest/Turkey. The competition goal is to detect other UAVs with vision systems and track them without breaking visual contact for 10 seconds. In every flight, more than 12 UAVs were flying in simultaneously. Our Autonomous System (Vechür-SIHA) is designed as a hexacopter with a rotating wing structure. The system can be controlled fully autonomously or manually. Vechür-SIHA can engage in dogfights through blocking or evasive maneuvers by detecting other nearby unmanned vehicles with the artificial intelligence-assisted visual detection system. In addition, the UAV will transmit the images and flight data received during the flight to the ground station in real-time using antennas that receive and transmit radio frequency signals. While our system position data is shared with the competition server connected via ethernet, other UAV data is received. The server also sends information about other UAVs to track and make evasive maneuvers. With the 2.4 GHz radio control, manual control can be encrypted and carried out at a sufficient distance. The system was simulated in a Robot operating system (ROS), and object detection algorithms were used for UAV detection for simulations and real-life applications.

Drones, also known as small, remotely-controlled unmanned aerial vehicles (UAVs), are used in a variety of societal roles such as law enforcement, medical, construction, search and rescue, parcel delivery, remote area exploration, topographic mapping, forest/water management, and inspection of large infrastructures such as power grids [1]. Their low cost and ease of operation have made drones accessible for recreational and entertainment purposes [2]. However, drones can be intentionally or unintentionally misused, posing a threat to the safety of others. For instance, an aircraft can be severely damaged if it collides with a consumer-sized drone, even at moderate speeds [3]. Additionally, an ingested drone can quickly disable an aircraft engine. The increasing occurrence of drone sightings in restricted airport areas is also a significant risk, leading to the total closure of airports and the cancellation of hundreds of flights [4]. Some hobbyist drone operators violate aviation safety regulations, sometimes without knowledge, leading to several near-misses and verified collisions with UAVs. Thus, research on drone detection has increased significantly [5,6] to counteract potential risks due to intrusion in restricted areas, either intentional or unintentional.

This paper addresses the design and evaluation of an automatic multi-sensor drone detection and tracking system using state-of-the-art machine-learning techniques. We extend the methods from conclusions and related literature recommendations [5,7] to enhance our development. In addition to effective detection, classification, and tracking methods, sensor fusion is also considered a critical open area to achieve greater accuracy and robustness compared to a single sensor. However, research in sensor fusion for drone detection is limited [7-10]. This work includes collecting and annotating a public dataset to train and evaluate the system. A lack of public reference databases serves as a benchmark for researchers [5]. Thus, we include three different consumer-grade drones in the dataset together with birds, airplanes, and helicopters, which constitutes the published dataset with the largest number of target classes (drone, bird,

airplane, and helicopter). In building the classes, we consider other flying objects that are likely to be mistaken for a drone [11,12]. Additionally, we address the system's classification performance as a function of the distance to the target, with annotations of the database including such information.

A preliminary version of this article appeared at a conference [13]. In this contribution, we substantially increase the number of reported results. For instance, we extensively analyze the effect of the internal parameters of different detectors on their performance for various sensors. We also report results with a radar module and provide comments about the fish-eye camera motion detector, all of which were missing in the previous publication. Additional results on the fusion of sensors are also provided, including an Appendix with complementary observations and visual examples. Furthermore, we provide new detailed information about the system architecture, hardware, and software employed, including details about implementation and design choices not included in the previous publication. We describe the related work in more detail.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 extensively describes the proposed system, including the architecture, hardware components, involved software, Graphical User Interface, and dataset. The experimental results are presented and discussed in Section 4. Finally, the conclusions are presented in Section 5.

Several sensors can be used for drone detection, such as radar (on several different frequency bands, both active and passive), cameras in the visible spectrum, cameras detecting thermal infrared emission (IR), microphones to detect acoustic vibrations, sensors to detect radio frequency signals to and from the drone and the controller (RF), and scanning lasers (Lidar) [8]. As explored in [14], even humans can be employed for the task, and animals can be trained for

2.1. Autonomous Drone Specifications

Vechür-UAV can be used at an altitude of 2 km, 35 minutes of flight time was calculated considering the power consumption of the motor and the 22000 mAh value of the battery. The selected motors are enough to accelerate the Vechür-SIHA, which weighs around 4 kg to 15 m/s.

A 2 MP wide-angle camera was used. The captured images are processed by the artificial intelligence algorithm working on the embedded system computer (Jetson Nx-Nvidia, USA), capable of 21 trillion operations per second using 48 tensor cores. Thus, a Vechür-SIHA is developed with the infrastructure capable of performing tasks that require high computing power, such as target detection, target maneuver estimation, and target locking and tracking. A deep learning object detection algorithm (Yolov4-Tiny) was used, and TensorRT optimization was performed to improve the processed frame per second. The system was capable of image processing at 22 FPS (frame per second), which is suitable for detecting UAVs through the wide-angle camera. Due to our data and specifications, the estimated minimum detection range was 5 meters, and the maximum was 50 meters. It is planned to transmit end-to-end encrypted flight

telemetry data at a range of 40 km and transmit it to the ground station in near real-time. In addition, the video transmission system is designed to transmit images at a range of about 4 km with delays of less than 30 milliseconds.

Vechür-SIHA constantly receives GPS data from the competition server and moves through to the other UAV's location. Suppose the UAVs detect the system and start analyzing trajectory, the number of elements with LSTM (Long-short term memory), and data received from the ground station. In that case, the system is also embedded with sub-systems that can follow the course using SLAM algorithms to control its route. SLAM is used for the shortest route calculation for following other UAVs.

2.2. Autonomous Drone Mechanical and Electronic System Design

In the mechanical design phase, the chassis diameter was determined as 600 mm, taking into account the dimensions of the electronic materials and the minimum distance between the propellers due to drone weight. The system consists of an embedded computer, flight controller, and battery pack. The chassis is designed in two layers to position the Electronic system, which has a developer kit, control card, and communication modules and is given in Figure 1.

Carbon fiber will be used in the drone's top-bottom plate, arms, and engine holder apparatus to provide durability and lightness. A modular flight computer holder has been designed on the bottom plate for easier removal and installation of the Jetson Xavier NX, which will be positioned in the center of the drone. The current breaker used as part of the security measures is shown in Figure 2 and is placed such that it may be rapidly intervened in case of a bad circumstance.

The system was designed with an appropriate deep-learning platform (Jetson Xavier NX). The electronic flight controller system was embedded with a 32-bit high-performance STM32F427 ARM Cortex M7 processor that compiles real-time information collected from sensors and other flight units into the NuttX RTOS operating system, providing safe autonomous/manual flight. Inside the cube, isolated from vibration and external factors, 3 IMUs and two barometric sensors communicating with SPI protocol provide highly accurate real-time information. In addition to 8 main and six auxiliary output pins, and also supports communication protocols such as UART, CAN, I2C, USB, DSM, and S-BUS. The Pixhawk Cube, Flight Control Board, includes a dedicated processor, an independent power supply, and an integrated backup system that includes protocols such as fail-safe and manual in-flight override (Figure 1).

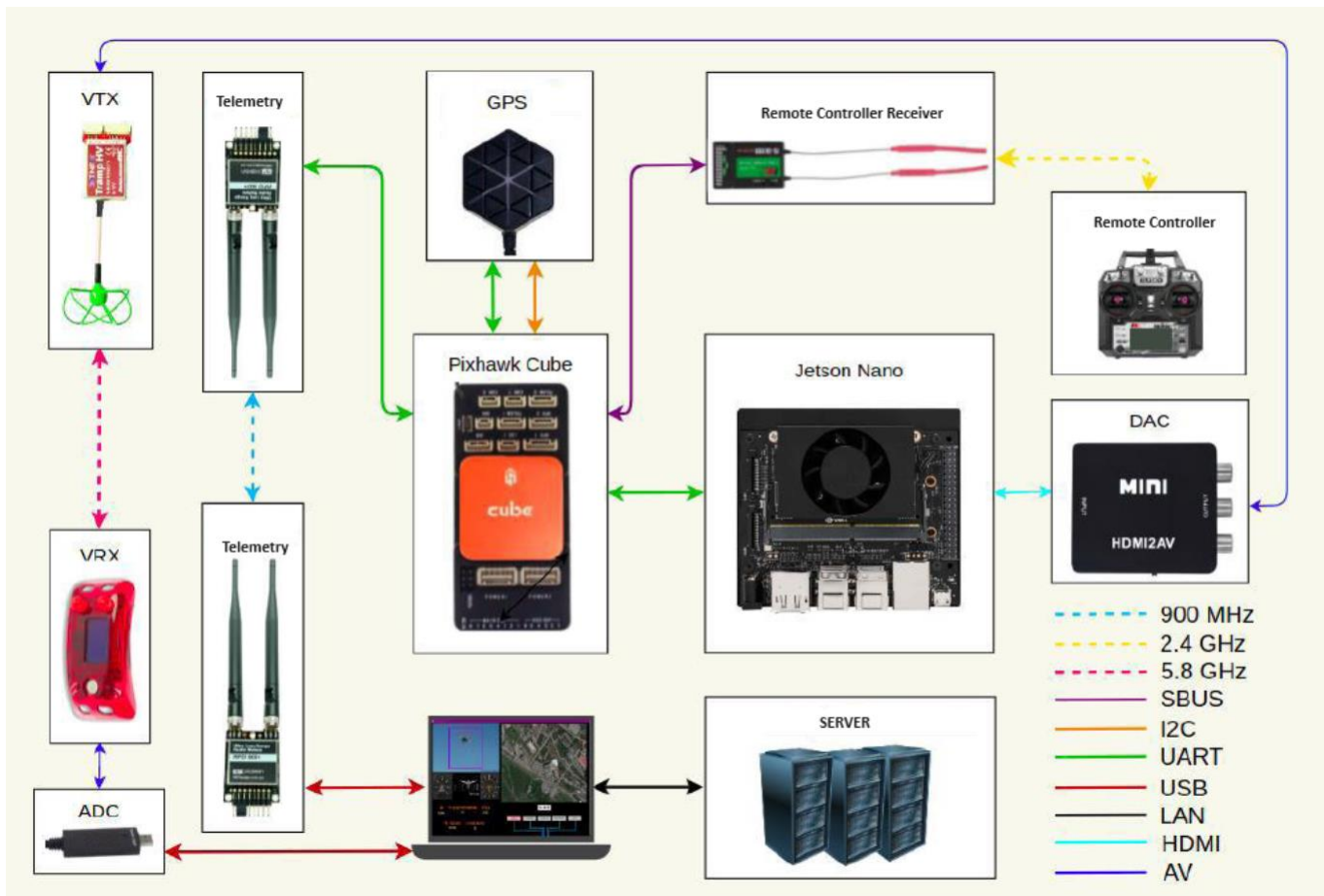


Figure 1. System and communication diagram

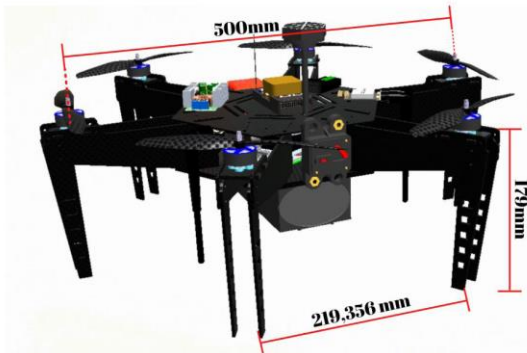


Figure 2. Autonomous drone mechanical design

Considering the weight of the 6-blade Vechür-SIHA system, as demonstrated in Figure 2, there are approximately 669 grams of thrust per hover engine. Thrust power was

measured with a thrustmeter for more accurate system balancing and power management.

Static and flow analyses of Vechür-SIHA were carried out using the 3D analysis program. The average density and weight of carbon fiber used in Vechür-SIHA are 1.76 g/cm³ and 0.198 g/m², respectively. The tensile strength of the carbon fiber is 3,530 MPa, the modulus of elasticity is 230 GPA, and the end stress is 1.5%, according to the manufacturer. According to the information in Figure 3, carbon fiber with Young's modulus of 230 GPA was selected. The maximum force at takeoff and during flight (assumed at full throttle) was derived from the bottom of the engine as a function of the engine's thrust in the static analysis of the Vechür-SIHA arms. Figure 3. shows that when the maximum force is applied, a maximum force of 8.46 MPa is generated on the arms. As a result of the analysis, it can be seen that the arms exceed the tensile strength of carbon fiber.

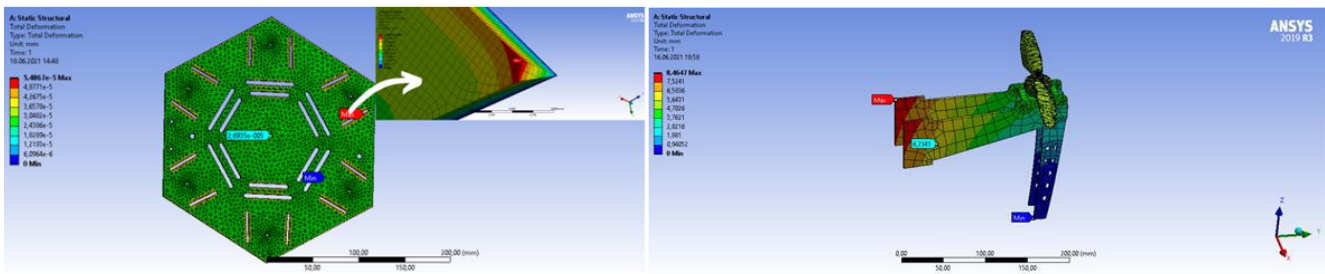


Figure 3. Static analysis of Vechür-SIHA brushless DC motor holder and landing gear



Figure 4. ROS simulated environment for drone detection

2.3. Robot Operating System(ROS) Simulation

The ROS Melodic, a well-established version, was used. A gazebo was used for environment simulation. Firstly, the necessary infrastructure was prepared for more than three rotary-wing UAVs to be able to fly (Figure 4) simultaneously. The SDF files of the UAV model, which are available in the hector_quadrotor packages used in the ROS environment, are designed according to demand and the capabilities of dogfighting competition. SITL software was used for establishing the Ardupilot-ROS connection, and the UAVs can perform autonomous flight according to the coordinates entered in the simulation environment. Autonomous detection of targets is ensured by integrating OpenCV libraries and darknet_ros structure into the Gazebo environment. Afterward, developed algorithms in the ROS environment were tested under real-world conditions.

2.4. Data Collection, Data Augmentation, and Object Detection-Tracking Algorithm

Artificial intelligence algorithm was used in both ROS and Real-world applications. Deep learning models like R-CNN, YOLO, and SSD that can generate real-time predictions are widely used [12]. In our system, FPS and optimization are very crucial. In a real-life environment, only prediction is not enough; simultaneously, the drone needs to be detected without losing contact. Jetson Nano (Nvidia, USA) is embedded inside the drone for real-time application. Jetson nano can be used in deep learning algorithms due to tensor cores, small size, and inference optimization. Because of the FPS needs, a small object problem Yolo Tinyv4 was selected. TenorRT and Onnx optimization can be used with the YOLO algorithm family. Also, Yolo Tiny models can detect smaller objects at a higher speed [12].

YOLOv4 Tiny Algorithm

✓ Input image:

- YoloV4 Tiny takes an input image as its initial input.

✓ Network architecture:

- YoloV4 Tiny uses a modified neural network architecture, which is typically a smaller and shallower version of the YoloV4 architecture to reduce computational complexity.
- It consists of a series of convolutional layers, followed by downsampling and upsampling layers to extract features and reduce the spatial dimensions of the image.

✓ Anchor boxes:

- YoloV4 Tiny uses anchor boxes to predict the bounding boxes for objects. These anchor boxes are predefined in terms of width and height to match the expected object sizes in the dataset.

✓ Object detection:

- The network predicts bounding boxes for objects and class probabilities at multiple scales (usually 13x13 and 26x26 grids).
- Each grid cell predicts a fixed number of bounding boxes (usually 3 or 6, depending on the configuration).
- For each bounding box, the network predicts the (x, y) coordinates of the bounding box's center relative to the grid cell, width, height, and class probabilities.
- The class probabilities represent the likelihood of the object belonging to a specific class (e.g., person, car, dog).

✓ Non-maximum suppression (NMS):

- After predictions are made at multiple scales, a post-processing step called non-maximum suppression is applied to remove duplicate and low-confidence detections.
- NMS selects the bounding box with the highest confidence for each object and removes overlapping boxes that have a high intersection-over-union (IoU) with the selected box.

✓ Output:

- The final output of YoloV4 Tiny is a list of bounding boxes, each associated with a class label and a confidence score.

✓ Object classification:

- YoloV4 Tiny can classify objects into predefined classes based on the highest class probability associated with each bounding box.

✓ Bounding box refinement:

- Optionally, the bounding box coordinates can be refined to improve the accuracy of object localization.

✓ Post-processing:

- The final detected objects can be drawn on the input image, and their class labels and confidence scores can be displayed.

✓ Evaluation and optimization:

- YOLOv4 Tiny's performance is evaluated using metrics like mean average precision (mAP).
- Model training and hyperparameter optimization are performed to improve detection accuracy.

YoloV4 tiny algorithm explained briefly. Implementing YoloV4 Tiny typically requires expertise in deep learning frameworks like TensorFlow or PyTorch, along with access to labeled training data for specific object detection tasks.

The system deep learning models were prepared according to the changing environment and fast movements [13-15]. Therefore, the data was augmented using several techniques to simulate different movements and angles (Figure 5). Data augmentation techniques included flipping, mirroring, adjusting brightness, blurring, gamma, and adding Gaussian noise. The resemblance of the movement effect in our real-time flying sequence led to the augmentation. Fast movements, shutter speeds, frames per second (FPS), weather conditions, and sun positions can impact real-time images. Our data consist of four thousand images. The augmentation was made limitedly to prevent overfitting of our model. In the competition, every team has a different drone design, which affects the model accuracy in real life.

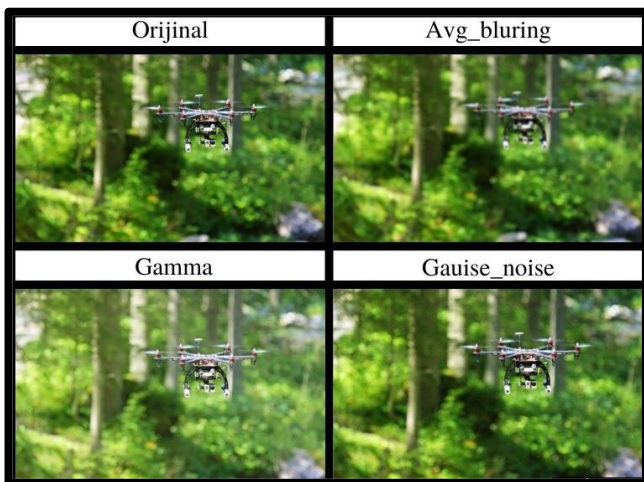


Figure 5. Data augmentations examples

Algorithm for Drone Detection and Tracking with YOLOv4 Tiny is explained briefly in the followint sequence:

✓ **Image acquisition and preprocessing:**

- Real-time imagery is captured using the drone's onboard camera.
- Captured images are preprocessed, resized, and normalized for input compatibility with the YOLOv4 Tiny model.

✓ **Model deployment and inference:**

- A pre-trained YOLOv4 Tiny model, optimized for real-time inference, is deployed.
- The model analyzes the preprocessed images for drone detection.

✓ **Object detection:**

- Detected drones are identified in the images, taking into account their presence and spatial coordinates.

✓ **Non-maximum suppression (NMS):**

- Non-maximum suppression is applied to refine drone detections, eliminating redundancy and preserving confidence.

✓ **Tracking initialization:**

- A tracking algorithm (e.g., SORT) is initialized, associating unique IDs with each detected drone for tracking continuity.

✓ **Drone tracking and prediction:**

- Continuous monitoring updates the drone's position and movement over time.
- Predictive capabilities estimate future drone positions based on historical trajectories.

✓ **Real-time ground station communication:**

- The tracking system transmits real-time information on detected drones, including their positions and IDs, to a ground station.

✓ **Visualization and alert mechanisms:**

- The tracked drones are visualized on the ground station's interface, while alert mechanisms promptly notify operators of any unauthorized or suspicious drone activity.

✓ **Collision avoidance (optional):**

- As needed, a collision avoidance system is integrated to maintain safe distances between the tracking drone and others, implementing avoidance actions when necessary.

✓ **Continuous operation and system resilience:**

- The system is designed for continuous operation throughout the drone's mission.
- It is equipped to handle system failures and ensure robust performance.

✓ **Mission completion and termination:**

- The algorithm concludes when the drone mission is successfully completed.

The drone detection and tracking algorithm with YOLOv4 Tiny involves capturing real-time imagery from a drone's camera, preprocessing the images, and running them through a pre-trained YOLOv4 Tiny model for object detection to identify other drones in flight. Non-maximum suppression is applied to refine the drone detections. Then, a tracking algorithm is initialized to continuously monitor and predict the movements of detected drones, assigning unique IDs to each. The tracked drone information is communicated to a ground station in real-time for visualization and alerts, ensuring the operators are aware of drone activity. Optionally, collision avoidance measures can be implemented. This system operates throughout the drone's flight mission and terminates upon mission completion. The algorithm's specific implementation will depend on the hardware, software, and application requirements.

2.4.1. LSTM (Long Short-Term Memory) tracking algorithm

The tracking of the detected object is a critical task. The control algorithm is dependent on the detection signal. Also, to perform the tracking, it is necessary to use an iterative neural network to establish a relationship between the object's previous steps and its current position. In the system, LSTM (Long Short-Term Memory) was preferred. LSTM has four gates: Forget, Input, Cell State, and Output. Gate of Forgetting: It is the gate that decides which information will be forgotten or kept. Information from the current input (X_t) and the previous hidden state (h_{t-1}) is subjected to the sigmoid activation function. Entrance Gate: It consists of two parts; First, the previous hidden state (h_{t-1}) and the current input (X_t) are passed through a sigmoid process to decide which values to update.

Then, the same two inputs are "tanh" activated to regulate the mesh and multiplied by the sigmoid output (it) to update the cell state (C_t). Cell state: Input from the previous cell state (C_{t-1}), multiplied pointwise by the output of the gate. If the forget output is 0, it discards the previous cell output (C_{t-1}). This output is punctually added with the input gate output to update the new cell state (C_t). The current cell state will be the entry for the next LSTM unit. Exit port: The hidden state contains information about previous entries and is used for prediction. The exit port regulates the current hidden state (h_t). The previous hidden state (h_{t-1}) and the current input (x) are passed to the sigmoid process. This output is multiplied by the output of the "tanh" function to get the current hidden state. The current state (C_t) and the current latent state (h_t) are the final outputs of a conventional LSTM unit.

The primary usage of LSTM in the project;

An algorithm has been developed using the LSTM model to produce solutions against scenarios that may prevent our UAV from being locked during the competition.

ROS and LSTM were used to simulate the drone's detection during flight. Algorithms were designed to create and

develop the Vechür-SIHA software to perform air combat maneuvers and to ensure that the system performs accurately. Drone detection was simulated in the ROS gazebo environment. The ROS communication diagram is demonstrated in Figure 6.

3. RESULTS AND DISCUSSION

The designed system can perform duties at an altitude of 2 km within 17 minutes of flight time. The selected engines are strong enough to accelerate the UAV, which weighs approximately 4 kg to 15 m/s. A camera with a minimum resolution of 2 MP and a wide angle of up to 170 degrees can detect other UAVs. The captured images will be processed with the artificial intelligence algorithm working on the embedded computer, which can perform 21 trillion operations per second with 48 tensor cores on it. Thus, a UAV with the infrastructure that can perform tasks that require high processing power, such as target detection, maneuver estimation, and tracking, was made without any error. The Vechür-SIHA can transmit flight telemetry data to the ground station at a range of 40 km, in an end-to-end encrypted manner and with very low delay. In addition, the video transmission system is designed to transmit images at a range of approximately 4 km with delays of less than 30 milliseconds. In addition to directing to the surrounding UAVs with the data received from the ground station, it will contain sub-systems that can follow the course with SLAM algorithms to control its route and prevent crashing. Tests were carried out during flight; By recording parameters such as flight time, delay, and the internal temperatures of the engines during a mission when the aircraft is fully loaded.

Today, it is undeniable that UAVs provide a significant advantage to countries in terms of reconnaissance, surveillance, and usage as a weapon system, asymmetrical effect, and cost-effectiveness. The primary evidence for the dimensions and sizes of these vehicles vary according to the desired performance and uses. The vehicle family has been expanded according to the variety of weapons and ammunition they carry. The study focused on UAV technologies and subfields [1,2].

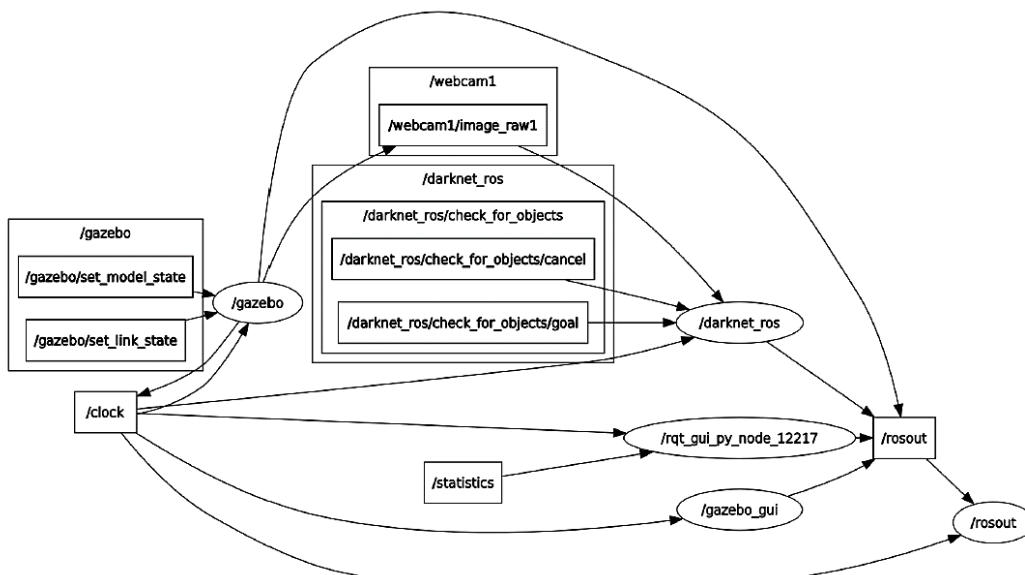


Figure 6. ROS communication diagram

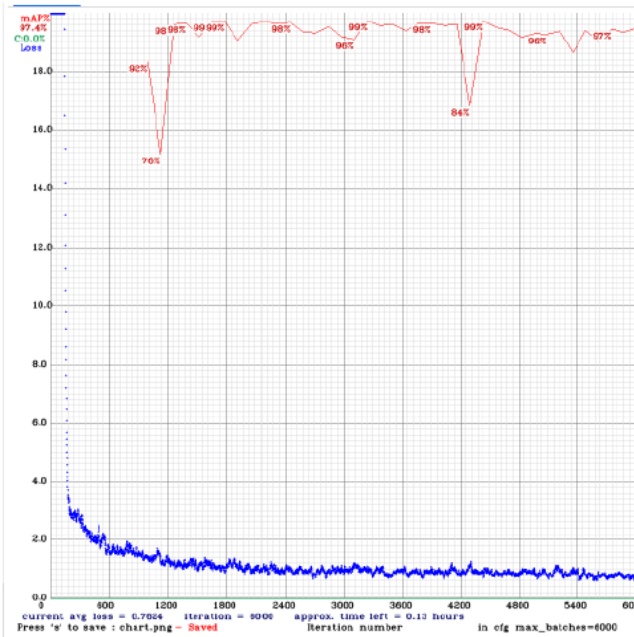


Figure 7. YOLOv4-tiny model training results.

Designed system ROS simulation results, Deep learning model prediction results, and system integration was demonstrated in this section. The real-time performance and accuracy comparison shows that the performance of the YOLO structure is better than the other models compared to the FPS and AP (Accuracy Points) [1-3].

Table 1. YoloV4 and YoloV4-Tiny model training results.

Model	Accuracy	Loss	FPS
YoloV4-Tiny	%97	%0.323	10.10
YoloV4	%98.5	%0.124	7
YoloV4-Tiny			TensorRT 24.99 FPS

Table 2. Hyperparameter optimization for YoloV4-tiny

Hyperparameter	Range or Options
Learning Rate	[0.001]
Batch Size	[16]
Number of Epochs	[120]
Backbone Architecture	[CSPDarknet53]
Anchor Box Scales	[[2, 4], [1, 3], [3, 5]]
Anchor Box Ratios	[[0.5, 1.0], [0.75, 1.0], [0.5, 1.5]]
Input Image Size	[416x416]
Data Augmentation	[Yes]
Weight Decay	[0.0001]
Dropout Probability	[0.2]
Optimizer	[Adam]
Learning Rate Scheduler	[StepLR]

The Darknet-53 architecture, consisting of 53 layers of convolution, is used to deepen the YOLOv3 network structure, and residual blocks have been added to this network. Instead of using the softmax function in YOLOv3, the logistic function is used for multi-label predictions. Another critical feature of YOLOv3 is its multiscale prediction, which improves the algorithm's ability to predict small objects. In 2020 the YOLOv4 algorithm was introduced, which is a further development of YOLOv3 [20]. Some new technologies, such as weighted residual links, cross-stepped partial links, and cross-mini-batch normalization, are built into this algorithm to improve the speed and accuracy of object detection. In this context, the YOLOv4-Tiny (416 x 416) architecture is used (Figure 7), which offers the highest real-time performance and prediction accuracy among the current models for small-sized objects suitable for our system.

This dataset was relatively complicated. The background in flight constantly changes. In our dataset, there are several different backgrounds with different drones. After model training, and hyperparameter optimization as shown in Table 2, TensorRT optimization was performed, and the fps was improved from 10 to 25 FPS (Table 1 and Figure 8).

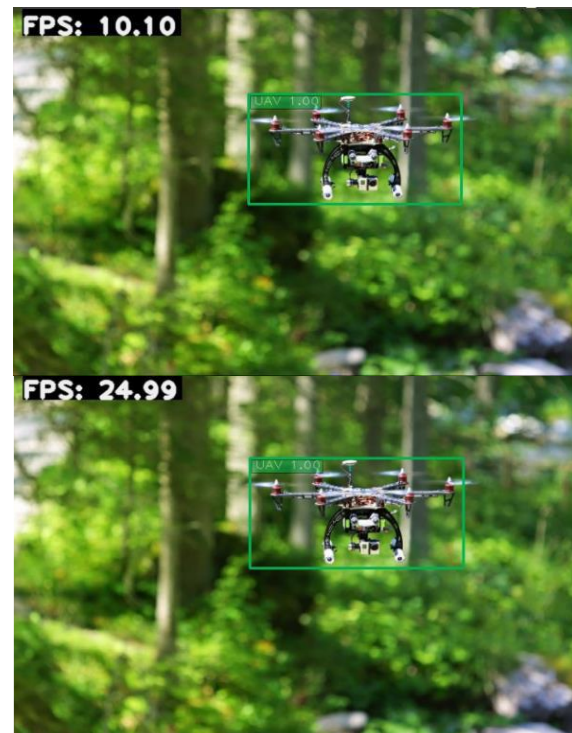


Figure 8. Model training results and TensorRT optimization real-time results

Using the LSTM model, an algorithm was developed to solve scenarios that could prevent the Vechür-SIHA from being locked during the competition.

Some sample scenarios;

- Image distortions,
- Errors in target detection,
- Confrontation with more than one target at a time, algorithm steps for preventing the tracking faults.

The simulations were carried out in ROS and then implemented in the system.

1. The location data of the objects obtained from the YOLOv4 Tiny neural network are obtained on the images.

2. When the target indicated by the target UAV determination algorithm is reached according to the location data shared by the server

a. Comparing GPS orientations of UAVs requested to be tracked even with the presence of more than one UAV in the field of view and the movements of the UAVs detected by the camera,

b. Single target is tracked without restrictions, as in Figure 9 a.

3. Positions detected on the second video frame are estimated by LSTM, as shown in Figure 9 b.

4. In Figure 9 c, the positions of the UAVs from the third video frame are compared to the positions estimated by the LSTM in the second frame, and the UAV with the highest percentage of agreement (IoU) is tracked.

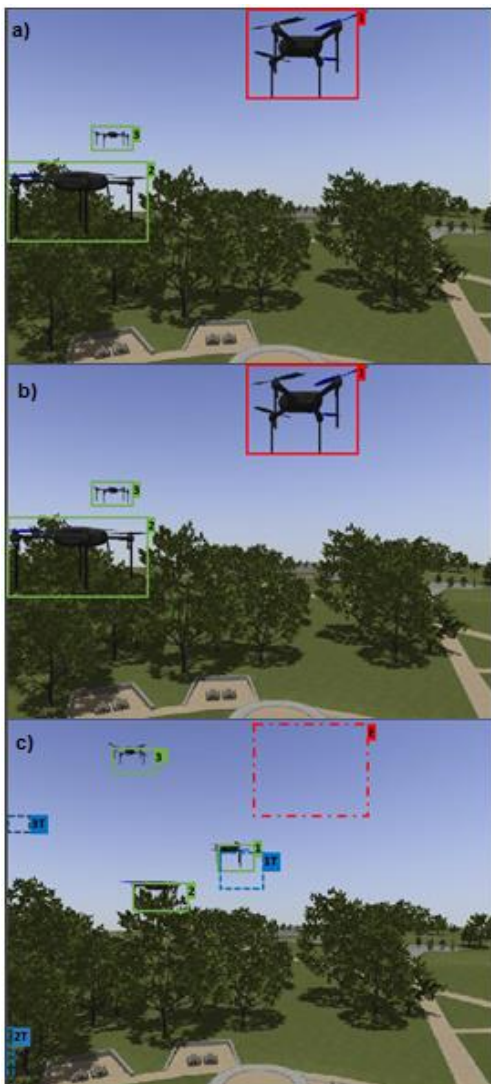


Figure 9. LSTM Location estimation with the red bounding box



Figure 10. Real-time application of Yolov4-tiny Drone detection algorithm.

Escape algorithm; According to the competition terms, Vechür-SIHA is obligated to avoid the ban operation of other UAVs during the competition. In this direction, the movements of the Vechür-SIHA are stored with the LSTM structure, and the outputs of the target tracking system of other UAVs are simulated (Figure 10). The stages of this process;

1) UAV GPS movement is similar to Vechür-SIHA for 5 seconds and is considered a threat, and escape maneuvers were made.

2) By calculating the estimated orientation of the Vechür-SIHA with the LSTM structure, the orientation of the locking quadrant of the other UAVs is estimated.

3) The escape operation is performed when Vechür-SIHA escapes in the opposite direction of the orientation predicted by LSTM.

4) However, when the Vechür-SIHA tries to detect competitor UAVs, the system decides on locking priority within other UAVs.

Real-time UAV detection was made with our developed Drone system. Deep learning and LSTM algorithm were used while Vechür-SIHA was flying. The real-time application of the UAV system example is shown in Figure 11.

LSTM Real-Time Locking Test: In this direction, the LSTM model has been trained, and solutions have been produced for scenarios that may occur simultaneously with more than one target. The trained LSTM model was tested on a YOLOv4-Tiny model with blue caps detected for convenience. Some of the steps performed in the tests;

- Identifying the closest target while in the field of view of more than one UAV at the same time,
- Compensating for frame losses in the target detection model,
- It can be listed as maintaining the visual-locking on the initially determined UAV and completing the tracking without interfering with other UAVs.



Figure 11. Vechür-SIHA system

Target detection and tracking tasks, which are the primary mission of the UAV system, were carried out at the Sakarya University of Applied Sciences football field and Bursa Yunuseli Airport, and the study's final results were obtained. The detection rate of the UAVs displayed in the trials reached 80%, and the follow-up process was carried out. However, the detection rate at Bursa Airport was recorded as 45%. The reasons for achieving such a low detection accuracy were determined as follows.

- Since remote UAVs are difficult to detect and are shown in very-low resolution by the camera and lens technology that is being used,
- Distortions in the images due to the vibrations and shifting of the field of view,
- Detection algorithms for such technical problems of the UAV system need to be implemented with suitable hardware systems,

The relatively low accuracy due to the detection's changing backgrounds made tracking targets challenging.

Therefore, the system was developed with object tracking algorithms to prevent these and improve accuracy. As a result of the trials, 4-10 second locking and tracking of other UAVs was accomplished and was shown in Figure 12.

The intersection of drone technology and deep learning has ignited a transformative era in aerial imagery analysis. This concise summary encapsulates key insights from two seminal papers [21, 22], highlighting the diverse and impactful applications of this fusion.



Figure 12. Real-life application and competition environment.

Researchers unveil the realm of real-time human action recognition in drone imagery, leveraging deep learning to monitor and identify human activities from aerial perspectives [23]. Another system pioneer the integration of deep learning with RGB and thermal imaging on drones, enhancing monitoring operations with improved situational awareness and precision [24,25].

These groundbreaking studies underscore the potential of deep learning in drone-based imagery analysis, with applications spanning security, infrastructure monitoring, and urban planning. Moreover, they emphasize the need for interdisciplinary collaboration, bridging computer vision, remote sensing, and robotics to advance drone systems.

4. CONCLUSION

The increasing use of commercial drones presents a pressing need for efficient tracking during flight, a complex task due to diverse embedded algorithms and dynamic backgrounds [1-2]. The developed UAV system successfully detects, tracks, and provides information about UAVs in flight, offering the potential to protect against unauthorized UAVs and enhance various applications. However, further improvements are required for more precise tracking and enhanced image acquisition systems [9].

Tracking UAVs in restricted areas holds paramount importance for several reasons. Firstly, it safeguards airspace safety by swiftly identifying and addressing potential hazards, benefiting air traffic and ground safety [15-16]. Secondly, it combats illicit activities such as terrorism and espionage involving drones [17-18]. Lastly, monitoring UAVs aids in managing the increasing congestion of airspace and preventing collisions, a critical concern given the rising number of UAVs [17-18].

Achieving UAV tracking relies on a suite of technologies, including radar, GPS, and computer vision. Radar provides essential data on UAV location, speed, and altitude, while GPS precisely pinpoints UAVs and predicts their trajectories. Computer vision, including object detection and identification, aids in UAV recognition [19].

Real-time data analysis through advanced algorithms and machine learning further enhances UAV tracking, helping identify unusual or hazardous activity [20]. Integration with security and air traffic control systems offers a comprehensive airspace overview, aiding authorities in informed decision-making and ensuring airspace safety and efficiency [22-24].

In summary, effective UAV tracking in restricted areas demands a multifaceted approach, combining diverse technologies and cutting-edge data analysis methods. This comprehensive system is indispensable for upholding airspace safety, security, and efficient management.

Author contributions: Ali Furkan KAMANLI: Conceptualization, Methodology, Software, Data curation, Writing- Reviewing and Editing.

Conflict of Interest: There is no conflict of interest.

Financial Disclosure: The study was funded by Sakarya University of Applied Sciences BAP with project number 078-2022.


REFERENCES

- [1] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (23 Apr 2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. Taiwan: Academia Sinica, Taiwan.
- [2] Huang, Z., Xu, W., & Yu, K. (09 Aug 2015). Bidirectional LSTM-CRF Models for Sequence Tagging. <https://arxiv.org/pdf/1508.01991.pdf>.
- [3] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. Department of the Computer Science University of Oxford.
- [4] Lee, H., & Kim, H. (2017). Trajectory Tracking Control of Multirotors from Modelling to Experiments: A Survey. *International Journal of Control, Automation and Systems* 15(1) (2017) 281-292.
- [5] Lon, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. California: UC Berkeley.
- [6] Redmon, J., & Farhadi, A. (8 Apr 2018). YOLOv3: An Incremental Improvement. Washington: University of Washington.
- [7] BAŞARAN, E. (Aralık 2017). PERVANE PERFORMANSININ ANALİTİK VE SAYISAL YÖNTEMLERLE HESABI. TOBB Ekonomi ve Teknoloji Üniversitesi.
- [8] Braun, S. (5 Jun 2018). LSTM Benchmarks for Deep Learning Frameworks. <https://arxiv.org/pdf/1806.01818.pdf>.
- [9] Hoffmann, G., & Waslander, S. (2009). Quadrotor Helicopter Trajectory Tracking Control. American Institute of Aeronautics and Astronautics.
- [10] Noh, H., Hong, S., & Han, B. (2015). Learning Deconvolution Network for Semantic Segmentation. Department of Computer Science and Engineering, POSTECH, Korea.
- [11] Radiocrafts. (2017). RF Modules Range Calculation and Test. By T.A.Lunder and P.M.Evjen, 1-13.
- [12] Jiang, Zicong, et al. "Real-time object detection method based on improved YOLOv4-tiny." *ArXiv abs/2011.04244* (2020): n. page.
- [13] S. Minaeian, J. Liu, and Y.-J. Son, "Effective and efficient detection of moving targets from a UAV's camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 497–506, Feb. 2018.
- [14] R. Opromolla, G. Fasano, and D. Accardo, "A vision-based approach to UAV detection and tracking in cooperative applications," *Sensors*, vol. 18, no. 10, 2018.
- [15] V. Walter, M. Saska, and A. Franchi, "Fast mutual relative localization of UAVs using ultraviolet led markers," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2018, pp. 1217–1226.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [17] R. Mitchell and I. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 5, pp. 593–604, May 2014.

- [18] Y. Tang et al., "Vision-aided multi-UAV autonomous flocking in GPSdenied environment," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 616–626, Jan. 2019
- [19] A. James, W. Jie, Y. Xulei, Y. Chenghao, N. B. Ngan, L. Yuxin, S. Yi, V. Chandrasekhar, and Z. Zeng, "Tracknet-a deep learning based fault detection for railway track inspection", 2018 International Conference on Intelligent Rail Transportation (ICIRT), pp. 1-5, 2018.
- [20] A.K. Singh, A. Swarup, A. Agarwal, and D. Singh, "Vision-based rail track extraction and monitoring through drone imagery", *ICT Express*, 5, 4, pp. 250-255, 2019
- [21] Azmat, U., Alotaibi, S. S., Abdelhaq, M., Alsufyani, N., Shorfuzzaman, M., Jalal, A., & Park, J. (2023). *Aerial Insights: Deep Learning-based Human Action Recognition in Drone Imagery*. IEEE Access.
- [22] Speth, S., Goncalves, A., Rigault, B., Suzuki, S., Bouazizi, M., Matsuo, Y., & Prendinger, H. (2022). Deep learning with RGB and thermal images onboard a drone for monitoring operations. *Journal of Field Robotics*, 39(6), 840-868.
- [23] Alam, S. S., Chakma, A., Rahman, M. H., Bin Mofidul, R., Alam, M. M., Utama, I. B. K. Y., & Jang, Y. M. (2023). RF-Enabled Deep-Learning-Assisted Drone Detection and Identification: An End-to-End Approach. *Sensors*, 23(9), 4202.
- [24] Zhang, C., Tian, Y., & Zhang, J. (2022). Complex image background segmentation for cable force estimation of urban bridges with drone-captured video and deep learning. *Structural Control and Health Monitoring*, 29(4), e2910.
- [25] Gupta, H., & Verma, O. P. (2022). Monitoring and surveillance of urban road traffic using low altitude drone images: a deep learning approach. *Multimedia Tools and Applications*, 1-21.

Determination of Electricity Production by Fuzzy Logic Method

*¹Beyza ÖZDEM, ²Muharrem DÜĞENÇİ, ³Mümtaz İPEK

^{1*} Corresponding Author, Department of Electrical and Electronics Engineering, Yaşar University, İzmir, Türkiye, ipek@sakarya.edu.tr 

² Department of Electrical and Electronics Engineering, Yaşar University, İzmir, Türkiye, gokhan.demirkiran@yasar.edu.tr 

³ Department of Industrial Engineering, Sakarya University, Sakarya, Türkiye, ipek@sakarya.edu.tr 

Abstract

With the increase in the need for electrical energy, production amount planning is of great importance in order not to experience restrictions in terms of use, to meet the required electricity production, and to evaluate the excess production efficiently. In this study, a generation forecasting model was created with the fuzzy logic method to determine the electricity generation. The created model is aimed to determine the electrical energy that needs to be produced daily by using the previous day's production amount, temperature, and season data. Three separate sets of data were used to test the fuzzy logic model built using information from the General Directorate of Meteorology and Energy Markets Operations Inc.. Fuzzy Logic was used to predict the data and the accuracy rates were found to be high. An improvement was observed when the accuracy rates were compared with the accuracy rates obtained in the Multiple Linear Regression Model. The accuracy rates of the model were initially examined using the Fuzzy Logic approach on weekdays and weekends, followed by a seasonal analysis and an assessment of the model's performance. As a result of the analysis, it was observed that the model worked with high accuracy in the autumn season and on weekend days.

Keywords: Energy, electrical energy, fuzzy logic

1. INTRODUCTION

Electric; it has undertaken important duties in transportation, regulation of production system, operation of industry, and meeting human needs. Mechanized production processes have been used in all sectors more effectively and efficiently thanks to electricity. With the use of electricity, it has been possible to develop new techniques, and raw materials that increase added value such as wood, metal, glass, paper, and chemicals can be converted into semi-finished or finished goods. Electricity, which has come to represent progress and economic growth, has begun to replace traditional energy sources in various contexts and has permeated every aspect of our lives. Electrical energy, which has been indispensable for social, economic, and industrial life since its inception, has had great importance in our lives, especially lighting, then mechanization, transportation, and heating.

Electricity, which has come to represent progress and economic growth, has begun to replace traditional energy sources in various contexts and has permeated every aspect of our lives. Electrical energy, which has been indispensable for social, economic, and industrial life since its inception, has had great importance in our lives, especially lighting, then mechanization, transportation, and heating [1].

When energy development in the world is examined; the technological transformation of the steam, engine and electrical energy for energy use has been realized. With the invention of the steam engine, the 18th century marked the first turning point for the use of energy. With this important step taken towards industrialization, renewable energy sources such as water, wood, sun, and wind, which are used as energy sources, were used. Renewable energy sources, which have been on the agenda recently, are also said to be the first energy sources used in history. In the process of the spread of coal, wood has become the primary energy source [2].

Production planning of this energy, which has an irreplaceable place in our lives, is also of great importance. Freelance generation companies, Energy Markets Operations Inc. (EMOI), companies whose operating rights have been transferred, and build-operate-transfer companies that produce electricity should also determine their production quantities using the right models to meet the demands. Accurate production estimations will guide important steps such as whether the demand can be met, how to proceed if the demand is not met, how much-installed power will be, and whether to invest in a power plant [3].

1.1. Literature Review

When the previous studies on electrical energy are examined, it is focused on electrical energy production and electrical energy production planning. In the research, it has been observed that generally focuses on electrical energy demand forecasting, and load forecasting and focuses on the electrical energy efficiency of certain devices or places. The Fuzzy logic method has been encountered in studies on electricity demand forecasting and load forecasting. Results showed that the accuracy rates were higher than other methods, especially as seen in the study of Najmi and Dalimi [4]. Particle swarm optimization [5], artificial neural networks [6], fuzzy logic [7], and adaptive network fuzzy inference systems (ANFIS) [8] were found to produce the greatest outcomes in prior studies. Fuzzy Logic method was preferred to solve our problem because it yielded high accuracy rates in studies on determining the production amount in the fuzzy logic method [9].

Karadağ Albayrak, Ö. examined the Estimation of Turkey's Renewable Energy Production with Artificial Neural Networks and ARIMA Model. The results were compared with the MAPE performance measure. An error rate of 13.1% was obtained for Artificial Neural Networks (MLP - 5-5-1) and 21.9% for ARIMA [32].

Olaru, L.M. et al. worked on electricity production and consumption modeling with fuzzy logic. It resulted in an average absolute error of 67.82 across all data sets, while the estimator based on the ARIMA model and MLP resulted in errors of 198.27 and 211.07, respectively [33].

In this study, a new model was created with the fuzzy logic method to determine electricity production and it was aimed to determine the electrical energy that should be produced daily by using the previous day's production amount, temperature and seasonal data. Estimating the production amount with high accuracy is of great importance in terms of production planning. Thanks to accurate forecasts, the producer companies will be able to produce as much electricity as necessary and plan the most efficient use of excess production, and the government will be able to allocate the necessary share to electricity production in investment plans.

2. MATERIALS AND METHODS

The Fuzzy Logic method has been preferred in solving our problem since fuzzy logic is not encountered in a model for determining the production and accuracy rates give high results in the fuzzy logic method.

In the Fuzzy Logic method, the process consists of 3 steps. These steps are fuzzification, fuzzy inference process, and defuzzification [10]. When using the method, the input data, which is precise information, is primarily blurred. The generated fuzzy inputs are calculated by going through the fuzzy inference process, taking into account the fuzzy logic rules. After this process, fuzzy outputs are obtained. Thanks to the clarification process, output data, which is precise information, is obtained. In a fuzzy system, the values of a fuzzy input execute all the rules in the knowledge pool that

have the fuzzy input as part of their antecedents. This process creates a new fuzzy set representing each output or solution variable. Defuzzification creates a value for the output variable from this new fuzzy set. Fuzzy Logic Stages are shown in Figure 1.

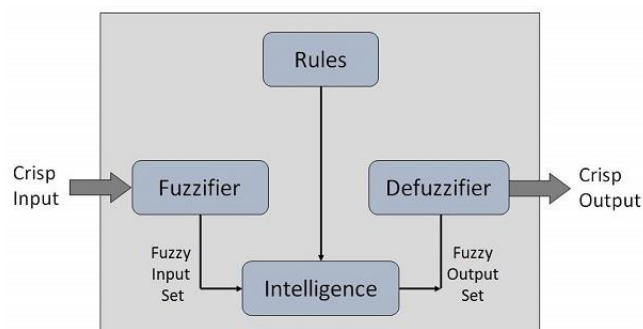


Figure 1. Fuzzy Logic Stages [11]

With the Fuzzy Logic method, a model was created in which the production amount, temperature, and seasonal inputs of the previous day and the production were determined. The generated electricity generation quantity model is shown in Figure 2. The results of the model tested with the data from EMOI and GDM were compared with the results of the Multiple Linear Regression method.

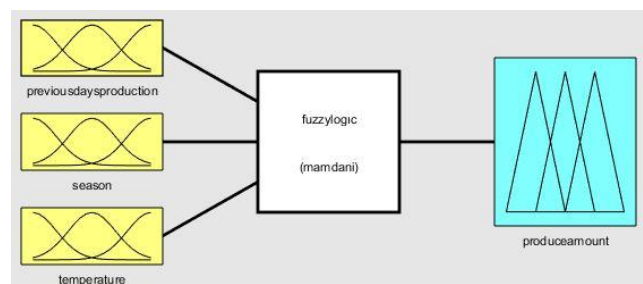


Figure 2. Fuzzy logic model of electricity production quantity

2.1. Materials

In the study, total electricity production amount obtained from natural gas, dam, lignite, river, imported coal, wind, solar, fuel oil, asphaltite coal, hard coal, biomass, naphtha, liquefied natural gas, international and waste heat from EMOI's Transparency Platform [12] used. The hourly data for 2020 and 2021 are compiled as the total daily electricity generation amount. Temperature data were compiled from GDM's Temperature Analysis [13] site by taking the average, lowest and highest values for each month in 2020 and 2021.

In February 2022, Turkey's total daily production data were obtained from EMOI's Transparency Platform. A total of 144 data were used, randomly selected from the years 2020 and 2021. The data given on the site on an hourly basis have been compiled into daily data. EMOI's total electricity production is obtained from natural gas, dam, lignite, river, imported coal, wind, solar, fuel oil, asphaltite coal, hard coal, biomass, naphtha, liquefied natural gas, international, and waste heat.

Temperature data were taken from the GDM Temperature Analysis website in February 2022. Average temperature,

lowest temperature, and highest temperature data for each month in 2020 and 2021 are compiled.

Descriptive statistics for the data used are shown in Table 1.

Table 1. Descriptive statistics of independent variables

Descriptive statistics	Production Amount of the Previous Day	Average Temperature	Lowest Temperature	Highest Temperature
Average	812497,2	14,5	9,2	20,7
Median	833233,3	13,9	8,4	20,3
Standard Deviation	104116,5	7,8	6,6	8,5

2.1.1. Methods

Membership functions have been created for the specified input and output parameters. The previous day's production quantity parameter was created with the Gaussian membership function and includes 7 clusters as very very low-very low-low-normal-high-very high-very high. The membership function for the previous day's production quantity parameter is shown in Figure 3. The data in the function is in MW and has been simplified to 1000.

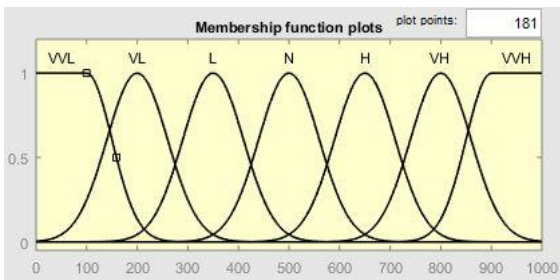


Figure 3. Membership function of production amount of the previous day

The season parameter is formed with the Gaussian membership function and includes 4 clusters winter-spring-summer-autumn. The membership function for the season parameter is shown in Figure 4. While creating the model, the month of December was considered the 1st month in the seasons.

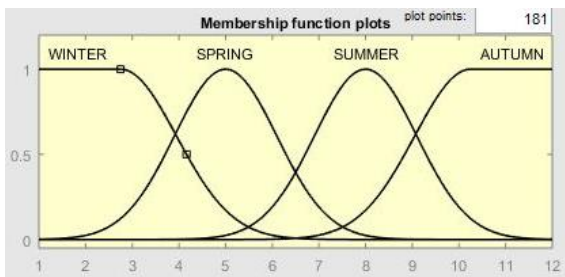


Figure 4. Membership function of the season

The temperature parameter is formed with the Gaussian membership function and includes 4 clusters as very low-low-high-very high. The membership function for the temperature parameter is shown in Figure 5.

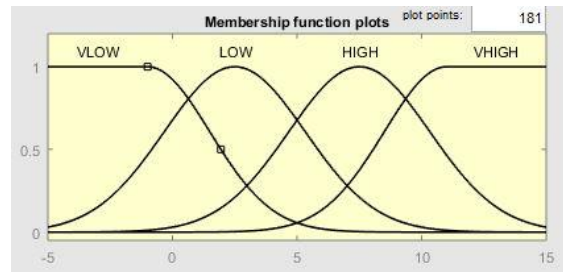


Figure 5. Membership function of temperature

The production quantity parameter is formed by triangular and trapezoidal membership functions and includes 7 clusters as very very low-very low-low-normal-high-very high-very high. The membership function for the production quantity parameter is shown in Figure 6. The data in the function is in MW and has been simplified to 1000.

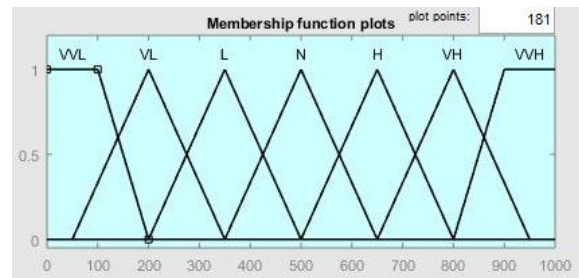


Figure 6. Membership function of production quantity

As the fuzzy inference method in the model, the Mamdani Fuzzy Inference Method, which was determined to be the most preferred in the literature studies, was used [14].

The rules in the Fuzzy Logic method were determined according to the number of clusters coming from the membership functions of the input parameters. Since there are 7 clusters in the Previous Day Production Amount parameter, 4 clusters in the seasonal parameter, and 4 clusters in the temperature parameter, 112 rules were created. The MATLAB login screen of the created rules is shown in Figure 7. After the rules were created, the model was tested in 3 different temperature categories with the previous day's production data obtained from EMOI.

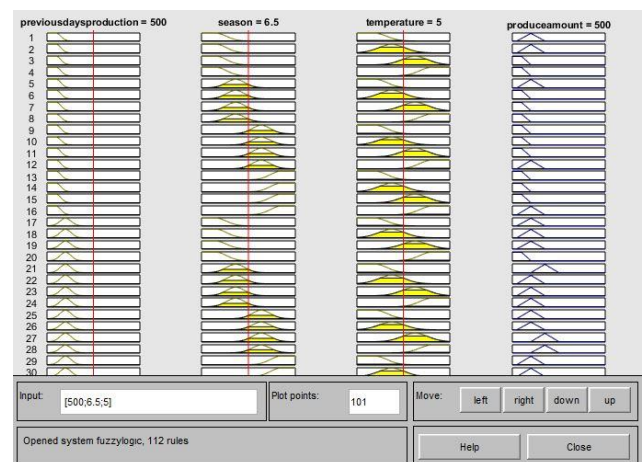


Figure 7. Rule base

Surfaces in the MATLAB Fuzzy Toolbox show the relationship between the input parameters and the output,

within the framework of the determined rules. From the surfaces showing how the two input parameters affect the output parameter, the relationship of temperature and seasonal parameters with the output is in Figure 8, the relationship between the previous day's production amount and seasonal parameters with the output Figure 9, the relationship of the previous day's production amount and temperature parameters with the output shown at Figure 10. In the figures, the production amount data and the previous day's production amount are simplified to 1000 and are in MW units. The temperature data are in Celsius, the numbers in the seasonal data represent the months and the first month represents the month of December.

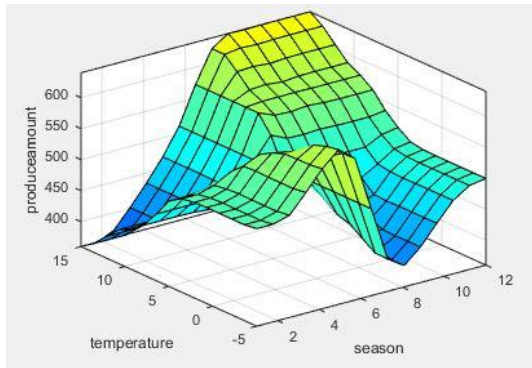


Figure 8. The effect of temperature and season on the amount of production

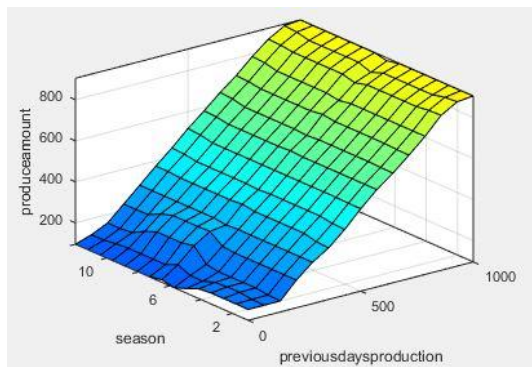


Figure 9. The effect of previous day's production and season on the amount of production

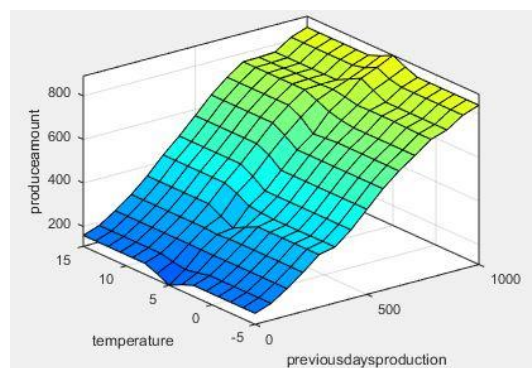


Figure 10. The effect of previous day's production and temperature on the amount of production

3. FINDING

The established model was tested with three different temperature data. The estimated data obtained and the actual

production data are shown in Figure 11, Figure 12, and Figure 13.

Results were compared with Multiple Linear Regression Results. For Average Temperature data, the accuracy rate for Multiple Linear Regression was 94.62%, while for Fuzzy Logic it was 96.32%. A 1.7% increase in accuracy rate was observed with Fuzzy Logic. A decrease of 11463.74 was observed in the RMSE value. For the Lowest Temperature data, the accuracy rate was 94.34% for Multiple Linear Regression and 95.52% for Fuzzy Logic. A 0.88% increase in an accuracy was observed with Fuzzy Logic. A decrease of 2088.57 was also observed in the RMSE value.

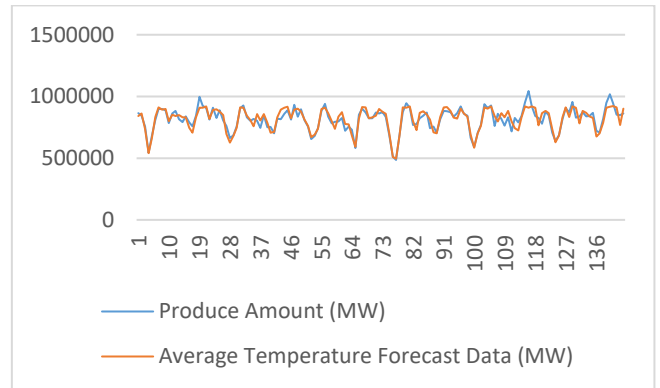


Figure 11. Forecast with average temperature data

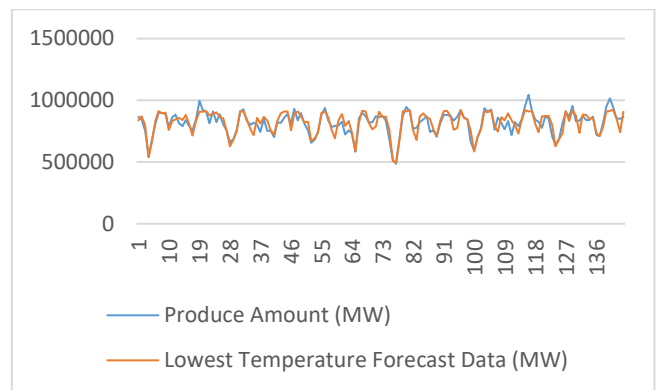


Figure 12. Forecast with lowest temperature data

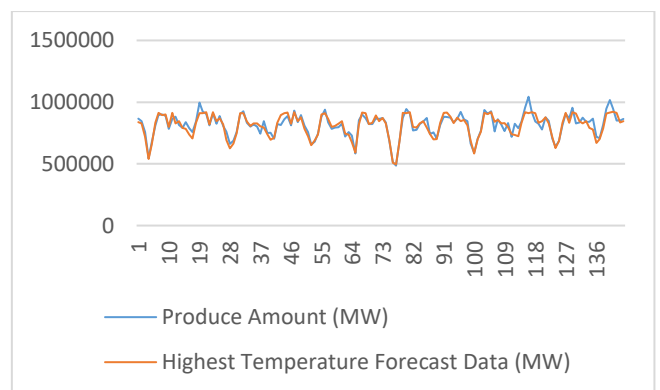


Figure 13. Forecast with highest temperature data

Peak Temperature data, the accuracy rate was 94.63% for Multiple Linear Regression and 96.78% for Fuzzy Logic. With Fuzzy Logic, an increase in accuracy rate of 2.15% was observed. A decrease of 14274.12 was also observed in the RMSE value. When evaluated in general, the Fuzzy Logic

method had high accuracy for each temperature data, and the best results were obtained in the fuzzy logic method tested with the highest temperature data.

Descriptive statistics of the actual production amount and the predictions made by the fuzzy logic method are shown in Table 2.

Table 2. Descriptive statistics of fuzzy logic predictions

Descriptive statistics	Actual Production	Average Temperature	Lowest Temperature	Highest Temperature
Average	816432,4	817520,8	818375,0	811125,0
Median	830045,1	837000,0	838000,0	830500,0
Standard Deviation	92779,9	92658,4	90918,3	92655,1

Descriptive statistics of the actual production amount and the estimates made by the multiple linear regression method are shown in Table 3.

Table 3. Descriptive statistics of multiple linear regression predictions

Descriptive statistics	Actual Production	Average Temperature	Lowest Temperature	Highest Temperature
Average	816432,4	816432,2	816432,5	816432,0
Median	830045,1	803530,0	802978,3	803591,6
Standard Deviation	92779,9	77769,8	77907,4	77787,6

The results are shown in Table 4.

Table 4. Results

Accuracy Ratios (%)	Multiple Linear Regression	Fuzzy Logic	Improvement
Average Temperatures	94,62	96,32	1,70
Lowest Temperatures	94,64	95,52	0,88
Highest Temperatures	94,63	96,78	2,15
	Average		1,58

Since the Fuzzy Logic model gives less error rate, three models tested using different temperature data were examined by distinguishing between weekdays and weekends in the data used to examine its performance. It was determined that 91 weekdays and 53 weekend days were used in randomly selected day data, and the accuracy rates were calculated for the forecast results obtained using the average temperature, lowest temperature, and highest temperature data. Obtained results are shown in Table 5.

When the weekdays and weekends are differentiated by looking at the accuracy ratios in Table 5, 96.89% in the weekend data in the model tested using the highest

temperature data; it is observed that it has the highest accuracy as 96.81% in weekday data. In the data tested using average temperature and lowest temperature data, the error rates were lower on weekdays than on weekend days, while the error rate was lower on weekends than on weekdays in the data tested using the highest temperature data.

Table 5. Analysis of fuzzy logic method by days

	Days	Accuracy Ratios (%)
Average Temperatures	Weekdays	96,76
	Weekend Days	95,70
Lowest Temperatures	Weekdays	96,21
	Weekend Days	94,22
Highest Temperatures	Weekdays	96,81
	Weekend Days	96,89

Since the Fuzzy Logic model gives less error rate, the data used to examine its performance are divided according to 4 seasons and 3 models tested using different temperature data are examined. Accuracy rates were calculated for the forecast results obtained using the average temperature, lowest temperature, and highest temperature data. Obtained results are shown in Table 6.

Table 6. Analysis of fuzzy logic method by seasons

	Temperature	Accuracy Ratios (%)
Winter	Average Temperatures	96,02
	Lowest Temperatures	95,35
	Highest Temperatures	96,07
Spring	Average Temperatures	96,25
	Lowest Temperatures	95,58
	Highest Temperatures	96,21
Summer	Average Temperatures	96,69
	Lowest Temperatures	96,46
	Highest Temperatures	96,70
Autumn	Average Temperatures	96,31
	Lowest Temperatures	94,71
	Highest Temperatures	97,53

Accuracy rates for Average Temperature data by season resulted in Summer > Autumn > Spring > Winter. Lowest Temperature data accuracy resulted in Summer > Spring > Winter > Autumn. Accuracy rates for Peak Temperature data resulted in Autumn > Summer = Winter > Spring. In general, the highest accuracy rate is seen in the model tested with the highest temperature data in the Autumn season.

4. DISCUSSION AND CONCLUSION

It is thought that this model, which was created with the Fuzzy Logic method, should be evaluated in terms of the accuracy rate of the analysis, the results it gives in the MAPE and RMSE criteria, and because it has better performance than the multiple linear regression analysis. With the fuzzy logic method, the accuracy rates were improved for all three-temperature data, but the highest increase was obtained in the model tested with the highest temperature data with a value of 2.15%. The model tested with the highest temperature data was examined in more detail on the basis of day and season. Although the accuracy rates on weekdays and weekends are very close to each other, it has been determined that 96.89% success is achieved on the weekend. In the seasonal analysis, the predictions obtained in the autumn season reached 97.53% accuracy. These findings reflect that our model has better predictive performance in the autumn season and on weekend days.

The results obtained have a higher accuracy rate than the results of previous studies on electricity production forecasting. Therefore, it should not be ignored that the model has high performance and can be used in electricity production planning.

The study, which is completed with the data received from EMOI and GDM, is a study on the daily production amount. The created model can be evaluated to determine the hourly production amount in EMOI. It is thought that accurate production estimations will guide important steps such as whether the demand can be met, how to follow if the demand is not met, how much the installed power will be, and the decision of whether to invest in a power plant.

Estimating the production amount with high accuracy is of great importance in terms of production planning. Thanks to accurate forecasts, the producer companies will be able to produce as much electricity as necessary and plan the most efficient use of excess production, and the government will be able to allocate the necessary share to electricity production in investment plans. It is envisaged that the model will support both power generation plants and government investment planning.

Financial Disclosure: The study did not receive any financial support.

Author contributions:

Conflict of Interest: This study was produced from the master's thesis titled "Determination and evaluation of electric energy generation strategy by fuzzy logic method", which was accepted in August 2022.

5. REFERENCES

- [1] Gök, K., *Electrical energy consumption, Turkey evaluation and analysis with analytical hierarchy process*, Master Thesis, Istanbul Technical University Energy Institute, 172, 2016.
- [2] Akgül, H. and Gözen, M., *An evaluation of the factors affecting the choice of resources in electricity production*, Journal of Humanities and Tourism Research, 10(4) : 919-938,2020.
- [3] Energy Markets Supervisory Authority, *Electricity Annual Sector Report*, Date of Access: 2022 Available: <https://www.epdk.gov.tr/Detay/Icerik/3-0-24-3/elektrikyillik-sektor-raporu>
- [4] Najmi, M. and Dalimi, R., *Electrical energy needs projection of Bangka Belitung province in 2019-2033 using fuzzy logic*, International Conference on Electrical Engineering and Computer Science, Bandung, 176-180, 2019.
- [5] Kazemzadeh, M. R., Amjadian, A. and Turaj, A., *A hybrid data mining driven algorithm for long term electric peak load and energy demand forecasting*, Energy, 204: 117948, 2020.
- [6] Olivera, E. M. and Olivera, F. L. C., *Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods*, Energy, 144: 776-788, 2018.
- [7] Yükseltan, E., Yücekaya, A. and Bilge A. H., *Forecasting electricity demand for Turkey: Modeling periodic variations and demand segregation*, Applied Energy, 193 : 287-296, 2017.
- [8] Çevik, H.H. and Çunkaş, M., *Short-term load forecasting using fuzzy logic and ANFIS*, Neural Computing Applications, 26 : 1355-1367, 2015.
- [9] Williams, S. and Short, M., *Electricity demand forecasting for decentralised energy management*, Energy and Environment, 1: 178-186, 2020.
- [10] Açıkgöz, N., *Improving EFQM excellence model measurement performance with fuzzy logic approach*, M.Sc., Sakarya University, Institute of Science and Technology, 94 (2019).
- [11] Tutorial Points, AI Fuzzy Logic Systems, Access: 2022, Available: https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_fuzzy_logic_system_s.htm
- [12] EMOI, *Transparency Platform-Real-Time Production Data*, Date of Access: 2022 Available:<https://seffaflik.epias.com.tr/transparency/uretlim/gerceklesen-uretim/gercek-zamanli-uretim.xhtml>
- [13] GDM, *Temperature Analysis*, Date of Access: 2022 Available:<https://www.mgm.gov.tr/veridegerlendirme/sicaklik-analizi.aspx>
- [14] Kişi, Ö., *Applicability of Mamdani and Sugeno fuzzy genetic approaches for modeling reference evapotranspiration*, Journal of Hydrology, 504 : 160-170, 2013.
- [15] Li, K., Ma, Z., Robinson, D., Lin, W. and Li, Z., *A data-driven strategy to forecast next-day electricity usage and peak electricity demand of a building portfolio using cluster analysis, Cubist regression models and Particle Swarm Optimization*, Journal of Cleaner Production, 273:123115, 2020.


- [16] Runge, J., Zmeureanu, R. and Cam, M., *Hybrid short-term forecasting of the electric demand of supply fans using machine learning*, Journal of Building Engineering, 20:101144, 2020.
- [17] Zadeh, L. A., *Fuzzy sets*, Information and Control, 8 : 338-353 (1965).
- [18] Zadeh, L. A., *Fuzzy algorithms*, Information and Control, 12: 94-102 (1968).
- [19] Çetin, E., *The application of fuzzy logic in production planning and its comparison with the current situation*, Master Thesis, Gazi University Institute of Science and Technology, 117, 2019.
- [20] Janarthanan, R., Balamurali, R., Annapoorani, A. and Vimala, V., *Prediction of rainfall using fuzzy logic*, Materials Today: Proceedings, 37: 959-963, 2021.
- [21] Incekara, C. Ö., *Turkey and EU's energy strategies and policies*, Journal of Turkish Operations Management, 3 : 298-313, 2019.
- [22] Özkan, E., Güler, E. and Aladağ, Z., *Choosing an appropriate estimation method for electrical energy consumption data*, Industrial Engineering, 31(2) :198-214, 2020.
- [23] Rezk, H., Inayat, A., Abdelkareem, M. A., Olabi, A. G. and Nassef, A. M., *Optimal operating parameter determination based on fuzzy logic modeling and marine predators algorithm approaches to improve the methane production via biomass gasification*, Energy, 239 : 122072, 2022.
- [24] Özdemir, O. and Kalinkara, Y., *Fuzzy logic: A content analysis of thesis and article studies between 2000-2020*, ACTA Infologica, 4(2) : 155-174, 2020.
- [25] Ross, T. J., *Fuzzy Logic With Engineering Applications*, 3rd ed., A John Wiley and Sons, New York, 602, 2010.
- [26] Pirbazari, A., Farmanbar, M., Chakravorty, A. and Rong, C., *Short-term load forecasting using smart meter data: a generalization analysis*, Processes, 8 : 484-505, 2020.
- [27] Busisiwe, R. L., Mbuyu, S. and Reginald, N. N., *A fuzzy logic based residential electrical energy optimization system based on time of use tariffs*, International Energy Journal, 21 : 415-426, 2021.
- [28] Çelik, A. N. and Özgür, E., *Review of Turkey's photovoltaic energy status: Legal structure, existing installed power and comparative analysis*, Renewable and Sustainable Energy Reviews, 134 : 1120344, 2020.
- [29] Gosmann, L., Geitner, C., Wieler, N., *Data-driven forward osmosis model development using multiple linear regression and artificial neural networks*, Computer and Chemical Engineering, 165 : 107933, 2022.
- [30] Şahin, H. and Esen, H., *The usage of renewable energy sources and its effects on GHG emission intensity of electricity generation in Turkey*, Renewable Energy, 192 : 859 – 869, 2022.
- [31] Uğur, D., *Mamdani Type Fuzzy Logic Based Greenhouse Climate Control System Design*, Master Thesis, Karamanoğlu Mehmetbey University Institute of Science, 63, 2021.
- [32] Karadağ Albayrak, Ö., *The forecasting of renewable energy generation for turkey by artificial neural networks and a autoregressive integrated movingaverage model -2023 generation targets by renewable energy resources*, Efficiency Magazine, 2023, Cilt:57, Sayı:1, pp.121-138
- [33] L. M. Olaru, A. Gellert, U. Fiore and F. Palmieri, *Electricity production and consumption modeling through fuzzy logic*, International Journal of Intelligent Systems published by Wiley Periodicals LLC, Int J Intell Syst. 2022, pp.8348–8364.

Statistical and Artificial Intelligence Based Forecasting Approaches for Cash Demand Problem of Automated Teller Machines

*¹Michele CEDOLIN, ²Deniz ORHAN, ³Müjde GENEVOİS

^{1*} Corresponding Author, Department of Industrial Engineering, American University of the Middle East, Kuwait, michele.cedolin@aum.edu.kw 

² Department of Industrial Engineering, Faculty of Engineering and Technology, Galatasaray Üniversitesi, Türkiye denizorhan012@gmail.com 

³ Department of Industrial Engineering, Faculty of Engineering and Technology, Galatasaray Üniversitesi, Türkiye, merol@gsu.edu.tr 

Abstract

The efficient management of cash replenishment in Automated Teller Machines (ATMs) is a critical concern for banks and financial institutions. This paper explores the application of statistical and artificial intelligence (AI) forecasting methods to address the cash demand problem in ATMs. Recognizing the significance of accurate cash predictions for ensuring uninterrupted ATM services and minimizing operational costs, we investigate various forecasting approaches. Initially, statistical methodologies including Autoregressive Integrated Moving Average (ARIMA) and Seasonal ARIMA (SARIMA) are employed to model and forecast cash demand patterns. Subsequently, machine learning techniques such as Deep Neural Networks (DNN) and Prophet algorithm are leveraged to enhance prediction accuracy. For the entire data set the average MAPE score obtained with ARIMA is 32.57% while this accuracy increased up to 29.26% with DNN and reached up to 26.77% with Prophet. By optimizing cash replenishment strategies based on accurate forecasts, financial institutions aim to simultaneously enhance customer satisfaction and reduce operational expenses. The findings of this study contribute to a comprehensive understanding of how statistical and AI-driven forecasting can revolutionize cash management in ATMs, offering insights for improving the efficiency and cost-effectiveness of ATM services in the banking sector.

Keywords: Cash Demand; Machine Learning; Forecasting; ARIMA; Prophet

1. INTRODUCTION

The effective organization of various facets is paramount to delivering top-tier service. Our research embarks on tackling a crucial issue, commencing with the identification of an optimal cash replenishment policy. The choice of cash replenishment strategy significantly impacts the overall outcome. Precision in cash predictions is pivotal; forecasts must strike a delicate balance between meeting consumer demands and not immobilizing excess funds in ATMs for extended periods. Achieving an efficient cash inventory management system necessitates harmonizing holding costs with customer service levels. Maintaining excessive inventory translates to substantial financial burdens, yet maintaining an appropriate inventory level is essential to cater to customer cash needs [1].

Hence, formulating a judicious cash management strategy is imperative. One solution involves infusing ATMs with minimal cash, but this approach carries drawbacks such as cash shortages, which lead to customer dissatisfaction—an unfavorable scenario. Conversely, loading ATMs with

excessive cash results in idle funds, which, remaining untraded on exchanges, can be construed as a loss for banks—the second undesirable scenario. Therefore, the objective is to avert over-replenishing cash. The most effective approach to optimize ATM cash management is to strike a balance between these constraints.

This study aims to establish effective forecasting techniques for ATMs. While the literature predominantly relies on conventional methods, the accuracy of predictions profoundly influences customer satisfaction with ATM services. Initially, diverse forecasting methodologies are applied to the NN5 dataset [2], encompassing statistical techniques like Exponential Smoothing, ARIMA and SARIMA [3] [4], alongside machine learning methods like neural networks [5][6]. It is worth noting that Prophet has not been previously applied in the context of ATM cash replenishment policies, differentiating our approach. To evaluate the outcomes, we employ the Mean Absolute Percentage Error (MAPE) metric. Nonetheless, the suitability of these conventional methods for meeting contemporary demands is open to debate. While these

methods yield promising results, the adoption of cutting-edge techniques such as the Prophet model can potentially deliver even more efficacious outcomes. This study illuminates how the utilization of state-of-the-art methods can yield optimal results in today's dynamic environment.

The paper comprises four additional sections. The second section offers a comprehensive review of prior research, the third section elucidates the methodologies and mathematical formulations underpinning our proposed approach, while the final section outlines application and results.

2. LITERATURE REVIEW

Forecasting the precise amount of cash required to meet daily customer demands, ensuring that a minimum cash level remains available until the next replenishment, presents a challenging issue. To address this problem, a data-driven machine learning technique has been employed for forecasting ATM replenishment quantities, offering a more accurate estimation of the optimal cash amount needed for ATM operations [7]. In recent years, machine learning techniques have become the predominant approach for resolving forecasting challenges. Another machine learning algorithm has been devised to address the prediction of daily cash withdrawals from ATMs, harnessing Artificial Neural Networks (ANN) to assert the predictability of daily cash withdrawals for specific ATMs. These cash withdrawals exhibit a seasonal pattern based on date and time parameters [8]. Machine learning methodologies, renowned for their effectiveness in forecasting problems, have witnessed significant advancement in recent times.

Notably, a significant study conducted by Van Anholt and colleagues [9] utilized authentic data from banks to address a multifaceted inventory-routing problem that encompassed pickups and deliveries, drawing inspiration from ATM replenishment procedures in the Netherlands. This research paves the way for advancements in the cash supply chain domain, focusing on enhancing cash management practices and cost reduction. In a parallel vein, Bolduc and co-authors [10] introduce a novel model aimed at minimizing the combined costs associated with transportation and inventory. This model determines which customers will be served by specific distributors and maps out the delivery routes for those handled by the private fleet.

In their study, Simutis and colleagues utilized Artificial Neural Networks (ANN) for predicting the daily cash requirements of 1225 ATMs. These predictions were generated through the integration of weekly and monthly seasonal patterns, as well as long-term trends spanning a two-year period. Furthermore, they introduced an optimization method to determine the most efficient cash replenishment strategy for each ATM. This approach factored in considerations such as cash expenses, uploading costs, and daily service expenses. They assessed the performance of their models in two scenarios, each featuring distinct interest rates and uploading costs, achieving a margin of error of less than 10% [11]. In 2008, Simutis and colleagues employed a versatile Artificial Neural Network (ANN) that incorporated a unique adaptive regularization term derived through cross-validation. Additionally, they

utilized the Support Vector Regression (SVR) algorithm to assess the accuracy of forecasting, utilizing both simulated and real data from 15 ATMs over a two-year period. Their evaluation focused on estimating the Mean Absolute Percentage Error (MAPE) for forecasting the daily cash demand for the subsequent 50 days. The results they obtained ranged from 15% to 28% for the first model and from 17% to 40% for the SVR model [12].

Brentnall and colleagues designed a forward-looking sequential prediction system for managing cash machine replenishment. They favored density forecasts over single-point forecasts and considered various factors such as seasonal patterns, first-order autocorrelation, and cash-out days. Their data was derived from 190 ATMs in the UK, spanning a two-year period. They conducted a comparison of different modeling approaches, including linear models, autoregressive models, structural time series models, and Markov-switching models. They used metrics such as the logarithmic score and continuous ranked probability score (CRPS) to evaluate these models. Their findings indicated that the selection of an appropriate model should be customized for each individual ATM [13].

Teddy and Ng (2011) introduced a novel approach called the pseudo self-evolving cerebellar model articulation controller (PSECMAC) model for forecasting ATM cash demand. They applied this model to a dataset comprising 111 daily ATM cash withdrawal series from the NN5 competition. Initially, they performed data preprocessing and then utilized the Monte Carlo Evaluation Selection (MCES) technique to select relevant features. Their evaluation was based on both one-step ahead and recurrent predictions, with the Symmetric Mean Absolute Percent Error (SMAPE) serving as the error metric. Their results yielded an average SMAPE of 27.25% for one-step ahead prediction and 27.60% for recurrent prediction, surpassing the performance of other global and local learning models [14]. Andrawis and colleagues (2011) meticulously selected nine final models from a vast array of 140 preprocessing combinations. These models were amalgamated to make predictions regarding ATM demand using the NN5 competition dataset. The selected models included the following: standard multilayer neural network, Gaussian process regression, Echo state network, Echo state network ensemble, ARMA, AR, multiple regression, Holt's exponential smoothing, and simple moving average. Initially, they carried out a time aggregation, converting the daily time series data into a weekly format. Following this, a thorough seasonality analysis was conducted, and they applied a deseasonalization method based on medians. The resulting SMAPE errors ranged from 18.94% to 23.77%, with a combined average of 18.95%. This remarkable performance secured them the championship in the NN5 competition among computational intelligence models and the second prize among both statistical benchmarks and computational intelligence models [15]. Venkatesh et al. (2014) focused their attention on identifying a common cash demand pattern among ATMs based on their day-of-the-week activity, with the objective of creating group forecasts. Their approach involved constructing individual time series models for each ATM. To achieve this, they discretized the seven continuous daily withdrawal seasonality parameters and quantified them

using the sequence alignment method (SAM). Subsequently, they employed four distinct neural network models: the general regression neural network (GRNN), multi-layer feedforward neural network (MLFF), group method of data handling (GMDH), and wavelet neural network (WNN) to estimate the cash demand for ATM centers within the NN5 competition dataset. The resulting SMAPE values for all models fell within the range of 18.44% to 21.10%, which is highly promising and among the top-performing results in the existing literature [16]. The outcomes of this study demonstrate an enhancement in the overall forecasting quality of solutions. Fallahtafi et al. (2022) categorized ATMs according to the accessibility and surrounding factors and predicted the cash demand of ATMs before and during COVID-19 pandemic. They concluded that ARIMA and SARIMA may provide high performance for short-term prediction while minimizing overfitting issue [17]. Recently, Sarveswararao et al. (2023) focused on modelling the chaos for Indian Commercial Bank ATMs. They used statistical, machine learning and deep learning techniques in their study and explored the hybrid layer techniques for the overall prediction performance [18].

3. MATERIALS & METHODS

3.1. ARIMA and SARIMA

ARIMA (autoregressive integrated moving average) is a statistical analysis model which employs time series data to optimize the analysis of data or forecast future trends. An autoregressive integrated moving average model is a type of regression analysis that determines how steady one dependent variable is in comparison to other changing variables. An ARIMA model has three constants in terms of p for autoregressive terms, d for the order of differencing, and q for the number of moving-average terms. Generally, ARIMA duration would be expressed as ARIMA (p, d, q) [19]. The general ARIMA model where $d = 1$ can be expressed as

$$Y_t = (1 + \varphi_1)Y_{t-1} + (\varphi_2 - \varphi_1)Y_{t-2} + \dots + (\varphi_p - \varphi_{p-1})Y_{t-p} - \varphi_p Y_{t-p-1} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} \quad (1)$$

Moreover, to overcome seasonality in the data SARIMA can be a useful method. SARIMA similarly not also used past data like ARIMA but also consider the seasonality of the pattern as a parameter. SARIMA has become more suitable for forecasting complex data spaces containing trends. SARIMA duration would be expressed (P, D, Q) m where m is the seasonality term. A multiplicative seasonal ARIMA model with non-seasonal orders and seasonal orders with seasonal period, is expressed by.

$$W_t = \nabla^d \nabla_s^D Y_t \quad (2)$$

Khanarsa and Sinapiromsaran (2017), focused on tackling the NN5 forecasting challenge through the application of ARIMA, SARIMA, and exponential smoothing models [20]. SARIMA models applied to certain subseries may be valuable tools in characterizing the structure of daily

withdrawals regardless of ATM location, according to the empirical findings [21].

3.2. Deep Neural Network (DNN)

Machine learning plays a significant role in literature when it comes to predicting optimal solutions. Neural networks have gained substantial prominence in the field of machine learning. Artificial Neural Networks (ANNs) and Simulated Neural Networks (SNNs) represent a subset of machine learning techniques that underpin deep learning methodologies. The figure below illustrates a multilayer feed-forward network, wherein each layer of nodes receives input from the preceding layers. A weighted linear combination is employed to amalgamate the inputs for each node. Subsequently, the outcome is modified by a nonlinear function before being presented as the output [22].

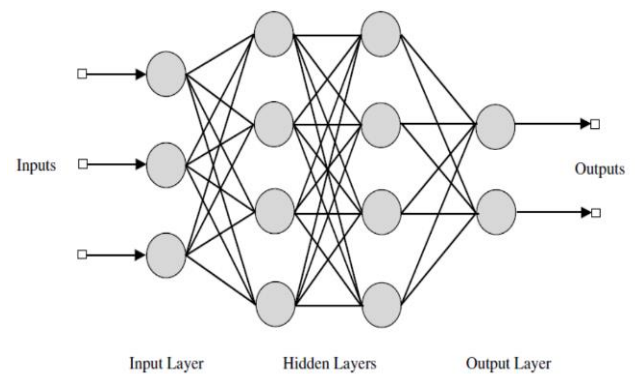


Figure 1. Multilayer Feedforward Neural Network

Figure 1. shows a DNN structure with a configuration of 3-4-4-2, where there are three input nodes and two output nodes. Neural networks offer a viable approach to predict cash demand. Venkatesh et al. (2014) employed four distinct neural networks, namely the General Regression Neural Network (GRNN), Multi-Layer Feed-Forward Neural Network (MLFF), Group Method of Data Handling (GMDH), and Wavelet Neural Network (WNN), to make projections regarding cash demand at ATM centers. Among these models, GRNN emerges as the most effective, as indicated by the SMAPE metric [23]. Atsalaki et al. (2011) employed a Multi-Layer Feed-Forward Neural Network model to predict future values within a set of 100 daily ATM cash withdrawal time series. This model provides outputs that accurately encapsulate the intricate dynamics of the time series data [24].

3.3. Prophet Method

Prophet, an open-source library developed by Facebook, is founded on decomposable models that encompass trends, seasonality, and holidays. Within the realm of machine learning, Prophet stands out as a highly valuable method. Introduced in 2017, it employs a fundamental modular regression model, often yielding satisfactory results with default settings. Moreover, analysts can selectively emphasize the components that are pertinent to their forecasting challenge and adjust as needed, as advocated by Taylor & Letham [25]. Prophet exhibits several distinctive

characteristics when compared to other machine learning techniques. Its approach to seasonality and holidays represents a key departure from conventional forecasting strategies. The model comprises three fundamental components: a trend term, a seasonal period term, and the consideration of holidays. In addition to its exceptional predictive capabilities, Prophet adeptly handles data characterized by periodic and cyclic fluctuations, as well as significant outlier values [26]. Given the multitude of features inherent to ATM data, which encompass seasonal variations and noise, the application of the Prophet method to estimate ATM withdrawals can transcend the limitations of standard forecasting models, yielding superior forecasting outcomes. While literature predominantly employs Prophet for purposes like production planning or sales forecasting, it can also prove to be a suitable solution for our cash demand estimation challenge. The mathematical formulation of Prophet is as

$$y(t) = g(t) + s(t) + h(t) + \varepsilon \quad (3)$$

Where $g(t)$ is a trend item, $s(t)$ is a seasonal change, $h(t)$ is the holiday factor, ε is an error term.

Trend parameters discussed in two ways. The first one is nonlinear growth which reaches a saturation point.

$$g(t) = \frac{c(t)}{1 + \exp(-(k + a(t)^T \delta)(t - (m + a(t)^T \gamma)))} \quad (4)$$

Where $C(t)$ is the time-varying capacity, k is the base rate, m is offset, $k + a(t)^T \delta$ is a growth rate of time varying, to connect the endpoints of segments $m + a(t)^T \gamma$ is adjusted as a offset parameter and δ is the change in the growth rate.

For linear growth, a piece-wise constant rate of growth ensures an efficient model most of the time.

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)^T \gamma) \quad (5)$$

Where k is the growth rate, δ is rate adjustments, m is offset and γ is to make the function continuous [12]. Seasonality models must be expressed as periodic functions of t . To imitate seasonality, Fourier terms are used in regression models by using sine and cosine terms.

$$s(t) = \sum_{n=1}^N \left(a_n \cos \cos \left(\frac{2\pi n t}{P} \right) + b_n \sin \sin \left(\frac{2\pi n t}{P} \right) \right) \quad (6)$$

Where P is the regular period that is expected, 365 for annual data and 7 for weekly data. $N=10$ and $N=3$ for yearly and weekly seasonality work well for most cases.

Some holidays are on certain dates in the year but some of them can vary every year. Therefore, the model needs to fit this change. Resuming that the impacts of vacations are independent. Assign each holiday a parameter κ_i , which is the corresponding change in the prediction, and an indicator function showing whether time t is during holiday i .

$$h(t) = Z(t)\kappa \quad (7)$$

4. APPLICATION & RESULTS

The data consists of two years' worth of daily cash withdrawal records from multiple automated teller machines situated in England. These records are just time series without any additional features or number of instances. These figures are derived from the NN5 competitiveness forecasts. This dataset encompasses 111 randomly selected ATMs located at different sites. The outline of the proposed approach is as follows.

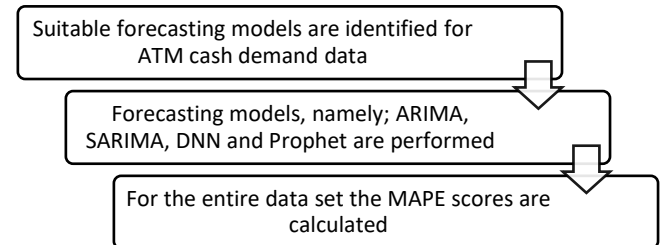


Figure 2. Proposed Approach

All preprocessing, analysis and forecasting models used in this study were implemented in a Colab Jupyter Notebook based on Python using numerous libraries such as TensorFlow and Keras. The algorithms were performed on a computer with a Windows 10 operating system and a CPU Core i7-6700 with 16 GB of memory.

Firstly, the data is preprocessed to be suitable for forecasting models. This preprocess step consists of eliminating the outliers and replacing the missing values. Outliers have the potential to impact both the standard deviation and the average of the dataset, introducing potential inaccuracies in forecasting models. Therefore, it's important to carefully examine the trend and seasonality of the data during preprocessing. When the data is stationary, missing values can be replaced with the arithmetic mean of the entire series. However, if there is a discernible trend, alternatives like the median or mode can be used for replacement. Typically, outliers are identified and removed by employing quartile values. Figure 3. depicts the box plot for the reduced NN5 data set.

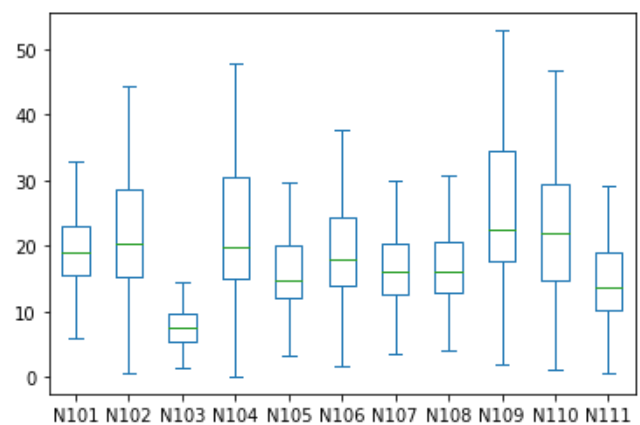


Figure 3. Box Plot for the Reduced Data Set

The subsequent phase in the forecasting process involves computing the autocorrelation (also known as serial correlation) and partial autocorrelation of the dataset.

Autocorrelation evaluates the impact of past observed values on the current value, in contrast to partial autocorrelation, which specifically measures the influence of recent observations on the current value. The term "lag" represents the number of prior observations considered in autocorrelation. The determination of the autoregressive and moving average process order is based on the autocorrelation function (ACF) and partial autocorrelation function (PACF). Figure 4. And Figure 5. display the ACF and PACF charts for the first ATM, respectively.

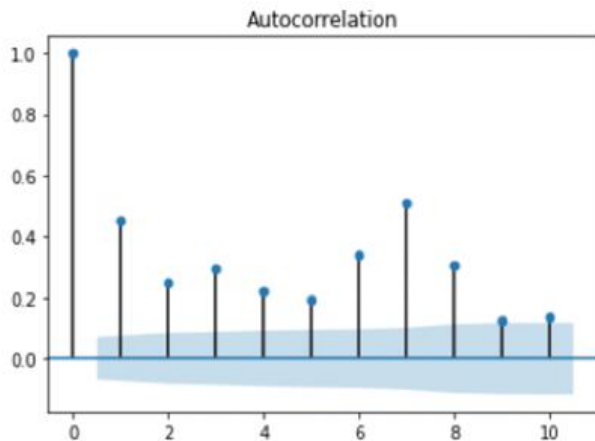


Figure 4. ACF values of the first ATM

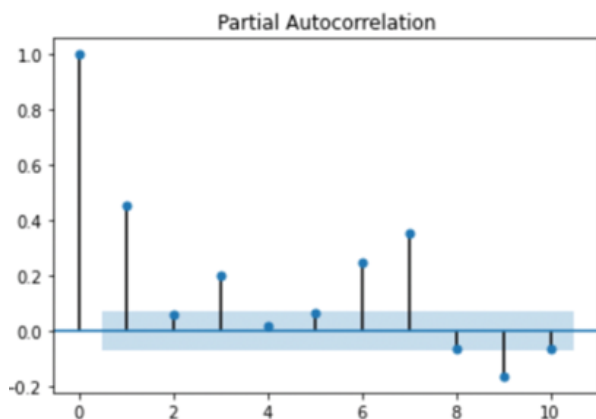


Figure 5. ACF values of the first ATM

The most pronounced correlations in both functions occur at a lag of seven, which can be directly attributed to the significant weekly seasonality of the dataset. This means that the day of the week plays a crucial role in the calculation process, indicating that the demand on Tuesdays and the demand on the following Tuesdays tend to exhibit similar patterns. These calculations were performed for each individual series, and in each case, the highest correlation value was observed at a lag of seven.

For addressing issues related to ATM cash replenishment, various methods are available, such as statistical approaches, machine learning techniques, and hybrid approaches. Following a thorough examination of existing literature, the

seasonality parameter, which holds significance in our specific dataset, led us to choose the SARIMA approach over statistical alternatives. Within the realm of machine learning, there is a wide array of neural network models, with the most used ones being the Artificial Neural Network (ANN) and Deep Neural Networks (DNN) demonstrating superior performance among other neural network techniques. Additionally, we also employed the most recent machine learning technique called Prophet in this study. Our approach, which compares ARIMA-SARIMA-DNN-Prophet, effectively handles the forecasting of cash quantities.

In our machine learning process, the model structure is established, permitting a maximum of three hidden layers. In neural networks, a hidden layer is positioned between the algorithm's input and output layers. Within this layer, the algorithm applies weights to the inputs and processes them through an activation function to produce the output. In essence, these hidden layers carry out nonlinear transformations on the input data fed into the network. It's worth noting that having an excessive number of hidden layers can significantly prolong the training process, to the extent that adequately training the neural network becomes impractical. Moreover, it can lead to overfitting, where the network memorizes the data instead of learning from it, resulting in poor performance when applied to test data. The chosen optimization method is Stochastic Gradient Descent (SGD) with a momentum, which consumes less memory than the traditional Gradient Descent (GD) algorithm. This is because SGD computes the derivative by considering only one data point at a time, and it achieves faster convergence thanks to the incorporation of momentum. Rectified Linear Unit (ReLU) is used as the activation function. Following table illustrates the hyperparameter tuning set used in the grid search for finding the best DNN settings.

Table 2. Hyperparameter Set for DNN

Window Size	range(7,28, step=7)
Batch Size	range(5,50, step=5)
Learning Rate	[1e-6, 1e-5, 1e-4]
Momentum	[0.3, 0.6, 0.9]
Epochs	range(100,500, step=50)

We opted to employ the model to predict the forthcoming 56 days by utilizing data from the initial 735 days. Training data and test data split is based on the NN5 competition requirement where researchers are asked to predict the next 56 days demand based on the given training data. We utilize the initial cash amount for a randomly selected ATM 103 as a reference point for comparing these processes in the chart below.

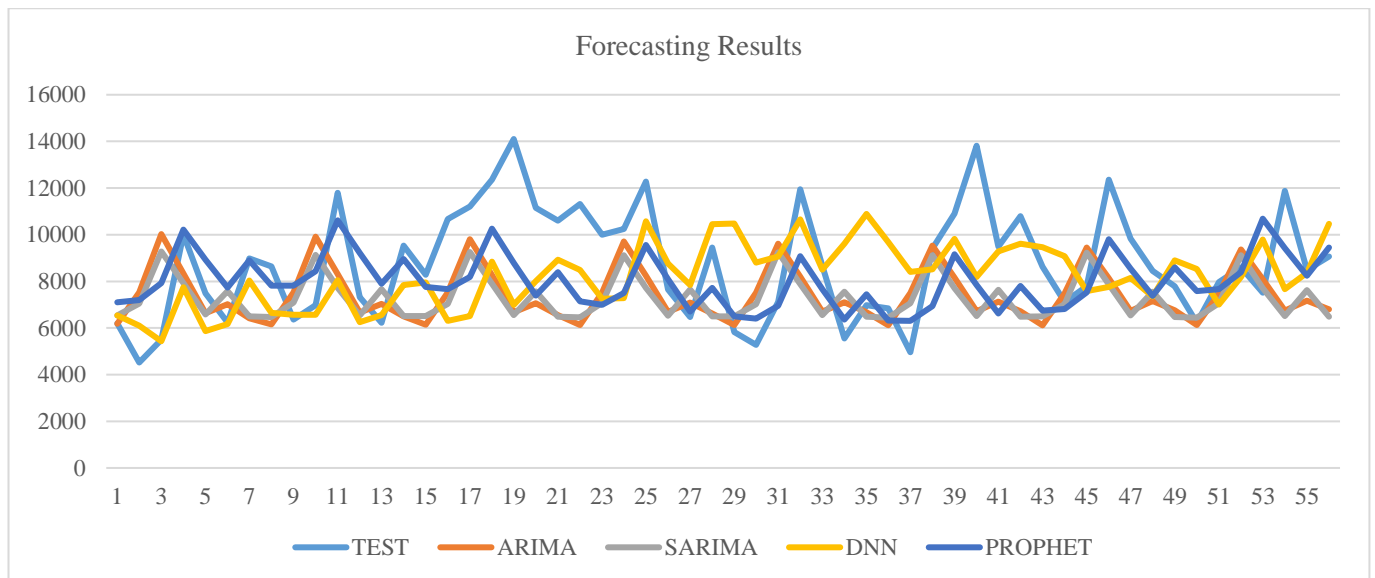


Figure 6. Forecasting outcomes for ATM 103

Where we see that Prophet model outperforms all the remaining models for ATM 103. Similarly, we performed all the forecasting models for the reduced data set, and we obtained the average MAPE scores below.

Table 3. MAPE Scores

Model	Average	St. Dev.	Median
ARIMA	32.57	22.03	26.29
SARIMA	33.69	22.05	27.78
DNN	29.26	16.70	25.49
PROPHET	26.77	14.59	23.50

As it is in Table 3., Prophet is the best forecasting model overall, followed by DNN. Figure 7 depicts the MAPE boxplots for all the series where the distributions of the errors may be observed.

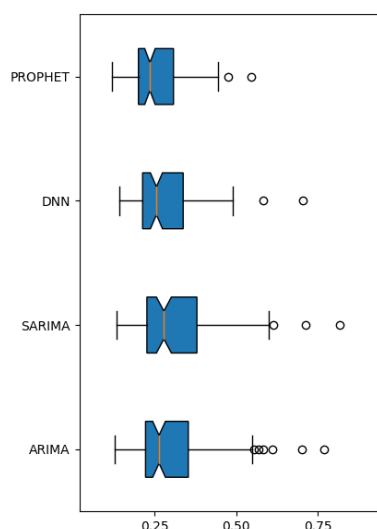


Figure 7. MAPE Boxplot

5. CONCLUSION

To sum up, in this study a decision support system has been developed to address the issue of ATM cash replenishment. Various methods have been thoroughly assessed from different angles, and their outcomes have been scrutinized using a metric system. The method with the lowest error rate was chosen.

In this study, the ARIMA, SARIMA, DNN and Prophet forecasting methods are employed. Machine learning based models increased accuracy significantly while reducing standard deviation among the series performance. We showed the significance of this problem and the importance of optimizing it through numerous benchmark scenarios from the existing literature.

Furthermore, as part of future work, the Vehicle Routing Problem (VRP) can be applied based on the forecasts generated by Prophet, ultimately leading to cost minimization. This approach aims to provide practical and effective solutions within the banking industry.

Author contributions:

Conflict of Interest: No conflict of interest was declared by the authors.

Financial Disclosure: The authors declared that this study has received no financial support.

REFERENCES


- [1] Baker, T., Jayaraman, V., Ashley, N. "A Data-Driven Inventory Control Policy for Cash Logistics Operations: An Exploratory Case Study Application at a Financial Institution," 2013, pp. 205-226.
- [2] <http://www.neural-forecasting-competition.com/NN5/> Retrieved October 19, 2023.
- [3] Catal, C., Fenerci, A., Ozdemir, B., & Gulmez, O. (2015). Improvement of demand forecasting models with special days. *Procedia Computer Science*, 59, 262-267.


- [4] Cedolin, M. and Erol Genevois, M. (2021), "An averaging approach to individual time series employing econometric models: a case study on NN5 ATM transactions data", *Kybernetes*, 51(9), 2673-2694.
- [5] Zapranis, A., & Alexandridis, A. (2009). Forecasting cash money withdrawals using wavelet analysis and wavelet neural networks. *International Journal of Financial Economics and Econometrics*.
- [6] Garcia-Pedrero, A., & Gomez-Gil, P. (2010, February). Time series forecasting using recurrent neural networks and wavelet reconstructed signals. In *2010 20th International Conference on Electronics Communications and Computers (CONIELECOMP)* (pp. 169-173). IEEE.
- [7] Asad, M., Shahzaib, M., Abbasi, Y., and Rafi, M. "A Long-Short-Term-Memory Based Model for Predicting ATM Replenishment Amount," in *2020 21st International Arab Conference on Information Technology (ACIT), Information Technology (ACIT), 2020 21st International Arab Conference On, 2020, 1-6*. <https://doi.org/10.1109/ACIT50332.2020.9300115>
- [8] Serengil, S. I., Özpınar, A., "ATM Cash Flow Prediction and Replenishment Optimization with ANN," 2019, pp. 402-408.
- [9] R. G. Anholt, L. C. Coelho, G. Laporte and I. F. A. Vis "An Inventory-Routing Problem with Pickups and Deliveries Aising in the Replenishment of Automated Teller Machines," 2016, pp. 1077-1091.
- [10] Bolduc, M.-C., Laporte, G., Renaud, J., and Boctor, F. F. "A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars," 2010, *European Journal of Operational Research*, 202(1), 122-130.
- [11] Simutis, R., Dilijonas, D., & Bastina, L. (2008). Cash demand forecasting for ATM using neural networks and support vector regression algorithms. *20th International Conference, Euro Mini Conference Continuous Optimization and Knowledge-Based Technologies*, 416-421.
- [12] Simutis, R., Dilijonas, D., Bastina, L., & Friman, J. (2007). A flexible neural network for ATM cash demand forecasting. *Cimmacs '07: WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, 163-168.
- [13] Brentnall, A. R., Crowder, M. J., & Hand, D. J. (2010a). Predicting the amount individuals withdraw at cash machines using a random effects multinomial model. *Statistical Modelling*, 10(2), 197-214.
- [14] Teddy, S. D., & Ng, S. K. (2011). Forecasting ATM cash demands using a local learning model of cerebellar associative memory network. *International Journal of Forecasting*, 27(3), 760-776.
- [15] Andrawis, R. R., Atiya, A. F., & El-Shishiny, H. (2011). Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. *International Journal of Forecasting*, 27(3), 672-688.
- [16] Venkatesh, K., Ravi, V., Prinzie, A., & Van den Poel, D. (2014). Cash demand forecasting in ATMs by clustering and neural networks. *European Journal of Operational Research*, 232(2), 383-392.
- [17] Fallahtafti, A., Aghaaminiha, M., Akbarghanadian, S., & Weckman, G. R. (2022). Forecasting ATM cash demand before and during the COVID-19 pandemic using an extensive evaluation of statistical and machine learning models. *SN Computer Science*, 3(2), 164.
- [18] Sarveswararao, V., Ravi, V., & Vivek, Y. (2023). ATM cash demand forecasting in an Indian bank with chaos and hybrid deep learning networks. *Expert Systems with Applications*, 211, 118645.
- [19] Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1994). *Time Series Analysis, Forecasting and Control*, Prentice Hall, Englewood Cliffs, N.J.
- [20] Khanarsa, P., Sinapiromsaran, K. "Multiple ARIMA subsequences aggregate time series model to forecast cash in ATM. 9th International Conference on Knowledge and Smart Technology: Crunching Information of Everything," 2017, 83-88. <https://doi.org/10.1109/KST.2017.7886096>
- [21] Gurgul, H., Suder, M. "Modeling of Withdrawals from Selected ATMs of the "Euronet" Network," 2013, pp. 65-82.
- [22] Hyndman, R.J., Athanasopoulos, G. "Forecasting: principles and practice," 2nd edition, 2018, OTexts: Melbourne, Australia. [OTexts.com/fpp2](https://www.otexts.com/fpp2). Accessed on 1.06.2022.
- [23] Venkatesh, K., Ravi, V., esd. "Cash Demand Forecasting in ATMs by Clustering and Neural Networks," 2014, pp. 383-392.
- [24] Atsalaki, I. G., Atsalakis, G. S., and Zopounidis, C.D. "Cash withdrawals forecasting by neural networks," *Journal of Computational Optimization in Economics and Finance*, 3(2), 2011, pp. 133-142.
- [25] Taylor, S. J., & Letham, B. "Prophet: Forecasting at scale," 2017, pp. 37-45.
- [26] Wang, D., Meng, Y., Chen, S., Xie, C., and Liu, Z. "A Hybrid Model for Vessel Traffic Flow Prediction Based on Wavelet and Prophet," 2021, pp. 1-16.

IoT-based Smart Home Security System with Machine Learning Models

*¹Selman HIZAL, ²Ünal ÇAVUŞOĞLU, ³Devrim AKGÜN

¹Department of Computer Engineering, Sakarya University of Applied Sciences, Sakarya, Türkiye, selmanhizal@subu.edu.tr 

²Department of Software Engineering, Sakarya University, Sakarya, Türkiye, unal@sakarya.edu.tr 

³Department of Software Engineering, Sakarya University, Sakarya, Türkiye, dakgun@sakarya.edu.tr 

Abstract

The Internet of Things (IoT) has various applications in practice, such as smart homes and buildings, traffic management, industrial management, and smart farming. On the other hand, security issues are raised by the growing use of IoT applications. Researchers develop machine learning models that focus on better classification accuracy and decreasing model response time to solve this security problem. In this study, we made a comparative evaluation of machine learning algorithms for intrusion detection systems on IoT networks using the DS2oS dataset. The dataset was first processed for feature extraction using the info-gain feature selection approach. The original dataset (12 attributes), the dataset (6 attributes) produced using the info gain approach, and the dataset (11 attributes) obtained by eliminating the timestamp attribute were then formed. These datasets were subjected to performance testing using several machine learning methods and test choices (10-crossfold, percentage split). The test performance results are presented, and an evaluation is performed, such as accuracy, precision, recall, and F1 score. According to the test results, it has been observed that 99.42% accuracy detection rates are achieved with Random Forest for IoT devices with limited processing power.

Keywords: information security; IoT; intrusion detection system; machine learning

1. INTRODUCTION

Due to the rapid development of information and communication in all sectors, numerous sensors, hardware components, and software programs exist. Today, IoT is widely used in many fields, such as industry, military, health, energy distribution, education, entertainment, agriculture, and transportation. IoT also has many specialized application areas in supply chain management, smart homes, smart cities, connected cars, and so on. With the decrease in the cost of IoT devices and the increase in their usage, they are also actively performed, especially in smart home systems. These systems make our homes smart and can be controlled with mobile applications. In addition to offering many conveniences to people, it also reveals some personal security concerns. Malicious attacks on IoT communication infrastructure have been increasing daily and bringing severe security problems in recent years. Especially since IoT devices need less computational capacity and energy consumption, security systems developed for IoT must comply with these requirements. But cybercriminals are increasingly focusing on these systems. For this reason, there is a need to develop security systems specific to these networks that will ensure the security of IoT networks.

Intrusion Detection Systems (IDS) have been developed in this area with many different methods.

Machine learning (ML) algorithms are widely used in security systems designed to secure IoT networks. Many studies in the literature use machine learning methods to achieve IoT system security. Some studies presented the recently developed methods and architectures to ensure IoT security. Hasan et al. [1] suggested an IDS using different ML algorithms in IoT sensor networks. Many methods, such as Random Forest, Artificial Neural Network, Support Vector Machine, Decision Tree, and Logistic Regression, are used to develop the system. Latif et al. [2] introduced an IDS for IoT-based industrial networks. It is possible to identify several threats to industrial networks, including denial of service (DoS), espionage data probing, scan, and malicious operation and control. A novel lightweight random neural network-based prediction model for IDS is suggested and compared to previous research. Kumar et al. [3] presented a new IDS based on a distributed ensemble design using fog computing for IoT networks. A double-layer structure is recommended in the proposed system, with K-Nearest Neighbors (KNN), eXtreme Gradient Boosting (XGBoost), and Naive Bayes used in the first layer and Random Forest techniques chosen in the second layer. Training and testing

processes were carried out in the UNSW cyber security lab in 2015 (UNSW-NB15), and the distributed smart space orchestration system (DS2oS) data sets and performance test results are presented. Reddy et al. [4] suggested an IDS to use in smart city applications. In the article, attacks were classified, and performance tests were carried out on the DS2oS data set. It has been reported that the proposed deep learning-based system provides a serious improvement for most attack types. Cheng et al. [5] proposed an IDS for IoT systems using a kind of convolutional neural network. For the training of the proposed system, two separate data sets were derived from the DS2oS data set, and optimal parameters were determined for labeled and unlabeled data. The proposed model is compared with many different methods, computation complexity analyses, and performance results are presented. It has been stated that it provides a serious improvement, especially on unlabeled data. Rashid et al. [6] developed a deep learning-based adversarial IDS for their IoT smart city applications. DS2oS data set is used, and different attack models are tested. The proposed model has been shown to achieve successful results in both binary and multi-class classification. Weinger et al. [7] worked with the publicly available Telemetry datasets of IoT (TON_IoT) and DS2oS datasets. They tested five different data augmentation methods on these datasets and showed that class imbalances have a negative impact on the detection rate. Chen et al. [8] have shown that their proposed DAGAN architecture can produce better results by preventing a marginal sample from being mispriced in industrial control systems. They have demonstrated this advantage in their experimental studies on DS2oS and Secure Water Treatment (SWaT) datasets. Mukherjee et al. [9] proposed an ML-based system for detecting attacks on the IoT device, which is now also referred to as smart. They tried classification models for two different cases on the DS2oS dataset. In the first case, Naïve Bayes had the lowest success rate, while in the second case, they achieved the highest prediction rates using Decision Tree and Random Forest. Amroui and Zouari [10] have proposed an architecture called Duenna to detect user behaviors that exhibit different behaviors, taking into account the use of devices within smart home systems by regular users. In this way, they have helped increase security against malicious individuals who threaten smart-home users and want to hijack the systems. Lysenko et al. [11] developed an ML-based IDS by analyzing the information in the network infrastructure packets that IoT devices use to communicate. They tested their flow-based models with the low computational cost for IoT devices using five different ML classification algorithms. For their study, they used traffic data from six different datasets. It was found that Random Forest (RF) performed the best, while Support Vector Machine (SVM) performed the worst. Hassan et al. [12] proposed a real-time method for detecting and mitigating Distributed Denial-of-Service (DDoS) attacks using the DS2oS and UNSW-NB15 datasets. They utilized fog computing and a machine learning approach based on KNN. Mendonça et al. [13] focused on a lightweight implementation of the IDS system using a model based on a sparse connected multi-layer perceptron structure. They gave the training and test time performance results to show the sparse model in addition to attack detection evaluations. Wahab [14] developed a deep learning model that

dynamically determines the depths of hidden layers and considers concept drift and data drift conditions in an IoT environment. Le et al. [15] proposed a model based on ensemble tree models, decision trees, and random forests. They used an online fine-tuning method for their deep learning model and drift detection methods. Also, they used the Shapley Additive Explanations (SHAP) to interpret the decision of the ensemble tree approach. Shobana et al. [16] proposed a new method for IoT smart city applications using a privacy-preserving model based on blockchain. They employed an optimization algorithm to optimize the hyperparameters of the hybrid deep neural network for IDS.

According to recently reviewed studies, many strategies were employed in IDS designs for IoT systems. Different IoT datasets were preferred for the training and testing of the developed systems. In this work, we focused on using the DS2oS data set. Common evaluation criteria were used in the examination of system performance. The results show that performance is strictly related to the data set and ML method. In addition, it is understood that the preprocessing operations on the data set also have an effect on the performance. It has been determined that high performances that do not reflect the truth are obtained in the performance results where the datasets containing repetitive recordings are used without preprocessing the datasets. In general, it can be concluded that tree-based system designs have higher performance.

The contribution of this article is as follows.

- This study evaluated the efficiency of different machine-learning algorithms for IoT networks in terms of IDS using the DS2oS dataset.
- The IDS's performance was evaluated with the reduced number of features using feature selection procedures.
- The IDS performance of different machine-learning algorithms in IoT networks in this work.
- We determined methods that provide high performance and low energy consumption by obtaining datasets with fewer features.

We organized the article as follows: First, we presented an evaluation by a literature review. Then, we explained the general structure of IDS in IoT home security, DS2oS dataset features, feature selection technique, and performance metrics. In the third part, we carried out performance tests. In the last part, we made evaluations and suggestions for future work.

2. BACKGROUND

In this section, we first describe IDS for IoT networks. Secondly, we introduce the DS2oS dataset, which is widely used in IoT security analysis, and its characteristics. Then, we presented the ML models that are used for detecting IDS in IoT. Also, we covered the methods for extracting features to be used on the dataset. Finally, we give the performance metrics to evaluate the model.

2.1. Intrusion Detection Systems for IoT

The variety of IoT devices and their richness of services make them indispensable parts of our daily lives. These devices can communicate with each other over a certain protocol, facilitating the necessary work and even making some decisions for us. With the rapid growth of IoT networks in the coming years, attacks will likely diversify. At the initial stage, cryptographic security mechanisms like authentication and encryption are insufficient due to the resource constraints of the devices, and there are several security vulnerabilities against attacks. Therefore, it is necessary to provide more advanced security by effectively detecting infiltration against attacks. Moreover, it is important to develop systems with less computational overhead for IoT devices compared to traditional IDS, as IoT devices are generally lightweight in terms of resources. Figure 1 shows IoT devices in a smart home system including a variety of ordinary appliances and devices that have sensors, connections, and the potential to communicate with other devices or the internet. These electronic devices have the purpose of making the home more intelligent, safer, and more valuable.

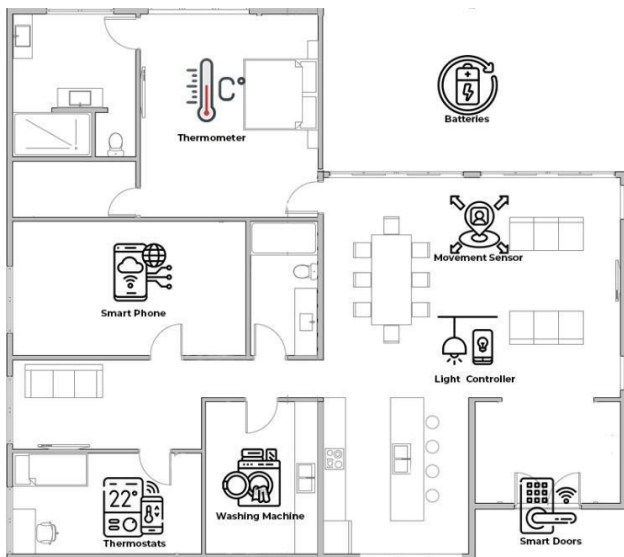


Figure 1. A Basic Sample of IoT Smart Home Plan

Today, many traditional security measures automatically detect threats with Artificial Intelligence and ML tools. Additionally, systems that can make necessary decisions to prevent attacks are available. For example, ML anomaly detection can be performed by automatically detecting upcoming threats. ML algorithms constantly work on IoT device traffic data to detect abnormal behavior. The notifications are generated when an anomaly is detected, and apps can be programmed to react automatically to specific irregularities.

2.2. Overview of DS2oS Dataset

The DS2oS dataset, was developed in 2008 by Oliver Pahl and extended in 2018 by François-Xavier Aubert with the module for anomaly detection as part of his bachelor thesis in computer science [17] [18]. DS2oS was created to ensure the privacy and security of IoT users. The DS2oS dataset is accessible to the public through Kaggle. During the dataset

development process, the system is trained to detect abnormal activity while taking into account normal user behavior. Sensors in a home and the actions of IoT devices at the application layer were utilized to create the dataset. Using a knowledge agent and virtual state layer (VSL) in the dataset architecture, the acquired data from IoT services may be shared with other IoT devices. Using a web interface or mobile application, users can give instructions to all IoT devices through a central administration system. Their actions are automatically logged in the data repository. A developer can also access the recorded data and publish new services.

/kaName/serviceName/variableName is the specific address used to access each node in the system. These nodes' types, including SmartDoors, Batteries, LightController, etc., as well as their locations, including entrance, kitchen, and bathroom, are also known. Four properties—serviceID (service1), accessednodeaddress (kaName/service1), operation (read, write), and timestamp (847690962, 1513093731) are used to define each connection.

The DS2oS dataset was produced using light controls, motion sensors, thermostats, washing machines, solar batteries, door locks, and smartphones, as seen in Figure 1. Four different IoT places (the house, two-room apartment, three-room apartment, and office) were tracked for a whole day while this dataset was being created, and traffic was recorded. Within each place, there are variations in the structure and procedures. It is discovered that there are the most DDoS attacks (5,780) and the fewest wrongSetUp (122) when only 3% of the created dataset is analyzed, as shown in Figure 2.

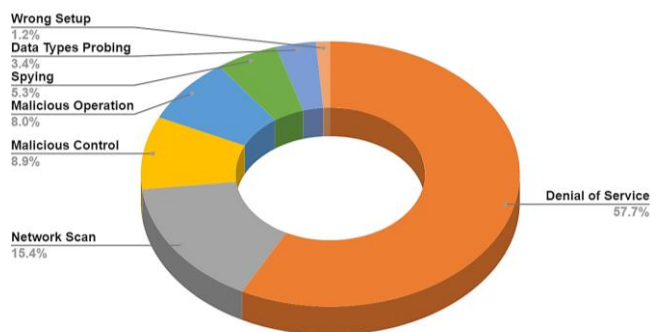


Figure 2. Anomalous Attack Types in the DS2oS Dataset

Table 1. The number of samples for source types

No	Number of Sources	Type	Number of Samples
1	22	lightControler	135,775
2	21	movementSensor	1,301
3	20	sensorService	85,196
4	6	batteryService	81,273
5	5	doorLockService	335
6	4	thermostat	5,980
7	3	washingService	47,986
8	3	smartPhone	106
Total	84		

The DS2oS dataset was generated from various devices. Table 1 shows the number of sources, types, and the number of samples.

The number of samples in the dataset is a total 357,952. The number of the normal trafficking dataset is 347,935 while 10,017 samples represent abnormal behavior. The class label and the number of the sample in the DS2oS dataset are given in Table 2.

Table 2. The number of samples for normal and anomaly attack types.

No	Label	Count
1	Normal	347,935
2	Scan	1,547
3	Malicious Operation	805
4	DoS	5,780
5	Spying	532
6	Data Probing	342
7	Wrong Setup	122
8	Malicious Control	889
	Total	357,952

Normal: Regular user activity behavior.

Denial of Service (DoS): The attacker transmits large amounts of traffic in order to disrupt services on IoT devices.

Network Scan: This attack first scans IoT devices to learn more about their networks before launching complex attacks to compromise security. IP address, port, and version scanning are often used scanning methods to gather device network information.

Malicious Control: This attack allows unauthorized access to IoT devices, allowing vital information to be accessed. These attacks are frequent cyberattacks in which the victim's system is compromised by malware, typically malicious software. Ransomware, malware, command and control, and other specialized attacks are all included in malicious software, sometimes known as viruses. For instance, hackers utilize the factory default login details of unprotected devices to infect thousands of IoT devices.

Malicious Operation: This type of attack occurs when IoT devices undertake operations that are not anticipated by them.

Spying: This is a sort of attack in which a hostile IoT device gains access to the sensitive information of others by exploiting system weaknesses.

Data Types Probing: This is a form of attack in which the attacker looks for weaknesses or vulnerabilities in an IoT device.

Wrong Setup: In this attack, a hacker might gain access to sensitive and important information about clients or the sector by taking advantage of an incorrect system setup.

2.3. Machine Learning Models

The ML models for classifying the DS2oS dataset is generally explained in this section. We used NaiveBayes, Random Tree, J48, and SVM. Here are the detailed working principles of the algorithms as follows.

Naive Bayes: As a probabilistic classifier, Naive Bayes applies Bayes' theorem for the decision rule and uses training data to determine the proper parameters to classify data. There are various algorithms to train classifiers, but the common point is that the features in the training set are assumed to be independent. Therefore, the form of the covariance matrices is diagonal. Table 3 shows the details of the features in the DS2oS dataset.

Table 3. The data type of the features in the DS2oS dataset.

No	Features	Type
1	sourceID	nominal
2	sourceAddress	nominal
3	sourceType	nominal
4	sourceLocation	nominal
5	destinationServiceAddress	nominal
6	destinationServiceType	nominal
7	destinationLocation	nominal
8	accessedNodeAddress	nominal
9	accessedNodeType	nominal
10	operation	nominal
11	value	continuous
12	timestamp	discrete
13	normality	nominal

C4.5 (J48) Algorithm: One of the decision tree algorithms is the C4.5 algorithm which is represented with J48 in Weka and is derived from the ID3 (Iterative Dichotomiser 3) algorithm. C4.5 uses an if-then set of rules converted from trained trees. It selects splitting attributes based on the information gain ratio. The algorithm's performance remains the same regardless of the amount of data to be trained.

Random Forest: Random Forest is a prominent ensemble learning algorithm employed in machine learning for enhancing predictive accuracy and reducing overfitting. This algorithm builds a collection of decision trees, each constructed from a different subset of the training data and employing random feature selection, and combines their outputs to make a final prediction. The aggregation process is mathematically represented as follows:

$$H(x) = \operatorname{argmax} \left(1/T \sum_{t=1}^T \Pi (c_t(x)) \right) \quad (1)$$

where $H(x)$ is the ensemble's final prediction, T represents the number of decision trees, $\left(\Pi (c_t(x)) \right)$ indicates the prediction of the t -th tree for input x , and the majority vote is used for classification tasks. By introducing randomness during the tree construction process, Random Forest mitigates overfitting and provides improved generalization,

making it a widely used tool in various ML applications. This algorithm's robustness and effectiveness have solidified its place as a fundamental component of ensemble methods in the field of data science and pattern recognition.

Bagging: Bagging, short for Bootstrap Aggregating, is a popular ensemble learning method widely used in machine learning and data mining. The primary objective of Bagging is to enhance the predictive performance and reduce the variance of base classifiers by generating multiple bootstrap samples from the training dataset and training a set of base classifiers on these samples. The final prediction is typically achieved through a majority vote (for classification) or averaging (for regression) of the individual base classifiers. The aggregation process is mathematically represented as

$$H(x) = \operatorname{argmax} \left(\sum_{i=1}^T w_i \cdot \Pi(c_i(x)) \right) \quad (2)$$

where $H(x)$ is the final ensemble prediction, $\Pi(c_i(x))$ is an indicator function evaluating the prediction of the i -th base classifier for input x , and w_i represents the weight assigned to each base classifier. The Bagging algorithm provides a powerful framework for improving the robustness and generalization of machine learning models, effectively reducing over fitting and enhancing classification accuracy. This method has been successfully applied in a variety of domains, making it a cornerstone of ensemble learning techniques in the field of data science and pattern recognition.

K-Star: The KStar algorithm is a well-established instance-based machine learning approach employed for feature selection in the field of data mining. It is particularly useful for classification tasks and is based on the k -nearest neighbor (k -NN) principle. The central idea behind KStar is to assess the relevance of each feature in a dataset by comparing the class distribution for the k -nearest neighbors of each instance with the class distribution for the entire dataset. The algorithm assigns a weight to each feature based on this comparison, enabling the selection of the most informative features. Mathematically, the weight (w_i) assigned to each feature (F_i) is computed as

$$w_i = 1/k \left(\sum_{j=1}^k N_{ij}/N_j \right) \quad (3)$$

where N_{ij} represents the number of instances in the k -nearest neighbors of instance i belonging to class j , and N_j is the total number of instances belonging to class j in the dataset. The features with higher weights are considered more relevant for classification. KStar offers a computationally efficient approach to feature selection and is commonly employed for improving the efficiency and accuracy of classification models in various research and practical applications.

2.4. Feature Selection

Feature selection was performed on the DS2oS dataset. As a result of this process, the number of features was reduced from 12 in the original data set to 6. Applying the feature extraction process is aimed at obtaining equivalent or higher performance values with a lower number of features. The

effect of feature selection on the performance was examined with the tests made with the data set with a reduced number of features. There are many different techniques in the literature as feature extraction methods. This study explains the Info Gain (IG) attribute selection method widely used in the literature.

Algorithm-independent relevant features are found using sorting in filtering-based feature extraction techniques. The algorithms have a lower computing load and provide results faster. The IG technique [19] minimizes dataset size and provides a small dataset with efficient and superior performance outcomes. Using this algorithm, 6 features were selected according to the IG method shown in Table 4. The IG method assigns ranks for each feature according to the importance determined by the algorithm. Feature reduction improves the algorithm's speed and ensures a straightforward tree search.

Table 4. The number of samples for normal and anomaly attack types

Ranked	No	Attribute
0.1543	2	sourceAddress
0.1521	8	accessedNodeAddress
0.1511	5	destinationServiceAddress
0.1478	1	sourceID
0.0902	11	value
0.0882	3	sourceType

Performance tests were made to evaluate the efficiency of the IoT-based traffic of the DS2oS dataset using various algorithms. Performance tests were conducted on the WEKA program using a variety of measures that are often used in the literature, such as Accuracy, Recall, Precision, and F-Measure [20], [21], [22]. The following provides an explanation of the values calculated in the complexity matrix, which is shown in Table 5:

TP (True-Positive): The amount of data in the dataset that is in the normal class and predicted in the normal class.

FN (False-Negative): The amount of data in the dataset in the normal class and predicted as an attack.

FP (False-Positive): The amount of data in the dataset that is in the attack class and normally estimated.

TN (True-Negative): The amount of data in the dataset in the attack class and estimated as an attack.

Accuracy: The percentage of correct predictions made by our model out of all the estimates is known as accuracy. This metric is calculated as the ratio of the number of samples properly identified by a data mining algorithm to the entire sample. The number of samples allocated from the data set for testing is utilized to calculate this value. The following formula is used to compute the value.

$$\text{Accuracy} = (TP + TN)/(TP + FN + FP + TN) \quad (4)$$

Recall: Recall measures the percentage of true positives that were accurately detected. The ratio of the number of items in the normal class and predicted as normal in the data set to all samples that are normal gives the sensitivity value. The value is calculated as follows.

$$Recall = TP / (TP + FN) \quad (5)$$

Precision: Precision indicates what percentage of positive predictions were correct. It is the ratio of the number of values classified as normal in the data set to the number of all samples predicted as normal. The value is calculated as follows.

$$Precision = TP / (TP + FP) \quad (6)$$

F-Measure: The F1 score, which represents the harmonic average of Precision and Recall, is a lesser-known performance metric.

This evaluation criterion generates a new value by combining precision and sensitivity. This value is calculated using the harmonic mean of the precision and sensitivity values obtained. The following formula is used to compute the value.

$$FMeasure = 2x \left(\frac{Recall \times Precision}{Recall + Precision} \right) \quad (7)$$

Table 5. Confusion Matrix for Performance Calculation.

Class/Attack Type		Predicted Class	
		Normal	Attack
True Class	Normal	TP	FN
	Attack	FP	TN

3. EXPERIMENTAL EVALUATIONS

This section explains how the other datasets were derived from the DS2oS dataset. Then we carried out performance testing processes with different test options and ML algorithms. We made various performance comparisons to evaluate the model and the selected features.

3.1. Experimental Setting

We used the Weka tool for training ML models and performance evaluations. Table 6 summarizes the hardware and software used to evaluate the models. Performance measurements were done on the Windows operating system, which runs on hardware with a CPU model, i5-11400H @ 2.70GHz processor with 16 GB memory. The GPU model is NVIDIA GeForce® GTX1650 with 4 GB memory.

Table 6. Experimental hardware and software environment.

Hardware / Software	Features
Operating System	Windows 10, 64-bit
Weka	3.8.6
CPU	i5-11400H @ 2.70GHz
RAM	16 GB
Video Graphics Card	NVIDIA GeForce® GTX1650

We derived two different datasets from the original dataset. The first dataset contains 11 attributes by removing the timestamp attribute from the original dataset. The other dataset contains six features produced using the Infogain evaluation method. Removed feature numbers are 4, 6, 7, 9, 10, 12. We used these two datasets and the original dataset for training the ML models. Duplicate checks on the first derived dataset revealed many repetitive data depending on the Timestamp attribute in the DS2oS dataset. Therefore, training ML models on the original dataset where the only differentiating feature is the timestamp for many features produces unrealistically high-performance results.

Figure 3 shows the options used for the training and testing. Our training and testing datasets contain three alternatives based on the selected features. We used the Weka tool to train and test Naive Bayes and J48 ML algorithms. There are various options, such as test split and k-fold cross-validation. In the Percentage split test option, 80% of the DS2oS dataset was used for training, and testing was carried out on the remaining 20%. In the K-fold method, the k value is selected as 10, the dataset is divided into ten different parts, and tests are performed on another part in each iteration. As a machine learning method, operations were carried out on all datasets and test options with NaiveBayes and J48 classifier algorithms, which are widely used in the literature.

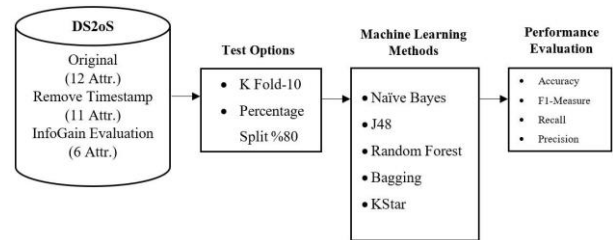


Figure 3. System Architecture and Performance Evaluation

3.2. Performance Comparisons

The results of all tests performed are presented in Table 7. When the results with the original dataset with 12 features are examined, there is no significant difference between the k-fold and percentage split test methods. Algorithms obtained with the J48 algorithm have higher performance than the NaiveBayes algorithm in both options. Since it is used as a decision attribute over the timestamp attribute in the J48 algorithm, the results have been obtained as weighted average performance values of 1.00. This shows that the system detects at a high rate by memorizing. Therefore, the timestamp property was deleted from the dataset, and a new 11-qualified dataset was obtained. In the tests made with this dataset, better results were obtained using the k-fold test option and the J48 algorithm.

The parameter values of the machine learning algorithms used in training and testing are given below. For Naive Bayes and K-Star algorithms, the batch size value was used as 100. For the J-48 algorithm, the confidence factor, batch size, number folds, and seed values were set as 0.25,100,3,1 respectively. In the random forest algorithm, the parameter values are batch size 100, max depth unlimited, num_ iterations 100 and seed 1. In the bagging method, the REPTree algorithm was preferred as the classifier and the

parameter values of bag size percent, batch size, num_iterations and seed were set as 100, 100, 10 and 1 respectively.

Table 7. The Performance Evaluation Results on DS2oS Dataset.

Dataset	Method	Acc.	Prec.	Recall	F1	AUC
percentage 80% 12-features	NB	96.93	0.99	0.97	0.98	1.00
	J48	100.00	1.00	1.00	1.00	1.00
	RF	100.00	1.00	1.00	1.00	1.00
	Bagging	99.88	1.00	1.00	1.00	1.00
	KStar	99.48	0.99	1.00	1.00	0.95
k-fold 10 12-features	NB	97.07	0.99	0.97	0.98	1.00
	J48	100.00	1.00	1.00	1.00	1.00
	RF	100.00	1.00	1.00	1.00	1.00
	Bagging	99.89	1.00	1.00	1.00	1.00
	KStar	99.54	1.00	1.00	1.00	0.96
percentage 80% 11-features	NB	96.98	0.99	0.97	0.98	0.99
	J48	99.39	0.99	0.99	0.99	1.00
	RF	99.39	0.99	0.99	0.99	1.00
	Bagging	99.29	0.99	0.99	0.99	1.00
	KStar	99.32	0.99	0.99	0.99	1.00
k-fold 10 11-features	NB	97.12	0.99	0.97	0.98	0.99
	J48	99.42	0.99	0.99	0.99	1.00
	RF	99.42	0.99	0.99	0.99	1.00
	Bagging	99.33	0.99	0.99	0.99	1.00
	KStar	99.35	0.99	0.99	0.99	1.00
percentage 80% 6-features	NB	96.16	0.99	0.96	0.97	0.99
	J48	98.71	0.99	0.99	0.98	0.99
	RF	99.28	0.99	0.99	0.99	1.00
	Bagging	99.19	0.99	0.99	0.99	1.00
	KStar	99.27	0.99	0.99	0.99	1.00
k-fold 10 6-features	NB	96.32	0.99	0.96	0.97	0.99
	J48	98.70	0.99	0.99	0.98	0.99
	RF	99.30	0.99	0.99	0.99	1.00
	Bagging	99.20	0.99	0.99	0.99	1.00
	KStar	99.29	0.99	0.99	0.99	1.00

The 6-selected features percentage split and k-fold validation results created by applying the IG method were close to each other for J48 and NaiveBayes. The accuracy value obtained with 11 attributes decreased from 99.42% to 98.69%. In Figure 4 and Figure 5, confusion matrices were obtained using the k-fold test option, and the J48 algorithm for datasets with 12 and 6 attributes is seen in the confusion matrix. As shown in Figure 4, the J48 algorithm only made an error for normal classification. Figure 5 shows that the J48 algorithm misclassified especially DoS, malicious operation, and data probe attacks for the dataset with six features.

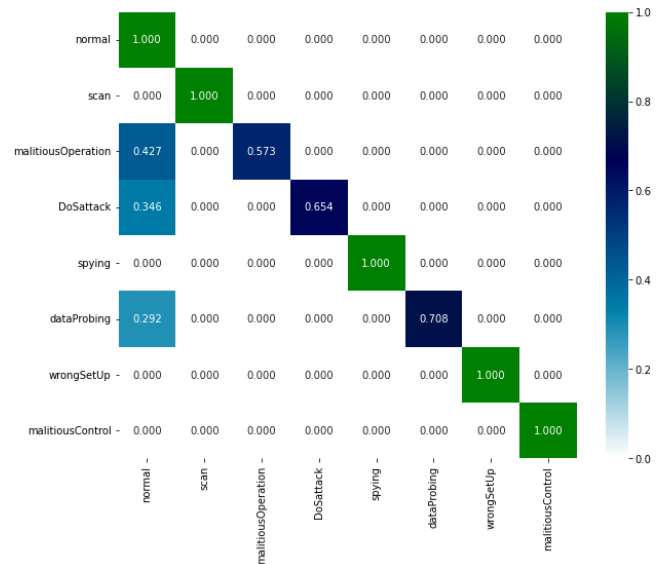


Figure 4. Confusion Matrix for Random Forest algorithm using selected 12 features

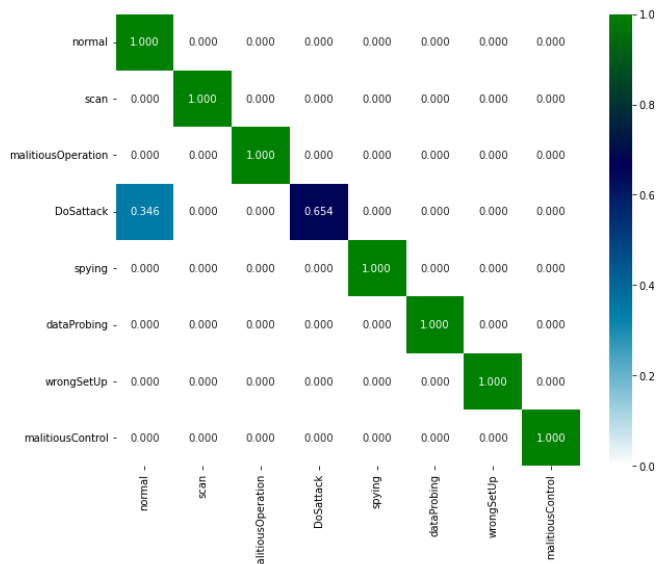


Figure 5. Confusion Matrix for Random Forest algorithm using selected 6 features

4. DISCUSSION

Comparisons of recent studies in the literature with the DS2oS dataset are presented in Table 8. When the results of the studies are analyzed, it is seen that high achievements around 99% are generally obtained. In the comparison with the proposed model, results with similar accuracy rates were obtained. In the performance tests performed without removing the timestamp attribute from the dataset, results close to 100% were obtained as shown in Table 6. Since timestamp is a unique value, it was observed that the tests performed without removing timestamp from the dataset resulted in unrealistic results. In addition, a dataset with 6 features was created with the info gain feature extraction method and a 99.30% success rate was obtained with the RF method. It is evaluated that achieving this performance over a 6-attribute data set can be used especially in IoT networks with low computational capacity.

Table 8. Comparison with recent studies on the DS2oS Dataset.

Authors	Model Tested	Best Model	Accuracy (%)	Precision (%)	Classification
Hasan et al. 2019 [1]	LR, SVM, DT, RF, ANN	RF	99.40	98.00	Multiclass
Latif et al. 2020 [2]	SVM, DT, ANN, RaNN	RaNN	99.20	99.08	Multiclass
Cheng et al. 2020 [5]	TCN, LSTM, SVM	HS-TCN	98.22	97.67	Multiclass
Reddy et al. 2020 [23]	Bayes Net, DT, NB, RF, DNN	DNN	98.28	97.00	Multiclass
Yadav et al. 2022 [24]	LR, RF, DT, ANN, KNN, AdaBoost	Adaboost	99.56	NA	Multiclass
Kushwah Et al. 2023 [25]	SVM, DT, LR, RF, ANN, AdaBoost	CatBoost	99.45	98.73	Multiclass
Paul Et al. 2023 [26]	Ensemble-DNN, RNN, CVT, DBN, TANN, F-SVM, DMM, DNN	Hybrid ML Model	99.80	99.50	Multiclass
Our Model 2023	NB, J48, RF, Bagging, K-Star	Random Forest	99.42	99.0	Multiclass

The scatter plot, a powerful tool in data exploration and analysis, enabled us to investigate the distribution and correlation of variables. Figure 6 shows a scatter plot which is a crucial aspect of our research to display the relationships and patterns in the DS2oS dataset. Distributions were obtained for normal and other attack types. When the distribution values in the dataset are examined, it is seen that the number of normal traffic samples is higher than the attack types and is distributed homogeneously. In other attack types, it was determined that the distributions were concentrated in certain regions due to the low number of samples.

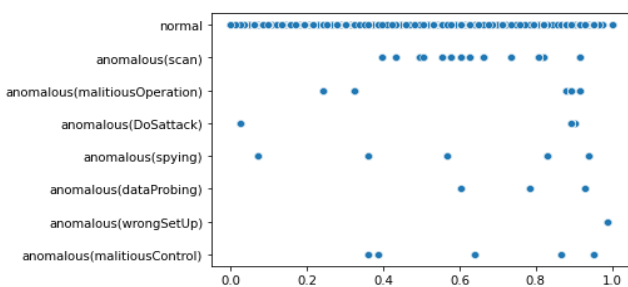


Figure 6. Distribution of attack types in the DS2oS

5. CONCLUSIONS

We performed a comparative assessment of machine learning techniques for intrusion detection systems on IoT networks using the DS2oS dataset. In the paper, DS2oS data set is introduced, the main principles of IDS are discussed, and details on the data set's attack types are provided. The feature selection approach, performance metrics utilized for evaluation, and comparison of ML algorithms are introduced. The dataset was processed using info-gain feature selection, resulting in 12 attributes, 6 attributes, and

11 attributes after eliminating the timestamp attribute. The datasets have been evaluated for performance using various machine learning algorithms and test configurations. Random Forest demonstrated its efficacy by achieving 99.42% accuracy detection rates for IoT devices with limited processing resources. The results show that using the timestamp value as a determining feature produced unhealthy results in terms of performance, so it was not used in the processed data sets. The test findings on the 6 attribute data sets acquired by the IG feature selection approaches have been proven effective, and good performance is achieved with fewer features. When we compare the obtained results, it is observed that the RF algorithm produces higher performance for all datasets. For future studies, the number of samples for specific classes, such as Wrong setup or Data Probing, is considerably low compared to the other classes. Advanced techniques such as GAN-based re-samplers can be trained to increase the number of samples.

5.1. Data Availability:

The DS2oS dataset is accessible at: <https://www.kaggle.com/datasets/francoisxa/DS2oStraffictcraces>

Author contributions: All authors contributed equally to the creation of the idea and the design.

Conflict of Interest: No conflict of interest was declared by the authors.

Financial Disclosure: This study is supported by Sakarya University of Applied Sciences, Scientific Research Projects under the grant number 105-2022.

REFERENCES

- [1] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. Hashem, "Attack and anomaly detection in iot sensors in iot sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, 2019.
- [2] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, "A novel attack detection scheme for the industrial internet of things using a lightweight random neural network," *IEEE Access*, vol. 8, pp. 89 337–89 350, 2020.
- [3] P. Kumar, G. P. Gupta, and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, pp. 9555–9572, 2021.
- [4] D. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik, and P. K. Singh, "Deep neural network based anomaly detection in internet of things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, p. e4121, 2021.
- [5] Y. Cheng, Y. Xu, H. Zhong, and Y. Liu, "Leveraging semisupervised hierarchical stacking temporal convolutional network for anomaly detection in iot communication," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 144–155, 2021.

- [6] M. M. Rashid, J. Kamruzzaman, M. M. Hassan, T. Imam, S. Wibowo, S. Gordon, and G. Fortino, "Adversarial training for deep learning-based cyberattack detection in iot-based smart city applications," *Computers & Security*, p. 102783, 2022.
- [7] B. Weinger, J. Kim, A. Sim, M. Nakashima, N. Moustafa, and K. J. Wu, "Enhancing iot anomaly detection performance for federated learning," *Digital Communications and Networks*, 2022.
- [8] L. Chen, Y. Li, X. Deng, Z. Liu, M. Lv, and H. Zhang, "Dual auto-encoder gan-based anomaly detection for industrial control system," *Applied Sciences*, vol. 12, no. 10, p. 4986, 2022.
- [9] I. Mukherjee, N. K. Sahu, and S. K. Sahana, "Simulation and modeling for anomaly detection in iot network using machine learning," *International Journal of Wireless Information Networks*, pp. 1–17, 2023.
- [10] N. Amraoui and B. Zouari, "Anomalous behavior detection based approach for authenticating smart home system users," *International Journal of Information Security*, vol. 21, no. 3, pp. 611–636, 2022.
- [11] S. Lysenko, K. Bobrovnikova, V. Kharchenko, and O. Savenko, "Iot multi-vector cyberattack detection based on machine learning algorithms: Traffic features analysis, experiments, and efficiency," *Algorithms*, vol. 15, no. 7, p. 239, 2022.
- [12] K. F. Hassan and M. E. Manaa, "Detection and mitigation of ddos attacks in internet of things using a fog computing hybrid approach," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, 2022.
- [13] R. V. Mendonça, J. C. Silva, R. L. Rosa, M. Saadi, D. Z. Rodriguez, and A. Farouk, "A lightweight intelligent intrusion detection system for industrial internet of things using deep learning algorithms," *Expert Systems*, vol. 39, no. 5, p. e12917, 2022.
- [14] O. A. Wahab, "Intrusion detection in the iot under data and concept drifts: Online deep learning approach," *IEEE Internet of Things Journal*, 2022.
- [15] T.-T.-H. Le, H. Kim, H. Kang, and H. Kim, "Classification and explanation for intrusion detection system based on ensemble trees and shap method," *Sensors*, vol. 22, no. 3, p. 1154, 2022.
- [16] M. Shobana, C. Shanmuganathan, N. P. Challa, and S. Ramya, "An optimized hybrid deep neural network architecture for intrusion detection in real-time iot networks," *Transactions on Emerging Telecommunications Technologies*, p. e4609, 2022.
- [17] M.-O. Pahl and F.-X. Aubet, "All eyes on you: Distributed multi-dimensional iot microservice anomaly detection," in *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 72–80.
- [18] F. Aubet and M. Pahl, "Ds2os traffic traces," 2018. [Online]. Available: <https://www.kaggle.com/datasets/francoisxa/ds2ostrafosttraffic>
- [19] S. Jadhav, H. He, and K. Jenkins, "Information gain directed genetic algorithm wrapper feature selection for credit rating," *Applied Soft Computing*, vol. 69, pp. 541–553, 2018.
- [20] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.
- [21] T. R. Patil, "Mrs. ss shrekar," performance analysis of j48 and j48 classification algorithm for data classification," *International Journal of Computer Science And Applications*, vol. 6, no. 2, 2013.
- [22] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Information Sciences*, vol. 340, pp. 250–261, 2016.
- [23] D. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik et al., "Deep neural network based anomaly detection in internet of things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies*, pp. 1–26, 2020.
- [24] P. K. Yadav and A. Kumar, "Analysis of Machine Learning Model for Anomaly and Attack Detection in IoT Devices," *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2022, pp. 387-392, doi: 10.1109/ICIRCA54612.2022.9985703.
- [25] R. Kushwah and R. Garg, "Anomaly Detection in IOT Site Using CatBoost," *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, Ravet IN, India, 2023, pp. 1-6, doi: 10.1109/ASIANCON58793.2023.10269881.
- [26] J. T. P. P. K. A. Paul, R. R. Chandran and P. P. Menon, "A Hybrid Machine Learning Approach to Anomaly Detection in Industrial IoT," *2023 3rd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, Kalady, Ernakulam, India, 2023, pp. 32-36, doi: 10.1109/ACCESS57397.2023.10199711.

Genetically Tuned Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor

Ali Tahir KARAŞAHİN

Karabük University, Faculty of Engineering, Department of Mechatronics Engineering, Karabük, Türkiye, tahirkarasahin@karabuk.edu.tr 

Abstract

In this paper, a linear quadratic regulator (LQR) controller operating according to the genetically tuned inner-outer loop structure is proposed for trajectory tracking of a quadrotor. Setting the parameters of a linear controller operating according to the inner-outer loop structure is a matter that requires profound expertise. Optimization algorithms are used to cope with the solution of this problem. First, the dynamic equations of motion of the quadrotor are obtained and modelled in state-space form. The LQR controller, which will operate according to the inner-outer loop structure in the MATLAB/Simulink environment, has been developed separately for 6 degrees of freedom (DOF) of the quadrotor. Since adjusting these parameters will take a long time, a genetic algorithm has been used at this point. The LQR controller with optimized coefficients and a proposed LQR controller-based study in the literature are evaluated according to their success in following the reference trajectory and their responses to specific control inputs. According to the results obtained, it was observed that the genetically adjusted LQR controller produced more successful outcomes.

Keywords: Quadrotor; LQR Controller; Inner-Outer Loop; Trajectory-Tracking; Genetic Algorithm

1. INTRODUCTION

Unmanned aerial vehicles, especially multi-propeller vehicles, are widely used in many industries, such as search and rescue, reconnaissance and surveillance, mapping, and inspection of power lines [1–5]. Multicopters used in these applications are generally produced with four propellers. The desired position or angle is achieved by applying different rotational speeds to the brushless dc motors placed on this four-propeller aircraft. However, since the dynamics of the quadcopter are nonlinear and inherently unstable, it is a problem to control.

Model-based or model-free linear or nonlinear controllers are being developed for quadcopters to follow the references given to them. Nonlinear controllers such as feedback linearization, adaptive sliding mode control [6] and backstepping [7] have been proposed to match the non-linear characteristics of quadcopters. However, to develop a nonlinear controller, quadcopters must be modelled to be suitable for all operating conditions. Since this is not possible in practice, adaptive control techniques [8]–[10] are used to solve the stated problem. Adaptive controllers incorporate approaches that can adapt to changes in the working environment. But to do this, they depend on the system model to be highly accurate. It has been shown that the attitude controller, which is developed with the model

reference adaptive control technique, successfully performs this operation on the hardware [11]. Reinforcement learning techniques are also used, which iteratively perform learning processes independently of the system model. In fact, there are studies to carry out the control process with those that remain intact against engine failures that occur in the aircraft with these techniques [12]. However, reinforcement learning processes cannot guarantee that the system will always produce stable responses. Various fuzzy logic controllers are also frequently used, regardless of the system model. Interval type-2 fuzzy logic controllers are used to develop a controller that is resistant to changes in system parameters or disturbances in the working environment [13]–[14]. Although the controllers developed with model-based or model-free techniques have some unique advantages, it is expected that they will be able to show the expected performance on the hardware. The control algorithm that will work on the hardware should be considered in some criteria, such as not bringing a high processing load, short response times, and short development processes.

For the reasons stated, proportional-integral-derivative (PID) and LQR techniques can be said to be advantageous at this point. In addition, in flight control software (PX4, etc.) commonly used for drones, the PID controller comes by default. The adjustment of the coefficients in the PID technique used in the quadcopter's position controller was

carried out by different optimization algorithm [15]–[20]. However, many of the control techniques developed for quadcopters are recommended for a part of the control system (usually as a replacement for the position controller). This practice is even more common, especially when performed experimentally on hardware. It is thought that there are areas with improvement potential due to this approach for systems with many problems, such as quadcopters.

Open-source autopilot software such as PX4 [21] or ArduPilot [22] developed for quadcopters has been developed with cascade PID controllers according to the inner-outer loop structure. The coefficients in the PID control technique are set to a certain degree by default. These coefficients should be adjusted by an expert who knows the behavior of the quadcopter according to the application in which it will operate. Adjusting the specified coefficients is both a matter requiring expertise and a serious time-consuming process.

In this paper, a genetically tuned LQR controller is proposed to increase the quadcopter's ability to follow the reference trajectory. The controller proposed for the quadcopter is designed according to the inner-outer loop structure, and the coefficients used in the LQR controller are optimized by a genetic algorithm (GA).

As a contribution to this paper, the integrative action and LQR-based controller coefficients working according to the inner-outer loop structure were optimized, and gains were obtained in the performance criteria determined according to the current study in the literature. In addition, the dynamic model of the quadcopter with the X configuration is derived simply and straightforwardly and linearized according to certain approaches, making it suitable for model-based controller development.

The article is organized as follows: In Section 2, the motion equations acting on the quadcopter are obtained according to the Newton-Euler formulas, the obtained motion equations are linearized according to the determined approaches, and the motion equations are presented in the state-space form by with the quadcopter inner-outer loop structure. In Section 3, the design of the LQR-based controller with integrative action according to the inner-outer loop structure has been carried out. In Section 4, the quadcopter is modelled in a MATLAB/Simulink environment according to the parameters of Parrot AR. Drone 2.0. The developed controller responses were tested according to the specified control inputs and following the reference trajectory. Additionally, the results of the simulation tests performed are shared. In Section 5, the test results are evaluated, and future work topics are mentioned.

2. MATERIALS AND METHODS

In this section, the equations of motion of the quadcopter will be derived according to Newton-Euler formulas. The equations of motion required for quadcopter body-fixed $\{B\}$ and inertial $\{I\}$ frames are shown in Figure 1. According to the inertial frame specified here, the position vector of the center of gravity of the quadcopter is expressed as $p =$

$[x \ y \ z]^T$. The Euler angle vector $n = [\varphi \ \theta \ \psi]^T$ in the body-fixed frame is denoted as the roll, pitch, and yaw angle, respectively. The angular velocity component is expressed as $\omega = [p \ q \ r]^T$. The quadcopter's equations of motion are expressed as follows [23]:

$$m\ddot{p} = -mg\vec{a}_3 + {}^I R_B F \quad (1)$$

$$I\dot{\omega} = -\omega \cdot I\omega + \tau \quad (2)$$

It is the rotation matrix used in the transformation from the ${}^I R_B$ body-fixed frame to the inertial frame in Equation 1.

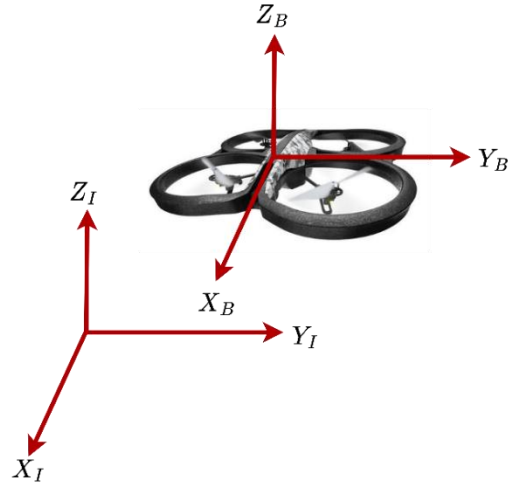


Figure 1. Reference frames are defined for the quadcopter model [24].

The unit vector in the body-fixed frame is $\{\vec{b}_1, \vec{b}_2, \vec{b}_3\}$ and in the inertial frame it is $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$. m shows the total mass of the quadcopter, g shows the gravity acceleration value, F shows the force produced by the motors, I show the 3×3 inertia matrix defined in the body-fixed frame, and τ shows the moments generated from the quadcopter. The rotation matrix used in the transformation from the body-fixed frame used in Equation 1 to the inertia frame. This rotation matrix performs the transformation between the specified axis sets [25]:

$${}^I R_B = \begin{bmatrix} c\theta c\psi & s\varphi s\theta c\psi - c\varphi s\psi & c\varphi s\theta c\psi + s\varphi s\psi \\ c\theta s\psi & s\varphi s\theta s\psi - c\varphi c\psi & c\varphi s\theta s\psi - s\varphi c\psi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \quad (3)$$

In Equation 3, s and c denote sine and cosine, respectively. It should be noted that the ${}^B R_I$ matrix [25] used in the controller design is different from the one specified in Equation 3. Angular rates of change $\dot{n} = [\dot{\varphi} \ \dot{\theta} \ \dot{\psi}]^T$ are obtained from the rotation rates occurring in the quadcopter body as follows [26]:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\varphi t\theta & c\varphi t\theta \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi s\theta & c\varphi s\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4)$$

In Equation 4: s , c , se and t denote sine, cosine, secant, and tangent, respectively. After a reference is given to the quadcopter, the relationship between the yaw moment and

thrust force obtained from the motors when it is in steady state can be defined as follows [27]:

$$T_i = c_{T_i} \Omega_i^2 \quad (5)$$

$$\tau_i = c_{\tau_i} \Omega_i^2 \quad (6)$$

The expression T_i in Equation 5 represents the thrust force produced by any motor, and c_{T_i} is used as the coefficient that realises the conversion between angular velocity and thrust force. The Ω value used in the same equation represents the angular velocity of the motors. Similarly, the expression τ_i in Equation 6 represents the yaw moment produced from any motor, c_{τ_i} is the coefficient that converts between angular velocity and yaw moment, and these coefficients can be calculated experimentally. In addition, these coefficients vary according to the propeller type, number of blades, profile, and air density. Equations 5 and 6 in the case of analyzing the thrust force and yaw moment on the z-axis [27]:

$$\tau_{\psi_i} = \frac{c_{\tau_i}}{c_{T_i}} T_i = c_i T_i \quad (7)$$

can be written as. The total thrust force (T) obtained from the motors on the quadcopter and the moments (τ_i) equations of the x, y and z axes are shown in Equation 8 [24]:

$$\begin{bmatrix} T \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ l & -l & -l & l \\ -l & -l & l & l \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (8)$$

In Equation 8, the rows represent the total thrust force obtained from the four motors and the other rows represent the moments on the x, y, and z axes, respectively. The l value indicates the distance from the centre of gravity of any motor in the quadcopter.

2.1. Linearization of the Quadcopter Model

When the sets of equations specified in Equations (1)-(8) are examined, it is seen that they contain nonlinear expressions. Since linear control techniques will be used in this study, these equation sets should be linearized. The equilibrium point for linearization was determined as the quadcopter's hover position ($p = [x \ y \ z]^T, n = [0 \ 0 \ 0]^T$). This is preferred for simplicity. With the small angles approach, the cosine values are assumed to be 1, and the sine and tangent Euler angles are accepted as themselves.

Within the scope of this study, an LQR-based controller with integrative action was developed according to the inner-outer loop structure. Therefore, six different controllers need to be designed. These controllers are the position in the x, y, and z axes designed to follow the angle references on φ, θ and ψ axes. Therefore, for each controller, its systems are expressed with a state-space approach [24]. T in Equation 9 represents the total thrust obtained from all motors. Equations of motion to consider when developing the height controller in the z axis [26]:

$$\ddot{z} \cong \frac{1}{m} (T - mg) \quad (9)$$

The state variables and inputs determined for this system are as follows:

$$x_z = [z \ \dot{z}]^T, u_z = T - mg \quad (10)$$

The state-space representation is as follows:

$$\dot{x}_z = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_z + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u_z \quad (11)$$

$$y_z = [1 \ 0] x_z \quad (12)$$

Changes in the x and y axes of the quadcopter also have effects on the roll and pitch angles in the inertia frame. We can express these changes as follows:

$$\ddot{\theta} \cong \frac{\tau_\theta}{I_y} \quad (13)$$

According to the specified motion equation, the state variable and inputs are as follows:

$$x_\theta = [\theta \ \dot{\theta}]^T, u_\theta = \tau_\theta \quad (14)$$

The state space representation consists of the following:

$$\dot{x}_\theta = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\theta + \begin{bmatrix} 0 \\ 1/I_y \end{bmatrix} u_\theta \quad (15)$$

$$y_\theta = [1 \ 0] x_\theta \quad (16)$$

The acceleration equation of motion occurring in the body-fixed frame is:

$${}^B a = \frac{F}{m} - {}^B R_1 g \vec{a}_3 - \omega \cdot {}^B v \quad (17)$$

In Equation 17, ${}^B v = [u \ v \ w]^T$ denotes the velocity value in the body-fixed frame, $\omega \cdot {}^B v$ denotes centripetal acceleration. If the x and y directions in the body-fixed frame are linearized, and the yaw angle ψ is assumed to be zero:

$${}^B a_x \cong \theta g \quad (18)$$

$${}^B a_y \cong -\varphi g \quad (19)$$

As per the definition ${}^B v$ stated in Equation 17:

$$\dot{u} \cong \theta g \quad (20)$$

$$\dot{v} \cong -\varphi g \quad (21)$$

can be edited. According to the obtained equations, the state variables to be designed for the x position can be arranged as follows:

$$x_x = [{}^B x_l \ u]^T, u_x = \theta \quad (22)$$

State-space representation to be used for reference position control on the X-axis:

$$\dot{x}_x = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_x + \begin{bmatrix} 0 \\ g \end{bmatrix} u_x \quad (23)$$

$$y_x = [1 \ 0] x_x \quad (24)$$

After linearizing the equations of motion occurring in the y-axis under certain assumptions, they can be expressed as follows:

$$\ddot{\varphi} \cong \frac{\tau_\varphi}{I_x} \quad (25)$$

The state variable and inputs, according to the specified motion equation are as follows:

$$x_\varphi = [\varphi \ \dot{\varphi}]^T, u_\varphi = \tau_\varphi \quad (26)$$

Its representation in state-space form is:

$$\dot{x}_\varphi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\varphi + \begin{bmatrix} 0 \\ 1/I_x \end{bmatrix} u_\varphi \quad (27)$$

$$y_\varphi = [1 \ 0] x_\varphi \quad (28)$$

According to Equations (19) and (21), the state variables and input for the controller that will perform the position control on the y axis can be used as follows:

$$x_y = [{}^B y_l \ v]^T, u_y = \varphi \quad (29)$$

State-space representation to be used for reference position control in the y axis is:

$$\dot{x}_y = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_y + \begin{bmatrix} 0 \\ -g \end{bmatrix} u_y \quad (30)$$

$$y_y = [1 \ 0] x_y \quad (31)$$

The angular acceleration occurring in the yaw axis can be linearized as follows:

$$\ddot{\psi} \cong \frac{\tau_\psi}{I_z} \quad (32)$$

State variable and input for the controller on this axis:

$$x_\psi = [\psi \ \dot{\psi}]^T, u_\psi = \tau_\psi \quad (33)$$

It can be expressed as. State-space representation to be used in the controller on this axis:

$$\dot{x}_\psi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\psi + \begin{bmatrix} 0 \\ 1/I_z \end{bmatrix} u_\psi \quad (34)$$

$$y_\psi = [1 \ 0] x_\psi \quad (35)$$

It can be used as. Thus, a quadcopter dynamic model is derived with six different state space representations. With the dynamic model obtained, it has become usable in understanding flight dynamics with some fixed inputs in the MATLAB/Simulink environment.

3. LQR CONTROLLER

The linear quadratic regulator is an optimal linear controller technique that performs optimization over system dynamics after all states are taken as feedback. If a design is to be made with the LQR controller, the system to be controlled should be modelled as follows:

$$\dot{x} = Ax + Bu \quad (36)$$

$$y = Cx + Du \quad (37)$$

The K matrix determined because of the optimization process is as follows:

$$u(t) = -Kx(t) \quad (38)$$

is implemented. Optimized function in LQR controller:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (39)$$

is defined as. Q refers to the weight matrix, and R refers to the control matrix specified in Equation 39. Calculation of the optimal coefficient is done as follows:

$$K = R^{-1} B^T P \quad (40)$$

The steady-state of the P matrix in Equation 40 is determined according to the Riccati equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (41)$$

An integrator is added to the LQR controller structure to eliminate perturbations and steady-state error. Thus, a more robust and steady-state error is eliminated against uncertainties. The *ref* value in Equation 42 represents the reference given to the controller. Integrator entry in LQR structure with integrative action:

$$\dot{\xi} = ref - y = ref - Cx \quad (42)$$

is implemented as. The integrator output is expressed as ξ . By adding the integrator to the LQR controller, the controller response is written as:

$$u(t) = -Kx(t) + k_1 \xi \quad (43)$$

State-space representation after adding the integrator process to the LQR controller [24]:

$$\begin{bmatrix} \dot{x} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} ref \quad (44)$$

is becoming. According to the representation in the newly formed state-space form [24]:

$$\bar{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} u \quad (45)$$

It is arranged in the form of matrix A and B. The optimal gain matrix is as follows [24]:

$$\bar{K} = [K \quad -k_1] \tag{46}$$

is expressed. In case the quadcopter is required to follow the position and angle reference given, the LQR controller is designed according to the inner-outer loop structure with the designed integrative action. The controller designed according to the inner-outer loop structure is shown in Figure 2.

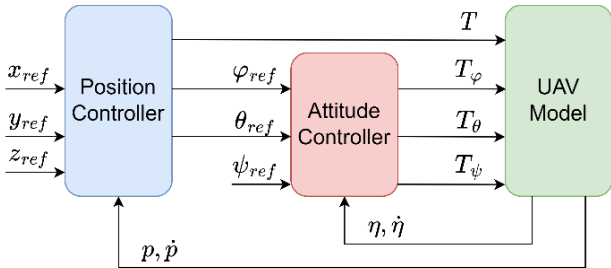


Figure 2. Inner-outer loop structure used for GA-LQR controller [24].

It is necessary to determine the Q and R matrix in the optimized function in the LQR controller. Determining this weight and control matrix by trial-and-error method takes considerable time. There are 18 coefficients in total that need to be set in the LQR controller for the x , y and z positions of the quadcopter and the roll, pitch, and yaw reference. These coefficients can be determined without losing time with optimization algorithms such as GA in the simulation environment.

3.1. Genetic Algorithm

The coefficients in the LQR controller, which is designed according to the inner-outer loop structure with integrative action, are adjusted with GA. GA is a search algorithm that tries to find the best result based on optimization. It was proposed by John Holland in 1975. It consists of natural selection, mutation, and crossover operators. GA works to find the values that will bring the given objective function to the best result. The flow diagram of the GA is shown in Figure 3.

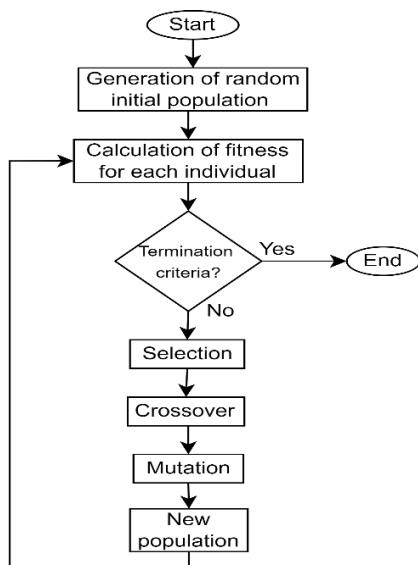


Figure 3. Genetic algorithm flow diagram.

A total of 18 coefficients that determine their success in tracking position and angle references have been optimized with GA. The gain matrix in Equation 46 is valid for position and angle controllers; the matrix K in this equation represents the state variables, and the matrix k_1 represents the integrative action value. For each controller, three coefficients must be used with two state variables and an integrative action value. Thus, 18 coefficients in total need to be determined for position and controllers. This optimization process was carried out using the dynamic model of the quadcopter in the MATLAB/Simulink environment. Root-mean-square error (RMSE) value of position errors is used for the fitness function used in GA. Function used for this operation:

$$f_{min} = rms(x) + rms(y) + rms(z) \tag{47}$$

In the GA working with the determined fitness function, the number of generations: 40, the number of populations: 20, the crossover ratio: 0.85, and the mutation rate: 0.20. GA was run on a computer equipped with AMD Ryzen 7 3700U processor and 16 GB RAM. With the specified GA parameters and hardware specifications, the run time was 907.42 seconds.

4. SIMULATION RESULTS AND DISCUSSION

The performance of the genetically tuned LQR controller was compared with a controller proposed in the literature [26]. The compared controller is designed with an inner-outer loop structure, LQR controller and integrative action. The coefficients of the controller were determined by trial-and-error method. In this study to evaluate the performance of the developed controller, its responses were observed by giving some references to the position and angle values. The parameters of the Parrot AR. 2.0 aircraft modelled in the MATLAB/Simulink environment are shown in Table 1.

Table 1. Physical parameters of Parrot AR. 2.0 quadcopter.

m (kg)	L (m)	I_x (kgm ²)	I_y (kgm ²)	I_z (kgm ²)
0.46	0.127	2.24e-4	2.90e-4	5.30e-4

In the MATLAB/Simulink environment, the response of the quadcopter to the applied input to observe the response to the control inputs is shown in Figure 4.

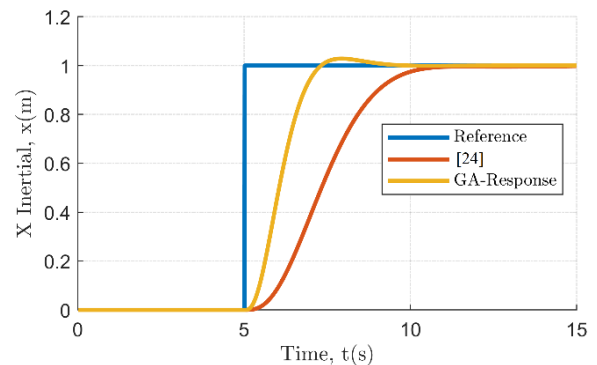


Figure 4. Reference response of the GA-LQR controller on the x -axis.

In this paper, the abbreviation GA-LQR is used for the proposed controller. The results obtained are shared in Table 4. Used in the tables: t_r represents the rise time (s) and M_p represents the overshoot (%). Accordingly, while the GA-LQR controller exhibited a faster rise performance to the 1-meter reference given on the x-axis, some overshoot was observed.

Table 4. Evaluation of the reference response in the x-axis.

Controller	t_r (s)	M_p (%)
[24]	2.9677	0
GA-LQR	1.3629	2.8549

The behavior of the quadcopter to reach the reference on the y-axis is shown in Figure 5.

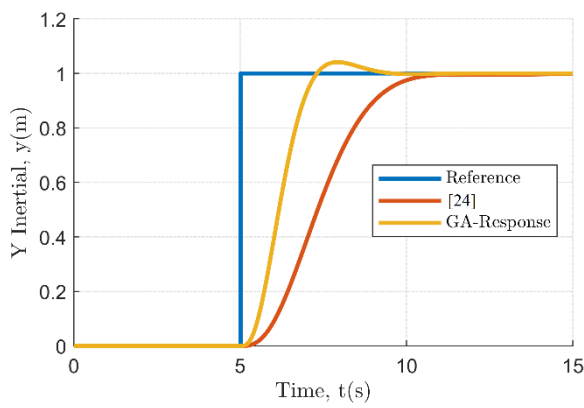


Figure 5. Reference response of the GA-LQR controller on the y-axis.

The developed GA-LQR controller exhibited similar behavior in the x and y axes. Obtained results are shown in Table 5.

Table 5. Evaluation of the reference response in the y-axis.

Controller	t_r (s)	M_p (%)
[24]	2.9677	0
GA-LQR	1.3714	4.1294

As in the x-axis, the y-axis also reached the reference in a shorter time. In addition, some overshoot occurred. When evaluated in terms of rise times, it was observed that there was a two-fold difference.

Another evaluation signal for position control was applied for the z-axis. The responses obtained after giving the quadcopter a 1-meter elevation reference are shown in Figure 6. The GA-LQR controller reached the reference in the z-axis in a shorter time than the other position references. Similarly, it can be said to be advantageous in terms of settling time.

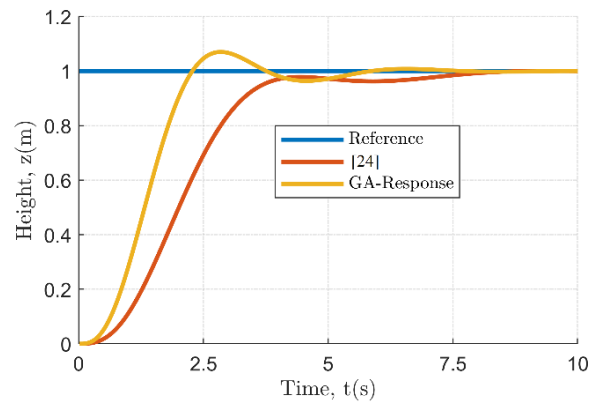


Figure 6. Reference response of the GA-LQR controller on the z-axis.

The results obtained on the z-axis are presented in Table 6.

Table 6. Evaluation of the reference response in the z-axis.

Controller	t_r (s)	M_p (%)
[24]	2.3437	0.0628
GA-LQR	1.3532	6.9580

The GA-LQR controller showed faster responses in terms of rise time in the x and y axes. However, it was found that a certain amount of overshoot occurred. After comparing the responses to position references, roll, pitch, and yaw angle reference responses were also analyzed. A roll angle reference of 0.2 radians was applied to the quadcopter, and the controller response is shown in Figure 7.

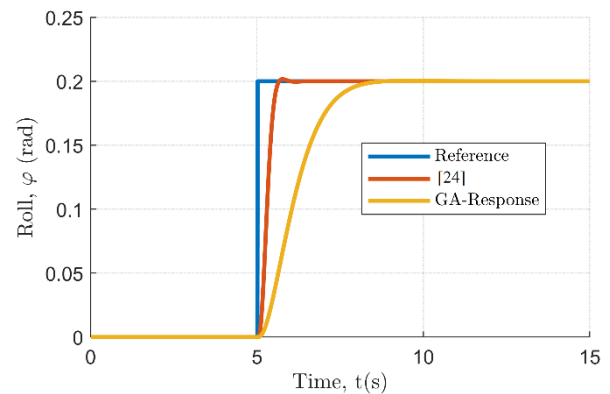


Figure 7. Reference response of GA-LQR controller for roll angle.

While the GA-LQR controller reached the roll reference angle later, no overshoot occurred this time. The results of the responses to the roll angle reference are presented in Table 7.

Table 7. Evaluation of the reference response in the roll axis.

Controller	t_r (s)	M_p (%)
[24]	0.3529	0.8185
GA-LQR	1.7875	0.1639

Similarly, the controller response to a pitch angle reference of 0.2 radians is shown in Figure 8.

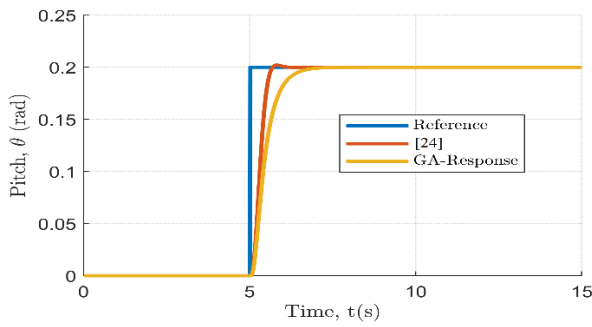


Figure 8. Reference response of GA-LQR controller for pitch angle.

It is observed that the response produced against the roll angle reference is produced in the same way for the pitch angle reference. The reactions occurring on the pitch axis are shared in Table 8.

Table 8. Evaluation of the reference response in the pitch axis.

Controller	t_r (s)	M_p (%)
[24]	0.3706	1.0163
GA-LQR	0.7955	0

When the responses at the pitch angle reference are analyzed according to Table 8, it is observed that no overshoot occurred in the GA-LQR controller. The responses of the controllers to the yaw angle reference are shown in Figure 9.

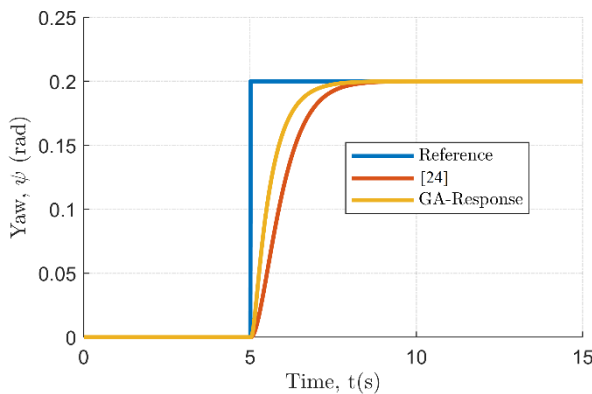


Figure 9. Reference response of the GA-LQR controller for yaw angle.

When the data obtained at the yaw angle were analyzed, it was determined that an overshoot did not occur in both controllers. The data obtained are shown in Table 9.

Table 9. Evaluation of the reference response in the yaw axis.

Controller	t_r (s)	M_p (%)
[24]	1.6421	0
GA-LQR	1.2079	0

4.1. Stability Analysis

The developed controller is expected to achieve success in certain performance criteria. However, stability analyses of the controller should also be performed. The stability of a system expressed in state-space representation can be analyzed by looking at the eigenvalues of the A matrix. Quadcopter controllers are arranged in a close-loop fashion with the coefficients determined by GA-LQR. The eigenvalues of the obtained through MATLAB system are shown in Table 10.

Table 10. Closed-loop eigenvalue analysis with optimized coefficients.

Axis Controller	Close-loop eigenvalues
x	$\{-1.3106 \pm 1.1906i, -4.8039\}$
y	$\{-1.1924 \pm 1.3667i, -2.4882\}$
z	$\{-1.3393 \pm 1.2512i, -2.0662\}$
roll	$\{-1.3840 \pm 0.7178i, -55.7223\}$
pitch	$\{-2.9947, -8.6620, -29.2386\}$
yaw	$\{-1.8593, -11.2946, -100.9108\}$

As seen in Table 10, all eigenvalues have negative values. It has been determined that the control system is stable because the real parts of the closed loop poles are negative.

After observing the success of the developed GA-LQR controller in tracking only one reference, it was also tested to track multiple references simultaneously. The GA-LQR controller is designed to track x, y and z position references, roll, pitch, and yaw angle references as stated in section 3. In this context, the ability to follow the trajectory given as a reference for x, y and z position controllers is shown in Figure 10. The trajectory tracking capability of the GA-LQR controller at x, y and z positions is demonstrated by the tests performed.

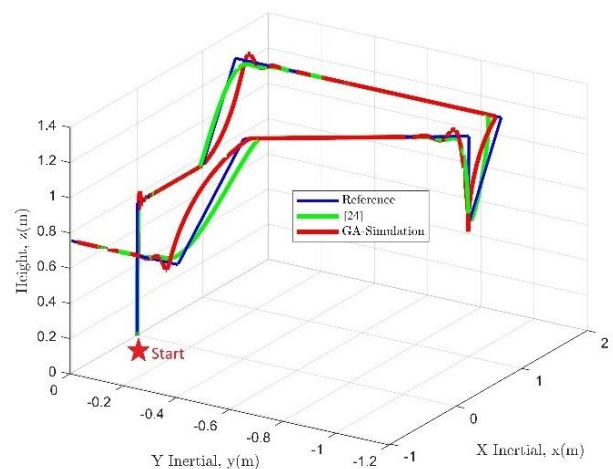


Figure 10. Control response of GA-LQR controller for trajectory tracking

The responses in the x-axis following the reference trajectory shown in Figure 11 are presented in Figure 12.

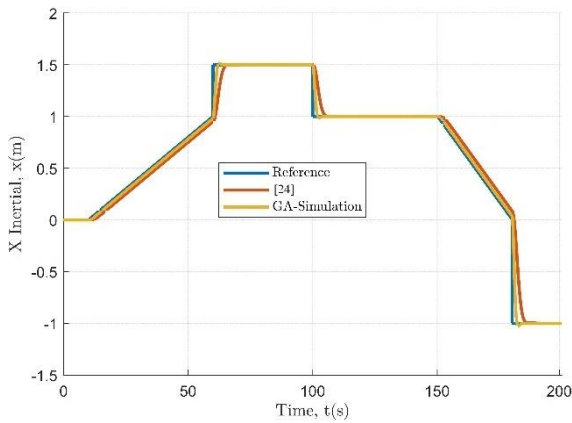


Figure 11. Control response of the GA-LQR controller in the x-axis for trajectory tracking

According to the data shown in Figure 11, it is observed that the GA-LQR controller provides superiority in terms of rise time in position references. However, some overshoots were also observed. The controller responses on the y-axis during the tracking of the reference trajectory are shown in Figure 12.

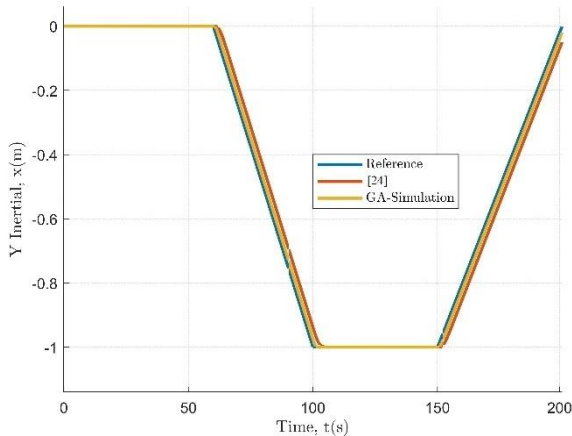


Figure 12. Control response of the GA-LQR controller in the y-axis for trajectory tracking.

It is again shown that the GA-LQR controller is successful in terms of rise time in the y-axis as well as in the x-axis. The responses in the z-axis during the tracking of the reference trajectory are shown in Figure 13.

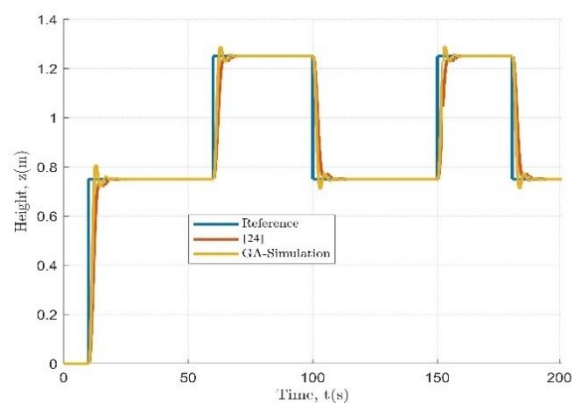


Figure 13. Control response of the GA-LQR controller in the z-axis for trajectory tracking.

When the responses occurring in the z-axis are analyzed, it is determined that the GA-LQR controller is more advantageous in terms of response time while overshoot occurs. The results of the evaluation according to the root-mean-square error (RMSE) criterion in the trajectory tracking scenario shown in Figure 10 in the Simulink environment are shown in Table 11.

According to the results shared in Table 11, the LQR controller with integrative action and genetic tuning follows the reference better than the inner-outer loop structure proposed in this study according to the root-mean-square error value.

Table 11. Results were obtained according to the root-mean-square error criterion

Controller	x (m)	y (m)	z (m)	Total
[24]	0.0865	0.0714	0.0556	0.2135
GA-LQR	0.0756	0.0143	0.0926	0.1825

As a result of the experiments carried out in MATLAB/Simulink environment for both tracking the reference in one axis and tracking multiple references, the reference tracking capability of the GA-LQR controller has been increased. It has been shown that not only more successful but also significant advantages, such as automatic detection of the controller coefficients, have been achieved.

According to the experimental results obtained from optimizing the LQR controller with GA, it was more successful. At this point, if the rise time, settling time or overshoot is essential for the area of use, the performance requirements will be met by adjusting the objective function of the GA according to this situation. Since such a situation was unimportant in the experiments, organizing the objective function according to the RMSE criterion.

In addition, the files of the quadrotor modelled in MATLAB/Simulink environment are shared at <https://github.com/atahirkarasahin/GA-LQR.git>.

5. CONCLUSION

In this paper, a genetically tuned LQR controller with integrative action operating according to the inner-outer loop structure is proposed. First, an integrator is added to the LQR controller and the system model in state-space form is adapted accordingly. With the addition of the integrator, perturbations and steady-state errors occurring in the LQR controller are eliminated. The developed GA-LQR controller is compared with a study in the literature in MATLAB/Simulink environment. With the proposed GA-LQR controller, an improvement in trajectory tracking is achieved with respect to the RMSE value. The genetically tuned LQR controller has achieved successful results both in terms of controller design time and according to the specified performance criteria. As a result of the tests performed, if the overshoot amount is an important criterion for the designer, different configurations can be realized by adjusting the

parameters in the GA to increase its success in this performance criterion.

Author contributions: Ali Tahir KARAŞAHİN: Conceptualization, Methodology, Software, Data curation, Writing- Reviewing and Editing.

Conflict of Interest: There is no conflict of interest.

Financial Disclosure: The study was funded by Sakarya University of Applied Sciences BAP with project number 078-2022.

REFERENCES

- [1] C.-C. Chang, J.-L. Wang, C.-Y. Chang, M.-C. Liang, and M.-R. Lin, "Development of a multicopter-carried whole air sampling apparatus and its applications in environmental studies," *Chemosphere*, vol. 144, pp. 484–492, 2016.
- [2] J. A. Paredes, J. González, C. Saito, and A. Flores, "Multispectral imaging system with UAV integration capabilities for crop analysis," in *2017 First IEEE International Symposium of Geoscience and Remote Sensing (GRSS-CHILE)*, 2017, pp. 1–4.
- [3] S. Anweiler and D. Piwowarski, "Multicopter platform prototype for environmental monitoring," *J Clean Prod*, vol. 155, pp. 204–211, 2017.
- [4] B. E. Schäfer, D. Picchi, T. Engelhardt, and D. Abel, "Multicopter unmanned aerial vehicle for automated inspection of wind turbines," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 244–249.
- [5] M. Stokkeland, K. Klausen, and T. A. Johansen, "Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 998–1007.
- [6] D. Lee, H. Jin Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *Int J Control Autom Syst*, vol. 7, pp. 419–428, 2009.
- [7] J. Farrell, M. Sharma, and M. Polycarpou, "Backstepping-based flight control with adaptive function approximation," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 6, pp. 1089–1102, 2005.
- [8] Z. Zuo and C. Wang, "Adaptive trajectory tracking control of output constrained multi-rotors systems," *IET Control Theory & Applications*, vol. 8, no. 13, pp. 1163–1174, 2014.
- [9] J. Spencer, J. Lee, J. A. Paredes, A. Goel, and D. Bernstein, "An adaptive pid autotuner for multicopters with experimental results," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7846–7853.
- [10] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor UAVs: A design trade study with flight evaluations," *IEEE Transactions on control systems technology*, vol. 21, no. 4, pp. 1400–1406, 2012.
- [11] E. A. Niit and W. J. Smit, "Integration of model reference adaptive control (MRAC) with PX4 firmware for quadcopters," in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 2017, pp. 1–6.
- [12] A. R. Dooraki and D.-J. Lee, "Reinforcement learning based flight controller capable of controlling a quadcopter with four, three and two working motors," in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, 2020, pp. 161–166.
- [13] C. Guzay and T. Kumbasar, "Aggressive maneuvering of a quadcopter via differential flatness-based fuzzy controllers: From tuning to experiments," *Appl Soft Comput*, vol. 126, p. 109223, 2022.
- [14] G. Unal, "Integrated design of fault-tolerant control for flight control systems using observer and fuzzy logic," *Aircraft Engineering and Aerospace Technology*, vol. 93, no. 4, pp. 723–732, 2021.
- [15] E. Yazid, M. Garrat, F. S.-2018 I. Conference, and undefined 2018, "Optimal PD tracking control of a quadcopter drone using adaptive PSO algorithm," in *ieeexplore.ieee.org*, 2018, pp. 146–151.
- [16] M. S. Can and H. Ercan, "Real-time tuning of PID controller based on optimization algorithms for a quadrotor," *Aircraft Engineering and Aerospace Technology*, vol. 94, no. 3, pp. 418–430, 2021.
- [17] Ş. YILDIRIM, N. ÇABUK, and V. BAKIRCIOĞLU, "Optimal Pid Controller Design for Trajectory Tracking of a Dodecarotor Uav Based On Grey Wolf Optimizer," *Konya Journal of Engineering Sciences*, vol. 11, no. 1, pp. 10–20, 2023.
- [18] I. Siti, M. Mjahed, H. Ayad, and A. El Kari, "New trajectory tracking approach for a quadcopter using genetic algorithm and reference model methods," *Applied Sciences*, vol. 9, no. 9, p. 1780, 2019.
- [19] M. J. Mahmoodabadi and N. R. Babak, "Robust fuzzy linear quadratic regulator control optimized by multi-objective high exploration particle swarm optimization for a 4 degree-of-freedom quadrotor," *Aerosp Sci Technol*, vol. 97, p. 105598, 2020.
- [20] M. N. Shauqee, P. Rajendran, and N. M. Suhadis, "Proportional double derivative linear quadratic regulator controller using improvised grey wolf optimization technique to control quadcopter," *Applied Sciences*, vol. 11, no. 6, p. 2699, 2021.
- [21] L. Meier, ... D. H.-2015 I. international, and undefined 2015, "PX4: A node-based multithreaded open-source robotics framework for deeply embedded platforms," *ieeexplore.ieee.org*.
- [22] "ArduPilot Documentation — ArduPilot documentation." Accessed: Jun. 13, 2023. [Online]. Available: <https://ardupilot.org/ardupilot/>

- [23] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot Autom Mag*, vol. 19, no. 3, pp. 20–32, 2012.
- [24] L. Martins, "Linear and nonlinear control of uavs: design and experimental validation," Master's Thesis, Instituto Superior Técnico, Lisbon, Portugal, 2019.
- [25] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, "Robotics," 2009, doi: 10.1007/978-1-84628-642-1.
- [26] L. Martins, C. Carneira, and P. Oliveira, "Linear quadratic regulator for trajectory tracking of a quadrotor," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 176–181, 2019.
- [27] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*. Cambridge university press, 2006.