



ISSN 2822-2385

AP JESS

Journal of Engineering  
and **Smart Systems**

VOLUME: 10

ISSUE: 1

YEAR: 2022

<https://dergipark.org.tr/en/pub/apjess/issue/68140>

## Volume 10 / 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, Turkey

### **Editors**

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

Dr. John YOO, Bradley University, USA

### **Editorial Board**

Dr. Abdullah Hulusi KÖKÇAM, Sakarya University, Turkey

Dr. Aydın MÜHÜRÇÜ, Kırklareli University, Turkey

Dr. Cengiz KAHRAMAN, Istanbul Technical University, Turkey

Dr. Elif Elçin GÜNAY, Sakarya University, Turkey

Dr. Gürcan YILDIRIM, Abant İzzet Baysal University, Turkey

Dr. Hacı Mehmet ALAKAŞ, Kirikkale University, Turkey

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

Dr. Mazin MOHAMMED, University Of Anbar, Iraq

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

Dr. Rakesh PHANDEN, Amity University Uttar Pradesh, India

Dr. Uğur Erkin KOCAMAZ, Bursa Uludağ University, Turkey

Dr. Tuğba TUNACAN, Abant İzzet Baysal University, Turkey

Dr. Valentina E. BALAS, Polytechnic University of Timisoara, Romania

### **Language Editor**

Dr. Hakan ASLAN, Sakarya University, Turkey

### **Editorial Assistants**

Selim İLHAN, Sakarya University, Turkey

İbrahim MUCUK, Sakarya University, Turkey

### **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/en/pub/apjess/issue/68140>

## 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](http://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
Prediction of the Ball Location on the 2D Plane in Football Using Optical Tracking Data	Anar Amirli, Hande Alemdar	1-8
Handwritten Digit Recognition With Machine Learning Algorithms	Kübra Gülgün Demirkaya, Ünal Çavuşoğlu	9-18
Determination of Optimum Pinch Point Temperature Difference Depending on Heat Source Temperature and Organic Fluid with Genetic Algorithm	Sadık Ata, Ali Kahraman, Remzi Şahin	19-29
Compositional correlation analysis of gene expression time series	Fatih Dikbaş	30-41
Determination of Chopped Fruits Freshness with High Accuracy by Using Electronic Nose	Bilge Han Tozlu	42-47
Realization of the Autonomous Driving System on the Experimental Vehicle	Namig Aliyev, Mehmet Turan Guzel, Oguzhan Sezer	48-56
A Novel Algorithmic Similarity Measure for Collaborative Filtering: A Recommendation System Based on Rating Distances	Şule Öztürk Birim, Ayça Tümtürk	57-69
Use of Reflection Coefficients and Decision Tree Algorithm for Rapid Classification of Hazardous Chemical Liquids	Ebru Efeoglu, Gurkan Tuna	70-77


# Prediction of the Ball Location on the 2D Plane in Football Using Optical Tracking Data

<sup>1</sup>Anar Amirli, <sup>\*2</sup>Hande Alemdar

<sup>1</sup> Department of Computer Science, Saarland University, Saarland, Germany

[anaramirli@gmail.com](mailto:anaramirli@gmail.com), 

<sup>\*2</sup> Department of Computer Engineering, Middle East Technical University, Ankara

Turkey, [alemdar@metu.edu.tr](mailto:alemdar@metu.edu.tr), 

## Abstract

Tracking the ball location is essential for automated game analysis in complex ball-centered team sports such as football. However, it has always been a challenge for image processing-based techniques because the players and other factors often occlude the view of the ball. This study proposes an automated machine learning-based method for predicting the ball location from players' behavior on the pitch. The model has been built by processing spatial information of players acquired from optical tracking data. Optical tracking data include samples from 300 matches of the 2017-2018 season of the Turkish Football Federation's Super League. We use neural networks to predict the ball location in 2D axes. The average coefficient of determination of the ball tracking model on the test set both for the x-axis and the y-axis is accordingly 79% and 92%, where the mean absolute error is 7.56 meters for the x-axis and 5.01 meters for the y-axis.

**Keywords:** Deep Neural Networks, Sports Analytics, Ball Tracking, Data Mining

## 1. INTRODUCTION

The rapid advancements in vision-based tracking and statistical tools have transformed many fields. These developments also break their way in sports analytics through many applications which offer new ways of in-detail analysis and observation to assess different aspects of both games and athletes' performance [1], [2]. As a result, sports analytics now has great importance for managers, athletes, sports experts, and even broadcasters since it enriches our knowledge about sports and leads to a more advanced and rich watching experience.

With its wide popularity and high revenue share, football benefits from all these developments the most. An extensive amount of research has already been done in football, from statistical properties of the game to game flow motifs [3]–[8]. Some studies focus on recognizing football events from the spatiotemporal soccer data. Khaustov and Mozgovoy [9] propose a rule-based system for identifying successful and unsuccessful passes and shots. Özdemir and Alemdar [10] develop a random forest classifier to identify corner kicks, free kicks, goals, and penalties. As in most team sports, understanding the strategies in football is a challenging task. It requires all kinds of relevant information, such as the

individual behavior of the players, their collective behavior as a team, and accurate ball location. In addition to sports analytics, new applications such as creating real-time game highlights for the audience experience are another emerging market that relies on accurate ball tracking. However, unlike some sports such as tennis, computer vision-based methods for frame-to-frame ball detection still remain beyond the state-of-the-art solutions for complex ball-centric sports since the ball is occluded most of the time. In football, detecting the ball's location is an even more challenging task mainly because of the nature of the game. Although there are several initiatives to equip the ball with a tracking chip, no such solution has been accepted by the governing organizations yet. One of the main challenges for ball tracking is that the size of the ball is relatively small compared to the vast field that needs to be monitored. Moreover, the players' interaction with the ball occurs in an unexpected way, and most importantly, the view is often occluded as the ball is lost behind the players.

To track the ball location in centimeter-level accuracy, a large number of very expensive cameras are needed. For example, the goal-line technology used to determine whether the ball has passed the goal line requires 14 cameras to detect

\* Corresponding Author

the goals, and that cannot track the ball in the field all the time.

Our study aims to provide a tool that can be used alongside current techniques to simplify and ensure more accurate ball-tracking. Our proposed approach uses players' formation during the game to estimate ball position. This formation-based approach focuses on analyzing within and between segment groups rather than the individual player activities. The main hypothesis we pursue is to deduce the key behaviors of ball movement in the dynamic game flow from spatial attributes such as players' speed and their positional distribution. This method has been motivated by our observations and experiments in deep learning-based approaches and our intuitive reasoning. Our model is designed to predict the ball's location on the 2D plane. Therefore, when the ball is flying, we aim to provide its projection on the 2D plane since this will give more valuable insights to the football professionals.

The rest of this paper is organized as follows. In the next section, we cite several related studies in the literature. In Section 3, we present our method to predict ball location from optical tracking data. In Section 4, we provide our experimental results on our real-world soccer optical tracking data set. Finally, we conclude with Section 5.

## 2. RELATED WORK

Given the importance of the location tracking of the ball in sports, there are several related studies in the literature. Kamble et al. provides a literature survey on the topic and identifies the need for multiple cameras as a challenge in ball tracking in football [11]. According to current regulations, it is impossible to equip the ball with wireless sensor devices; therefore, all of the existing studies that consider official football match data use computer vision-based approaches. There are two main methodological tracks: i) using broadcast videos and ii) having a fixed camera setup in the stadium. More recently, the use of drones [12] has also been suggested, yet it is not as common. Several studies focus on the detection of the ball only, whereas others also propose a trajectory for the ball.

Cardenas and Zuniga propose a two-stage algorithm [13]. In the first stage, they first extract a set of candidate objects from the segmented image. Then they filter objects that do not look like a ball using several features. In the second stage, each ball candidate's features obtained in the previous stage are combined with the dynamics model to form a trajectory. Then all the possible trajectories are ranked. Lhoest [14] proposes a similar two-stage approach starting with a ball detector followed by tracking. A deep convolutional neural network for image segmentation is used for detection, and a Kalman filter-based approach is used for tracking. In [15], an extended Kalman filter is used after the ball detection stage. Naidoo and Tapamo [16] propose another similar two-stage approach that contains soccer ball detection based on coarse analysis and filtering. Ren et al. use an 8-camera system to track the ball's location, and they provide results on a relatively small dataset that consists of a couple of minutes long video footage [17]. When they use a

buffer size of 50 frames, the detection rate is 68.5% only. A deep-learning-based system is also proposed in [18] and [19] for CCTV footage videos. Leo et al. [20] present a multi-step algorithm to detect the ball in image sequences acquired from fixed cameras. Candidate ball regions are selected by probabilistic analysis of locally affine invariant regions around distinctive points.

Durus works on the broadcast videos to track the ball to make tactical analyses [21]. He proposes to detect the ball first and then employs a particle filtering-based approach to track the ball and recover the ball's trajectory. This method requires the ball to be present and visible in the scene in all frames. Komorowski et al. use a deep neural network-based detector for the ball and players detection in high-resolution broadcast recordings [22]. The model produces a ball confidence map together with the position of the detected ball. To improve the discriminability of the ball, the feature pyramid network design pattern is used. In that way, lower-level features with a higher spatial resolution are combined with higher-level features with a bigger receptive field. In [23], a ball detection algorithm is presented. Ball candidates are first extracted using features based on the shape, color, and size. For selecting the best candidate, they use object area, centroid, bounding box, and minor and major axes features with a rule-based algorithm to eliminate the non-ball objects. Niu et al. [24] present an approach for discovering the ball states rather than its actual trajectory to automatically find the attacking patterns by the teams using broadcast videos.

In this study, we propose a machine learning-based approach to relieve the need for the increased number of cameras just for the ball tracking and use the players' and referee's behavior instead to determine the actual location of the ball in football. In our approach, even though the system cannot recognize the ball object, we are able to predict its location since we use the players' and referee's behavior. We train and evaluate our results on a dataset that contains data from a complete season. To the best of our knowledge, this study is unique in its attempt to locate the ball by using the players' and the main referee locations.

## 3. MATERIALS AND METHODS

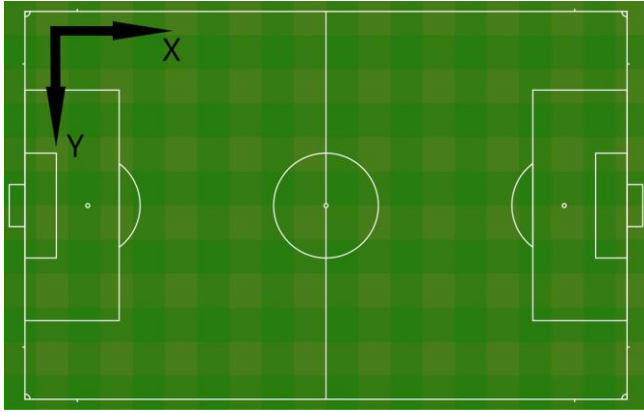
The state-of-art real-time two-camera player tracking system SentioScope, developed by Sentio, collects data from Turkish Super League (TSL) matches [25]. Using this data, we created a dataset for each game to analyze. For each second, position data of players of both teams and the ball in a rectangular coordinate system are saved in this dataset  $\mathcal{D}$ . We identify the home team as  $\mathcal{H}$  and away team as  $\mathcal{A}$ . The ball is labeled as  $\mathcal{B}$ , and the main referee is denoted as  $\mathcal{R}$ . The dataset for a match  $M$  is constructed as follows:

$$\mathcal{D}_M = \{c_i^t = (x_i^t, y_i^t) \mid \forall i \in \mathcal{H} \cup \mathcal{A} \cup \mathcal{B} \cup \mathcal{R}, \quad (1) \\ t = 1, 2, \dots, T_M\}$$

where  $c_i^t$  is the coordinates of the  $i^{\text{th}}$  object (player, referee, or ball) at timestep  $t$ ,  $x_i^t$  is the x-axis coordinate and  $y_i^t$  is the

y-axis coordinate.  $T_M$  is the maximum number of seconds in the match  $M$ .

The dataset contains data for 300 matches of the Turkish Football Federation Super League 2017-2018 season. It encloses the speed and location information of players and the main referee. As the raw data is collected using optical tracking cameras, it suffers from previously mentioned flaws to precisely track the ball [26].



**Figure 1.** Optical tracking software's coordinate system

In the dataset, spatial information of the ball is implicitly available during the frames when some player owns the ball. When the ball is not in play due to the game's pauses, we are also not interested in ball location since it does not have a value. Therefore, we are particularly interested in moments where the ball is in possession of a player. Those are the moments the ball is occluded the most, making computer vision-based ball tracking challenging.

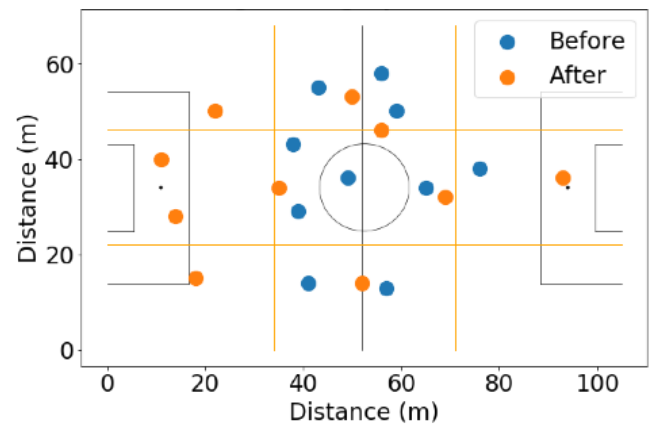
In the following sections, we introduce the formulation of our approach, which can also be applied to other team sports which possess the notion of ball possession, such as handball, basketball, and American football. We first propose a segment-based representation method that handles the ordering problem of the features. After that, we describe our feature extraction methodology. Finally, we present our neural network model to predict the ball location.

### 3.1. Segment-based Representation

There are 11 players for each team in a typical football match, adjusting their positions according to the ball location. Therefore, their collective behavior gives a good indication of the ball's location. For each player on the pitch, the feature set could be represented in a vector. Thus, the collection of individual player attributes forms a matrix that can be used in a machine learning task. However, the number of players can change due to certain events in the game, such as red cards or injuries. Also, due to the substitutions of players, the identities of the players may change. Moreover, in each game, there are different teams and different players playing in different formations. For all these reasons, it is impossible to find a correct ordering for the individual players to be represented in the feature matrix.

In order to address these problems and players' positional interchanges and capture the flow of players' movement, we suggest a data representation method using a role-based approach. The main idea of the proposed method is to divide the pitch into segments and assign players to these segments. This method enables us to set a common data representation regardless of the team and player identities; thus, we can use the same representation for all the matches.

We separate the football pitch into different segments on each axis and assign each player to the corresponding segment on each axis by assessing their movements for the most recent minutes. After grouping players, we use players' coordinates, and speed attributes to extract features, such as average characteristics and attributes of outlier players in the groups with faster speed or slowest speed, for example.



**Figure 2.** Visualization of scaling of average positions

To find out the players' segments, simple averaging of their movements alone is not enough. Football has its own well-established play-book, such as the tendency of teams to keep their formation structure when the opposition team owns the ball. To this end, when segment assignment is carried out, the average positions of a team are calculated over time steps when the rival team has the ball. The average positions are calculated over fixed-width overlapping sliding windows of 15 minutes with a step size of one minute. Due to the averaging, the positions tend to be grouped towards the middle of the field, as depicted in Figure 2 with blue dots. We observe that although the average distribution of the players may show some pattern of formation, it lies around the middle of the field in a squeezed form. In order to represent the average distribution of player formation across the whole pitch, we scale the positions according to the full field size. For each player  $i$ , the scaled coordinate  $c'_i$  is calculated as follows:

$$c'_i = \delta_2 - \frac{[(\delta_2 - \sigma) - (\delta_1 + \sigma)](\alpha - c_i)}{\alpha - \beta} \quad (2)$$

where  $c_i$  is the actual coordinate of player  $i$  for a given axis for a given time step,  $\delta_1$  and  $\delta_2$  are the boundaries of the segment,  $\alpha$  is the coordinate of the player that has the maximum value,  $\beta$  is the coordinate of the player that has the minimum value and  $\sigma$  is the variance of the player coordinates. In this way, for each time step, players' average positions on each axis are scaled to the range  $[\delta_1 + \sigma, \delta_2 -$



$\sigma$ ] based on the dispersion of the players' coordinate distribution. In this way, we obtain the new scaled positions as shown in orange dots in Figure 2.

Updating average positions in overlapping windows and grouping players at each minute based on their scaled averages allow us to capture the trends in players' movements even when a player migrates to different positions during the game or when two players swap their positions. In order to ensure the best representative groups, we partition the field into smaller segments making the area for each segment large enough to host at least one player at each frame. If there are too many segments, most of them will be empty most of the time since players usually stay towards the center and go to some of these segments for a short amount of time, especially the ones at the corners. Their short presence there does not change their overall average position much. For this reason, empirically, we divided the pitch into three different sections on each axis. Segment division on each axis separately helps deduce positional information about players' placement jointly on both axes and mitigate the sparsity of the feature vector introduced by empty segments. The boundaries of these segments are provided in Table 1 and Table 2 for the x and y axes, respectively.

**Table 1.** Segment groups along the x-axis and their boundaries

Segments	Boundaries (m)
Vertical Back (VB)	0, 34
Vertical Middle (VM)	34, 71
Vertical Front (VF)	71, 105

**Table 2.** Segment groups along the y-axis and their boundaries

Segments	Boundaries (m)
Horizontal Top (HT)	0, 22
Horizontal Middle (HM)	22, 46
Horizontal Bottom (HB)	46, 68

We use these segments to assign a role to each player using their scaled average coordinates. Results of the proposed role assignment method reflect the players' positional distribution properly. Having a fixed segment representation also allows us to order the features using the segments. This approach also makes it possible to represent players' data dynamically in a fixed order as they simultaneously change their roles during a match. The order of feature representation can simply be initialized in the form of a set of segments as  $\mathcal{S} = \{HT, HM, HB, VB, VM, VF\}$ . Instead of features calculated individually for each player in that setting, we have features extracted for set the of players in each segment group.

### 3.2. Feature Extraction

In our feature set, we consider many aspects of the football game to obtain the best set of features that can be used to predict the ball location. To begin with, the direction of the game flow depends on the movement of the player who possesses the ball, whose position, in turn, depends on the positional distribution of the other players and their spatial values, such as location, speed, and direction. In order to build a feature set that can help to map from feature space to the game flow at any given moment, we should consider all these spatial features. To capture the relevant connection among all the role groups and team groups (i.e., home team and away team), we calculate features using the groups. Furthermore, we also perform the feature extraction on the combined set of both teams to find the possible interactions among teams. We define our features on different sets:  $\mathcal{H}$  and  $\mathcal{A}$  are the set of the home team and away team's players except for the goalkeepers, respectively.  $\{\mathcal{S}_i \cap \mathcal{H}\}_i^{|\mathcal{S}|}$  and  $\{\mathcal{S}_i \cap \mathcal{A}\}_i^{|\mathcal{S}|}$  represent the set of players for each segment group in each team, and the set  $\mathcal{H} \cup \mathcal{A}$  contains all the players except for the goalkeepers. We represent goalkeepers and the main referee separately. For achieving the unity of expression, we define them as sets that contain a single element. We denote the goalkeepers for home and away teams as  $\mathcal{G}_{\mathcal{H}}$  and  $\mathcal{G}_{\mathcal{A}}$ , respectively. We denote the referee set as  $\mathcal{R}$ .

The speed is also one of the crucial components that provide insight into the ball's location. Teams can develop counterattacks or play with slow tactical passes just before an attack, or when a player dribbles the ball, he runs or sprints to pass his rival. All of these behavior patterns can be used to predict the location of the ball. In order to incorporate this into our prediction model, we categorize players into distinct speed groups. Empirically, we devised two groups. These groups are identified as *Low Intensity* (the speed is less than or equal to 3.5 m/s) and *High Intensity* (the speed is greater than 3.5 m/s). We observed that these speed groups show different characteristics in their relation to the ball's coordinates. For example, the distance of the *Low Intensity* (*LI*) group to the ball is usually more than that of the *High Intensity* (*HI*) group.

The movement direction is another essential component of motion when it comes to finding the ball's location. Thus, for each time step, we calculate the direction of the average movement for a group of players  $dir(G)$  as follows:

$$dir(G) = sign\left(\sum_i c_i^t - c_i^{t-1}\right), \forall i \in G \quad (3)$$

Ball control is an essential element in the football rulebook. Hence, to gain more control of the ball, players approach each other, eventually approaching the ball. The distribution of players gets denser as the game gets close to one of the goal lines. Capturing the form of players' positional distribution is an essential auxiliary element for defining ball location. Thus, the average position of each player group is calculated on each axis separately for each player group.

$$avg(c_p) = \frac{1}{N} \sum_{p=1}^N (c_p), \forall p \in G \quad (4)$$

However, the average position is not helpful when some players in that group are grouped closely, and the remaining players are relatively remote. For this reason, we also use the variance of the player coordinates

$$var(c_p) = \frac{1}{N} \sum_{i=1}^N (c_i - avg(c_p))^2, \forall p \in G \quad (5)$$

for different groups for players such as home team players,  $\mathcal{H}$ , away team players,  $\mathcal{A}$ , all players,  $\mathcal{H} \cup \mathcal{A}$  and player groups found with a density-based clustering (DBSCAN) approach [27]. We apply DBSCAN to have a more robust distribution representation by finding clustered groups of players. The algorithm starts from an arbitrary point in the group  $G$  and finds the cluster of neighborhood points where

at least  $N_{\min}$  of them are directly density-reachable from this arbitrary point with respect to  $\epsilon$  such that

$$|\{p \in G: d(x_{p_i}, x_{p_j}) \leq \epsilon\}| \geq N_{\min} \quad (6)$$

where  $d(\cdot)$  is a distance function. Density-based clustering allows us to find the player clusters where players are closer to each other. The center of this cluster is often close to the location of the ball. In our approach, we used  $\epsilon = 15\text{m}$  and  $N_{\min} = 7$  for the set  $\mathcal{H} \cup \mathcal{A}$ , and  $\epsilon = 15\text{m}$  and  $N_{\min} = 4$  for the  $\mathcal{H}$  or  $\mathcal{A}$  since their group size is smaller than the union set. In addition to DBSCAN features, we also extract features for specific target groups. For example, the referee's coordinates and speed were extracted by considering the fact that referee movements can be determinant since the referee often stands next to positions to resolve any dispute on the pitch. A compact representation of all the features we have extracted is provided in Table 3. In total, we use 251 different features per time step.

**Table 3.** List of features.  $G$  is the player set,  $G^+$  is the set of players with HI speed,  $G^-$  is the set of players with LI speed,  $G^*$  represents the cluster set of players that is found by DBSCAN.  $v_p$  is the speed and  $c_p$  is the coordinate of player  $p$ .

$G = \mathcal{H} \text{ or } \mathcal{A}$	$G = \{\mathcal{S}_i \cap \mathcal{H}\}_i^{ \mathcal{S} } \text{ or } \{\mathcal{S}_i \cap \mathcal{A}\}_i^{ \mathcal{S} }$	$G = \mathcal{H} \cup \mathcal{A}$	$G = \mathcal{G}_{\mathcal{H}} \text{ or } \mathcal{G}_{\mathcal{A}}$	$G = \mathcal{R}$
$avg(c_p) \forall p \in G$	$avg(c_p) \forall p \in G$	$avg(c_p) \forall p \in G$	$c_p, \forall p \in G$	$c_p, \forall p \in G$
$avg(v_p) \forall p \in G$	$avg(v_p) \forall p \in G$	$avg(v_p) \forall p \in G$	$v_p, \forall p \in G$	$v_p, \forall p \in G$
$dir(G)$	$dir(G)$	$var(c_p) \forall p \in G$	$dir(G)$	$dir(G)$
$var(c_p) \forall p \in G$	$var(c_p) \forall p \in G$	$avg(v_p) \forall p \in G^*$	$d(c_p, avg(c_i)),$	$d(c_p, avg(c_i)),$
$avg(c_p) \forall p \in G^+$	$avg(c_p) \forall p \in G^+$	$var(c_p) \forall p \in G^*$	$\forall p \in \mathcal{G}_{\mathcal{H}}, \forall i \in \mathcal{H}$	$\forall p \in \mathcal{R}, \forall i \in \mathcal{H} \cup \mathcal{A}$
$avg(v_p) \forall p \in G^+$	$avg(v_p) \forall p \in G^+$		$d(c_p, avg(c_i)),$	$d(c_p, avg(c_i)),$
$avg(c_p) \forall p \in G^-$	$avg(c_p) \forall p \in G^-$		$\forall p \in \mathcal{G}_{\mathcal{A}}, \forall i \in \mathcal{A}$	$\forall p \in \mathcal{R}, \forall i \in G^*$
$avg(v_p) \forall p \in G^-$	$avg(v_p) \forall p \in G^-$			
$avg(c_p) \forall p \in G^*$				
$avg(v_p) \forall p \in G^*$				
$var(c_p) \forall p \in G^*$				
$min(c_p) \forall p \in G$				
$min(v_p) \forall p \in G$				
$max(c_p) \forall p \in G$				
$max(v_p) \forall p \in G$				
$c_i, i = \underset{p \in G}{\operatorname{argmin}}(v_p)$				
$c_i, i = \underset{p \in G}{\operatorname{argmax}}(v_p)$				
$v_i, i = \underset{p \in G}{\operatorname{argmin}}(c_p)$				
$v_i, i = \underset{p \in G}{\operatorname{argmax}}(c_p)$				

#### 4. PERFORMANCE EVALUATION

We use two separate artificial neural network regression models for predicting ball location along the x-axis and y-axis. In our experimental setup, we use training, validation, and test sets containing data from 243 (81%), 27 (9%), and 30 (10%) matches respectively. We randomly select the matches using their unique identifiers from the whole dataset that contains 300 matches. The matches in the training set were used in the training stage of the neural network models. We used 27 matches as a validation set to perform the

hyperparameter optimization. After the hyperparameter tuning is finished. We tested the performance of the final tuned model on the test set. The match data in the test set were used only at that stage.

The models have been trained with dropout [28]. Moreover, an early-stopping technique has been implemented to avoid overfitting [29]. In our setting, we use L2 loss as the main loss function. Furthermore, mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of

determination  $R^2$  evaluation metrics are used as well in order to carry out fair performance evaluation.

$$MAE = \frac{1}{N} \sum_i |y_i - f_i| \quad (7)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_i |y_i - f_i|^2} \quad (8)$$

$$R^2 = 1 - \frac{\sum_i (f_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2} \quad (9)$$

To avoid saturation of activation function, we normalized the target variables using min-max normalization. However, we scaled back the predicted outputs to the normal scale when we did the performance measurement. That method helped our training network to converge to a better local optimum.

The final proposed model is built on a deep neural network through a series of experiments. We optimized the depth of our neural network, the number of hidden nodes, the activation function, the optimization algorithm, and the learning rate of the optimization algorithm. All of the hyperparameter optimizations are performed using the validation set. As a result, we use the rectified linear unit activation [30] as our nonlinear activation function with a gradient-based stochastic optimization algorithm with Adam optimizer [31] with a batch size of 65 and a learning rate of 0.01. The depths of the neural networks are 7 for the x-axis prediction model and 5 for the y-axis prediction model. The number of hidden nodes is 251 for both models.

According to our experimental evaluation, the result of the coefficient of determination on the train set of the x-axis and y-axis are 85% and 94%, respectively. The performance on the test set is 79% and 92% for the x-axis and y-axis, respectively. Our results indicate that the method performs well in its generalization ability. The mean absolute error of each model on the test set was calculated to gain more insight regarding the ball's location on the pitch. On the test, the error is slightly above 7.56 meters on the x-axis while it is 5.01 meters on the y-axis. The mean squared error yields marginally higher results than the mean absolute error. The most significant errors occur in cases when the ball transaction happens unexpectedly and when the ball changes its position from one player to another over a long distance, which often occurs during the long passes and shots on goal.

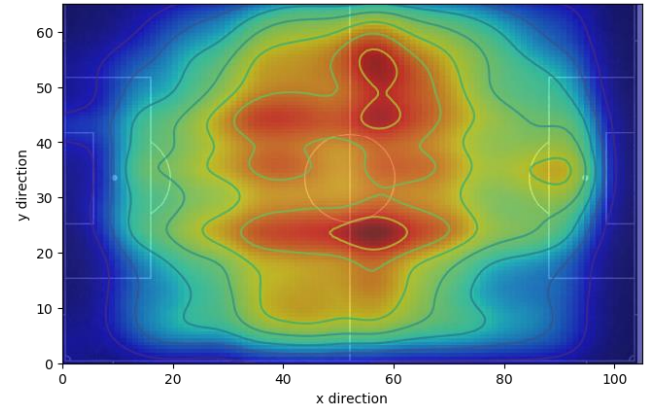
**Table 4.** Regression results on the x-axis

Dataset	MAE	RMSE	$R^2$
Train	6.47	9.90	84.64
Validation	7.39	11.23	80.11
Test	7.56	11.46	79.41

**Table 5.** Regression results on the y-axis

Dataset	MAE	RMSE	$R^2$
Train	4.12	6.23	93.56
Validation	4.83	6.96	92.74
Test	5.01	7.19	92.16

Overall, we can see that our approach performs particularly better on the y-axis. This is because the width of the pitch is almost half of its length and also, the teams are not spread over the y-axis as they try to cover distance mostly on the x-axis in order to reach the opposition goal. The full evaluation results for the x-axis and y-axis are provided in Table 4 and Table 5, respectively.

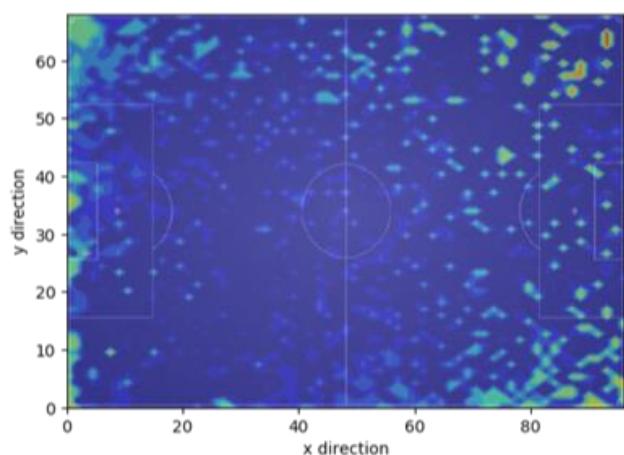


**Figure 3.** Heatmap of the predicted coordinates  $c'_i = (x'_i, y'_i)$  when the Euclidean distance between the actual  $c_i = (x_i, y_i)$  and the predicted coordinate is bigger than the Euclidean distance between the mean absolute error of test set on both axes.

We also visualize our model performance using a spatial representation with heatmaps. In Figure 3, we present the heatmap for the density of the predicted coordinates,  $c'_i = (x'_i, y'_i)$ , when the Euclidean distance between the actual coordinate,  $c_i = (x_i, y_i)$ , and the predicted coordinate is bigger than the Euclidean distance between the mean absolute error of test set on both axes, i.e.,  $\sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2} > \sqrt{x_{MAE}^2 + y_{MAE}^2}$ . This gives us an impression about which parts of the pitch the “faulty predictions” occur the most. Since the ball is at the center region on average, the errors also happen at that region.

In Figure 4, we provide the heatmap of the density distribution of all the errors using the Euclidean distance between the prediction and the actual coordinate. This heatmap visualizes the magnitude of errors the model makes. We observe that regions closer to the goals, corners, wings are the places where the model makes the largest prediction error. The results represented here are consistent with our observations that the model initially struggles to adapt when the goalkeeper starts the game with a long shot, a corner kick is taken, or when there is an unexpected change of attack from one wing to another.

Overall, with these two heatmaps, we provide insights about both the number of errors and the magnitude of errors across the pitch. Additionally, we provide an animated image of our method's performance on real-world match data together with our codebase used for obtaining the results presented in this study at <https://github.com/anaramirli/predict-soccer-ball-location>.



**Figure 4.** Heatmap of the density distribution of all the errors based on the Euclidean distance between the prediction and the actual coordinate.

## 5. CONCLUSIONS

We presented our approach to track the ball location in football, especially when it is occluded. We showed that the ball's coordinates could be estimated from optical tracking data by using machine learning models. In addition to the results obtained by using neural networks, we also present novel ways of extracting features that are the most significant for predicting the ball's location.

The ability of our model's predictions on the y-axis is around 5 meters. The results for the x-axis (~7.5 m) are not as good as that of the y-axis model. The model for x-axis struggles, especially when a goalkeeper starts the game or a player takes a free kick. The model prediction focuses on the player groups on the x-axis rather than a player who is with the ball. During the typical long passing when the ball rapidly changes location halfway through the other side, we observed that the regression models could not identify these changes for the first few frames. However, the achieved prediction rate is good enough to apply it to the existing system as an additional tool and increase their performance.

For our future study, we will explore the ways of improving the performance of the models by employing some auxiliary models such as the detection of the ball from the detection of the game events such as free kick, corner, penalty. The regression models we use for prediction ball location can be combined with the detected event's field lines and thus generalize performance more precisely. Moreover, the accuracy of the model can be improved with more detailed features. It is also important to mention that in further studies, we can utilize Adversarial Generative Networks (GAN) [32] to eliminate the shortcoming of input space representation of individual player attributes that we face in traditional neural networks. Our future work will also focus on pixel-wise detection of the ball location using conditional Pixel2Pixel GAN [33] architecture by incorporating individual features of players together with that of group-based features that we proposed in this study.

**Author contributions:** Concept – H.A., A. A.; Data Collection and Processing - A.A.; Literature Search - H.A., A. A.; Writing - H.A., A. A.

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

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

## REFERENCES


- [1] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews, "Large-Scale Analysis of Soccer Matches Using Spatiotemporal Tracking Data," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2015-Janua, no. January, pp. 725–730, 2014.
- [2] B. Skinner and S. J. Guy, "A method for using player tracking data in basketball to learn player skills and predict team performance," *PLoS One*, vol. 10, no. 9, pp. 1–15, 2015.
- [3] P. Lucey, D. Oliver, P. Carr, J. Roth, and I. Matthews, "Assessing team strategy using spatiotemporal data," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. Part F1288, pp. 1366–1374, 2013.
- [4] C. Perin, R. Vuillemot, C. D. Stolper, J. T. Stasko, J. Wood, and S. Carpendale, "State of the Art of Sports Data Visualization," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 663–686, 2018.
- [5] A. Rusu, D. Stoica, E. Burns, B. Hample, K. McGarry, and R. Russell, "Dynamic visualizations for soccer statistical analysis," *Proc. Int. Conf. Inf. Vis.*, pp. 207–212, 2010.
- [6] D. Sumpter, *Soccermatics: Mathematical Adventures in the Beautiful Game*. Bloomsbury Publishing Plc, 2016.
- [7] L. Gyarmati, H. Kwak, and P. Rodriguez, "Searching for a Unique Style in Soccer," 2014, pp. 5–8.
- [8] L. Y. Wu, A. J. Danielson, X. J. Hu, and T. B. Swartz, "A contextual analysis of crossing the ball in soccer," *J. Quant. Anal. Sport.*, vol. 17, no. 1, pp. 57–66, 2021.
- [9] V. Khaustov and M. Mozgovoy, "Recognizing events in spatiotemporal soccer data," *Appl. Sci.*, vol. 10, no. 22, pp. 1–12, 2020.
- [10] E. Özdemir and H. Alemdar, "Predicting soccer events from optical tracking data," *26th IEEE Signal Process. Commun. Appl. Conf. SIU 2018*, pp. 1–4, 2018.
- [11] P. R. Kamble, A. G. Keskar, and K. M. Bhurchandi, "Ball tracking in sports: a survey," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 1655–1705, 2019.
- [12] A. E. Abulwafa, A. I. Saleh, H. A. Ali, and M. S. Saraya, "A fog based ball tracking (FB2T) system using intelligent ball bees," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 11, pp. 5735–5754, 2020.
- [13] D. G. Cardenas and M. D. Zuniga, "Bullet-Proof Robust Real-Time Ball Tracking," in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016, pp. 1–8.
- [14] A. Lhoest, "Deep Learning for Ball Tracking in Football Sequences," University of Liège, 2020.
- [15] H. D. Najeeb and R. F. Ghani, "Tracking Ball in Soccer Game Video using Extended Kalman Filter," *Proc. 2020 Int. Conf. Comput. Sci. Softw. Eng. CSASE 2020*, pp. 78–82, 2020.


- [16] W. C. Naidoo and J. R. Tapamo, "Soccer video analysis by ball, player and referee tracking," *ACM Int. Conf. Proceeding Ser.*, vol. 204, pp. 51–60, 2006.
- [17] J. Ren, J. Orwell, G. A. Jones, and M. Xu, "Tracking the soccer ball using multiple fixed cameras," *Comput. Vis. Image Underst.*, vol. 113, no. 5, pp. 633–642, 2009.
- [18] J. Komorowski, G. Kurzejamski, and G. Sarwas, "BallTrack: Football ball tracking for real-time CCTV systems," *Proc. 16th Int. Conf. Mach. Vis. Appl. MVA 2019*, 2019.
- [19] P. R. Kamble, A. G. Keskar, and K. M. Bhurchandi, "A deep learning ball tracking system in soccer videos," *Opto-electronics Rev.*, vol. 27, no. 1, pp. 58–69, 2019.
- [20] M. Durus, "Ball Tracking and Action Recognition of Soccer Players in TV Broadcast Videos," *Technische Universität München*, 2014.
- [21] J. Komorowski, G. Kurzejamski, and G. Sarwas, "Footandball: Integrated player and ball detector," *VISIGRAPP 2020 - Proc. 15th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 5, pp. 47–56, 2020.
- [22] M. Leo, P. L. Mazzeo, M. Nitti, and P. Spagnolo, "Accurate ball detection in soccer images using probabilistic analysis of salient regions," *Mach. Vis. Appl.*, vol. 24, no. 8, pp. 1561–1574, 2013.
- [23] J. Hossein-Khani, H. Soltanian-Zadeh, M. Kamarei, and O. Staadt, "Ball detection with the aim of corner event detection in soccer video," *Proc. - 9th IEEE Int. Symp. Parallel Distrib. Process. with Appl. Work. ISPAW 2011 - ICASE 2011, SGH 2011, GSDP 2011*, pp. 147–152, 2011.
- [24] Z. Niu, X. Gao, and Q. Tian, "Tactic analysis based on real-world ball trajectory in soccer video," *Pattern Recognit.*, vol. 45, no. 5, pp. 1937–1947, 2012.
- [25] S. Baysal and P. Duygulu, "Sentioscope: A Soccer Player Tracking System Using Model Field Particles," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1350–1362, 2016.
- [26] E. K ulah and H. Alemdar, "Quantifying the value of sprints in elite football using spatial cohesive networks," *Chaos, Solitons and Fractals*, vol. 139, 2020.
- [27] M. Daszykowski and B. Walczak, "Density-Based Clustering Methods," *Compr. Chemom.*, vol. 2, pp. 635–654, 2009.
- [28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [29] L. Prechelt, "Early stopping - But when?," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 *LECTU*, pp. 53–67, 2012.
- [30] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, vol. 15, pp. 315–323.
- [31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [32] I. J. Goodfellow et al., "Generative Adversarial Nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2014, pp. 2672–2680.
- [33] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.



## Handwritten Digit Recognition With Machine Learning Algorithms

\*<sup>1</sup>Kübra Gülgün Demirkaya, <sup>2</sup>Ünal Çavuşoğlu

<sup>1</sup>Sakarya University, Computer Engineering [kubra.demirkaya@ogr.sakarya.edu.tr](mailto:kubra.demirkaya@ogr.sakarya.edu.tr), 

<sup>2</sup>Sakarya University, Software Engineering, [unalc@sakarya.edu.tr](mailto:unalc@sakarya.edu.tr), 

### Abstract

Nowadays, the scope of machine learning and deep learning studies is increasing day by day. Handwriting recognition is one of the examples in daily life for this field of work. Data storage in digital media is a method that almost everyone is using nowadays. At the same time, it has become a necessity for people to store their notes in digital media and even take notes directly in the digital environment. As a solution to this need, applications have been developed that can recognize numbers, characters, and even text from handwriting using machine learning and deep learning algorithms. Moreover, these applications can recognize numbers, characters, and text from handwriting and convert them into visual characters. This project, investigated the performance comparison of machine learning algorithms commonly used in handwriting recognition applications and which of them are more efficient. As a result of the study, the accuracy was 98.66% with artificial neural network, 99.45% with convolutional neural network, 97.05% with K-NN, 83.57% with Naive Bayes, 97.71% with support vector machine and 88.34% with decision tree. This study also developed a handwriting recognition system for numbers similar to these mentioned applications. A desktop application interface was developed for end users to show the instant performance of some of these algorithms and allow them to experience the handwriting recognition system.

**Keywords:** Machine Learning, Handwriting Recognition, Deep Learning, MNIST

### 1. INTRODUCTION

Today, the applications of artificial intelligence and data science are advancing very rapidly and are now used in almost all fields. Most of these applications include machine learning and deep learning. One of the application areas is handwriting recognition as an evolving field. Handwriting recognition enables communication between machines and humans and is a field that aims to facilitate this communication. A lot of information is now stored in digital media. In daily life, almost everyone has started storing their data and notes in digital media and using electronic diaries. They tend to take notes in the digital environment using the keyboard, touch screen, and smart pens of smartphones and tablets that they always have with them. Handwriting recognition systems are needed more and more every day for the following reasons: For data to be stored in digital media with recognition once handwritten, for data previously written to be transferred to digital media as optical characters. With handwriting recognition systems beginning to evolve out of this need, today's technology has moved away from keyboards to touch systems, and writing in the digital environment has begun to be done with handwriting instead of the small keyboards of touch screens. In addition, handwritten text documents, lettering on signs, etc. have begun to be transferred to digital in sectoral areas. Handwriting recognition systems are constantly evolving and being integrated into intelligent systems. Many

electronic tablets and digital agendas are now offered to users with handwriting recognition system software and smart pens. At the same time, in addition to everyday use, these systems are also being used in areas such as education, health, and security, which are evolving every day. In banks, handwriting recognition systems are used in areas such as reading check amounts and forms, and reading and sorting incoming mail addresses [1]. In addition, studies are being conducted daily for the systems to be used by children in smart boards and tablets, mainly to be used in education and to support teaching.

Handwriting recognition is the computer recognition of handwritten letters, numbers and characters. This process, which is very simple for a human, is difficult for computers. In other words, making sense of lines, symbols, and their combined shapes at the word level is difficult for computers. Handwriting features such as the presence of characters that are different in many languages, the fact that each person has different handwriting, and the presence of combined handwriting make it difficult for computer systems to recognize handwriting. This topic is not yet fully developed and it is an area of limited efficiency [2]. When this technology is developed, which is mainly used in tablet computers and for which there are already examples, it will be possible to store and organize any handwritten information in a digital environment without using a keyboard. Handwriting recognition technologies can be

\* Corresponding Author

studied in two different groups. One is interactive (online) and the other non-interactive (offline) systems. Interactive systems are systems that recognize and classify by following the movements of the writing tool as it writes the handwriting. They are usually used in devices with a touch surface such as tablets, phones, smartboards, etc. Every movement during writing is controlled by the device. This is an important factor in increasing the accuracy rate. The disadvantage of these systems is that they have to give instant results and they have to be fast running systems as they have to keep up with the writing speed [2]. Non-interactive systems are an attempt to recognize the information previously written on paper by digitizing it using methods such as photography and scanning. These systems are not expected to produce instantaneous output, so there are ways to work more slowly. However, since handwriting may not be very smooth, especially in old handwritten documents, the accuracy rates in the recognition process are low. To increase this rate, extensive pre-processing on the photo is required first. However, as a first step, the text should be divided into sections. Lines, sentences, words and characters should be divided according to their size, then preprocessing steps should be applied to each divided part before classification [1]. The advantage of this method is that it can be used for all kinds of documents that have existed for years and should be transferred to electronic media.

In this study, detailed information about the different machine learning and deep learning algorithms used in the field of handwriting recognition was given and experiments were conducted to measure their performance in this field, discover the most efficient parameters and compare the success rates on the MNIST dataset. The results obtained at the end of the experiments were compared with the results of similar studies in this field. In addition, using some of these algorithms, a system for recognizing digits from non-interactive handwriting, similar to handwriting recognition applications, was developed and an example usage method suitable for everyday use was demonstrated. A desktop application was developed for the end users of the application. In the second part of this study, similar studies on the topic in the literature are explained. In the third section, the dataset used, the development environment, the method and the evaluation criteria are explained in detail. In the fourth section, the details of the research conducted are explained. Detailed information is given about the design of each model created and the algorithm used in the model. In the fifth section, the interface application created is explained. In the sixth chapter, the results obtained at the end of the experiments are explained and the comparison of the success rates of the models in this study with similar studies are included. In the seventh chapter, the findings and evaluations obtained from the study are included in this direction.

## 2. RELATED STUDIES

In the work of Mohd Razif Shamsuddin and his colleagues who analyzed machine learning models for the MNIST dataset; They obtained 2 different versions of the MNIST dataset, grayscale and binary (black and white), with data preprocessing. They also performed the model training with these 2 different datasets. In the grayscale dataset; The

accuracy rate was 99.4% with the CNN model, 94% with the random forest model, and 94% in the Extremely Randomized Trees model. In the binary data set; The accuracy rate was 90.1% with the CNN model, 91% with the Random Forest model, and 92% with the Extremely Randomized Trees model. As a result of these experiments, they showed the effect and importance of proper selection of data preprocessing step and methods in machine learning on success rate [3]. Abien Fred M. Agarp created an architecture for machine learning by combining convolutional neural networks and support vector machines for image classification problems. In addition to the model where he used the ReLu function as the activation function in the CNN structure and then added a support vector machine, he created another model where the same CNN structure determined the activation function of the output layer as a softmax function. 2 models named "CNN-Softmax" and "CNN-SVM"; trained with MNIST and Fashion-MNIST datasets and compared the results. In the study, 60000 training data and 10000 test data from both datasets were used. With the structure of "CNN-Softmax", 99.23% success rate in MNIST dataset and 91.86% success rate in FashionMNIST dataset; With the structure of "CNN-SVM", it achieved 99.04% success rate in MNIST dataset and 90.72% in Fashion-MNIST dataset [4].

Mine Altınay Günler Pirim argued that the output weights in the hidden layers in the structure of neural network used in handwriting recognition can be used to extract the feature vectors of the image. For this purpose, it used MNIST and USPS datasets in its experiments with MATLAB; using the artificial neural network, the support vector machine and the Euclidean distance classifier algorithm as the classifier, the success rates were measured. Using the structure developed in the study, it achieved success rates of 99.64% in the support vector machine model with the MNIST dataset, 98.01% in the Euclidean distance classifier model, and 98.56% in the artificial neural network model. With the USPS dataset, it achieved 97.47% success rates in the support vector machine model, 94.52% in the Euclidean distance classifier model, and 94.18% in the artificial neural network model [5]. Aoudou Salouhou, tested and compared deep learning algorithms for image classification and handwriting recognition using Fashion-MNIST, MNIST, CIFAR-10 and Arabic datasets in his study. To classify Arabic characters, the Arabic dataset created by Ahmed and his friends was used [6]. He used Deep Neural Network, Convolutional Neural Network and recursive neural network structures from Deep Learning algorithms. In his experiments with the MNIST dataset, the deep neural network model provided 99.53% accuracy rate, the convolutional neural network model provided 99.88% accuracy rate, and the iterative neural network model provided 99.05% accuracy rate. In his experiments with the Arabic dataset; The deep neural network model provided 96.48% accuracy, the convolutional neural network model provided 99.00% and the recursive neural network model provided 96.94% accuracy [7].

In her studies on handwriting recognition with machine learning algorithms, Rabia Karakaya conducted tests with support vector machine, decision tree, random forest, artificial neural network, K-nearest neighbor algorithm and

K-mean algorithm using MNIST dataset. She conducted her work using Scikit Learn library and tools. In the test results of the models trained with MNIST dataset using all 60000 data, she achieved 90% accuracy rate by using polynomial kernel function in support vector machine model. It also achieved 87% accuracy in the decision tree model, 97% in the random forest model, and 97% in the artificial neural network model. In the K-Nearest Neighbor algorithm model, it achieved 96% accuracy in 865.932 seconds of test time and 98% accuracy in the K-Mean algorithm model [8]. In their work, Shubham Sanjay Mor and his colleagues developed a system and an Android application to recognize handwritten characters and numbers using the EMNIST dataset and the CNN structure. At the end of the experiments they conducted in the CNN model they created, they used the "Adamax" function as an optimization parameter with which they achieved the highest performance. The model they created recognizes 62 handwritten characters and has an accuracy rate of 87.1% [9].

### 3. MATERIAL AND METHOD

#### 3.1. MNIST Dataset

The dataset used for training and testing this system is the MNIST (Modified National Institute of Standards and Technology) dataset. It is a dataset created to process and make sense of the image. It consists of images of handwritten numbers, each of which is 28x28 pixels in size. It contains 60,000 training (train) and 10,000 verification (test) images as classification [10]. It is a commonly preferred dataset in studies in similar domains.

#### 3.2. Libraries

The software libraries used in this study are TensorFlow, Scikit Learn, Keras and Numpy libraries. In the following sections of this study, some mathematical values are used to make comparisons in performance evaluations and analysis. These mathematical values defined in the mentioned libraries and their definitions are as follows:

**Confusion Matrix:** Confusion Matrix or error matrix is a performance measurement method for machine learning classification algorithms. It gives information about the accuracy of the predictions. It is a table that contains 4 combinations of predictions and actual values [11]. With the help of this matrix, we can calculate values like precision, recall, support, accuracy, specificity, and F1 score.

**Accuracy:** It is a value that indicates the accuracy rate. It is the ratio of predictions classified as correct to all predictions.

		Classifier's Prediction	
		Positive	Negative
Real Value	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Figure 1. Confusion Matrix

**Precision:** It is the value that indicates how much of the positively predicted data was predicted correctly. The higher this ratio is, the more accurate the predictions were [12]. It is calculated as;

$$TP/(TP + FP) \quad (1)$$

**Recall:** It is the ability of the classifier to find all positive samples. It is a metric that shows how much of the data that should be positively predicted is positively predicted [12]. It is calculated as;

$$TP/(TP + FN) \quad (2)$$

**F1 Score:** The F1 score can be interpreted as a weighted harmonic average of the Precision and Recall scores. It is a measure of the level of performance exhibited by the model. It is often used to compare models.

#### 3.3. Data Processing

Data sets are processed through multiple phases, as with any MA study courses. Data pre processing, cutting, feature extraction and classification are the key procedures employed in the image classification and handwriting recognition applications.

**Preprocessing data:** Data preparation is performed ready for analysis and subsequent phase; such operations as correction, conversion, cleaning, decrease of size, standardization and noise reduction. In handwriting recognition data the data preparation processes required might be stated in the following way:

**Thresholding:** This is the way the image is converted into a binary picture, namely a black and white one. The basic objective is to highlight the image and identify the item with this technique. Pixels are calculated as black or White according to the supplied threshold value, depending on the type of threshold approach used [13].

**Noise reduction:** Refers to processes carried out to further clear the picture. The highlights can be sharpened in texts by a technique that is needed. It aims to produce a clear picture through this approach. Methods such as a medium filter, median filter, Gaussian smoothing filter and masking approach can minimize noise.

**Normalization:** Slope correction is an additional name for standardization, one of the image processing procedures. This procedure corrects the curvature of the text in handwritten text photographs and eliminates the skew. Histograms are used to identify and correct curvature and path, as with many other image processing procedures. In the early parts of the normalization operations, Bosinosvic and Srihari Method (BSM) is commonly used [14]. The angle between the horizontal axis in the text correction and the horizontal axis of the written text is known as the slope, and the angle between the vertical axis in the text correction and the vertical axis of the written text is known as the slant. Normalization is another term for handwriting correction using slant and slope [15].

**Feature extraction:** Each distinguishing feature in the photos may be designated as an attribute. Attribute information is composed of numerical data derived from the separation of features from the picture [16]. Approaches like histograms, projection-based methods, Fourier and Wavelet transforms, or defining letters as a set of basic shapes like curves and lines are utilized at this step [2]. The information collected from this stage has a direct impact on the recognition stage, and the qualities of this information have a direct impact on the recognition stage's efficiency.

**Classification:** During the classification phase, the attributes of the data in the picture are compared to the classes in the database to determine which class the picture belongs to. Many various approaches are utilized at this step, including template matching, neural networks, classification algorithms, statistical learning, and structural learning. The data sets employed are critical for this stage's high performance and accuracy, and they should be prepared to contain as many samples and kinds as feasible.

### 3.4. The Proposed Model

The implementation steps in the project are as follows:

- 1) The project contains necessary libraries such as Tensorflow, Keras, Scikit Learn, and Numpy.
- 2) The project includes the MNIST dataset from the Tensorflow-Keras library.
- 3) The dataset was labeled and split into training and test data.
- 4) The data was subjected to preprocessing processes.
- 5) Following the preparation processes, the data was normalized and characteristics were retrieved.
- 6) Algorithms to be utilized with libraries have been defined.
- 7) For each method, the parameters to be utilized during training were determined in the most appropriate method for the dataset. It was intended to get the best possible results.
- 8) The dataset was trained using the "Fit" function, and model training was completed. Scikit Learn library was widely utilized during learning in K-NN, SVM, Decision Tree, and Naive Bayes algorithms, while Tensorflow and Keras libraries were employed in neural networks.
- 9) The model was evaluated using the "Predict" function on a portion of the dataset that had been put aside for testing.

- 10) Finally, to assess the performance of the trained and tested model, the classification report and confusion matrix were generated and shown as a report using the "classification report" and "confusion matrix" functions of the Scikit Learn library's "metrics" module. Precision, recall, f1 score, support, and accuracy scores generated from the confusion matrix findings are shown in tabular form in the classification report. This table contains enough information to allow you to examine and evaluate the performance results.

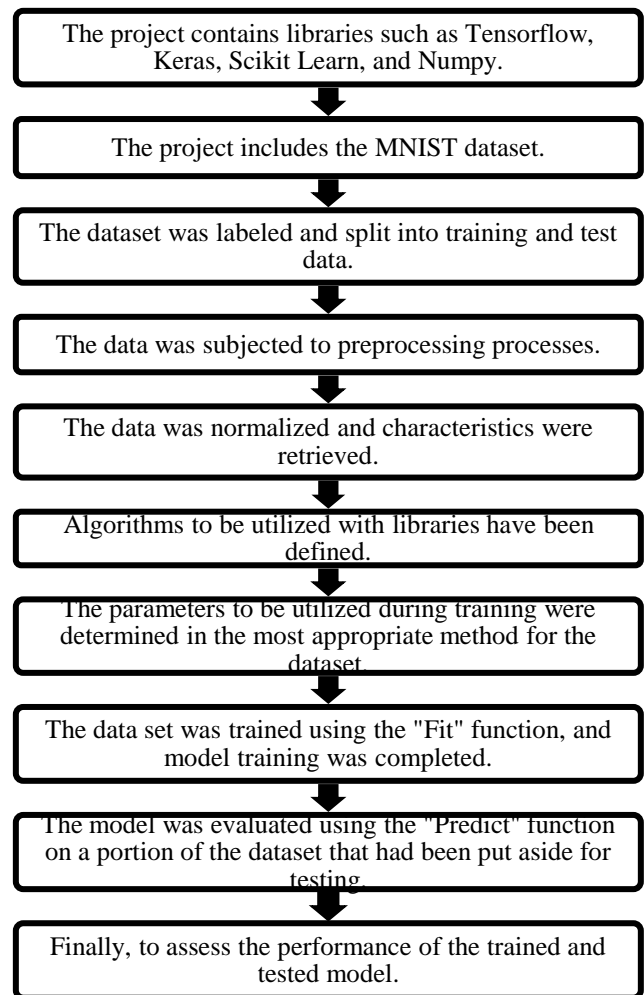


Figure 2. The Implementation Steps

## 4. RESEARCH AND PERFORMANCE RESULTS

The tests carried out, the models constructed using the methods utilized, the hyper-parameter modifications, and the performance results are all detailed in depth in this section. All 70000 bits of data from the data set were used, with 60000 reserved for training and 10000 designated for testing, and the ratios of these portions were kept consistent throughout all models.

### Artificial Neural Network – ANN

Deep learning is based on neural networks, which were formed by modeling the neuron structure of humans and adapting this neuron structure to machines. The human neuron structure was used to generate features such as brain

structure, learning, and information use. To detect real-world relationships, artificial neural networks have been built. They can also conduct classification, pattern recognition, grouping, and estimate. They may process many inputs and generate results [17]. The network structure is formed by the combination of artificial neural network cells within the framework of particular rules. Layers are generated when these brain networks, or neurons, join together. Receiving inputs and sending outputs are handled by cells and layers. As a result, they are linked to the outside world [18]. This structure's layers are divided into three sections: the input layer, the concealed layer, and the output layer.

"Multiple-layer perceptron neural network (MLPNN)" refers to artificial neural networks that have one or more hidden layers in addition to the input and output layers. MLPNN structures are artificial neural network structures that are employed and represented in many artificial neural network research nowadays [19]. The study's artificial neural network is a three-layer fully linked network. In order to build the neural network, the "Sequential" model from the Keras library's "model" module was used. According to the "Sequential" paradigm, a neural network will be formed.

The "Dense" class was then added to the "layers" module to generate neural network layers. The number of neurons in the layers was calculated using these modules, and the neural network was built. Neurons in the neural network's hidden layers have activation functions [19]. A neuron's activity is determined by activation functions. As a result, they are extremely important in neural networks. Different functions are selected based on where they will be utilized and how they will effect performance. The most preferred and recommended activation functions in neural network layers are "relu", "sigmoid" and "softmax" functions. In the neural networks in this study, while the "relu" activation function is used in the input layers, the "softmax" function, which is the most preferred in multiple classification problems, is used in the output layers. The "Softmax" function generates values between [0,1] and these values show the probability that each input belongs to a class. After creating the neural network structure, it was compiled with the necessary parameters.

**Table 1.** Parameters applied to the ANN model in this study

Parameters	Values applied in this model
Batch Size	256
Epochs	40
Dropout Rate	(0.5,0.2)
Learning Rate	0.001
Activation Function	Relu
Activation Function (output layer)	Softmax
Loss Function	Categorical cross entropy
Metric	Accuracy

Layer (type)	Output Shape	Param #
dense_2 (Dense)	(None, 512)	401920
dropout_1 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
dropout_2 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 10)	5130
Total params: 669,706		
Trainable params: 669,706		
Non-trainable params: 0		

**Figure 2.** Architecture of the designed ANN model

For optimization, tests were carried out with 3 different parameters. These are the "Adam", "Adadelta" and "RMSprop" parameters. Finally, the model was trained and tested. The results obtained after 40 epochs are as follows:

**Table 2.** Comparison of success rates of optimization parameters

Optimization Parameters	Accuracy Rate
Adadelta	%85.39
Adam	%98.66
Rmsprop	%98.50

### Convolutional Neural Network – CNN

Convolutional neural networks are the form in which artificial neural networks were developed for image data training to be performed using convolution. It is a neural network with a convolutional mathematical operation in some of the layers. In these neural networks, a convolution structure is created by convolution, pooling and smoothing steps just before the ANN structure [20].

The neural network structure has been started with the "Sequential" model, as described in the previous title. "Convolution2D" function was used for convolution in the convolution layer and "Maxpooling2D" function was used in the pooling layer. These two layers can be added as many times as desired. Finally, the "Flatten" function has been used for flattening. Activation functions are used in convolution layers as in neural network structures. In this study, the "relu" function was preferred in neurons of all convolution layers. After the flattening layer, the convolution structure was connected to the ANN structure.

"Batch\_size" and "epochs" values given as parameters to the "compile" function in neural networks indicate how long the training will take. The "epochs" value represents the number of rounds of the training, which determines how many times the data will be shown to the network. The "batch\_size" value is a measure of how much data will be received in each epoch. By changing these values, it is possible to improve the test results and increase the accuracy rate.

In the convolutional neural network model designed in this study, the hyper-parameters were determined as follows:



**Table 3.** The parameters applied to the CNN model in this study

Parameters	Values applied in this model
Batch Size	128
Epochs	30
Dropout Rate	(0.25,0.5)
Learning Rate	0.001
Activation Function	Relu
Activation Function (output layer)	Softmax
Loss Function	Categorical cross entropy
Metric	Accuracy

Optimization parameters given in the compilation section of the created CNN structure also affect the accuracy rate. In this study, 2 different trainings were made using the "adam" and "adadelta" functions and the test results were obtained. All variables except the "optimizer" parameter were kept constant. The architecture of the proposed model is shown in the report obtained with the "summary" function below.

**Table 4.** Comparison of accuracy rates of tested parameters

Optimization Parameters	Accuracy	Loss
Adam	0.994	0.304
Adadelta	0.864	0.548

```

Model: "sequential_3"
-----
Layer (type)                Output Shape              Param #
-----
conv2d_10 (Conv2D)          (None, 26, 26, 16)       160
max_pooling2d_7 (MaxPooling2 (None, 13, 13, 16)       0
conv2d_11 (Conv2D)          (None, 11, 11, 32)       4640
conv2d_12 (Conv2D)          (None, 9, 9, 64)         18496
conv2d_13 (Conv2D)          (None, 7, 7, 128)        73856
max_pooling2d_8 (MaxPooling2 (None, 3, 3, 128)       0
dropout_2 (Dropout)         (None, 3, 3, 128)        0
flatten_1 (Flatten)         (None, 1152)              0
dense_2 (Dense)             (None, 256)               295168
dropout_3 (Dropout)         (None, 256)               0
dense_3 (Dense)             (None, 10)                2570
-----
Total params: 394,890
Trainable params: 394,890
Non-trainable params: 0
    
```

**Figure 4.** Architecture of the designed convolutional neural network model

### Naive Bayes Algorithm

Naive Bayes classification algorithm is an algorithm based on Bayes' theorem, that is, probability and using probability calculation. This algorithm works by calculating the probability of belonging to each class for each new data and classifying it according to the highest probability value [21]. Very efficient results can be achieved even with low

computation time. It can work well on unbalanced datasets. Bayes' theorem equation is given below.

$$P(A/B) = (P(B/A) \times P(A))/P(B) \quad (3)$$

The "naive\_bayes" module in the Scikit Learn library has been added for the Naive Bayes algorithm. There are 3 most important submodules that can be used for the Naive Bayes algorithm in the Scikit Learn library. These are: "GaussianNB", "MultinomialNB", "BernoulliNB". These modules are methods that can be used in different data sets and they work by using Gaussian, Multinomial, Bernoulli distribution methods. The ones that can be used in the MNIST dataset in this study are the "GaussianNB" and "MultinomialNB" methods. As a result of the tests, a higher accuracy rate was obtained with the "MultinomialNB" method. Finally, the training and testing process was carried out, the test results are as follows:

**Table 5.** Accuracy rate of Naive Bayes model

Model	Accuracy
Naive Bayes (MultinomialNB)	0.835

### Decision Trees

Decision trees are algorithms that are frequently used in classification problems, which divide a data set into smaller classes by putting certain decision rules and querying within the framework of these rules. It consists of three basic parts called node, branch and leaf. Each variable is defined as a node. They are easily understandable structures. It can be used for processing both categorical and numerical data. The most important disadvantage of decision trees is that the tree structure cannot be read and followed if very complex trees are produced. In addition to these, there is a possibility of over-fitting [22]. Important points in decision trees; how the division will take place, how the branching will take place, that is, how the tree structure will be created and according to which algorithm. There are many decision tree algorithms, but in order to achieve high accuracy, the most suitable algorithm for the data set and the problem should be selected [23].

In order to use decision trees, the "tree" module of the Scikit Learn library has been added to the project. For classification, the "DecisionTreeClassifier" submodule is included from this module. When creating a model with the "DecisionTreeClassifier" class, the "criterion" parameter must be specified. With this parameter, the tree forming criterion and the function that will create the division criterion of the data are determined. In this study, the "criterion" parameter was chosen as "Gini".

**Table 6.** Parameters applied to the Decision Tree model in this study

Parametres	Values applied in this model
Criterion	gini
Splitter	best

Class Weight	balanced
Maximum Depth	15
Minimum Samples Split	2
Minimum Samples Leaf	1
Minimum Weight Fraction	0.0
Leaf Decrease	0.0

Finally, the training and testing process was carried out, the test results are as in Table 7:

**Table 7.** Accuracy rate of Decision Tree model

Model	Accuracy
Decision Tree	0.8834

### Support Vector Machine – SVM

Support vector machines that can perform well on both linear and non-linear data are easy-to-implement algorithms. High success results can be obtained with support vector machines in many data sets [24].

Support vector machines are a method that creates support vectors based on existing class data and classifies new data accordingly. Support vector machines farthest removed from any two points dealt with in the dataset will allow to decide between the two classes is work to create a border [25]. Support Vector Machines are divided into two according to the linear separability and non-separability of the data set. One of the most important points in support vector machines is kernel functions. The appropriate kernel function should be selected according to whether the data set can be separated linearly or not [26]. These kernel functions greatly affect success. The most commonly used kernel functions are "linear", "rbf", "poly" and "sigmoid". While the "linear" kernel function is used for linearly separable data sets, "rbf" is used for non-linearly separable data sets.

The "SVC" (Support Vector Classification) module under the "svm" module from the Scikit Learn library for the support vector machine was included in the study. "poly", "rbf" and "linear" kernel functions were tested with the MNIST dataset. As a result of the test, in this study, the "poly" kernel function was preferred in accordance with the MNIST data set.

The results after the training and testing processes are as in Table 8:

**Table 8.** Comparison of the accuracy rates of the parameters applied to the SVM model and the kernel functions tested

Kernel Function	Gamma value	C value	Accuracy Rate
poly	scale	1	%97.71
linear	scale	1	%94.04
rbf	scale	1	%97.92

### K-Nearest Neighbor Algorithm – K-NN

K-NN is a supervised learning algorithm that provides learning by looking at the nearest neighbor data around the evaluated data. It is frequently preferred because it can be used in very wide areas, can provide high success results, and is easy to understand.

The two most important parameters affecting the success rate of the algorithm are the number of neighbors (k) and the distance criterion, which is the value of the distance to the neighbors [27]. The data is classified by looking at the nearest neighbors as many as the k value determined in the algorithm. The data selected as the nearest neighbor is selected according to the parameter determined as the distance criterion. The most common distance criterion; Minkowski, Euclidean and Manhattan distances are used [28].

The K-nearest neighbor algorithm is a powerful algorithm because it is simple and resistant to noisy training data. The disadvantage of this algorithm is that it needs very large memory space in large data set, as it stores all data and accounts.

**Table 9.** Variation of accuracy rate according to the parameters applied to the K-NN model and the k value

Number of neighbors (k)	Distance criterion	Accuracy rate
1	minkowski	%96.91
2	minkowski	%96.27
3	minkowski	%97.05
4	minkowski	%96.82

In this study, the k value, which represents the number of neighbors, was determined as 3 with the "n\_neighbors" parameter. The "metric" parameter, which is the distance criterion, was determined as "minkowski", that is, the Minkowski distance was used.

The results after training and testing are as follows:

**Table 10.** Accuracy rate of K-NN model

Model	Number of neighbors (k)	Distance criterion	Accuracy
K-NN	3	minkowski	0.9705

## 5. USER INTERFACE APPLICATION

In this project, an interface application was created for the end user by using the "PyQt5" library created for the graphical user interface of the Python programming language. The application has been developed with the aim of taking a handwritten number using the mouse as input data and classifying this data through previously trained and recorded models.

In the "Modeller" tab of the application, the parameter values of the trained algorithms used in the training are listed and presented to the user's information. Users click on the Draw

button, write a number in the drawing area window opened with the mouse and save the entry with the "Kaydet (Save)" button. If it is written incorrectly and it is desired to repeat

the drawing process, the drawing screen is returned to its original state by clicking the "Temizle (Clear)" button.

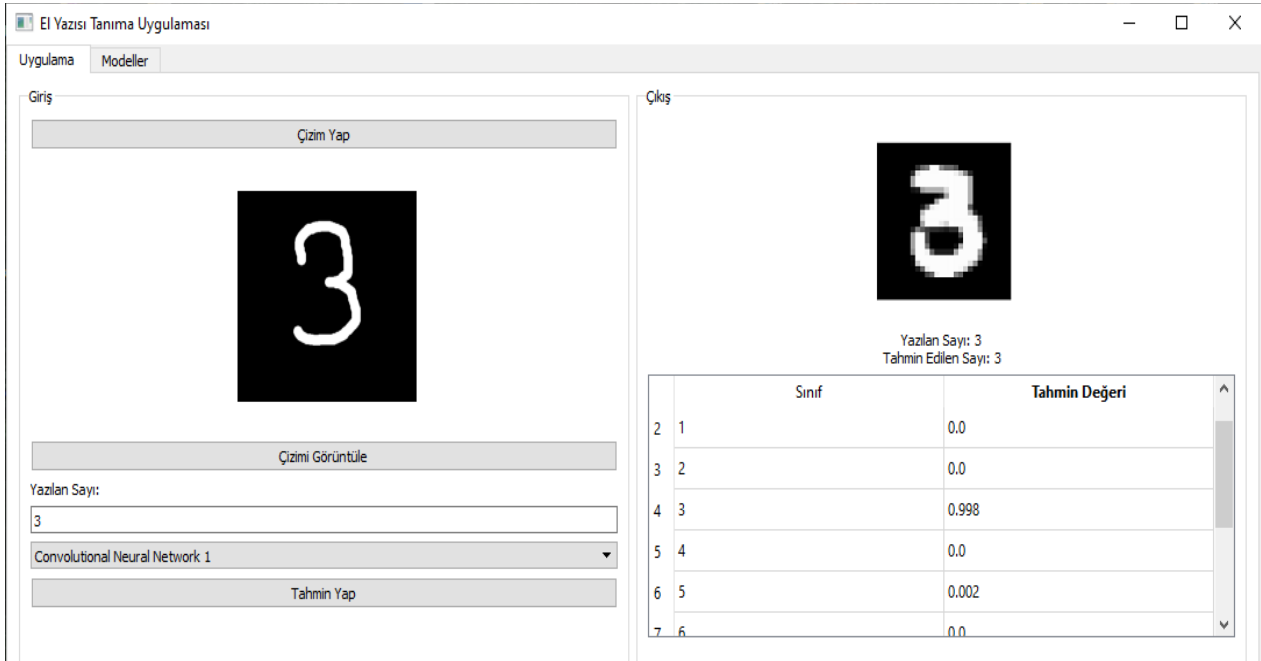


Figure 5. Interface implementation classification process result



Figure 6. Interface application drawing window

Then, with the "Çizimi Görüntüle (View Drawing)" button on the main screen, the input data is displayed in the main window. In the next step, users enter the "Yazılan Sayı (Actual Number):" information in the relevant place.

The users, among the algorithms previously described in this study; details of models trained using Convolutional Neural Network, Multiple-layer perceptron neural network, Naive Bayes, Support Vector Machine, K-Nearest Neighbors and Decision Tree algorithms can be accessed in the "Modeller" tab. Then, on the main screen, they select the model they want to be used in the classification process. The classification process is started with the "Tahmin Yap (Predict)" button. After the classification process is finished, "Sınıflandırma tamamlandı (Classification is complete)." information message is received.

Then, in the "Çıkış (Output)" pane, a representative data from the MNIST dataset is displayed as the classification result in the appropriate label. In the "Tahmin Edilen Sayı (Predicted Number):" section, the label of the classification result is shown. In the lower part, the results table and prediction rates are shown.

## 6. EXPERIMENT RESULTS

The results obtained from the test results are shown in Table 11. According to the results of the experiment performed with the MNIST data set in handwriting recognition processes, an accuracy rate of over 80% was obtained in all models. The highest accuracy rate is 99.45% and this rate was obtained with the CNN-Adam model. After the convolutional neural network models, the ANN-Adam model provided the highest accuracy rate with 98.66%. In addition, in the same data set, with the number of 3 neighbors, the K-NN model provided 97.05% accuracy and the SVM-poly model 97.71% accuracy. High success has been achieved in these models as well.

Table 11. Accuracy rate comparison of models

Model	Accuracy Rate
ANN-RMSprop	%98.50
ANN-Adam	%98.66
CNN-Adadelta	%86.44
CNN-Adam	%99.45
K-NN	%97.05
Naive Bayes	%83.57
SVM-poly	%97.71
Decision Tree	%88.34

Neural networks, which have not been studied much due to hardware deficiencies until recently, often show higher success than other classification algorithms with today's hardware.

Algorithm and method selection in handwriting recognition systems depends on many factors such as hardware, working environment, interactive or non-interactive systems. However, considering the developing technology and hardware, the use of neural networks in this field will provide high efficiency and accuracy.

In other classification algorithms, success rates can be increased by using different hyper-parameters and applying detailed data preprocessing. These algorithms are frequently preferred in handwriting recognition, show high success and are already used in many software in this field.

Comparison of the accuracy rates of the models created in similar studies in the literature with the accuracy rates of the models created in this study are given in Tables 12, 13, 14, 15.

**Table 12.** Comparison of the K-NN model with similar studies

Reference (Year)	Dataset	Accuracy Rate
M. A. G. Pirim (2017)	MNIST	%98.01
M. A. G. Pirim (2017)	USPS	%94.52
R. Karakaya (2020)	MNIST	%96
This Study	MNIST	%97.05

**Table 13.** Comparison of ANN-Adam model with similar studies

Reference (Year)	Dataset	Accuracy Rate
M. A. G. Pirim (2017)	MNIST	%98.56
M. A. G. Pirim (2017)	USPS	%94.18
A. Salouhou (2019) – Derin Sinir Ağı	MNIST	%99.53
A. Salouhou (2019) – Yinelemeli Sinir Ağı	MNIST	%99.05
R. Karakaya (2020)	MNIST	%97
This Study	MNIST	%98.66

**Table 14.** Comparison of the CNN-Adam model with similar studies

Reference (Year)	Dataset	Accuracy Rate
M. R. Shamsuddin ve ark. (2019)	MNIST	%99.4
M. R. Shamsuddin ve ark. (2019)	MNIST (binary)	%90.1
A. Salouhou (2019)	MNIST	%99.88
A. F. M. Agarap (2017) - CNN-Softmax	MNIST	%99.23
S. S. Mor ve ark. (2019)	EMNIST	%87.1
This Study	MNIST	%99.45

**Table 15** Comparison of the SVM-poly model with similar studies

Reference (Year)	Dataset	Accuracy Rate
M. A. G. Pirim (2017)	MNIST	%99.64
M. A. G. Pirim (2017)	USPS	%97.47
R. Karakaya (2020)	MNIST	%90
A. F. M. Agarap (2017) - CNN-SVM	MNIST	%99.04
This Study	MNIST	%97.71

In addition to these experiments, an end-user desktop interface application has been developed for the handwriting recognition system. The application was developed using the Python programming language and its interface library, the PyQt5 library. The application is designed to make handwritten digit prediction with all algorithms used in the experiments according to selection.

## 7. CONCLUSION AND EVALUATION

In this study, the achievements and performances of machine learning algorithms in handwriting recognition processes were examined. An interface has been developed for test studies by sampling recognition models with different algorithms. These algorithms were selected as ANN, CNN, K-NN, Naive Bayes algorithm, SVM and decision trees, all of them were discussed in detail and experiments were carried out on the MNIST dataset for each of them. During these experiments, Python programming language and accordingly; Keras and Tensorflow libraries for MNIST dataset, CNN and ANN structures; Scikit Learn library was used for test result reports of K-NN, Naive Bayes algorithm, SVM, decision trees and models. The result reports and accuracy value obtained using the Scikit Learn library

metrics were used to compare the success rates of the algorithms.

In the study, a total of 6 different algorithms were used. ANN and CNN models were the most successful when the models were compared in terms of accuracy. A success rate of 98.66% was achieved with the ANN-Adam model and 99.45% with the CNN-Adam model. Experimental results have shown that; Neural networks are more successful than other algorithms studied. However, neural network architecture and selected activation functions significantly affect the performance. In traditional classification algorithms, it is observed that the K-NN and SVM models can achieve a success rate of over 97%.


As a result almost all classification algorithms were examined with the MNIST data set, unlike similar studies in the literature. Experiments were carried out with many different parameters in all of them, and it was aimed to create the most efficient combination.


## REFERENCES


- [1] I. S. MacKenzie and K. Tanaka-Ishii, Text entry systems: mobility, accessibility, universality. San Francisco, Calif: Morgan Kaufmann, 2007. doi: 10.1016/B978-0-12-373591-1.X5000-1.
- [2] P. Duygulu, "El Yazısı Tanıma," in Bilişim Ansiklopedisi, Papatya Yayıncılık, 2006.
- [3] M. R. Shamsuddin, S. Abdul-Rahman, and A. Mohamed, "Exploratory Analysis of MNIST Handwritten Digit for Machine Learning Modelling," Communications in Computer and Information Science, vol. 937, pp. 134–145, 2019, doi: 10.1007/978-981-13-3441-2\_11.
- [4] A. F. M. Agarap, "An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification," arXiv, pp. 5–8, 2019.
- [5] M. A. Günler Pirim, "Neural Network Based Feature Extraction for Handwriting Digit Recognition," Ankara, 2017.
- [6] A. El-Sawy, M. Loey, and H. El-Bakry, "Arabic Handwritten Characters Recognition using Convolutional Neural Network," WSEAS Transactions on Computer Research, vol. 5, pp. 11–19, 2017.
- [7] A. Salouhou, "Deep Learning Approaches in Handwriting Character Recognition and Image Classification," Istanbul, 2019.
- [8] R. Karakaya, "Makine Öğrenmesi Yöntemleriyle El Yazısı Tanıma," Sakarya, 2020.
- [9] S. S. Mor, S. Solanki, S. Gupta, S. Dhingra, M. Jain, and R. Saxena, "Handwritten Text Recognition: With Deep Learning and Android," International Journal of Engineering and Advanced Technology, vol. 8, no. 2, pp. 172–178, 2019.
- [10] "THE MNIST DATABASE of handwritten digits." url: <http://yann.lecun.com/exdb/mnist/> (accessed Nov. 08, 2020).
- [11] J. M. Banda, R. A. Angryk, and P. C. Martens, "Steps Toward a Large-Scale Solar Image Data Analysis to Differentiate Solar Phenomena," Solar Physics, vol. 288, no. 1, pp. 435–462, 2013, doi: 10.1007/s11207-013-0304-x.
- [12] "Scikit-Learn." url: [https://scikit-learn.org/stable/modules/model\\_evaluation.html#precision-recall-f-measure-metrics](https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics) (accessed Dec. 20, 2020).
- [13] "Thresholding Process." url: <http://www.atasoyweb.net/Otsu-Esik-Belirleme-Metodu> (accessed Dec. 02, 2020).
- [14] A. Vinciarelli and J. Luetttin, "A New Normalization Technique for Cursive Handwritten Words," Pattern Recognition Letters, vol. 22, no. 9, pp. 1043–1050, 2001, doi: 10.1016/S0167-8655(01)00042-3.
- [15] B. Yılmaz, "Design of A Mobile Device Application with Handwriting Recognition to Make Learning Easy For Students Who Have Learning Disabilities," Istanbul, 2014.
- [16] H. H. Çelik, "Recognition of Handwritten Numerals by Using Neural Network," Istanbul, 1999.
- [17] O. A. Erdem and E. Uzun, "Turkish Times New Roman, Arial, And Handwriting Characters Recognition by Neural Network," journal of the Faculty of Engineering and Architecture of Gazi University, vol. 20, no. 1, pp. 13–19, 2005.
- [18] H. A. Şahin, "Comparison of Artificial Neural Networks and Different Optimization Methods," Samsun, 2020.
- [19] E. Öztemel, Yapay Sinir Ağları. İstanbul: Papatya Yayıncılık, 2012. [Online]. Available: [http://papatyabilim.com.tr/PDF/yapay\\_sinir\\_aglari.pdf](http://papatyabilim.com.tr/PDF/yapay_sinir_aglari.pdf)
- [20] B. Ma, X. Li, Y. Xia, and Y. Zhang, "Autonomous deep learning: A genetic DCNN designer for image classification," Neurocomputing, vol. 379, pp. 152–161, 2020, doi: 10.1016/j.neucom.2019.10.007.
- [21] E. Yalçın, "Binary-Data Multi-Criteria Recommender Systems Based on Naive Bayes Classifier," Eskişehir, 2016.
- [22] M. W. Berry and M. Browne, Eds., Lecture Notes In Data Mining. World Scientific, 2006.
- [23] F. Köktürk, "Comparing Classification Success of K-Nearest Neighbor, Artificial Neural Network and Decision Trees," Zonguldak, 2012.
- [24] N. Turdaliev, "Destek Vektör Makineleri ile Otel Öneri Sistemi," 2018.
- [25] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge University Press, 2008. Accessed: Feb. 10, 2021. [Online]. Available: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>
- [26] E. Dağdeviren, "El Yazısı Rakam Tanıma İçin Destek Vektör Makinelerinin ve Yapay Sinir Ağlarının Karşılaştırması," 2013.
- [27] G. Sakkis, "A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists."
- [28] E. Taşçı and A. Onan, "The Investigation of Performance Effects of K-Nearest Neighbor Algorithm Parameters on Classification," in XVIII. Akademi Bilişim Konferansı, 2016, p. 8.

# Determination of Optimum Pinch Point Temperature Difference Depending on Heat Source Temperature and Organic Fluid with Genetic Algorithm

\*<sup>1</sup>Sadık Ata, <sup>2</sup>Ali Kahraman, <sup>3</sup>Remzi Şahin

<sup>1</sup> KTO Karatay University, Faculty of Engineering and Natural Sciences, Department of Mechanical Engineering, sadik.ata@karatay.edu.tr, 

<sup>2</sup>Necmettin Erbakan University, Faculty of Engineering, Department of Mechanical Engineering, akahraman@erbakan.edu.tr, 

<sup>3</sup>KTO Karatay University, Faculty of Engineering and Natural Sciences, Department of Mechanical Engineering, remzi.sahin@karatay.edu.tr, 

## Abstract

In this study, the effect of evaporator pinch point temperature difference ( $\Delta T_{PP,e}$ ) value in Organic Rankine Cycle (ORC) on system performance was determined. Under different applications of ORC, optimum  $\Delta T_{PP,e}$  value has been determined in ORC systems designed with different heat source temperatures. By changing the  $\Delta T_{PP,e}$  value, the heat input provided to the system, the mass flow of organic fluid, the evaporation pressure and the enthalpy drop in the turbine are affected. In thermodynamic optimization, the objective function is determined as turbine power maximization. Genetic algorithm optimization technique is used. Within the scope of low and high temperature ORC applications, the optimum  $\Delta T_{PP,e}$  value of different organic fluids under 10 different heat source temperatures (Low, 90-130 °C; High, 250-290 °C) has been determined. Low temperature organic fluids have been selected from dry, isentropic, wet and new-generation categories. High temperature organic fluids have been selected from the alkane, aromatic hydrocarbon, and siloxane categories. The effect of  $\Delta T_{PP,e}$  on fluids of different categories was determined for low and high temperature ORCs. It has been determined that taking the  $\Delta T_{PP,e}$  value constant regardless of the heat source temperature and organic fluid causes performance loss in ORC.

**Keywords:** Genetic Algorithm, Low-High Organic Fluids Optimum, Pinch Point, Organic Rankine Cycle, Thermodynamic Optimization

## 1. INTRODUCTION

The Organic Rankine Cycle (ORC) works like the Rankine cycle as its working principle, the difference is that an organic fluid other than water is used. The fluid used in ORC has a lower boiling point and a higher vapor pressure than water and can therefore be used in low temperature heat sources to generate electricity. The organic fluid is selected to best match the heat source according to its different thermodynamic properties, resulting in higher efficiency of both the process and the expander.

In this study, the performance of organic fluids was determined depending on the heat source temperature under low and high temperature applications of ORC. The optimum evaporator pinch point temperature difference ( $\Delta T_{PP,e}$ ) was determined for each heat source temperature.  $\Delta T_{PP,e}$ ; It is defined as the difference between the evaporator pinch point temperature ( $T_{P,e}$ ) and the evaporation temperature of the organic fluid. It has been observed that this value ( $\Delta T_{PP,e}$ ), which was taken as a constant in most of

the previous studies, seriously affects the ORC performance. Important studies on this subject are summarized.

Wu et al. [1] conducted a study on the determination of  $\Delta T_{PP,e}$  and  $\Delta T_{PP,c}$  in ORC designed using mixing fluids. They considered exergo - economic performance, which is the ratio of annual total cost to net power, as an evaluation criterion. They stated that the increase of  $\Delta T_{PP,e}$  rapidly increased exergo economic performance, but reached the best performance at optimum  $\Delta T_{PP,e}$  value. They concluded that the optimum  $\Delta T_{PP,e}$  for mixing fluids should be between 3-6 °C.

Yu et al. [2] developed a method that can instantly determine the organic fluid and working conditions in ORC depending on the  $\Delta T_{PP,e}$ . They defined the  $\Delta T_{PP,e}$  formed in the preheater and the  $\Delta T_{PP,e}$  formed in the evaporator for this aim. They determined that the maximum power is reached when there is a suitable difference between the heat source inlet temperature and the critical temperature of the fluid, and the fluid evaporates near the critical region.

\* Corresponding Author



Liu et al. [3] performed a performance analysis for geothermal different heat source temperatures in the ORC system they designed using R245fa. The effect of  $\Delta T_{PP,e}$  on system performance has been determined. Net power, turbine size parameter, volume flow rate and total thermal conductivity were calculated. It has been determined that  $\Delta T_{PP,e}$  is inversely proportional to total thermal conductivity and net power. It has been stated that the optimum  $\Delta T_{PP,e}$  is associated with the heat source inlet temperature, and low  $\Delta T_{PP,e}$  provides high net power. As a result of the change of heat source inlet temperature between 80-180 °C, it has been determined that  $\Delta T_{PP,e}$  increased from 2 °C to 21 °C.

Kaşka et al. [4] conducted a study on the energy and exergy analysis of the Organic Rankine-Brayton combined cycle. They found that it is important to determine the optimum  $\Delta T_{PP,e}$  temperature in heat exchangers where heat source and work fluid heat transfer occurs in ORC design. They stated that while the heat transfer to the evaporator increases linearly with the increase of the  $\Delta T_{PP,e}$  value, the thermal efficiency of the ORC decreases, but depending on the  $\Delta T_{PP,e}$  value, the net power produced by the ORC is the optimum point.

Sun et al. [5] examined the effect of  $\Delta T_{PP,e}$  on thermodynamic performance within the scope of geothermal ORC applications. They stated that  $\Delta T_{PP,e}$  is an important parameter for thermodynamic and economic performance. They have determined that low  $\Delta T_{PP,e}$  will provide more turbine net power but have a negative effect on the economy as it will increase the heat transfer area. For heat source applications higher than 130 °C, it has been determined that ORC produces 1.7-2.6% more power with every 1 °C decrease in  $\Delta T_{PP,e}$ .

Bademlioglu et al. [6] studied the effect of  $\Delta T_{PP,e}$  on exergy performance in ORC. The effect of changing  $\Delta T_{PP,e}$  between 5-20 °C on systems prepared using different organic fluids has been determined. They stated that depending on the  $\Delta T_{PP,e}$  and the organic fluid, the irreversibility in the evaporator can be reduced by 62.32%.

Wang et al. [7] have worked on  $\Delta T_{PP,e}$  optimization using the Analytical Hierarchy Process (AHP) - Entropy method in ORC systems. As a result of the study, they stated that they reached the maximum power output with R141b and the maximum thermal efficiency and exergy efficiency values with R11.

Sarkar [8] worked on  $\Delta T_{PP,e}$  design and optimization for maximum heat recovery in ORC. He developed a method that can determine  $\Delta T_{PP,e}$  and  $\Delta T_{PP,c}$  instantaneously. Best results have been achieved in ammonia fluid in terms of low mass flow requirement, high exergy efficiency and low turbine size at optimum points. In terms of high-power output and heat recovery efficiency, it performed better in isopentane fluid.

Jankowski et al. [9] determined the optimum  $\Delta T_{PP,e}$  value in ORC systems using the multi-objective approach technique. They worked on two objective functions: economy and environment. At the end of their studies, they reached the optimum  $\Delta T_{PP,e}$  between 7-10 °C by using R245fa fluid.

Imran et al. [10] conducted an optimization study by aiming thermal efficiency maximization and unit investment cost minimization with NSGA-II method. Evaporation pressure, superheating temperature and  $\Delta T_{PP,e} - \Delta T_{PP,c}$  values were chosen as design parameters.

In the section below, the differences of the number of objective functions in optimization with GA are examined. In some studies, the objective function was determined through a single parameter in GA optimization. The objective functions; Bian et al. [11] determined the heat transfer area as the ratio of the total net power output, and Long et al. [12] decided the total exergy efficiency. Gutierrez et al. [13] accepted gross annual profit as an objective function, Han et al. [14] as a total irreversibility loss, Pierobon et al. [15] as a thermal efficiency, Agromayor et al. [16] as a second law efficiency. Finally, Andreasen et al. [17], Fiaschi et al. [18] and Kai et al. [19] used the net power as the objective function and studied both the optimum fluid selection and the thermodynamic optimization of the system with GA.

In this study, thermodynamic optimization has been made in order to find the optimum  $\Delta T_{PP,e}$  point for ORC designed using different fluids. As can be seen from the literature studies, it is stated that the maximum turbine power is not obtained due to the absorption of heat in the evaporator at the point where the thermal efficiency reaches its maximum. It has been determined that while the thermal efficiency decreases with the increase of  $\Delta T_{PP,e}$  value, the turbine power is not in the same trend. It was observed that the turbine power of the system started to decrease after a certain  $\Delta T_{PP,e}$  value. With the change of  $\Delta T_{PP,e}$ , the heat input required to be provided to the system increased, however, the mass flow rate of the organic fluid increased. But at the same time, with the change of  $\Delta T_{PP,e}$ , the evaporation pressure decreased and the enthalpy difference in the turbine decreased. It has been determined that the turbine power of the system starts to decrease at the point where the decrease in the enthalpy difference is more than the increase in ORC mass flow rate.

Therefore, it was observed that the optimum  $\Delta T_{PP,e}$  point depends on the organic fluid and the heat source temperature. In the studies, it was determined that taking a constant  $\Delta T_{PP,e}$  value caused a certain amount of error in the analysis results. In this study, the optimum  $\Delta T_{PP,e}$  point of organic fluids in different categories at different heat source temperatures under various ORC applications was determined. These applications; geothermal, low temperature solar, waste heat and biomass-high temperature solar. Organic fluids have been selected for low temperature ORC from dry, isentropic, wet and new-generation organic fluids. In high temperature ORC, fluids have been chosen from among alkanes, aromatic hydrocarbons and siloxanes.

For low temperature ORC;

- Geothermal Energy Applications ( $T_{h,i} = 90, 100, 110$  °C)
- Low Temperature Solar Energy Applications ( $T_{h,i} = 120, 130$  °C)

For high temperature ORC;

- Waste Heat Applications ( $T_{h,i} = 250, 260, 270$  °C)

- Biomass and High Temperature Solar Energy Applications ( $T_{h,i} = 280, 290 \text{ }^\circ\text{C}$ )

Li [20], in his review study, examined the organic fluid performance under different application areas (geothermal, low temperature solar, waste heat and biomass-high temperature solar) of ORC according to the heat source temperatures.

By using the temperature values determined for these applications, the effect of optimum pinch point temperature on turbine power maximization on different fluids has been determined.

In previous studies, it was observed that the  $\Delta T_{PP,e}$  value was taken as constant. However, the optimum  $\Delta T_{PP,e}$  value changes depending on the heat source temperature and the organic fluid. Based on these two factors, it is aimed to make an optimization study by determining the turbine power maximization purpose under the optimum  $\Delta T_{PP,e}$ . By determining the optimum  $\Delta T_{PP,e}$  points for different applications of ORC, it is aimed to reach higher system performances in thermodynamic analysis, modeling and optimization studies conducted by the researchers.

## 2. MATERIALS AND METHODS

### 2.1. Thermodynamic Analysis

Engineering Equation Solver (EES) was used for thermodynamic analysis and optimization of ORC. Energy

and mass equations for  $\Delta T_{PP,e}$  is introduced to EES, boundary conditions are entered for optimization using EES and genetic algorithm interface.

Table 1 and Table 2 summarizes the thermophysical and safety-environmental properties of fluids for low and high temperature ORC fluids respectively. The thermophysical properties of the fluid are taken from the "ASHRAE Standard 34" table. [21].

General definitions and equations (1-4) for the system are given below.

**Mass balance** (Total Mass Input = Total Mass Output);

$$\sum \dot{m}_{input} = \sum \dot{m}_{out} \quad (1)$$

**Energy balance** (Total Energy Input = Total Energy Output);

$$\sum E_{input} = \sum E_{out} \quad (2)$$

$$\dot{Q} - \dot{W} = \dot{m} * (h_{in} - h_{out}) \quad (3)$$

**Exergy balance** (Total Exergy input = Final Exergy + Exergy Consumption + Exergy Destruction);

$$\dot{E}x_{in} = \dot{E}x_f + \dot{E}x_c + \dot{E}x_d \quad (4)$$

**Table 1.** Thermophysical and safety-environmental properties of fluids for low-temperature ORC.

Fluids	R601	R601a	R141b	R123	R152a	R134a	R1234yf	R1234ze
Type	Dry		Isentropic		Wet		New-Generations	
Molecular mass (g/mol)	72.15	72.15	116.95	152.93	66.05	102	114.04	114.04
Normal Boiling Points ( $^\circ\text{C}$ )	36.1	27,8	32	27,8	-24	-26.1	-29.3	-18.8
Critical Temperature ( $^\circ\text{C}$ )	196.6	187.2	204.4	183.7	113.3	101.1	94.85	109.52
Critical Pressure (MPa)	3.37	3.38	4.21	3.66	4.52	4.06	3.38	3.63
ASHRAE 34 safety group	A3	A3	n.a	B1	A2	A1	*A2L	*A2L
ODP	0	0	0.12	0	0	0	0	0
GWP	20	20	725	77	124	1430	4	6

**Table 2.** Thermophysical and safety-environmental properties of fluids for high-temperature ORC.

Fluids	n-octane	cyclohexane	benzene	toluene	MM	D4
Type	Alkanes		Aromatic Hydrocarbons		Siloxanes	
Molecular mass (g/mol)	114.23	84.161	78.108	92.138	162.4	296.6
Normal Boiling Points ( $^\circ\text{C}$ )	125	80	80	110	100.4	175
Critical Temperature ( $^\circ\text{C}$ )	296	280	289	319	245	312
Critical Pressure (MPa)	2.49	4.075	4.89	4.12	1.91	1.33
ASHRAE 34 safety group	n.a	A3	B2	A3	n.a	n.a
ODP	n.a	0	0	0	n.a	n.a
GWP	n.a	low	low	2.7	n.a	n.a

\*A2L; low toxicity and mildly flammable

In the energy analysis of the components in the system, the equations used for pump work (5), evaporator heat input (6), turbine work (7), the amount of heat discharged from the condenser (8) are given below (Isentropic efficiencies of turbine and pump,  $\eta_t$  and  $\eta_p$ , respectively).

$$W_p = (h_2 - h_1) = (h_{2s} - h_1)/\eta_p \quad (5)$$

$$Q_e = (h_3 - h_2) \quad (6)$$

$$W_t = (h_3 - h_4) = (h_3 - h_{4s})\eta_t \quad (7)$$

$$Q_c = (h_4 - h_1) \quad (8)$$

The equations used for net work (9) and thermal efficiency (10) in the system are given below.

$$W_{net} = W_t - W_p \quad (9)$$

$$\eta_{th} = W_{net}/Q_e \quad (10)$$

The irreversibility equations used for the pump (11), evaporator (12), turbine (13) and condenser (14) in the exergy analysis of the components in the system are given below. The average temperatures of the heat source and cooling water are given in Equation 15-16.

$$i_p = T_0(s_2 - s_1) \quad (11)$$

$$i_e = T_0[(s_3 - s_2) - (h_3 - h_2)/T_h] \quad (12)$$

$$i_t = T_0(s_4 - s_3) \quad (13)$$

$$i_c = T_0[(s_1 - s_4) + (h_4 - h_1)/T_c] \quad (14)$$

$$T_h = (T_{h,i} - T_{h,o})/\ln(T_{h,i} - T_{h,o}) \quad (15)$$

$$T_c = (T_{c,i} - T_{c,o})/\ln(T_{c,i} - T_{c,o}) \quad (16)$$

The equations used for total irreversibility (17), consumed exergy (18) and exergy efficiency (19) in the system are given below.

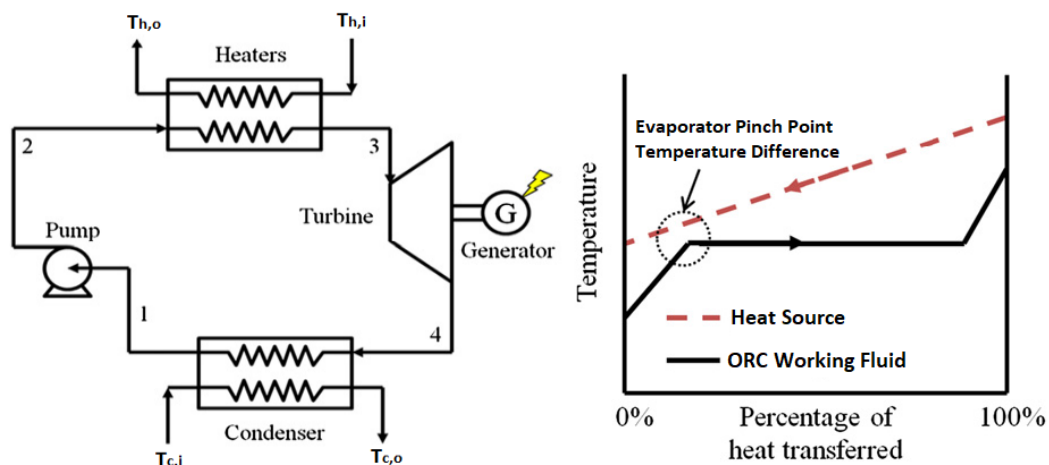
$$i_{Total} = i_p + i_e + i_t + i_c \quad (17)$$

$$e_{consumed} = [1 - T_0/T_h]Q_e + W_p \quad (18)$$

$$\eta_{II} = 1 - i_{Total}/e_{consumed} \quad (19)$$

The working principle of ORC and the demonstration of  $\Delta T_{PP,e}$  is given in Figure 1. The evaporator and condenser energy balance relations (Eq.20-26) are given below. The explanations of the symbols in these equations are given below.

- $T_{p,e}$ : Evaporator pinch point temperature
- $T_{3,f}$ : Evaporation temperature
- $\Delta T_{PP,e}$ : Evaporator pinch point temperature difference
- $T_{p,c}$ : Condenser pinch point temperature;
- $T_{1,g}$ : Condensation temperature
- $\Delta T_{PP,c}$ : Condenser pinch point temperature difference



**Figure 1.** ORC Working Principle and Demonstration of evaporator pinch point temperature difference ( $\Delta T_{PP,e}$ ) [22]

Evaporator energy balance

$$\dot{m}_{ORC} * (h_3 - h_2) = \dot{m}_h * Cp * (T_{h,i} - T_{h,o}) \quad (20)$$

$$\dot{m}_{ORC} * (h_3 - h_{3,f}) = \dot{m}_h * Cp * (T_{h,i} - T_{p,e}) \quad (21)$$

$$\Delta T_{PP,e} = (T_{p,e} - T_{3,f}) \quad (22)$$

Evaporator effectiveness ( $\varepsilon$ )

$$\varepsilon = \frac{Q}{Q_{max}} = \frac{\dot{m}_h * Cp * (T_{h,i} - T_{h,o})}{\dot{m}_h * Cp * (T_{h,i} - T_2)} = \frac{(T_{h,i} - T_{h,o})}{(T_{h,i} - T_2)} \quad (23)$$

Condenser energy balance

$$\dot{m}_{ORC} * (h_{4a} - h_1) = \dot{m}_c * Cp * (T_{c,o} - T_{c,i}) \quad (24)$$

$$\dot{m}_{ORC} * (h_{1,g} - h_1) = \dot{m}_c * Cp * (T_{p,c} - T_{c,i}) \quad (25)$$

$$\Delta T_{pp,c} = (T_{1,g} - T_{p,c}) \quad (26)$$

For the thermodynamic analysis of ORC, the following assumptions are employed.

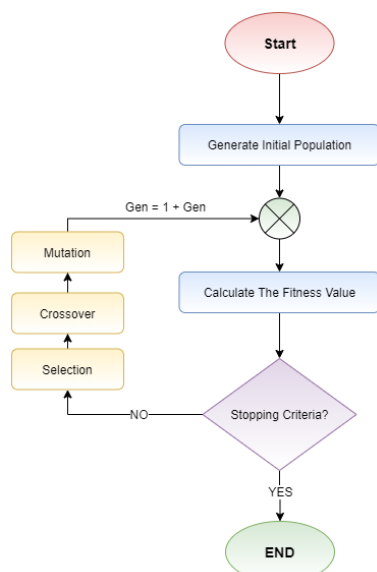
- All processes are under steady state.
- Pressure losses in the evaporator and condenser are neglected. Losses in pipelines are neglected.
- In the analysis, all equipment is considered adiabatic and it is assumed that there is no heat transfer between its surfaces and the environment.
- Potential and kinetic energy changes have been neglected.
- Low-temperature ORC heat source temperatures: 90, 100, 110, 120 and 130 °C
- High-temperature ORC heat source temperatures: 250, 260, 270, 280 and 290 °C
- Heat source mass flow rate is 0.27 kg/s.
- Isentropic efficiency of the turbine and the pump are 75%.
- Evaporator effectiveness is 75%
- Cooling water inlet temperature ( $T_{c,i}$ ) 27 °C.
- Dead point pressure and temperature, respectively,  $P_0$ : 100 kPa and  $T_0$ : 25 °C

## 2.2. Thermodynamic Optimization with GA

In this study, the effect of  $\Delta T_{pp,e}$  on ORC was determined by Genetic Algorithm (GA). Tournament selection method was used for the optimization of the simple ORC with the genetic algorithm. Control parameters for optimization are shown in below. Flow diagram of GA's working principle is shown in Figure 2.

Control parameters of GA for the optimization:

- Population size is 65.
- Maximum generations are 256.
- Crossover probability is 0.7.
- Mutation probability is 0.175.
- Selection process is "Tournament".



**Figure 2.** Flow chart of the genetic algorithms.

Thermodynamic optimization is performed using genetic algorithm. The lowest turbine power in the system is 1 kW; the highest turbine power has been set as 10 kW and 50 kW for low and high temperature ORC respectively. The primary working conditions are selected as decision variables which include evaporating pressure ( $P_{eva}$ ),  $\Delta T_{pp,e}$ ,  $\Delta T_{pp,c}$  and superheating temperature ( $T_{sup}$ ). Since organic fluids in different fluid categories are used in the design, the limit values for evaporation pressure have been determined at different ranges. In this way, better results were obtained in optimization. Table 3 summarizes the logical bounds for four decision variables for low-high temperature ORC respectively.

Based on the energy balance and the definition of evaporator and condenser pinch point temperature difference, other following constraints are considered in the optimization. Thermodynamic optimization was applied separately for 3 different heat source temperatures. Therefore, the limitations that should be related to the heat source temperature are also specified.

- $1 \text{ kW} < W_T < 10 \text{ kW}$  (for low-temperature ORC)
- $1 \text{ kW} < W_T < 50 \text{ kW}$  (for high-temperature ORC)
- $T_{eva} + \Delta T_{pp,e} < T_{h,i}$
- $T_{eva} + \Delta T_{pp,e} < T_{critical}$
- $T_{eva} + T_{sup} < T_{h,i}$
- $T_{c,i} + \Delta T_{pp,c} < T_{con}$
- $T_{eva,min} : 70 \text{ °C}$

By changing the  $\Delta T_{pp,e}$  value, the heat input provided to the system, the mass flow of organic fluid, the evaporation pressure and the enthalpy drop in the turbine are affected. Four important parameters are affected by the change of  $\Delta T_{pp,e}$  value in ORC system. These are; the heat input provided to the system, the mass flow of organic fluid, the evaporation pressure and the enthalpy drop in the turbine. It has been determined that the turbine power of the system starts to decrease at the point where the decrease in the enthalpy difference is more than the increase in ORC mass flow rate. Therefore, the objective function in this study was determined as turbine power maximization.

Objective Function;

- $f(x)$ : max (WT); Turbine power maximization

where  $x = \{P_{eva}, \Delta T_{pp,e}, \Delta T_{pp,c}, T_{sup}\}$  subjected to lower bound  $< x <$  upper bound.

## 3. MODEL VALIDATION

In order to determine the accuracy of the data obtained using GA, two studies investigated within the scope of literature research were used. The net power values determined by using three different organic fluids under the same design parameters were compared for two different studies in Table 4. When Table 4 is examined, it is seen that the thermodynamic model prepared can be used successfully

**Table 3.** Logical bounds for four decision variables for low and high temperature ORC.

Low-temperature ORC				
Organic Fluids	Evaporating Pressure ( $P_{eva}$ ) (kPa)	$\Delta T_{PP,e}$ (°C)	$\Delta T_{PP,c}$ (°C)	$T_{sup}$ (°C)
R601	$260 < P_{eva} < 410$	$1 < \Delta T_{PP,e} < 15$	$1 < \Delta T_{PP,c} < 10$	$0 < T_{sup} < 20$
R601a	$330 < P_{eva} < 510$			
R141b	$300 < P_{eva} < 470$			
R123	$350 < P_{eva} < 550$			
R152a	$1840 < P_{eva} < 4250$			
R134a	$2100 < P_{eva} < 3900$			
R1234yf	$2000 < P_{eva} < 3300$			
R1234ze	$1600 < P_{eva} < 3410$			
High-temperature ORC				
Organic Fluids	Evaporating Pressure ( $P_{eva}$ ) (kPa)	$\Delta T_{PP,e}$ (°C)	$\Delta T_{PP,c}$ (°C)	$T_{sup}$ (°C)
n-octane	$200 < P_{eva} < 400$	$1 < \Delta T_{PP,e} < 40$	$1 < \Delta T_{PP,c} < 10$	$0 < T_{sup} < 20$
cyclohexane	$590 < P_{eva} < 1150$			
benzene	$550 < P_{eva} < 1100$			
toluene	$270 < P_{eva} < 480$			
D4	$50 < P_{eva} < 130$			
MM	$460 < P_{eva} < 1270$			

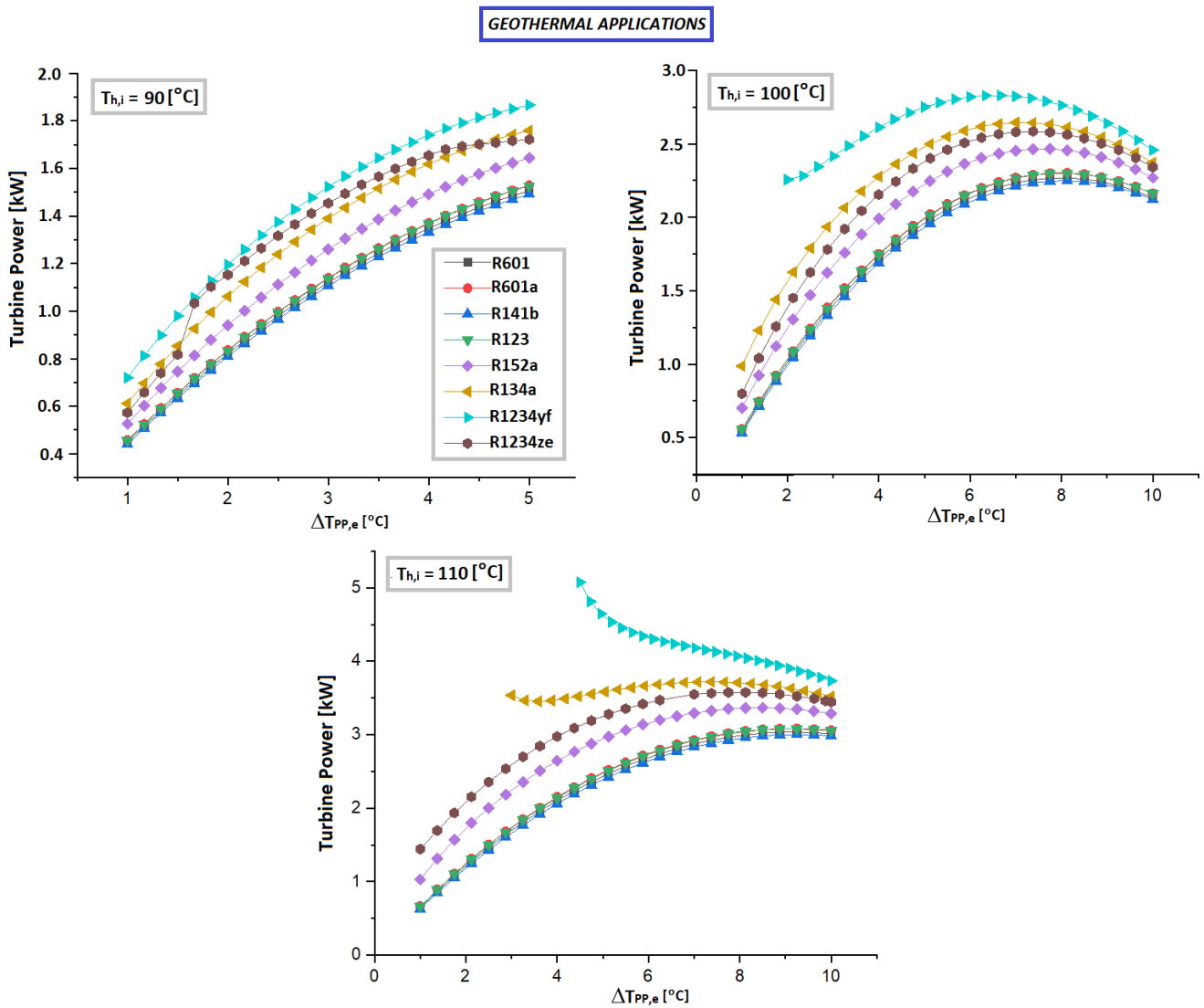
**Table 4.** Comparison of important optimization results with literature under same design parameters (GA).

Design Parameters	Heat Source Temperature: 150 °C; Heat Sink Temperature: 20 °C; $\Delta T_{PP,e} + \Delta T_{PP,c} = 20$ °C Turbine and pump isentropic efficiency: 85% and 80%				Evaporation Temperature: 80 °C $\Delta T_{PP,e} = 8$ °C Turbine and pump isentropic efficiency: 80% and 70%	
Organic Fluids	R113		R11		R245fa	
Performance Parameters	Present Study	Literature [9]	Present Study	Literature [9]	Present Study	Literature [11]
Net Power (kW)	73.12	73.91	70.24	70.93	50.2	51.0

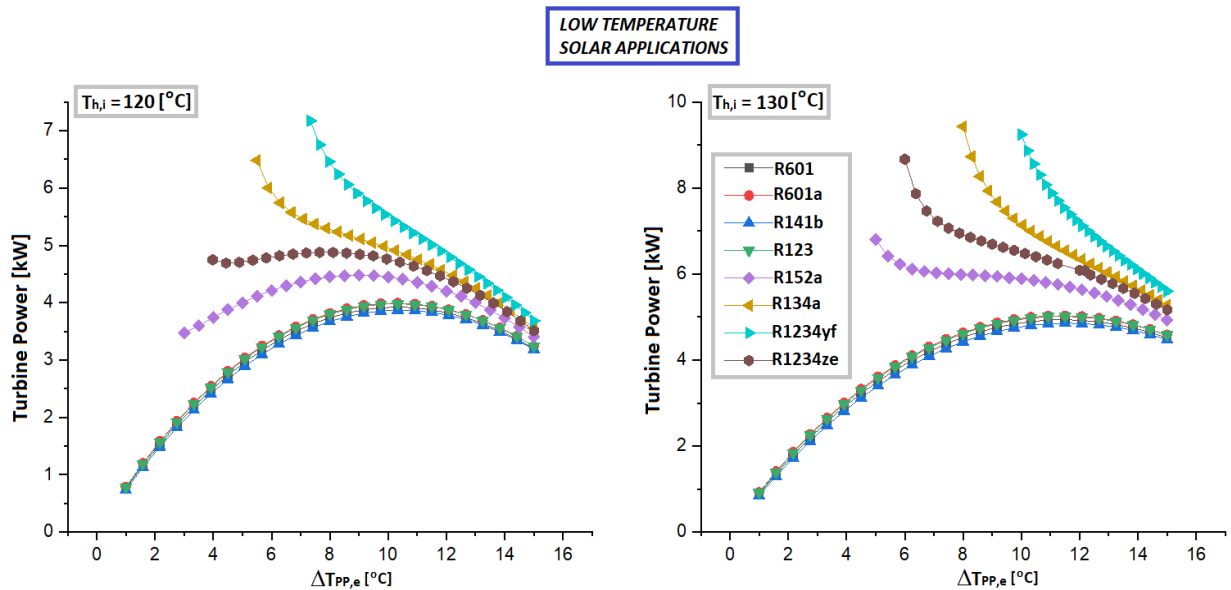
#### 4. RESULT AND DISCUSSION

Figures 3 and 4 show the effect of  $\Delta T_{PP,e}$  change on turbine power in geothermal and low temperature solar energy applications of ORC, respectively. When GA optimization results are evaluated for low-temperature ORC;

- It has been determined that the turbine power decreases at the point where the enthalpy difference decrease is more than the mass flow increase at the other heat source temperatures except 90 °C. Net power increased as  $\Delta T_{PP,e}$  increased, since mass flow rate increase was greater than enthalpy difference decreases at 90 °C.
- It was observed that the allowable  $\Delta T_{PP,e}$  value according to the minimum evaporator temperature under 90 °C heat source temperature is 5 °C maximum.
- In low-temperature applications of ORC, the highest turbine power has been reached in the system with R1234yf. Also, ORC systems with R1234ze at 90 °C and R134a at 100 and 110 °C performed better.
- While  $\Delta T_{PP,e}$ 's effect on turbine power tends to be similar in dry and isentropic fluids, it is very different in wet and new-generation fluids.
- It is seen that the turbine power starts to decrease after a certain  $\Delta T_{PP,e}$  value at all heat source temperatures except 90 °C heat source temperature.
- Especially in low temperature solar energy applications, for wet fluid and new-generation organic fluids, the effect of  $\Delta T_{PP,e}$  on turbine power is different than other fluids.
- In dry and isentropic fluids, low turbine power was obtained at low  $\Delta T_{PP,e}$  values. As  $\Delta T_{PP,e}$  increased, the turbine power value increased and decreased after a certain value due to the ORC mass flow rate and enthalpy drop in turbine.
- However, in wet fluid and new-generation organic fluids, a high turbine power value was achieved at the minimum  $\Delta T_{PP,e}$  value allowed by the optimization limit values and it was observed that the turbine power remained at the same rate or started to decrease directly as  $\Delta T_{PP,e}$  increased.
- Due to the low critical temperature of wet and new-generation fluids, the minimum  $\Delta T_{PP,e}$  point increased as the heat source temperature increased.



**Figure 3.** Effect of  $\Delta T_{PP,e}$  change on turbine power for 90, 100 and 110 °C heat source temperatures in ORC's geothermal applications.



**Figure 4.** Effect of  $\Delta T_{PP,e}$  change on turbine power for 120 and 130 °C heat source temperatures in ORC's low temperature solar applications.



The optimum  $\Delta T_{PP,e}$  points where the maximum turbine power is obtained under 5 different heat source temperatures of 8 different fluids are summarized in Table 5. It is noteworthy that the  $\Delta T_{PP,e}$  value is the same in all fluids at

90 °C heat source temperature. In addition, it has been determined that dry and isentropic fluids have the same  $\Delta T_{PP,e}$  value at other temperatures.

**Table 5.** Determination of optimum  $\Delta T_{PP,e}$  value for different fluids under different heat source temperatures for low temperature ORC applications.

$T_{h,i}$	Optimum $\Delta T_{PP,e}$ (°C)							
	R601a	R601	R141b	R123	R152a	R134a	R1234yf	R1234ze
90 °C	5							
100 °C	8.125				7.75	7	6.67	7.38
110 °C	9.25				8.5	7.375	4.5	8.125
120 °C	10.33				9	5.5	7.33	3.98
130 °C	11.5				5	8	10	6

Figures 5 and 6 show the effect of  $\Delta T_{PP,e}$  change on turbine power in waste heat and biomass-high temperature solar energy applications of ORC, respectively. When GA optimization results are evaluated for high-temperature ORC;

- In high temperature applications of ORC, the highest turbine power was achieved in the siloxanes group.
- The highest turbine power has been reached in the system with MM. It has been observed that benzene and toluene, which are aromatic hydrocarbons, perform worse than other fluids.
- It is seen that the turbine power starts to decrease after a certain  $\Delta T_{PP,e}$  value at all heat source temperatures except MM fluid.
- Since MM has a lower critical temperature compared to other fluids, as the heat source temperature increased, the minimum  $\Delta T_{PP,e}$  point increased.

The optimum  $\Delta T_{PP,e}$  points where the maximum turbine power is obtained under 5 different heat source temperatures of 6 different fluids are summarized in Table 6. It was stated that very close  $\Delta T_{PP,e}$  values were obtained in fluids in the same fluid group. It is seen that MM, which has a lower critical temperature compared to other fluids, has a lower optimum  $\Delta T_{PP,e}$  value from 280 °C.

In the last part of the study, the loss of performance due to constant  $\Delta T_{PP,e}$  values were investigated. It was seen from the literature research that the constant  $\Delta T_{PP,e}$  value in low and high temperature ORC's was taken as 5 and 20 °C, respectively. In systems where the heat source temperature is higher than 90 °C, it is seen that taking  $\Delta T_{PP,e}$  as constant 5 °C causes performance loss. In low temperature ORC systems, it is seen that the performance loss increases as the heat source temperature increases. There was less performance change in high temperature ORC systems compared to low temperature systems. Performance comparison of all fluids used in thermodynamic design was

made under constant and optimum  $\Delta T_{PP,e}$ . On average, 38.7% and 5.9% higher turbine power was achieved for low and high temperature applications, respectively, in the optimum  $\Delta T_{PP,e}$  condition. An example of performance comparison from low and high temperature applications is given below.

- At 120 °C, the turbine power under constant  $\Delta T_{PP,e}$  (5 °C) in ORC system with R141b is 2.863 kW, while it is 3.871 kW under optimum  $\Delta T_{PP,e}$  (10.33 °C). Under optimum  $\Delta T_{PP,e}$ , 35% performance increase was determined.
- At 270 °C, the turbine power under constant  $\Delta T_{PP,e}$  (20 °C) in ORC system with benzene is 21.47 kW, while it is 22.42 kW under optimum  $\Delta T_{PP,e}$  (27 °C). Under optimum  $\Delta T_{PP,e}$ , 4.42% performance increase was determined.

## 5. CONCLUSIONS

In this study, the effect of optimum  $\Delta T_{PP,e}$  value on ORC performance was determined. Optimum  $\Delta T_{PP,e}$  values were determined under different applications by thermodynamic optimization with turbine power maximization.

In low temperature ORC applications,

- The performance of dry, isentropic, wet and new-generation fluid groups was compared.
- The highest turbine power has been reached in the ORC system with R1234yf.
- While the effect of  $\Delta T_{PP,e}$  on turbine power has a similar tendency in dry and isentropic fluids, it has been different due to the low critical temperature of wet and new-generation fluids.
- It was stated that the optimum  $\Delta T_{PP,e}$  value for dry and isentropic fluids depends on the heat source temperature. In wet and new-generation fluids, it was determined that the optimum  $\Delta T_{PP,e}$  value depends on both the heat source temperature and the organic fluid.

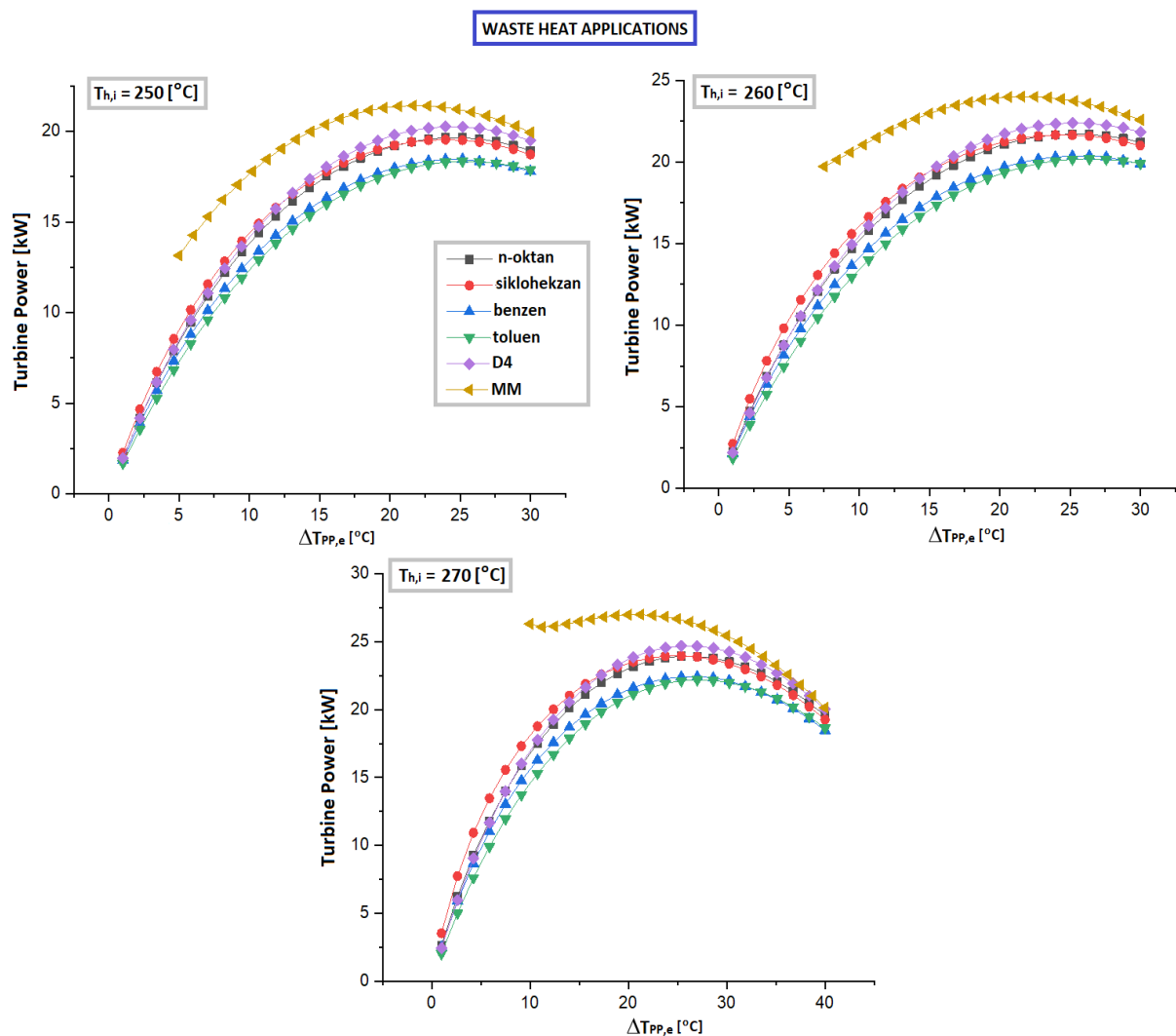
- For 90 °C, it is seen that the optimum  $\Delta T_{PP,e}$  value is the same for all fluids. In ORC systems designed at heat source temperatures above 90 °C, firstly, optimum  $\Delta T_{PP,e}$  values should be determined.

In high temperature ORC applications,

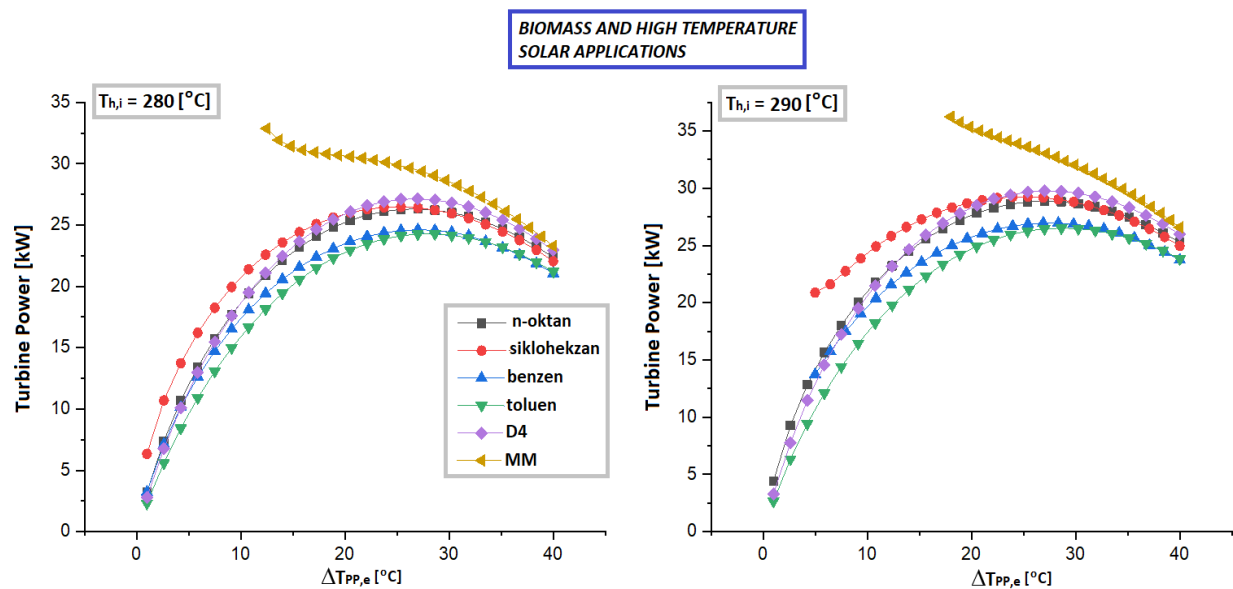
- The performance of alkanes, aromatic hydrocarbon and siloxane fluid groups were compared.
- The highest turbine power has been reached in the ORC system with MM.
- Since MM has a lower critical temperature than others, as the heat source temperature increased, the minimum  $\Delta T_{PP,e}$  value increased.
- Cyclohexane appears to be the fluid most affected by  $\Delta T_{PP,e}$  exchange. It is stated that it has higher turbine power than others at low  $\Delta T_{PP,e}$  values.

It has been determined that the effect of  $\Delta T_{PP,e}$  on turbine power is greater in low temperature ORC systems. Studies in which the  $\Delta T_{PP,e}$  value was taken as constant regardless of the heat source temperature and organic fluid were examined. It has been determined that 38.7% and 5.9% more turbine power will be achieved, respectively, for low and high temperature applications under optimum  $\Delta T_{PP,e}$ .

Taking constant  $\Delta T_{PP,e}$ , which has a very important place in ORC performance, causes seriously erroneous results in studies. Using optimum  $\Delta T_{PP,e}$  values determined depending on the heat source temperature and organic fluid will help achieve higher ORC performances. In thermodynamic analysis, modeling and simulation studies, it is recommended to determine the optimum  $\Delta T_{PP,e}$  value first.



**Figure 5.** Effect of  $\Delta T_{PP,e}$  change on turbine power for 250, 260 and 270 °C heat source temperatures in ORC's waste heat applications.



**Figure 6.** Effect of  $\Delta T_{PP,e}$  change on turbine power for 280 and 290 °C heat source temperatures in ORC's biomass and high temperature solar applications.

**Table 6.** Determination of optimum  $\Delta T_{PP,e}$  value for different fluids under different heat source temperatures for high temperature ORC applications.

$T_{h,i}$	Optimum $\Delta T_{PP,e}$ (°C)					
	n-octane	sikloheksan	benzen	toluen	MM	D4
250 °C	23,96	23,96	25,17	25,17	21,67	23,96
260 °C	25,17	23,96	26,38	26,38	21,56	25,17
270 °C	25,38	25,38	27	27	21,25	25,38
280 °C	27	25,38	27	28,63	12,5	27
290 °C	27	25,42	26,88	28,63	18	27

## REFERENCES

- [1] S. Y. Wu, S. M. Zhou, and L. Xiao, "The determination and matching analysis of pinch point temperature difference in evaporator and condenser of organic rankine cycle for mixed working fluid," *Int. J. Green Energy*, vol. 13, no. 5, pp. 470–480, 2016, doi: 10.1080/15435075.2014.966371.
- [2] H. Yu, X. Feng, and Y. Wang, "A new pinch based method for simultaneous selection of working fluid and operating conditions in an ORC (Organic Rankine Cycle) recovering waste heat," *Energy*, vol. 90, pp. 36–46, 2015, doi: 10.1016/j.energy.2015.02.059.
- [3] X. Liu, Y. Zhang, and J. Shen, "System performance optimization of ORC-based geo-plant with R245fa under different geothermal water inlet temperatures," *Geothermics*, vol. 66, pp. 134–142, 2017, doi: 10.1016/j.geothermics.2016.12.004.
- [4] Ö. Kaşka, O. Bor, and N. Tokgöz, "Energy and exergy analysis of an organic rankine-brayton combined cycle," *J. Fac. Eng. Archit. Gazi Univ.*, vol. 33, no. 4, pp. 1201–1213, 2018, doi: 10.17341/gazimmfd.416420.
- [5] J. Sun, Q. Liu, and Y. Duan, "Effects of evaporator pinch point temperature difference on thermo-economic performance of geothermal organic Rankine cycle systems," *Geothermics*, vol. 75, no. February, pp. 249–258, 2018, doi: 10.1016/j.geothermics.2018.06.001.
- [6] A. H. Bademlioglu, R. Yamankaradeniz, and O. Kaynakli, "Exergy analysis of the organic rankine cycle based on the pinch point temperature difference," *J. Therm. Eng.*, vol. 5, no. 3, pp. 157–165, 2019, doi: 10.18186/THERMAL.540149.
- [7] J. Wang, M. Diao, and K. Yue, "Optimization on pinch point temperature difference of ORC system based on AHP-Entropy method," *Energy*, vol. 141, pp. 97–107, 2017, doi: 10.1016/j.energy.2017.09.052.
- [8] J. Sarkar, "Generalized pinch point design method of subcritical-supercritical organic Rankine cycle for maximum heat recovery," *Energy*, vol. 143, pp. 141–150, 2018, doi: 10.1016/j.energy.2017.10.057.
- [9] M. Jankowski, A. Borsukiewicz, K. Szopik-Dępczyńska, and G. Ioppolo, "Determination of an optimal pinch point temperature difference interval in ORC power plant using multi-objective approach," *J. Clean. Prod.*, vol. 217, pp. 798–807, 2019, doi: 10.1016/j.jclepro.2019.01.250.
- [10] M. Imran, B. S. Park, H. J. Kim, D. H. Lee, M. Usman, and M. Heo, "Thermo-economic optimization of Regenerative Organic Rankine Cycle for waste heat

- recovery applications,” *Energy Convers. Manag.*, vol. 87, pp. 107–118, 2014, doi: 10.1016/j.enconman.2014.06.091.
- [11] S. Bian, T. Wu, and J. F. Yang, “Parametric optimization of organic rankine cycle by genetic algorithm,” *Appl. Mech. Mater.*, vol. 672–674, pp. 741–745, 2014, doi: 10.4028/www.scientific.net/AMM.672-674.741.
- [12] R. Long, Y. J. Bao, X. M. Huang, and W. Liu, “Exergy analysis and working fluid selection of organic Rankine cycle for low grade waste heat recovery,” *Energy*, vol. 73, pp. 475–483, 2014, doi: 10.1016/j.energy.2014.06.040.
- [13] C. G. Gutiérrez-Arriaga, F. Abdelhady, H. S. Bamufleh, M. Serna-González, M. M. El-Halwagi, and J. M. Ponce-Ortega, “Industrial waste heat recovery and cogeneration involving organic Rankine cycles,” *Clean Technol. Environ. Policy*, vol. 17, no. 3, pp. 767–779, 2015, doi: 10.1007/s10098-014-0833-5.
- [14] Z. Han, Y. Yu, and Y. Ye, “Selection of working fluids for solar thermal power generation with organic rankine cycles system based on genetic algorithm,” *ICMREE 2013 - Proc. 2013 Int. Conf. Mater. Renew. Energy Environ.*, vol. 1, pp. 102–106, 2013, doi: 10.1109/ICMREE.2013.6893624.
- [15] L. Pierobon, M. Rokni, U. Larsen, and F. Haglind, “Thermodynamic analysis of an integrated gasification solid oxide fuel cell plant combined with an organic Rankine cycle,” *Renew. Energy*, vol. 60, pp. 226–234, 2013, doi: 10.1016/j.renene.2013.05.021.
- [16] R. Agromayor and L. O. Nord, “Fluid selection and thermodynamic optimization of organic Rankine cycles for waste heat recovery applications,” *Energy Procedia*, vol. 129, pp. 527–534, 2017, doi: 10.1016/j.egypro.2017.09.180.
- [17] J. G. Andreasen, U. Larsen, T. Knudsen, L. Pierobon, and F. Haglind, “Selection and optimization of pure and mixed working fluids for low grade heat utilization using organic rankine cycles,” *Energy*, vol. 73, pp. 204–213, 2014, doi: 10.1016/j.energy.2014.06.012.
- [18] D. Fiaschi, A. Lifshitz, G. Manfrida, and D. Tempesti, “An innovative ORC power plant layout for heat and power generation from medium- to low-temperature geothermal resources,” *Energy Convers. Manag.*, vol. 88, pp. 883–893, 2014, doi: 10.1016/j.enconman.2014.08.058.
- [19] Z. Kai, Z. Mi, W. Yabo, S. Zhili, L. Shengchun, and N. Jinghong, “Parametric Optimization of Low Temperature ORC System,” *Energy Procedia*, vol. 75, pp. 1596–1602, 2015, doi: 10.1016/j.egypro.2015.07.374.
- [20] G. Li, “Organic Rankine cycle performance evaluation and thermoeconomic assessment with various applications part I: Energy and exergy performance evaluation,” *Renewable and Sustainable Energy Reviews*, 53, pp. 477–499, 2016, doi: 10.1016/j.rser.2015.08.066.
- [21] J. M. Calm and G. C. Hourahan, “Refrigerant data update,” *HPAC Heating, Piping, AirConditioning Eng.*, vol. 79, no. 1, pp. 50–64, 2007.
- [22] T. Ho, S. S. Mao, and R. Greif, “Comparison of the Organic Flash Cycle (OFC) to other advanced vapor cycles for intermediate and high temperature waste heat reclamation and solar thermal energy,” *Energy*, vol. 42, no. 1, pp. 213–223, 2012, doi: 10.1016/j.energy.2012.03.067.

## NOMENCLATURE

$\Delta T_{PP,e}$ : Evaporator pinch point temperature difference

$\Delta T_{PP,c}$ : Condenser pinch point temperature difference

W<sub>p</sub>: Pump Work

W<sub>t</sub>: Turbine Work

W<sub>net</sub>: Net Work

Q<sub>e</sub> : Evaporator heat load

Q<sub>c</sub> : Condenser heat load

i<sub>p</sub>: Pump irreversibility

i<sub>e</sub>: Evaporator irreversibility

i<sub>t</sub>: Turbine irreversibility

i<sub>c</sub>: Condenser irreversibility

i<sub>total</sub>: Total irreversibility

T<sub>h</sub>: Average heat source temperature

T<sub>c</sub>: Average coling water temperature

T<sub>h,i</sub>: Heat source inlet temperature

T<sub>h,o</sub>: Heat source output temperature

T<sub>c,i</sub>: Cooling water inlet temperature

T<sub>c,o</sub>: Cooling water output temperature

$\eta_{th}$  : Thermal efficiency

$\eta_{II}$  : Exergy efficiency


$\eta_p$  : Pump isentropic efficiency

$\eta_t$  : Turbine isentropic efficiency

## Compositional correlation analysis of gene expression time series

\*<sup>1</sup>Fatih Dikbaş

<sup>1</sup>Pamukkale University, Civil Engineering Department, Denizli, Turkey

[f\\_dikbas@pau.edu.tr](mailto:f_dikbas@pau.edu.tr) 

### Abstract

Accurate determination of temporal dependencies among gene expression patterns is crucial in the assessment of functions of genes. The gene expression series generally show a periodic behavior with nonlinear curved patterns. This paper presents the determination of temporally associated budding yeast gene expression series by using compositional correlation method. The results show that the method is capable of determining real direct or inverse linear, nonlinear and monotonic relationships between all gene pairs. Pearson's correlation values between some of the gene pairs have shown negative or very weak relationships ( $r \approx 0$ ) even though they were found to be strongly associated. Inversely, a high positive  $r$  value was obtained even though the genes are inversely related as determined by the compositional correlation approach. Comparisons with Pearson's correlation, Spearman's correlation, distance correlation and the simulated annealing genetic algorithm maximal information coefficient (SGMIC) have shown that the presented compositional correlation method detects important associations which were not found by the compared methods. Supplementary materials containing the code of the used software together with some extended figures and tables are available online.

**Keywords:** Combinatorics, Compositions of  $n$ , Compositional correlation, Gene expression association, *Saccharomyces Cerevisiae*

### 1. INTRODUCTION

"The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents" wrote H. P. Lovecraft at the beginning of his cult story "The Call of Cthulhu" [1]. This was long before human mind managed to invent supercomputers to try to calculate correlations among data sets (huge or small) by using various correlation coefficients. Pearson's Correlation Coefficient (simply called correlation or  $r$ ) [2] - which was introduced when Lovecraft was only five years old - might still be the most widely used statistical measure for assessing relationships between data series.

In its nature, Pearson's correlation is a measure of linear association. Currently it is very hard to find a single issue of a scientific journal that does not include the word 'correlation'. Pearson's correlation is also used in the analysis of high-throughput data (such as genotype, genomic, imaging, and others) [3, 4], although the relationships are generally nonlinear. This tendency of using Pearson's correlation for non-linear associations still widely exists in literature despite clear warnings about its improper use: Correlation is misleading [5]; good correlation does not automatically imply good agreement [6]; risk of producing spurious correlations when analyzing non-independent variables is very large [7, 8]. The unintended and generally unnoticed misleading results are caused by the approach used in calculation of the correlation itself where the

averages of the whole series are used for assessing relationship. In fact, the average value of a data series is a single value which does not reflect the variations within the data series. In fact, the variations in data might have great importance in the determination of associations with other data series. Unfortunately, in association studies, there is still an inability in completely correlating the contents of data sets caused by inappropriate implementation of the currently used approaches or the inappropriate methodology of the used approach itself. Gene expression over time is a continuous process and can be considered as a continuous curve or function [9]. Genome-wide association studies try to determine associated gene pairs by comparing the expression series of each gene [10, 11]. Most of the studies use Pearson's correlation for attempting to find associations among the genes by comparing the expression series but Pearson's correlation is a measure of linear association and gene expression series generally show a periodic behavior with nonlinear curved patterns. Therefore, it is not surprising that the widely used Pearson method is generally reported to be less efficient than the compared methods in finding gene pairs of multiple relationships. It must be kept in mind that the efficiencies of different methods vary with the data properties to some degree and a pre-analysis is generally advised to identify the best performing method. A comprehensive comparison of gene association methods was provided by Kumari et al. [12].

The compositional correlation method used in this study takes its name from the term composition in number theory and combinatorics. The details of the method are presented

in the Materials and Methods section. Compositional correlation is based on the foundations of the two-dimensional correlation method developed by the author for assessing the degree and direction of relationships between matrices [13, 14]. These methods were developed when it was noticed by the author that, in some cases, the Pearson's correlation value decreases even though the estimations become closer to the observations in the hydrological modelling studies. Instead of considering the averages of the compared series, the compositional correlation approach considers the averages of all parts of all possible compositions of the data series. The comparison plots of calculated compositional variance, covariance and correlation values generate clouds that allow a comprehensive visual inspection of all obtained results on a single graph. The variance and covariance clouds also provide an opportunity for the comparison of the results with the Pearson's correlation. The purpose of this study is to present the implementation of the compositional correlation method in the determination of the yeast genes sharing similar temporal expression patterns. The aim was to provide strong clues for determining the functions of undefined yeast genes. The general trends of molecular events are directly associated with the timing of global gene expression patterns. Therefore, determination of the genes sharing similar temporal expression patterns is the first important step in the validation of functional implications of the inferred expression patterns [15].

Understanding the temporal relationships between the expression profiles of genes is crucial in determining the causes, functions and consequences of the biological processes like the cell cycle [16]; identifying the roles of genes in the stages of developmental processes of organisms [17, 18]; determination of genetic relatedness among various species [19]; investigating the functions of individual genes by exploring genetic interactions [20], and developing drugs to cure diseases by identifying genes that act in response to a certain disease [21]. Consequently, as the presented results also suggest, the compositional correlation method seems to be a very appropriate method for finding the associated genes by a complete comparison of the expression series, a task which is impossible to be made manually because of thousands of genes to be compared.

## 2. MATERIALS AND METHODS

This study presents for the first time in literature, the implementation of the compositional correlation method for gene expression time series data where the association levels of all yeast (*saccharomyces cerevisiae*) genes are determined. The details of the data used in the study are provided in the Results section below. The first introduction of the compositional correlation method was made in an association study between polynomial functions for which the Pearson's correlation failed because of nonlinearity of the polynomials [22]. The previous findings have shown that the compositional correlation method determines both the inversely and directly related portions of the examined functions. Therefore, gene expression series become very appropriate observations for the compositional correlation method because of their nonlinear structure which also

sometimes show an alternating (sometimes increasing and sometimes decreasing) behavior.

The main idea of the compositional correlation is that the association between two series might be better defined by the cumulative contribution of their parts but not the averages of the whole series especially when the series have varying (nonlinear, alternating, periodic etc...) behavior. If A is any set of positive integers, a composition of n with parts in A is an ordered collection of one or more elements in A whose sum is n [23]. The integer n is the number of observations in one of the compared series in the correlation case.

In number theory and combinatorics, a partition of a positive integer n, also called an integer partition, is an expression representing n as a sum of positive integers [24-27]. If order matters, which is also the case in the gene expression time series, the sum becomes a composition.

Each component of a composition is called a part of the composition. For example, the compositions of 3 are [1, 1, 1], [1, 2], [2, 1] and [3]. Similarly, if a sample data series has n (a positive integer) elements, the compositions of the series can be determined by dividing the series into parts. Each part should have at least two elements ( $m \geq 2$ ) for calculating compositional correlation. The compositions for  $2 \leq n \leq 10$  with parts  $\geq 2$  are shown in Table 1. The number of elements in each part are shown in brackets and the total number of compositions,  $t_n$ , is shown in the right column.

The number of possible compositions increases rapidly with n. The total numbers of compositions shown in the right column of Table 1 is a Fibonacci sequence ( $t_n = F_{n-1}$ ). The Fibonacci numbers are defined by  $F_{n+1} = F_n + F_{n-1}$  where the rate  $F_{n+1} / F_n$  rapidly tends to the golden ratio known as  $\phi = (1+\sqrt{5})/2 = 1.618...$  [28]. This means that there is golden ratio between the total number of compositions for two consecutive integers when the minimum number of observations in each part is equal to 2 ( $m = 2$ ) [29].

**Table 1.** All possible compositions with parts  $\geq 2$  for  $2 \leq n \leq 10$

n	All possible compositions with parts $\geq 2$	$t_n$
2	[2]	1
3	[3]	1
4	[2, 2]; [4]	2
5	[2, 3]; [3, 2]; [5]	3
6	[2, 2, 2]; [2, 4]; [3, 3]; [4, 2]; [6]	5
7	[2, 2, 3]; [2, 3, 2]; [2, 5]; [3, 2, 2]; [3, 4]; [4, 3]; [5, 2]; [7]	8
8	[2, 2, 2, 2]; [2, 2, 4]; [2, 3, 3]; [2, 4, 2]; [2, 6]; [3, 2, 3]; [3, 3, 2]; [3, 5]; [4, 2, 2]; [4, 4]; [5, 3]; [6, 2]; [8]	13
9	[2, 2, 2, 3]; [2, 2, 3, 2]; [2, 2, 5]; [2, 3, 2, 2]; [2, 3, 4]; [2, 4, 3]; [2, 5, 2]; [2, 7]; [3, 2, 2, 2]; [3, 2, 4]; [3, 3, 3]; [3, 4, 2]; [3, 6]; [4, 2, 3]; [4, 3, 2]; [4, 5]; [5, 2, 2]; [5, 4]; [6, 3]; [7, 2]; [9]	21
10	[2, 2, 2, 2, 2]; [2, 2, 2, 4]; [2, 2, 3, 3]; [2, 2, 4, 2]; [2, 2, 6]; [2, 3, 2, 3]; [2, 3, 3, 2]; [2, 3, 5]; [2, 4, 2, 2]; [2, 4, 4]; [2, 5, 3]; [2, 6, 2]; [2, 8]; [3, 2, 2, 3]; [3, 2, 3, 2]; [3, 3, 2, 2]; [3, 2, 5]; [3, 3, 4]; [3, 4, 3]; [3, 5, 2]; [3, 7]; [4, 2, 2, 2]; [4, 2, 4]; [4, 3, 3]; [4, 4, 2]; [4, 6]; [5, 2, 3]; [5, 3, 2]; [5, 5]; [6, 2, 2]; [6, 4]; [7, 3]; [8, 2]; [10]	34



### 2.1. Calculation of Compositional Correlation

The name of the compositional correlation method is based on the term composition in number theory and its value is calculated by using compositional variance and compositional covariance [22]. Compositional variance is a cumulative measure of how far the numbers in a time series spread from the averages of the part they belong in a composition. The compositional variance of a scalar time series for any composition with  $k$  parts is defined by the following equation:

$$\text{Var}_c(A) = \frac{\sum_{i=1}^k \sum_{j=1}^{n_i} (A_{i,j} - \bar{A}_i)^2}{n} \quad (1)$$

In the above equation:

A: A scalar vector;

$n$ : The number of data in A;

$\text{Var}_c(A)$ : The compositional variance of the vector [A] for the current composition;

$k$ : The number of parts in the current composition for which the correlation is being calculated;

$n_i$ : The number of data in part  $i$ ;

$A_{i,j}$ : The  $j^{\text{th}}$  data in  $i^{\text{th}}$  part of vector A;

$\bar{A}_i$ : The arithmetic mean of the  $i^{\text{th}}$  part of vector A;

Compositional covariance is a measure of how changes in the part averages of a time series are associated with changes in the part averages of a second time series. This approach enables a better consideration of the contribution of local associations among the observed series. It is based on the idea that any observation might be more related with the average of the neighboring values than it is related with the average of the whole series. The compositional covariance is negative when the relationship between the part averages is inverse and it is positive when the relationship is direct. Higher compositional covariance indicates a stronger association. The compositional covariance between scalar matrices A and B is defined by the following equation:

$$\text{Cov}_c(A, B) = \frac{\sum_{i=1}^k \sum_{j=1}^{n_i} (A_{i,j} - \bar{A}_i)(B_{i,j} - \bar{B}_i)}{n} \quad (2)$$

where  $B_{i,j}$  is the  $j^{\text{th}}$  data in  $i^{\text{th}}$  part of vector B and  $\bar{B}_i$  is the arithmetic mean of the  $i^{\text{th}}$  part of vector B;

Covariance is a scale dependent dimensioned measure and its value increases when a variable is increased in scale. Correlation is a scaled and dimensionless version of covariance and it takes values between  $-1$  and  $1$ . A correlation of  $\pm 1$  indicates perfect linear association and  $0$  indicates no linear relationship. Based on the above definitions of compositional variance and covariance, the compositional correlation is defined as follows:

$$r_c = \frac{\text{Cov}_c(A, B)}{\sqrt{\text{Var}_c(A)\text{Var}_c(B)}} \quad (3)$$

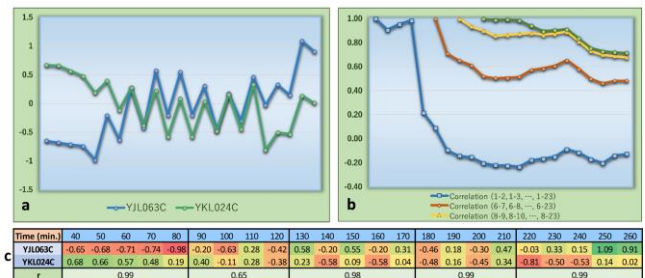
The following equation can also be used for calculating the compositional correlation directly:

$$r_c = \frac{\sum_{i=1}^k \sum_{j=1}^{n_i} (A_{i,j} - \bar{A}_i) (B_{i,j} - \bar{B}_i)}{\sqrt{\left[ \sum_{i=1}^k \sum_{j=1}^{n_i} (A_{i,j} - \bar{A}_i)^2 \right] \left[ \sum_{i=1}^k \sum_{j=1}^{n_i} (B_{i,j} - \bar{B}_i)^2 \right]}} \quad (4)$$

## 3. RESULTS

### 3.1. Application on gene expression data

The gene expression dataset used in this study consists of the results of the *cdc15* experiment made by Spellmann et al. [30]. The expression data was provided by Reshef et al. [31]. Before explaining the implementation process of the compositional correlation, an example is presented for showing how unreliable the Pearson's correlation ( $r$ ) might be when comparing time series data (Figure 1). The selected gene pair is YJL063C and YKL024C. Both expression series have 23 observations ( $n = 23$ ) and  $r = -0.12$  between the whole time series of the genes. The correlation value is 1 for the first two pair (1-2) and the correlation gradually decreases to  $-0.12$  for the whole data range (observation 1-23) ( $r = 1$  for the range 1-2;  $r = 0.91$  for the range 1-3;  $r = 0.95$  for the range 1-4; . . . and  $r = -0.12$  for the range 1-23 as shown with the blue line in Figure 1b). The correlation between the first five observations is 0.99 and  $r$  suddenly decreases to 0.22 when the first six observations are considered. Value of  $r$  continues to have very low values for the remaining ranges and does not get positive values for the ranges from (1-7) to (1-23).



**Figure 1.** (a) The expression time series of budding yeast genes YJL063C and YKL024C, (b) the variation of Pearson's correlation with the selected data range and (c) the expression series and the correlations between the parts of the BCC of the genes YJL063C and YKL024C.

When the first five observations are ignored (for which  $r = 0.99$ ) the  $r$  values vary between 1 (for the range 6-7) and 0.48 (for the range 6-23) (Figure 1b). The figure clearly shows that correlation only gets negative values when the first five values which nearly have a perfect positive correlation are included in the calculation of correlation. The expression time series of the sample genes always increase and decrease together for all smallest subsections ( $n = 2$ ) (Figure 1a) indicating a very strong quantitative relationship but the Pearson's correlation does not reflect this behavior.

The compositional correlation method calculates correlations for all compositions of the compared series and the values tend to be higher when the parts of the compositions are highly correlated. For example, the above gene pair is one of the numerous gene pairs determined to have a very low value of Pearson's correlation while most of

the compositional correlations are very high (up to 0.92). For this pair, the best correlated composition (BCC) is [5, 4, 5, 4, 5]. When the 23 observations are divided into five parts (which form the BCC) as shown in Figure 1c,  $r = 0.65$  for the second part and  $r \geq 0.98$  for the remaining parts. The high  $r$  values in all parts point out a strong direct relationship between the series as depicted by the time series graph (Figure 1a) while the  $r$  value for the whole series is interestingly a negative number close to zero indicating an inverse weak relationship. The high value of the compositional correlation for the gene pair indicates the apparent direct relationship. The above example shows the influence of sample size on the value of correlation.

The compositional correlation approach enables detection of this type of relationship by considering the cumulative influence of all possible sample compositions for the compared series. If there is no composition producing high correlations (positive or negative) for the compared parts, then the association between the compared series is definitely weak.

### 3.2. Compositional correlations between 4381 gene expression series

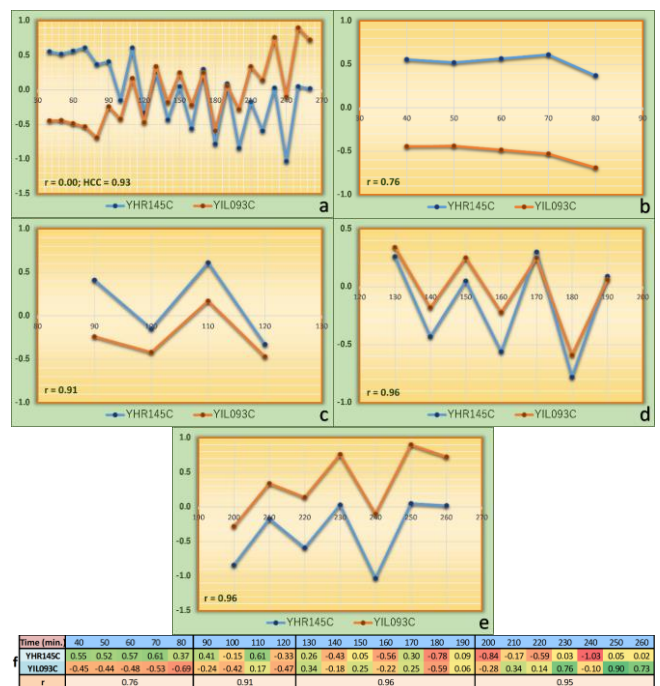
The compositional correlations between the available expression series of all pairs of 4381 budding yeast (*saccharomyces cerevisiae*) genes were calculated. Compositional correlations against time were also calculated for each gene. The expression time series for each gene consists of 23 observations ( $n = 23$ ) and the number of all possible gene pairs is 9,594,390. For each gene pair, all possible compositional correlations were calculated by considering the minimum number of observations in each part of a composition to be at least 4 (a total of 250 compositional correlations for each pair when  $m = 4$ ). The total number of calculated compositional correlations is over 2.3 billion. All of the calculated compositional correlations were not stored as output because this would significantly slow down the file generation process and increase the requirement of storage space. Only the highest compositional correlation (HCC),  $r$ , the lowest compositional correlation (LCC), best correlated composition (BCC) and the worst correlated composition (WCC) values for all possible data series pairs were written to the output file. The output file is provided for download via the following link for enabling further investigation. The file contains clues for determining the relationships and functions of the hundreds of yeast genes which are still unidentified.

<https://www.dropbox.com/s/lqqnmaf9h6g1rr/Compositional.Correlations.Spellman.m4.rar>

Among all the compared gene pairs, the highest compositional correlation (0.993) was obtained between the genes YDL003W and YDR097C for the composition [7, 4, 8, 4] (Table S1 in the Supplementary Material provides the complete list of the gene pairs with HCC values over 0.9). For this gene pair,  $r = 0.985$  and  $LCC = 0.942$  and the small difference between HCC and LCC indicates a very strong relationship all through the observed period (Figure S1 in the

Supplementary Material). LCC is higher than 0.9 for 777 gene pairs while it is over 0.85 for 5202 gene pairs. The lowest LCC value (-0.982) was obtained for the pair YIL141W and YMR031C for the composition [9, 4, 5, 5] (Table S2 provides the complete list of the gene pairs with lowest compositional correlation (LCC) values under -0.9). For this pair,  $r = -0.932$  and  $HCC = -0.791$  (Figure S2). HCC is lower than -0.9 for 146 gene pairs while it is less than -0.85 for 1355 gene pairs. The HCC values are over 0.9 for 31185 (0.325%) gene pairs (Table S1) while the Pearson's correlation is over 0.9 for only 2684 (0.0027%) of the gene pairs. For example,  $HCC = 0.93$  for the composition [5, 4, 7, 7] of the gene pair YHR145C and YIL093C while the composition [23] is the WCC and gives  $LCC = 0.00$  which is the Pearson's correlation (Figure 2).

This gene pair is one of the 58 gene pairs for which the  $HCC > 0.9$  while  $-0.1 < r < 0.1$  (Table S3). Figures 2b - 2e show the expressions of the genes for each part of the BCC and Figure 2f shows the expression values together with the Pearson's correlations for each part of the BCC. Even though the Pearson's correlation for the whole series is zero, the Pearson's correlations for each part of the BCC are very high (between 0.76 and 0.96; 3/4 of them being over 0.9). This shows that the Pearson's correlation for the gene pair is misleading because there seems to be a very strong direct relationship between the genes as shown by the time series graph and the very high compositional correlation value.

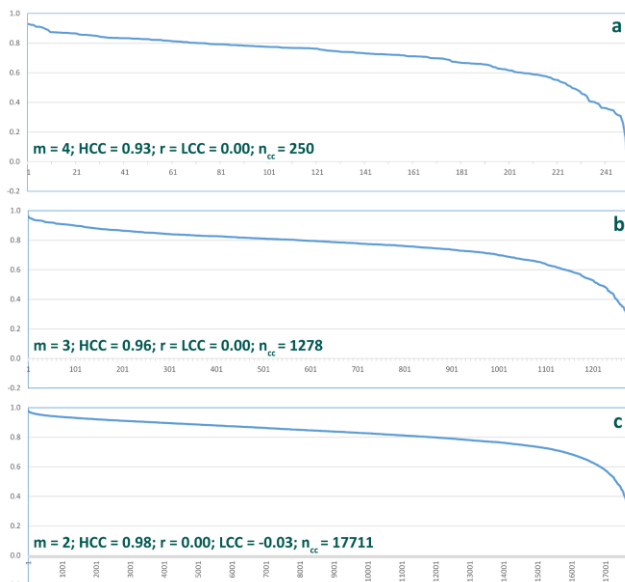


**Figure 2.** (a) Expression time series for the genes YHR145C and YIL093C; (b) first 5 expression pairs (40 to 80 minutes); (c) 4 expression pairs from 90 to 120 minutes; (d) 7 expression pairs from 130 to 190 minutes and (e) 7 expression pairs from 200 to 260 minutes and (f) the expression series and the correlations between the parts of the BCC of the genes YHR145C and YIL093C.

Table S4 presents the 250 compositional correlations between the genes YHR145C and YIL093C for  $m = 4$ . For the same gene pair, the 1278 compositional correlations for  $m = 3$  (Table S5) and 17711 (the maximum number of

possible compositions) compositional correlations for  $m = 2$  (Table S6) are calculated separately for investigating the variation of compositional correlation for the gene pair. The compositions for  $m = 3$  and  $m = 4$  are subsets of the compositions for  $m = 2$  and their compositional correlation values remain within the compositional correlation range obtained for  $m = 2$  (Figures 3a, 3b and 3c). The compositional correlation range is the difference between HCC and LCC and these values are available for each gene pair.

All compositional correlations are higher than  $r$  (which is 0.00) when  $m = 3$  and  $m = 4$ . Similarly, when  $m = 2$ , 17709 of the 17711 compositional correlations (99.99%) are higher than  $r$ . The results for  $m = 2$  and  $m = 3$  show that the obtained results for  $m = 4$  provide sufficient information on the compositional correlation structure of the investigated gene expression series. This indicates that there is a strong direct numerical relationship between the expressions all through the investigated period and might point out the existence of a functional relationship even though the Pearson's correlation is zero.



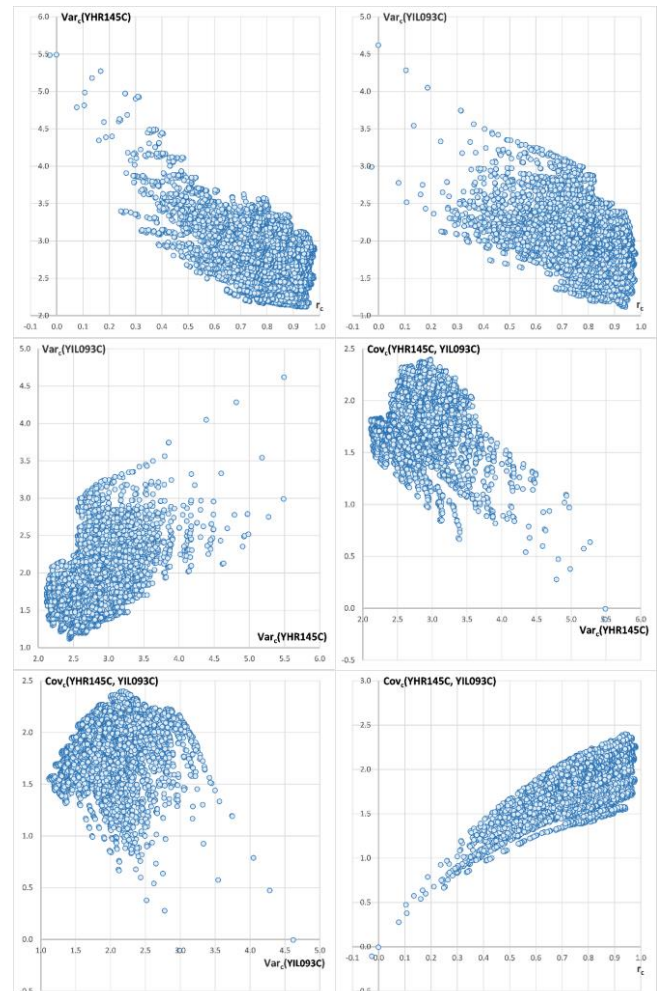
**Figure 3.** The compositional correlations obtained for the gene pair YHR145C and YIL093C when  $m = 4$  (a),  $m = 3$  (b) and  $m = 2$  (c). The HCC,  $r$ , LCC and the number of compositional correlations ( $n_{cc}$ ) are indicated on each figure.

The variance and covariance clouds of the genes YHR145C and YIL093C shown in Figure 4 also validate that the Pearson's correlation is far from representing the association between the genes. The Pearson's correlation is a point at the far end (the point at the origin) of the tail of the compositional covariance cloud shown in the bottom right panel. The Pearson's correlation does not indicate the strong direct (positive) relationship between the genes shown by all the panels in the figure.

### 3.3. Inverse relationships

As in all association studies, determination of inverse relationships between genes might also be as important as determining direct relationships for assessing expression balance of proteins [32]. Pearson's correlation is below -0.9

for 554 of the investigated gene pairs, while 12373 gene pairs have a LCC value below -0.9 indicating that there might be much more inversely related yeast gene pairs than the Pearson's correlation points out (Table S2). The genes YBR146W and YJR045C are one of the many inversely related gene pairs that Pearson's correlation fails to detect. For this pair, HCC =  $r = 0.00$  while LCC = -0.93 for the WCC which is [4, 4, 6, 4, 5]. Figure 5 shows the expression time series for this pair together with the correlations for each part of the WCC.



**Figure 4.** The variance and covariance clouds obtained for the gene pair YHR145C and YIL093C when  $n = 23$  and  $m = 2$ .

The negative correlations for each part are very close to -1 (ranging between -0.88 and -1.00) indicating a strong inverse relationship but the combined parts produce a zero Pearson's correlation falsely proposing that the genes have no relation. The inverse relationship is also apparent in the time series graph but it is practically impossible to generate graphs and determine these types of relationships manually when there are millions of pairs of genes. The computational procedure of the CompCorr software enables determination of these relationships by calculating compositional correlations for all possible compositions.

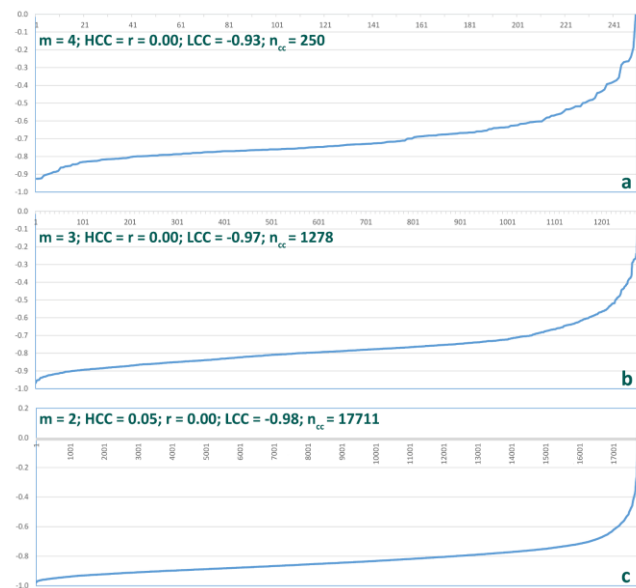
Tables S7, S8 and S9 respectively show in ascending order, the compositional correlations obtained for the genes YBR146W and YJR045C by taking  $m = 4$ ,  $m = 3$  and  $m = 2$ .



When  $m = 3$  and  $m = 4$ , all compositional correlations are lower than  $r$  (which is 0.00) and when  $m = 2$ , 17709 of the 17711 compositional correlations (99.99%) are lower than  $r$  (Figure 6a, 6b and 6c). These results imply that these genes might have a strong inverse functional relationship even though Pearson's correlation is zero.

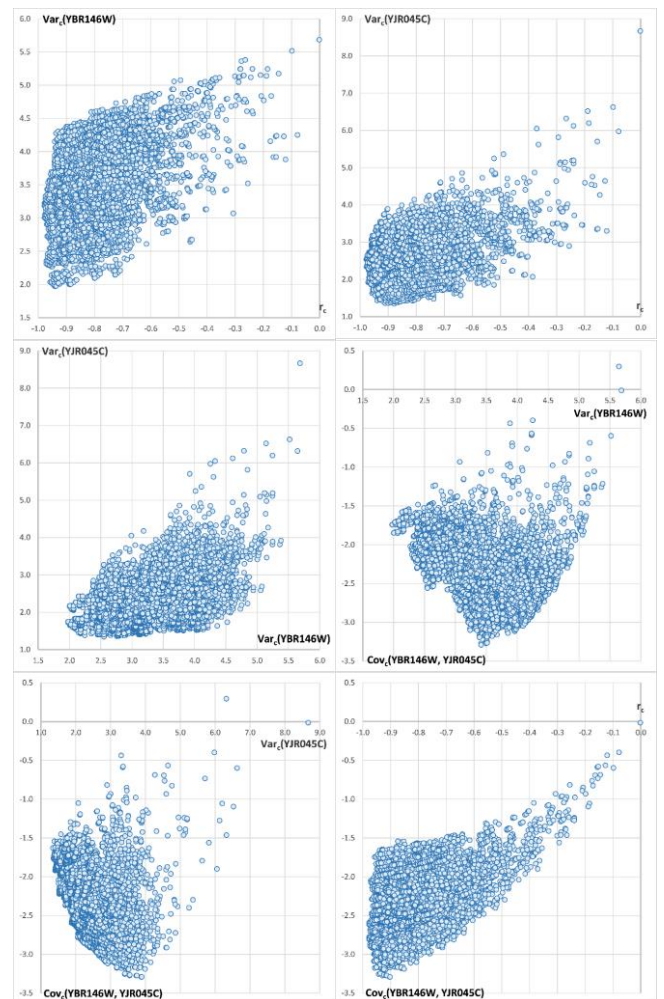


**Figure 5.** (a) Expression time series for the genes YBR146W and YJR045C; (b) first 4 expression pairs (40 to 70 minutes); (c) 4 expression pairs from 80 to 110 minutes; (d) 6 expression pairs from 120 to 170 minutes; (e) 4 expression pairs from 180 to 210 minutes and (f) 5 expression pairs from 220 to 260 minutes and (g) the expression series and the correlations between the parts of the WCC of the genes YBR146W and YJR045C.



**Figure 6.** The compositional correlations obtained for the gene pair YBR146W and YJR045C when  $m = 4$  (a),  $m = 3$  (b) and  $m = 2$  (c). The HCC,  $r$ , LCC and the number of compositional correlations ( $n_{cc}$ ) are indicated on each figure.

The variance and covariance clouds of the genes YBR146W and YJR045C in Figure 7 show that the Pearson's correlation does not correctly point out the association between the genes. The Pearson's correlation is a point at the far end (the point at the origin) of the tail of the compositional covariance cloud shown in the bottom right panel and it is far from representing the strong direct (positive) relationship between the genes shown by all the panels in the figure. The strong indirect relationship between the gene expression series through the whole observation period is also validated by the covariance clouds in Figure 7.



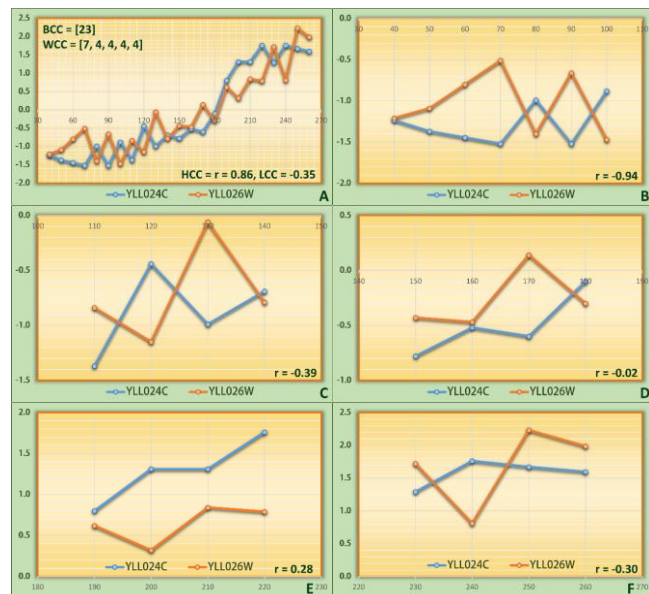
**Figure 7.** The variance and covariance clouds obtained for the genes YBR146W and YJR045C when  $n = 23$  and  $m = 2$ .

The above gene pairs are only two examples among the thousands of pairs with probable functional relationships determined by the compositional correlation method. Some other selected examples with high compositional correlation but significantly lower  $r$  values are presented in Figure S3. Each graph in the figure includes HCC,  $r$  and LCC values together with the BCC's and WCC's. Table S10 shows the expression values of the gene pairs and the correlations for all parts of the BCC's for each pair in Figure S3.

### 3.4. Directly or Inversely Related?

Another feature of compositional correlation is its ability to determine the existence of inverse relationships when the series have a general direct relationship (or vice versa). An

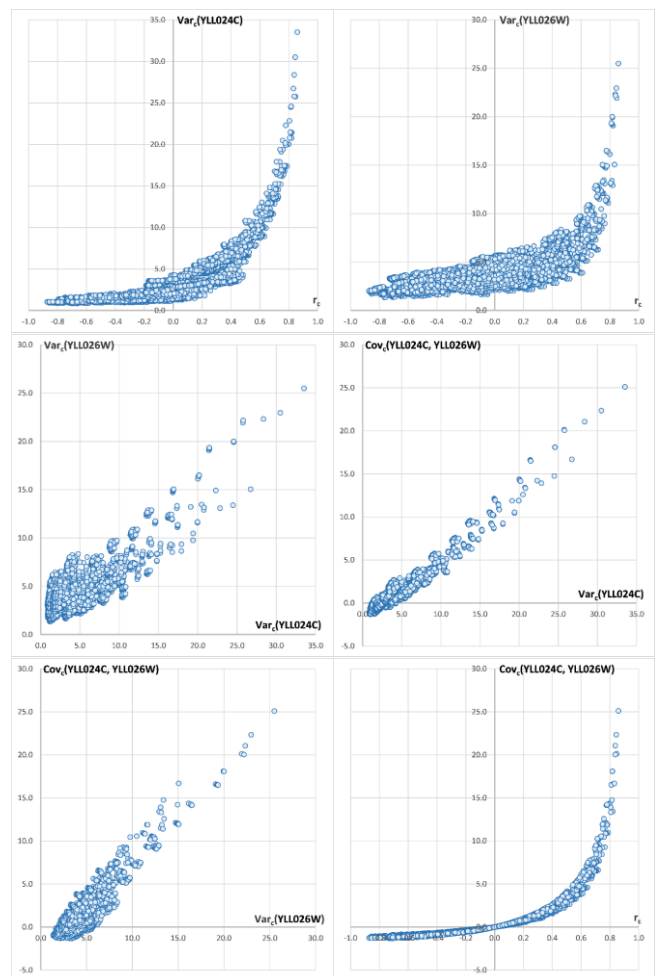
example of this feature is the gene pair YLL024C and YLL026W. Figure 8a shows that the genes in this pair both have a similarly increasing expression trend and the value of Pearson's correlation (0.86) also validates this behavior, but the expression values are always inversely related between all observation points (when one is increasing, the other is decreasing) only except for the minutes between 180 and 190 where they both increase and between the minutes 250 and 260 where they both decrease. This inverse expression behavior is determined by looking at the differences between the HCC and LCC values. For this pair, the LCC value is -0.35 indicating that there is an inverse relationship for some compositions of the gene expression series.



**Figure 8.** The genes YLL024C and YLL026W have a similar expression trend with  $r=0.86$  (a) but their expression patterns have inverse relationship (b-f).

Calculation of correlations for the parts of the WCC ([7, 4, 4, 4, 4]) of the pair also points out the inverse relationship between the genes (Figures 8b-8f). The  $r$  values of the parts of the WCC varies between -0.94 and 0.28 (4/5 being negative). In fact, 10585 of the 17711 compositions (59.8%) have a negative compositional correlation when  $m = 2$  (Figure 9). The dispersion of the variance and the covariance clouds are very small for these genes. This shows that both genes are closely related. The variation of the compositional correlation between 0.859 and -0.871 indicates that the genes have both a positive and a negative relationship. The compositions containing long parts produce positive compositional correlations as expected (the genes have an overall similar pattern) and the compositions composed of shorter parts produce negative compositional correlations because the smaller parts are inversely related.

Some other selected examples of gene pairs with apparent inverse relationships which cannot be detected by calculating  $r$  are shown in Figure S4. The expression values of the sample pairs and the  $r$  values for each part of the BCC for each pair are presented in Table S11. The  $r$  values of the parts (the majority being between -0.95 and -1.00) validate that the pairs might have a strong functional inverse relationship even though the  $r$  values obtained for the whole series vary between -0.01 and -0.53.



**Figure 9.** The variance and covariance clouds obtained for the genes YLL024C and YLL026W when  $n = 23$  and  $m = 2$ .

#### 4. COMPARISONS WITH OTHER CORRELATION METHODS

Determination of genes with similar or concordant (and also inversely related) expression profiles is crucial in the determination of the functions of the genes. If a method fails to find some of the harmonious gene pairs, then it will be much more difficult to make decisions on the functions of the genes as in the case of *saccharomyces cerevisiae* for which there are still hundreds of unidentified genes even though its genome was sequenced nearly 25 years ago. In this section, the performance of the compositional correlation method is compared with four widely used correlation methods which are the Pearson's correlation, Spearman's correlation, distance correlation and SGMIC which was proposed as an algorithm for the precise calculation of the maximal information coefficient [33].

Genes generally exhibit varying expression behaviors through time. Detecting this behavior by using standard correlation approaches is generally not possible because the series are considered as a whole and the different behaviors in subsections are not detected and considered. For example, the expressions of the gene pair YMR296C and YOL032W (Figure S5ac) first slightly decrease together between minutes 40 and 70. Then they begin to fluctuate together with a very similar pattern until minute 170.

Finally, after minute 180, YOL032W shows an increasing and YMR296C shows a decreasing trend while they still have a very similar expression profile. Even though the time series graph for this pair shows that there might be a strong relationship between these two genes, this behavior was not detected by Pearson's (0.048), Spearman's (0.082) and distance correlations (0.386) which pointed out a weak or no relation. The SGMIC value for this gene pair is 0.566 which seems to be a better estimate but might easily be ignored among the millions of SGMIC values for all the possible gene pairs as it also does not point out a significant relationship.

The compositional correlation method successfully determined the relationship between these two genes with an HCC value of 0.943. For another 58 gene pairs, the HCC values are higher than 0.9 for which the Pearson's correlation remains between -0.1 and 0.1 (Figure S5). The HCC, Pearson's correlation, LCC, Spearman's correlation, distance correlation and SGMIC values for these 58 gene pairs are presented in Table 2. The minimum and maximum values for each statistic are indicated in bold.

The obtained compositional correlations are much higher than the correlations of the compared methods only except for the SGMIC value of the pair YBR082C and YOR262W (Figure S5av). The distance correlation value (0.505) for this pair is the second highest distance correlation among the 58 pairs. It is known that the distance correlation is zero if and only if the random variables are statistically independent but it cannot be claimed that the distance correlation always exactly determines the strength of statistical dependence. The distance correlations for none of the compared pairs are close to zero showing that all of the 58 gene pairs are statistically dependent (Figure S5).

The Pearson's and Spearman's correlations fail to detect the statistical dependence between the genes and they produce values close to zero for all the compared pairs in Table 2. This result is caused by the fact that both methods generally fail to detect the relationship when the trend line for one series is increasing while the trend line for the other series is decreasing with nearly the same angle even though the series have a strong relationship. The presented gene pairs are good examples of this deficiency of Pearson's and Spearman's correlation approaches.

The second-best performance after the compositional correlation is shown by the SGMIC method with an average SGMIC value of 0.511 but all SGMIC values except for 0.932 obtained for the pair YBR082C and YOR262W are under 0.8 showing that the SGMIC results for these gene pairs are not sufficiently maximal to be noticed.

The presented comparative results and the supporting figures provide satisfactory proof for the statistical dependence between the compared gene pairs. There are many more strongly related gene pairs detected by the compositional correlation method with very low Pearson's correlation values close to zero. For example, the number of gene pairs with a compositional correlation value over 0.8 but Pearson's correlation between -0.2 and 0.2 is 5999. Consequently, the

results of this study provide the yeast researchers a very narrowed down target for defining the functions of the genes, a task which seems to be impossible by using conventional correlation measures that fail to detect real relationships between genes showing alternating but dependent behavior through the course of time.

## 5. COMPARISON OF THE RESULTS WITH EXISTING LITERATURE

A genome-wide association analysis on *Saccharomyces Cerevisiae* is required for both identifying new genes and exploring the extent to which genetic background influences mechanism, because the majority of functional studies on *Saccharomyces Cerevisiae* are carried out in a small number of laboratory strains that do not represent the rich diversity found in this species [34]. Global gene expression of *Saccharomyces cerevisiae* is also investigated in order to identify the correlation between redox potential profiles and gene expression patterns and enables locating genes that could be modulated by altering culture redox potential during VHG ethanol fermentation [35]. Another potential use of the whole-genome sequences is mapping the genetic basis of phenotypic variation through genome-wide association (GWA) studies, with the benefit that associated variants can be studied experimentally with greater ease [36]. Genome-wide comparative analysis are primarily based on genomic sequence information although differences among organisms are often attributed to differential gene expression [37].

Genes whose expression varies differentially and periodically over the cell cycle might be identified by both experimental and computational methods. Aside from the aforementioned uses of genome-wide gene expression analysis, principal-oscillation-pattern (POP) analysis which is a multivariate and systematic technique for identifying the dynamic characteristics of a system from time-series data, can be used to infer oscillation patterns in gene expression [38]. The gene YDL003W is reported by many researches as one of the cell-cycle genes in *Saccharomyces Cerevisiae*. The results obtained in this paper indicate that there are 31 genes determined by the compositional correlation method to have HCC values with the gene YDL003W which are higher than 0.9. These genes are listed in Table 3 together with the HCC,  $r$  and LCC values. The obtained results show that, all genes listed in Table 3 are also reported as cell-cycle genes by several studies. A detailed summary on these methods were provided by de Lichtenberg et al. [39] and the whole lists of the cell-cycle genes reported in these studies is provided by Wang et al. [38]. The findings of the compositional correlation method as presented in this paper show that the compositional correlation method both compares well with existing computational methods and experiments, and it also determines complementary knowledge in addition to information provided by other approaches because it also detected cell-cycle genes not determined by all compared methods. This comparison proves that the compositional correlation method can be used reliably not only in the determination of cell-cycle genes but also other genome wide association studies, but still, the users must be warned against type I errors which should be



checked as always before making final decisions on their association studies.

**Table 2** Comparisons of correlation methods for gene pairs with compositional correlations over 0.9 while  $-0.1 < r < 0.1$

Gene 1	Gene 2	HCC	Pearson	LCC	Spearman	Dist.Cor.	SGMIC
YCL063W	YPL203W	0.926	<b>0.098</b>	<b>0.098</b>	-0.031	0.357	0.546
YGR244C	YKR072C	0.908	0.095	0.095	-0.038	0.363	0.528
YLR109W	YLR441C	0.913	0.094	0.094	-0.028	0.274	0.445
YDR375C	YDR449C	0.906	0.093	0.093	0.026	0.326	0.548
YLL034C	YPL118W	0.908	0.091	0.091	0.029	0.297	0.652
YMR093W	YNL252C	0.905	0.089	0.089	0.126	0.320	0.441
YPR060C	YPR158W	0.904	0.084	0.084	0.124	0.338	0.510
YKL150W	YNL007C	0.941	0.080	-0.028	0.138	0.352	0.667
YKL122C	YLR109W	0.913	0.079	0.079	-0.052	0.301	0.520
YJR054W	YOL052C	0.903	0.079	0.077	0.059	0.295	0.398
YGR244C	YLR075W	0.913	0.078	0.078	0.066	0.357	0.586
YIL093C	YLR197W	0.914	0.078	0.078	0.001	0.419	0.791
YFR050C	YGL189C	0.919	0.078	0.078	0.053	0.350	0.544
YNL005C	YOR095C	0.903	0.076	0.076	0.091	0.302	0.423
YJL063C	YOL097C	0.910	0.073	0.073	0.006	0.353	0.423
YIL070C	YLR344W	0.907	0.072	0.072	-0.036	0.291	0.545
YLR203C	YPL048W	0.916	0.069	0.069	0.072	0.380	0.464
YGL049C	YMR186W	0.905	0.068	0.068	0.048	0.252	0.361
YGL120C	YJL063C	0.901	0.068	0.068	0.015	0.400	0.559
YJL125C	YNL252C	0.936	0.064	0.064	0.019	0.329	0.360
YDR489W	YPR158W	0.903	0.063	0.063	0.065	0.270	0.418
YLR354C	YNL007C	0.912	0.063	0.033	0.133	0.317	0.586
YEL039C	YER027C	0.905	0.058	-0.112	0.192	0.289	0.455
YGR244C	YOR300W	0.902	0.051	0.051	0.062	0.309	0.456
YGR244C	YHR145C	0.936	0.050	0.050	-0.027	0.447	0.735
YNL135C	YOR325W	0.921	0.050	0.050	0.076	0.308	0.490
YER156C	YLR109W	0.920	0.050	0.050	-0.056	<b>0.208</b>	0.316
YBL066C	YDR231C	0.902	0.048	0.048	0.198	0.409	0.464
YMR296C	YOL032W	<b>0.943</b>	0.048	0.048	0.082	0.386	0.566
YCR056W	YOL032W	<b>0.900</b>	0.044	0.044	0.088	0.330	0.453
YLR203C	YPR085C	0.922	0.040	0.040	0.012	0.367	0.321
YIL093C	YLR175W	0.908	0.035	0.035	-0.055	0.386	0.697
YGL219C	YNL007C	0.903	0.031	<b>-0.179</b>	0.132	0.403	0.436
YBL101W-B	YLR069C	0.923	0.028	0.028	0.029	0.366	0.482
YLL026W	YML008C	0.909	0.024	0.024	-0.005	0.288	0.510
YJL063C	YPR062W	0.902	0.023	0.023	-0.146	0.393	0.761
YDL022W	YJL029C	0.919	0.016	0.016	-0.047	0.353	0.667
YDR231C	YLR185W	0.924	0.016	0.016	<b>0.200</b>	0.386	0.588
YCR056W	YPL118W	0.913	0.012	0.012	-0.008	0.290	0.351
YJL063C	YOR300W	0.940	0.007	0.007	-0.052	0.341	0.493
YHR145C	YIL093C	0.929	-0.001	-0.001	-0.135	0.407	0.588
YLL034C	YLR203C	<b>0.900</b>	-0.008	-0.008	-0.029	0.380	0.592
YJL063C	YMR102C	0.912	-0.022	-0.022	-0.037	0.341	0.367
YBR183W	YHR216W	0.911	-0.030	-0.030	0.064	<b>0.508</b>	0.775
YHL035C	YNL007C	0.905	-0.036	-0.177	0.177	0.367	0.586
YGL221C	YNL007C	0.921	-0.040	-0.123	0.113	0.332	0.618
YER049W	YGR048W	0.901	-0.041	-0.041	0.003	0.309	0.463
YBR082C	YOR262W	0.917	-0.044	-0.105	0.044	0.505	<b>0.932</b>
YLR109W	YPR110C	0.925	-0.047	-0.047	-0.115	0.241	0.348
YCL014W	YDL110C	0.901	-0.060	-0.090	<b>-0.204</b>	0.398	0.649
YGR097W	YLL026W	0.904	-0.062	-0.062	-0.002	0.274	0.332
YLR109W	YOR300W	0.911	-0.063	-0.063	-0.087	0.256	0.426
YGR228W	YOR310C	0.904	-0.073	-0.073	-0.110	0.251	<b>0.259</b>
YLR138W	YPL118W	0.917	-0.076	-0.076	-0.116	0.332	0.324
YGR244C	YLR293C	0.924	-0.081	-0.081	-0.144	0.350	0.559
YDR509W	YGR254W	0.908	-0.082	-0.082	-0.187	0.371	0.499
YKL024C	YLR109W	0.907	-0.089	-0.089	-0.176	0.301	0.495
YPL118W	YPR048W	0.906	<b>-0.094</b>	-0.094	-0.127	0.293	0.288
	<b>Max:</b>	0.943	0.098	0.098	0.200	0.508	0.932
	<b>Min:</b>	0.900	-0.094	-0.179	-0.204	0.208	0.259
	<b>Average:</b>	0.913	0.024	0.010	0.009	0.340	0.511

**Table 3** The genes determined to have HCC values with the gene YDL003W higher than 0.9

GENE	HCC	r	LCC	BCC	WCC
YDR097C	0.9928	0.9851	0.9422	7, 4, 8, 4	4, 5, 5, 5, 4
YJL115W	0.9723	0.9454	0.8718	6, 4, 9, 4	4, 5, 5, 5, 4
YGR044C	0.9692	0.8275	0.7788	7, 5, 7, 4	4, 5, 5, 5, 4
YOL017W	0.9618	0.9565	0.8425	7, 6, 5, 5	4, 5, 6, 4, 4
YGR152C	0.9554	0.9220	0.8242	10, 9, 4	4, 5, 5, 5, 4
YKLO45W	0.9514	0.9266	0.7459	6, 6, 6, 5	4, 5, 5, 5, 4
YDL101C	0.9453	0.9246	0.7959	7, 4, 8, 4	4, 5, 5, 9
YOL007C	0.9440	0.9234	0.6680	7, 4, 8, 4	4, 5, 5, 5, 4
YLL002W	0.9423	0.9337	0.6919	8, 4, 6, 5	4, 5, 5, 5, 4
YHR110W	0.9416	0.9296	0.7608	7, 4, 8, 4	4, 5, 5, 5, 4
YJR148W	0.9388	0.8685	0.5928	8, 5, 5, 5	4, 5, 5, 5, 4
YDL211C	0.9387	0.7795	0.5522	4, 4, 11, 4	4, 5, 5, 9
YIL066C	0.9374	0.8455	0.6212	7, 5, 7, 4	4, 5, 5, 9
YIL076W	0.9361	0.7787	0.5249	8, 5, 4, 6	4, 5, 14
YLR049C	0.9358	0.7527	0.5078	7, 4, 8, 4	4, 5, 5, 5, 4
YLR194C	0.9343	0.8638	0.7242	10, 9, 4	4, 4, 6, 5, 4
YDL103C	0.9338	0.8097	0.7082	6, 6, 6, 5	4, 5, 5, 5, 4
YJL074C	0.9321	0.8373	0.6543	5, 8, 6, 4	4, 5, 6, 4, 4
YPL256C	0.9321	0.9066	0.6296	6, 4, 7, 6	4, 5, 5, 9
YLL022C	0.9310	0.8986	0.6626	8, 5, 5, 5	4, 5, 5, 5, 4
YOR114W	0.9305	0.8335	0.5588	8, 5, 6, 4	4, 5, 5, 9
YGR151C	0.9291	0.8723	0.6938	8, 10, 5	4, 5, 5, 5, 4
YLR121C	0.9275	0.8503	0.4697	8, 11, 4	4, 5, 5, 9
YML027W	0.9264	0.8891	0.5626	11, 8, 4	4, 5, 5, 9
YDL127W	0.9248	0.7899	0.6389	4, 7, 7, 5	5, 4, 5, 5, 4
YLR183C	0.9214	0.9156	0.6761	13, 5, 5	4, 5, 5, 5, 4
YOL090W	0.9202	0.9060	0.6185	8, 11, 4	4, 5, 5, 5, 4
YPR175W	0.9188	0.8733	0.7006	5, 5, 4, 4, 5	4, 5, 6, 4, 4
YGR189C	0.9063	0.8866	0.5842	10, 9, 4	4, 5, 5, 5, 4
YLR286C	0.9049	0.7052	0.6415	4, 10, 9	4, 5, 5, 5, 4
YMR029C	0.9042	0.5445	0.4129	6, 4, 4, 4, 5	4, 5, 14

## 6. THE COMPCORR SOFTWARE

The CompCorr software developed in Python for implementing the compositional correlation method is freely provided together with this manuscript. The software accepts an Excel file containing the data series as input and generates a text file as output containing the compositional correlations. The number of compositional correlations calculated for each pair varies according to the length of the data series and the minimum number of accepted values in each part of the compositions. The compositions were determined by using the ruleGen function which generates all interpart restricted compositions of  $n$  by using restriction function  $\sigma$  [40]. For each composition, the compositional correlation is determined by using Equation 4.

## 7. CONCLUSION

The results obtained in this study have shown that the compositional correlation method is very successful in determining linear, nonlinear, direct and indirect relationships between gene expression series and that the method has a great potential of being applied in all areas of science. Comparisons with widely used and well-established correlation methods also validated the results of the study. Taken together, the presented findings could be applied quite reliably in studies aimed at determining the functions of

specific yeast genes for which the functions are still undefined. However, the current results were obtained by using available expressions of 4381 of the yeast genes which are estimated to be around 6000. Therefore, future research might include the remaining genes for finding more relationships.

In the light of the findings on the gene expression data series, it is evident that the method may enable possibilities for numerous important discoveries and will contribute to the improvement of our understanding of correlation as a new way of finding associations. The usefulness and benefit of the compositional correlation method lies in the approach that the variation of average through the observations is considered instead of considering the average of the whole series. The author also hopes that the method and the results presented in this manuscript will also provide important clues and the tools to the biologists trying to find the functions for the genes of many other organisms. The software code developed for implementing the compositional correlation method is freely provided as a supplement together with the manuscript.

## 8. SUPPLEMENTARY MATERIAL

The online Supplementary Material contains all the Figures S1 to S5, the Tables S1 to S12 and the Python code of the CompCorr software. The software is provided under the terms of the GNU Free Documentation License, Version 1.3. The Supplementary Material is available for download in the following link:

<https://www.dropbox.com/s/ai8r590sz2e8aw6/Supplementary.Material.pdf?dl=0>

**Author contributions:** Concept – F.D.; Data Collection &/or Processing – F.D.; Literature Search – F.D.; Writing – F.D.

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

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

## REFERENCES

- [1] H. P. Lovecraft. (1928, February) The Call of Cthulhu. *Weird Tales*. 159-178.
- [2] K. Pearson, "Note on Regression and Inheritance in the Case of Two Parents," *Proceedings of the Royal Society of London*, vol. 58, no. 347-352, pp. 240-242, January 1, 1895 1895, doi: 10.1098/rspl.1895.0041.
- [3] J.-L. Magnard et al., "Biosynthesis of monoterpene scent compounds in roses," *Science*, vol. 349, no. 6243, pp. 81-83, 2015, doi: 10.1126/science.aab0696.
- [4] Y. X. R. Wang, K. Jiang, L. J. Feldman, P. J. Bickel, and H. Huang, "Inferring gene-gene interactions and functional modules using sparse canonical correlation analysis," (in en), *Ann. Appl. Stat.*, vol. 9, no. 1, pp. 300-323, 2015/03 2015, doi: 10.1214/14-AOAS792.
- [5] J. M. Bland and D. G. Altman, "Statistical methods for assessing agreement between two methods of clinical measurement," *Lancet*, vol. 1, no. 8476, pp. 307-310, 1986. [Online]. Available:

- <http://www.scopus.com/inward/record.url?eid=2-s2.0-0022624332&partnerID=40&md5=7814d6e99afa1a58edebf08387536f8c>.
- [6] M. B. I. Lobbes and P. J. Nelemans, "Good correlation does not automatically imply good agreement: The trouble with comparing tumour size by breast MRI versus histopathology," *European Journal of Radiology*, vol. 82, no. 12, pp. e906-e907, 2013, doi: 10.1016/j.ejrad.2013.08.025.
- [7] M. T. Brett, "When is a correlation between non-independent variables "spurious"?", *Oikos*, vol. 105, no. 3, pp. 647-656, 2004, doi: 10.1111/j.0030-1299.2004.12777.x.
- [8] L. Duan, W. N. Street, Y. Liu, S. Xu, and B. Wu, "Selecting the Right Correlation Measure for Binary Data," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 2, p. Article 13, 2014, doi: 10.1145/2637484.
- [9] N. Coffey and J. Hinde, "Analyzing time-course microarray data using functional data analysis - A review," *Statistical Applications in Genetics and Molecular Biology*, Review vol. 10, no. 1, 2011, Art no. 23, doi: 10.2202/1544-6115.1671.
- [10] J. Zhang et al., "Genome-wide association study for flowering time, maturity dates and plant height in early maturing soybean (*Glycine max*) germplasm," *BMC Genomics*, vol. 16, no. 1, p. 217, 2015/03/20 2015, doi: 10.1186/s12864-015-1441-4.
- [11] X. Zhang, F. Zou, and W. Wang, "Efficient algorithms for genome-wide association study," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 4, p. Article 19, 2009, doi: 10.1145/1631162.1631167.
- [12] S. Kumari et al., "Evaluation of Gene Association Methods for Coexpression Network Construction and Biological Knowledge Discovery," *PLoS One*, vol. 7, no. 11, p. e50411, 2012, doi: 10.1371/journal.pone.0050411.
- [13] F. Dikbaş, "A novel two-dimensional correlation coefficient for assessing associations in time series data," *International Journal of Climatology*, vol. 37, no. 11, pp. 4065-4076, 2017, doi: <https://doi.org/10.1002/joc.4998>.
- [14] F. Dikbaş, "A New Two-Dimensional Rank Correlation Coefficient," *Water Resources Management*, vol. 32, no. 5, pp. 1539-1553, 2018/03/01 2018, doi: 10.1007/s11269-017-1886-0.
- [15] S.-J. Chou et al., "Analysis of spatial-temporal gene expression patterns reveals dynamics and regionalization in developing mouse brain," *Sci. Rep.*, vol. 6, no. 1, p. 19274, 2016/01/20 2016, doi: 10.1038/srep19274.
- [16] E. Martinez, K. Yoshihara, H. Kim, G. M. Mills, V. Trevino, and R. G. W. Verhaak, "Comparison of gene expression patterns across 12 tumor types identifies a cancer supercluster characterized by TP53 mutations and cell cycle defects," *Oncogene*, Original Article vol. 34, no. 21, pp. 2732-2740, 05/21/print 2015, doi: 10.1038/onc.2014.216.
- [17] J. A. Bubier et al., "Integration of heterogeneous functional genomics data in gerontology research to find genes and pathway underlying aging across species," *PLoS One*, vol. 14, no. 4, p. e0214523, 2019, doi: 10.1371/journal.pone.0214523.
- [18] D. I. Scheffer, J. Shen, D. P. Corey, and Z. Y. Chen, "Gene expression by mouse inner ear hair cells during development," *Journal of Neuroscience*, vol. 35, no. 16, pp. 6366-6380, 2015, doi: 10.1523/JNEUROSCI.5126-14.2015.
- [19] J. Delfini et al., "Population structure, genetic diversity and genomic selection signatures among a Brazilian common bean germplasm," *Sci. Rep.*, vol. 11, no. 1, p. 2964, 2021/02/03 2021, doi: 10.1038/s41598-021-82437-4.
- [20] A. R. Marderstein, E. R. Davenport, S. Kulm, C. V. Van Hout, O. Elemento, and A. G. Clark, "Leveraging phenotypic variability to identify genetic interactions in human phenotypes," *The American Journal of Human Genetics*, vol. 108, no. 1, pp. 49-67, 2021/01/07/ 2021, doi: <https://doi.org/10.1016/j.ajhg.2020.11.016>.
- [21] M. Perros, "A sustainable model for antibiotics," *Science*, vol. 347, no. 6226, pp. 1062-1064, 2015, doi: 10.1126/science.aaa3048.
- [22] F. Dikbaş, "Compositional Correlation for Detecting Real Associations Among Time Series," in *Academic Researches in Mathematic and Sciences*, Z. Yildirim Ed., 1 ed. Ankara: Gece Kitaplığı, 2018, pp. 27-46.
- [23] S. Heubach and T. Mansour, "Compositions of n with parts in a set," *Congressus Numerantium*, vol. 168, p. 127, 2004.
- [24] G. E. Andrews, *The Theory of Partitions (Encyclopedia of Mathematics and its Applications)*. Cambridge: Cambridge University Press, 1984.
- [25] G. E. Andrews and K. Eriksson, *Integer Partitions*. Cambridge: Cambridge University Press, 2004.
- [26] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*. Oxford university press, 1979.
- [27] J. J. Watkins, *Number theory: a historical approach*. Princeton University Press, 2013.
- [28] A. P. Stakhov, "The golden section in the measurement theory," *Computers and Mathematics with Applications*, vol. 17, no. 4-6, pp. 613-638, 1989, doi: 10.1016/0898-1221(89)90252-6.
- [29] L. Lindroos, "Integer Compositions, Gray Code, and the Fibonacci Sequence," 2012.
- [30] P. T. Spellman et al., "Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, no. 12, pp. 3273-3297, 1998. [Online]. Available: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0031742022&partnerID=40&md5=212944b877cb8836ca1f33a585f0b8c9>.
- [31] D. N. Reshef et al., "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518-1524, 2011, doi: 10.1126/science.1205438.
- [32] V. Subbarayan et al., "Inverse relationship between 15-lipoxygenase-2 and PPAR- $\gamma$  gene expression in normal epithelia compared with tumor epithelia," *Neoplasia*, vol. 7, no. 3, pp. 280-293, 2005, doi: 10.1593/neo.04457.
- [33] Y. Zhang, S. Jia, H. Huang, J. Qiu, and C. Zhou, "A novel algorithm for the precise calculation of the maximal information coefficient," *Sci. Rep.*, Article vol. 4, 2014, Art no. 6662, doi: 10.1038/srep06662.
- [34] M. Sardi et al., "Genome-wide association across *Saccharomyces cerevisiae* strains reveals substantial

- variation in underlying gene requirements for toxin tolerance," *PLoS Genet.*, vol. 14, no. 2, p. e1007217, 2018, doi: 10.1371/journal.pgen.1007217.
- [35] C. G. Liu, Y. H. Lin, and F. W. Bai, "Global gene expression analysis of *Saccharomyces cerevisiae* grown under redox potential-controlled very-high-gravity conditions," (in eng), *Biotechnol J*, vol. 8, no. 11, pp. 1332-40, Nov 2013, doi: 10.1002/biot.201300127.
- [36] C. F. Connelly and J. M. Akey, "On the prospects of whole-genome association mapping in *Saccharomyces cerevisiae*," (in eng), *Genetics*, vol. 191, no. 4, pp. 1345-1353, 2012, doi: 10.1534/genetics.112.141168.
- [37] S. Bergmann, J. Ihmels, and N. Barkai, "Similarities and Differences in Genome-Wide Expression Data of Six Organisms," *PLoS Biol.*, vol. 2, no. 1, p. e9, 2003, doi: 10.1371/journal.pbio.0020009.
- [38] D. Wang, A. Arapostathis, C. O. Wilke, and M. K. Markey, "Principal-Oscillation-Pattern Analysis of Gene Expression," *PLoS One*, vol. 7, no. 1, p. e28805, 2012, doi: 10.1371/journal.pone.0028805.
- [39] U. de Lichtenberg, L. J. Jensen, A. Fausbøll, T. S. Jensen, P. Bork, and S. Brunak, "Comparison of computational methods for the identification of cell cycle-regulated genes," (in eng), *Bioinformatics*, vol. 21, no. 7, pp. 1164-71, Apr 1 2005, doi: 10.1093/bioinformatics/bti093.
- [40] J. Kelleher, *Encoding Partitions as Ascending Compositions*. NUI, 2005 at Department of Computer Science, UCC., 2005.

# Determination of Chopped Fruits Freshness with High Accuracy by Using Electronic Nose

\*<sup>1</sup>Bilge Han Tozlu

<sup>1</sup>Department of Electrical Electronics Engineering, Hitit University, Çorum, Türkiye

[bilgehantozlu@hitit.edu.tr](mailto:bilgehantozlu@hitit.edu.tr), 

## Abstract

Especially for the last 20 years many studies have been made in the field of health, chemistry and food with the electronic nose which is a very popular research area thanks to development of sensor technology. In these studies, high achievements have been obtained in many subjects ranging from disease detection to identification of bacterial species and from determination of food quality to aroma separation. In the study, the variation of the freshness of some fruits prepared for eating was examined with an electronic nose according to the days. Fruits as chopped melon, peach, banana and uncut strawberries were put on plates separately and all of them were stored at the room temperature for 5 days. A low cost electronic nose has been made with the MQ branded 11 gas sensors. Fruits were sniffed 15 times for each day at the same time zone. Sensors' data were transferred to the computer via 2 Arduino Uno microcontroller cards and recorded to computer with an interface program made up in the LabVIEW environment. Then, features were extracted from the obtained data taken from the sensors, and the extracted features were classified by using the k-Nearest Neighbors and Neural Network Classification Algorithms. Best classification accuracies were obtained as; 93.28% for melon, 80.80% for peach, 84.80% for banana and 75.78% for strawberry.

**Keywords:** Electronic nose, fruit freshness, classification

## 1. INTRODUCTION

Electronic nose (e-nose) is a device that can recognize odors which were introduced previously. An e-nose generally consists of the following parts; (a) an odor box that the odor which is desired to be recognised, will enter, (b) a sensor block which exists in this box, composed of gas sensors that they can detect the amounts of gases in the composition of the incoming odor, (c) an analog-digital converter unit which converts the electrical signals of these electrochemical sensors to digital data, (d) a classification algorithm which extracts features of the collected data and classifies, and (e) a computer where these operations are performed.

Today, many scientists generate e-noses to distinguish and recognize odors in health, food and chemical area for commercial or academic purposes. There have been many successful studies in the field of health. The diseases like lung cancer[1, 2] heart diseases[3], respiratory system diseases[4, 5], Alzheimer's and Parkinson's diseases[6, 7] [5,6] and diabetes[8, 9] were diagnosed with high accuracy from the patients' breath by using an e-nose. Electronic nose studies have been carried out in the cosmetic field in order to differentiate real and fake perfume[10] as well as to determine a special perfume[11].

In the food and beverage area; numerous studies have been done in the evaluation of quality of foods or beverages by

using an electronic nose. The quality or aroma of many foods such as tomatoes[12], peaches[13] and beverages such as fruit juice[14], wine[15] and tea[16] have been determined with the e-nose in food industry. Moreover, there are many e-nose studies available on the open literature to determine the shelf life of milk[17] and freshness of food such as fish[18], peach[19], egg[20], etc.

In this study; daily changes of the smell of chopped melon, peach, banana and uncut strawberry have been observed with electronic nose. An electronic nose consisting of 11 gas sensors has been made for this study. The sensor array has been placed in an odor (sample) box. Electrical outputs of the sensors have been converted to digital data via an analog-digital converter and the data have been recorded to the computer. The smells of fruits were recorded for 5 days. Then classification process has been carried out. The output of the study can be useful for commercial kitchens, food factories and electronic kitchen products.

## 2. MATERIAL AND METHOD

### 2.1. Customized Electronic Nose Setup

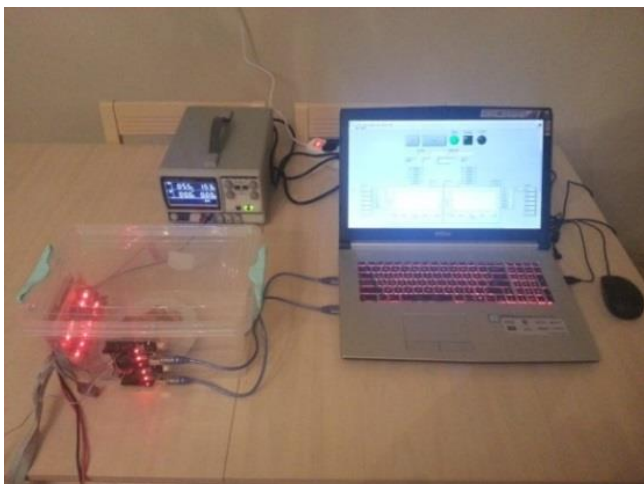
In the study, primarily the sensor block of the electronic nose was generated by using the gas sensors listed in Table1.



**Table 1.** List of Gas Sensors

Sensor No	Sensor Name	Target Gas
1	MQ-2	Methane, Butane, LPG, Smoke
2	MQ-3	Alcohol, Ethanol, Smoke
3	MQ-4	Methane, CNG Gas
4	MQ-5	Natural Gas, LPG
5	MQ-6	LPG, Butane Gas
6	MQ-7	Carbon Monoxide
7	MQ-8	Hydrogen Gas
8	MQ-9	Carbon Monoxide, Flammable Gasses
9	MQ-131	Ozone
10	MQ-135	Air Quality (CO, Ammonia, Benzene, Alcohol, Smoke)
11	MQ-137	Ammonia

MQ branded gas sensors have been used with their kit in this study. The sensor array has been generated by placing these sensors to 6x10 cm board as shown in Figure 1. In Figure 2, the manufactured electronic nose system is given. The sensor array has been placed inside a storage container and the cables are pulled out of an airtight hole. The supply voltages of the sensor kits have been provided from a power supply and the analog outputs of the sensors have been connected to the analog inputs of 2 Arduino Uno cards. The sensors outputs that are electrical voltages, were converted to digital data by Arduino cards and the data have been transferred to the computer via USB port.

**Figure 1.** Sensor array**Figure 2.** Customized electronic nose system

Sensors' data have been recorded with a data collection software which was prepared in LabVIEW. The presence and quantity of the relevant gases in the smell of the fruits which placed in the sample box, have been sensed by the related sensor and recorded to the computer. And obtained data have been classified after features extracted in MATLAB software.

## 2.2. Data Collection and Analysis

In this study; strawberries were washed to be eaten 1 day after they were picked and their daily odors were recorded with an e-nose, while other fruits were chopped on the first day they were bought from the market and their odors from chopping to spoiling was studied. The odors of each fruit were taken 15 times in a day along 5 days. All records were taken between 20.00-23.00 hours under conditions of 22-24 °C temperature and 60%-70% humidity. A sniffing cycle consisted of placing the chopped fruit plate into the previously ventilated sample box, closing the lid of the box and starting the odor recording software program. The period of sniffing cycle was determined as 120 seconds and 10 data were taken per second from the sensors. As a result of 1200 data recorded from each of 11 gas sensors, a data matrix which is consisted 11x1200 data were obtained in a sniffing cycle. A 3 dimensional matrix having 15x11x1200 dimensions were obtained daily for each fruit. End of the 5 days, a total of 300 different odors were recorded, 75 for each fruit.

For the classification process, firstly, Kurtosis, Skewness, Sum, Average, Hilbert Transformation and Variance of the Derivative (VD) methods were applied to 11 sensors' data of each fruit's 75 odor records and the features of these data were extracted. How these features are calculated[21] is given in Table 2.

In the formulas given in Table 2,  $x_i$  is the  $i^{\text{th}}$  sample of a trial  $x$ ,  $\bar{x}$  is the mean value of  $x$ ,  $x'$  is the derivative of  $x$ ,  $x'_i$  is the  $i^{\text{th}}$  sample of  $x'$ ,  $\bar{x}'$  is the mean value of  $x'$ ,  $L$  is the length of a trial,  $a_{\text{mean}}$  and  $a_{\text{std}}$  are the mean and standard deviation of real part of  $\hat{s}(t)$  respectively.

The kNN classification method is an extremely useful method that is frequently used in classification problems. In kNN classification, unknown specimen is classified according to nearest k neighbours. In this method; the metric distances of the training trials to the test trial are calculated, after that nearest neighbors in k quantity are taken into account and the test trial is labeled to be the class of the largest number of these training data. Ideal k number was determined with random sub-sampling method[21].

The neural network classification is also a method that is frequently used in classification studies. In this method; the data in the input layer is processed with the activation functions in the neurons in the hidden layer or layers, and the results of neurons in hidden layer transferred to the output layer. The transitions of the data between the layers are made according to certain weights. Finally, the data coming to the



output layer with the determined weights are assigned to a class as a result of the evaluation made in the output layer. In this study, a single layer is used in the hidden layer.

**Table 2.** Methods of Extracting Features

Kurtosis	$\frac{\frac{1}{L} \sum_{i=1}^L (x_i - \bar{x})^4}{\left(\frac{1}{L} \sum_{i=1}^L (x_i - \bar{x})^2\right)^2}$
Skewness	$\frac{\frac{1}{L} \sum_{i=1}^L (x_i - \bar{x})^3}{\left(\sqrt{\frac{1}{L} \sum_{i=1}^L (x_i - \bar{x})^2}\right)^3}$
Sum	$\sum_{i=1}^L (x_i)$
Mean	$\frac{1}{L} \sum_{i=1}^L (x_i)$
VD	$\frac{1}{L} \sum_{i=1}^L (x'_i - \bar{x}')^2$
Hilbert Transform	$H\{s(t)\} = \hat{s}(t) = s(t) * \frac{1}{\pi t}$ $= \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{s(\tau)}{t - \tau} d\tau = a + jb$ $a_{mean} = \sum_{a=1}^L \frac{a_1 + a_2 + \dots + a_L}{L}$ $a_{std} = \sqrt{\frac{(a_1 - a_{mean})^2 + \dots + (a_L - a_{mean})^2}{L - 1}}$

It was used classification accuracy (CA), sensitivity (SE) and specificity (SF) metrics for evaluating performance of the classifiers. CA is the percentage expression of the ratio of the number of correctly classified trials to the total number of trials (Eq. 1). SE and SF were also calculated as follows (Eq. 2 and Eq. 3):

$$CA = \frac{CCT}{TT} \times 100 \quad (1)$$

$$SE = \frac{TP}{TP + FN} \times 100 \quad (2)$$

$$SF = \frac{TN}{TN + FP} \times 100 \quad (3)$$

CCT, TT, TP, TN, FP, FN are abbreviations of; the number of correctly classified trials, the total number of considered trials, the number of positive samples correctly classified, the number of negative samples correctly classified, the number of positive samples incorrectly classified, the number of negative samples incorrectly classified, respectively.

Features are extracted from the training data set. By using all features, the feature that gives the highest cross validation accuracy (CVA) is determined. Then, the remaining features are added next to this feature by trying all combinations, and the features used for the highest CVA are recorded. This process is done as much as the determined number (100 times for this study) of cross validations with randomly selected sub-training and validation sets. In the cross validation process given in Figure 3, the most used feature and the features which were used more than half of the highest usage count, are selected as the effective features of the classification.

The classification process was carried out in 5 classes for 5 different days. Randomly selected 10 of the 15 data which was recorded in each day, used as training data and the remaining 5 were used as test data. 5 of the 10 training data were used as sub-learning set by selecting randomly, and the other 5 were used as validation data. Feature selection was done as described above. Then, using the determined features, 5 data allocated for the test were classified by kNN and NN classification algorithms.

Thus, one classification process was completed and classification accuracy, sensitivity and specificity were recorded. In order to increase the reliability of the classification accuracy, the classification process mentioned above was performed 100 times for different training-test sets. The arithmetic mean of the results of 100 classifications was accepted as the classification result.

### 3. RESULTS

In this study, four fruits' five days odors have been classified by using an electronic nose. Totally 300 odors 75 for each fruit have been sniffed, and gas sensors' data has been recorded to computer. All sniffing cycles composed of 2 minutes records. 1200 values for each of 11 gas sensors have been got in a sniffing cycle. At the classification stage; 50 of 75 data were used as training data (10 of the 15 per day) and the other 25 of them were classified to 5 classes with kNN and NN classification algorithms. Classification flow diagram is given in Figure 3.

The classification results for each fruit are as follows:

#### For melon;

An average of 93.28% classification accuracy (CA), 85.40% sensitivity (SE) and 97.00% specificity (SF) were achieved at the end of 100 classification cycles with kNN algorithm. By using NN algorithm, 84.53% CA, 66.60% SE and 97.45% SF were achieved at the end of 100 classification cycles. Most effective feature is  $x_{mean}$  of Hilbert Transform of MQ-4 gas sensor. Error matrix of a randomly selected

classification among 100 different classifications is given in Table 3.

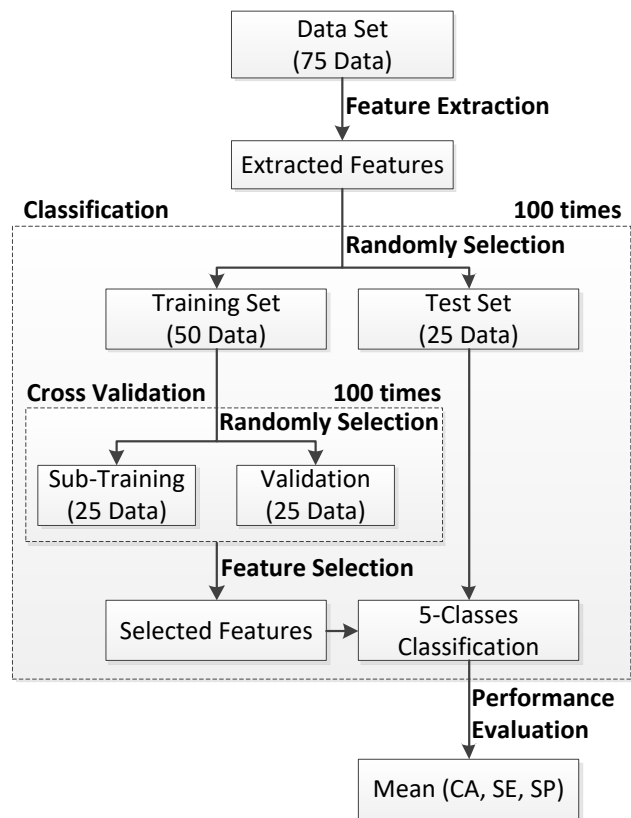


Figure 3. Classification Flow Diagram

Table 3. Error Matrix

		Predicted				
		1.Day	2. Day	3. Day	4. Day	5. Day
Actual	1. Day	4	0	0	0	0
	2. Day	0	5	0	0	0
	3. Day	1	0	5	0	0
	4. Day	0	0	0	4	0
	5. Day	0	0	0	1	5

In Figure 4, extracted features graph is given for random one cycle of 100 cycles. Symbols in given Figure 4-7 represent the days as follow:

- + : 1. Day
- ◇ : 2. Day
- : 3. Day
- \* : 4. Day
- : 5. Day

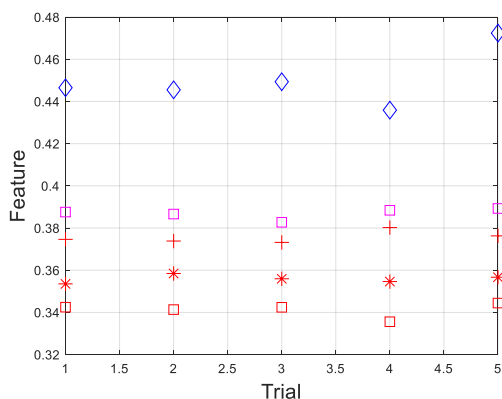


Figure 4. The graph of extracted feature of the chopped melon

**For peach;**

An average of 80.80% CA, 95.00% SE and 93.75% SF were achieved with kNN algorithm and 76.52% CA, 89.00% SE and 91.05% SF were achieved by using NN algorithm at the end of 100 classification cycles. Most effective feature is  $x_{mean}$  of Hilbert Transform of MQ-3 gas sensor's data. Error matrix of a randomly selected classification among 100 different classifications is given in Table 4. In Figure 5, extracted feature graph is given for one classification.

Table 4. Error Matrix

		Predicted				
		1.Day	2. Day	3. Day	4. Day	5. Day
Actual	1. Day	5	1	0	0	0
	2. Day	0	4	0	0	0
	3. Day	1	0	5	0	0
	4. Day	0	0	0	2	1
	5. Day	0	0	0	3	4

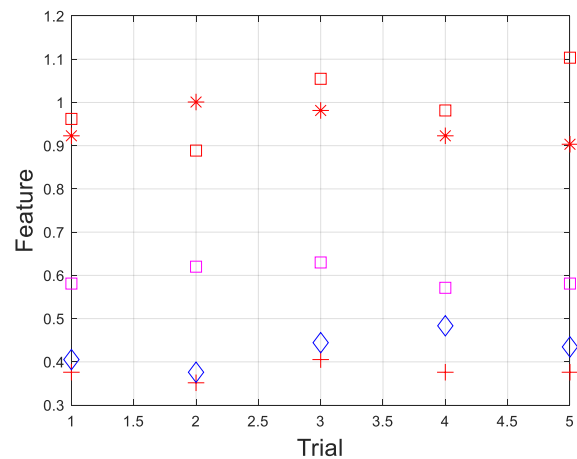


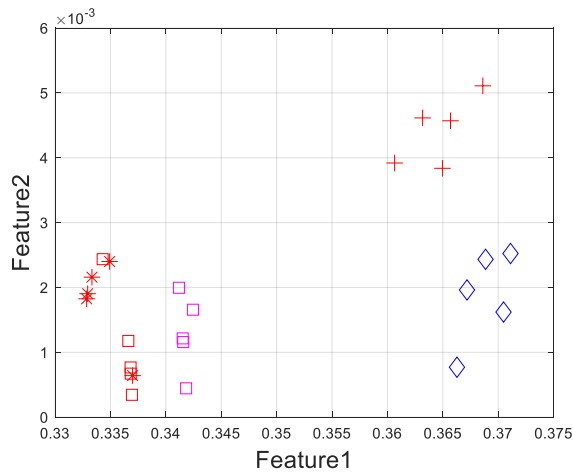
Figure 5. The graph of extracted feature of the chopped peach

**For banana;**

An average of 84.80% CA, 83.40% SE, 96.80% SF were achieved with kNN algorithm, and 75.79% CA, 83.20% SE, 94.95% SF were achieved with NN algorithm at the end of 100 classification cycles. Most effective features are  $x_{mean}$  of Hilbert Transform of MQ-4 gas sensor's data and  $x_{std}$  of Hilbert Transform of MQ-131 gas sensor's data. Error matrix of one among 100 classifications is given in Table 5. Extracted feature graph for one of the classifications is given in Figure 6.

Table 5. Error Matrix

		Predicted				
		1.Day	2. Day	3. Day	4. Day	5. Day
Actual	1. Day	3	0	0	0	0
	2. Day	2	5	0	0	0
	3. Day	1	0	5	0	0
	4. Day	0	0	0	3	0
	5. Day	0	0	0	2	5



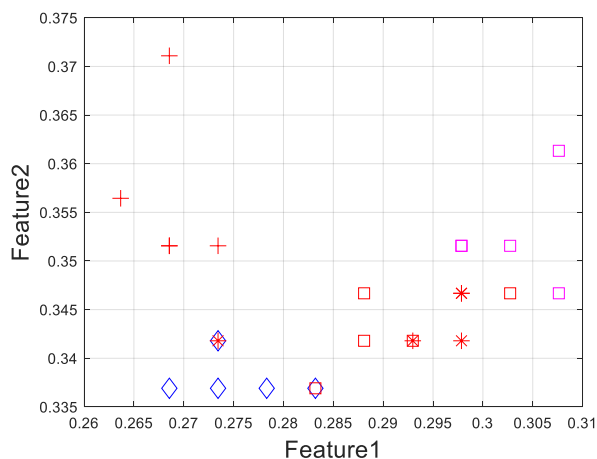
**Figure 6.** The graph of extracted features of the chopped banana

#### For strawberry;

An average of 75.78% CA, 92.80% SE, 96.00% SF were achieved with NN algorithm, and 65.72% CA, 93.00% SE, 97.90% SF were achieved with kNN algorithm at the end of 100 classification cycles. Most effective features are  $x_{\text{mean}}$  and  $x_{\text{std}}$  of Hilbert Transform of MQ-2, MQ-3 and MQ-4 gas sensors' data. Error matrix of one among 100 classifications is given in Table 6. Extracted feature graph for one of the classifications is given in Figure 7.

**Table 6.** Error Matrix

		Predicted				
		1. Day	2. Day	3. Day	4. Day	5. Day
Actual	1. Day	5	0	0	0	0
	2. Day	0	4	0	0	0
	3. Day	0	0	2	0	0
	4. Day	0	1	3	3	2
	5. Day	0	0	0	2	3



**Figure 7.** The graph of extracted features of the strawberry

## 4. DISCUSSION AND CONCLUSION

In this study, the freshness of the fruits was determined with high accuracy with an electronic nose. To start with, a low cost electronic nose was made up. Then, melon, banana and peach were chopped to plates separately and unchopped

strawberries placed to another plate, and these fruits were kept in room temperature. These fruits' odors were sniffed with the e-nose in the same time zone (between 20.00-23.00 hours) for 5 days. A total of 75 odor data for each fruit, 15 per day, were recorded into the computer. Finally, 25 of data were classified with two classifier algorithms by introducing randomly selected 50 of data as training data, after extracting features. The success rates and related classifier types of classifications, which are more successful are given in Table 7.

**Table 7.** Classifications Summary

Fruit	CA (%)	SE (%)	SF (%)	Classification Algorithm
Melon	93.28	85.40	97.00	kNN
Peach	80.80	95.00	93.75	kNN
Banana	84.80	83.40	96.80	kNN
Strawberry	75.78	92.80	96.00	NN

While the kNN classification algorithm can detect freshness and spoilage in melon, peach and banana fruits with high accuracy, it was not very successful in strawberry. Strawberry classification could be done with NN with less success than other fruits. The reason of the low classification performance in strawberry is considered to be that it is not chopped. As a matter of fact, it has been scientifically proven that chopped fruits will lose their freshness faster and spoil earlier[22]. It is predicted that strawberry freshness determination will be made with higher accuracy with e-nose by using chopped strawberries in the new studies in this field.

The information of how many days ago the fruit was chopped can be useful for commercial kitchens, food factories and perhaps smart refrigerators to be developed. So, the study has to be developed by increasing the variety of fruits and vegetables. In addition, in scientific studies, it has been suggested to use electronic gas sensors instead of commercial mass spectrometers or gas chromatography in detecting the freshness of fruits [22]. Undoubtedly, the odors emitted by the fruits change daily and these can be detected by the gas chromatography method. However, the gas chromatography method is quite costly, laborious and long-term work compared to the electronic nose method. With the electronic nose, a situation can be detected so quickly from an odor, and this method can be integrated into industry very easily.

This study has some limitations: The low number of data is one of the negative aspects of the study. Although high accuracy classification results show that the spoilage of fruits can be monitored daily by their fragrances, working with larger data sets in future studies will further reinforce the reliability of the method. The other limitation is lack of gas chromatography of fruit spoiling before this study for choosing related gas sensors.

Presentation at a meeting: A part of this study has been shared as an oral presentation at 3rd International Conference on Advanced Engineering Technologies (ICADET) in Bayburt, TURKEY on 19-21 September 2019


## REFERENCES

- [1] A. D'Amico vd., "An investigation on electronic nose diagnosis of lung cancer", *Lung Cancer*, c. 68, sy 2, ss. 170-176, May. 2010, doi: 10.1016/j.lungcan.2009.11.003.
- [2] R. F. Machado vd., "Detection of Lung Cancer by Sensor Array Analyses of Exhaled Breath", *Am J Respir Crit Care Med*, c. 171, sy 11, ss. 1286-1291, Haz. 2005, doi: 10.1164/rccm.200409-1184OC.
- [3] B. H. Tozlu, C. Şimşek, O. Aydemir, ve Y. Karavelioglu, "A High performance electronic nose system for the recognition of myocardial infarction and coronary artery diseases", *Biomedical Signal Processing and Control*, c. 64, s. 102247, Şub. 2021, doi: 10.1016/j.bspc.2020.102247.
- [4] S. Scarlata, G. Pennazza, M. Santonico, C. Pedone, ve R. A. Incalzi, "Exhaled breath analysis by electronic nose in respiratory diseases", *Expert Review of Molecular Diagnostics*, c. 15, sy 7, ss. 933-956, Tem. 2015, doi: 10.1586/14737159.2015.1043895.
- [5] N. Fens vd., "Exhaled breath profiling enables discrimination of chronic obstructive pulmonary disease and asthma", *Am. J. Respir. Crit. Care Med.*, c. 180, sy 11, ss. 1076-1082, Ara. 2009, doi: 10.1164/rccm.200906-0939OC.
- [6] J.-P. Bach vd., "Measuring Compounds in Exhaled Air to Detect Alzheimer's Disease and Parkinson's Disease", *PLOS ONE*, c. 10, sy 7, s. e0132227, Tem. 2015, doi: 10.1371/journal.pone.0132227.
- [7] U. Tisch vd., "Detection of Alzheimer's and Parkinson's disease from exhaled breath using nanomaterial-based sensors", *Nanomedicine*, c. 8, sy 1, ss. 43-56, Eki. 2012, doi: 10.2217/nnm.12.105.
- [8] S. Esfahani, A. Wicaksono, E. Mozdiak, R. P. Arasaradnam, ve J. A. Covington, "Non-Invasive Diagnosis of Diabetes by Volatile Organic Compounds in Urine Using FAIMS and Fox4000 Electronic Nose", *Biosensors (Basel)*, c. 8, sy 4, Ara. 2018, doi: 10.3390/bios8040121.
- [9] A. Bermak ve M. Hassan, "Noninvasive Diabetes Monitoring with Electronic Nose", Mar. 2016, c. 2016, s. HBPP2776. doi: 10.5339/qfarc.2016.HBPP2776.
- [10] J. Gebicki, B. Szulczynski, ve M. Kaminski, "Determination of authenticity of brand perfume using electronic nose prototypes", *Meas. Sci. Technol.*, c. 26, sy 12, s. 125103, Eki. 2015, doi: 10.1088/0957-0233/26/12/125103.
- [11] A. Carrasco, C. Saby, ve P. Bernadet, "Discrimination of Yves Saint Laurent perfumes by an electronic nose", *Flavour and Fragrance Journal*, c. 13, sy 5, ss. 335-348, Eyl. 1998.
- [12] X. Huang, S. Pan, Z. Sun, Y. Wei-tao, ve J. H. Aheto, "Evaluating quality of tomato during storage using fusion information of computer vision and electronic nose", *Ağu. 2018*, [Çevrimiçi]. Erişim adresi: <https://doi.org/10.1111/jfpe.12832>
- [13] "Evaluation of peach quality indices using an electronic nose by MLR, QPST and BP network", *Sensors and Actuators B: Chemical*, c. 134, sy 1, ss. 332-338, Ağu. 2008, doi: 10.1016/j.snb.2008.05.008.
- [14] "Qualification and quantisation of processed strawberry juice based on electronic nose and tongue", *LWT - Food Science and Technology*, c. 60, sy 1, ss. 115-123, Oca. 2015, doi: 10.1016/j.lwt.2014.08.041.
- [15] M. Aleixandre, J. M. Cabellos, T. Arroyo, ve M. C. Horrillo, "Quantification of Wine Mixtures with an Electronic Nose and a Human Panel", *Front. Bioeng. Biotechnol.*, c. 6, 2018, doi: 10.3389/fbioe.2018.00014.
- [16] B. Tozlu, H. I. Okumus, ve C. Simsek, "Online Quality Classifying With Electronic Nose For Black Tea Production.", *International Journal of Academic Research*, c. 6, sy 4, 2014.
- [17] S. Labreche, S. Bazzo, S. Cade, ve E. Chanie, "Shelf life determination by electronic nose: application to milk", *Sensors and Actuators B: Chemical*, c. 106, sy 1, ss. 199-206, Nis. 2005, doi: 10.1016/j.snb.2004.06.027.
- [18] S. Güney ve A. Atasoy, "Freshness Classification of Horse Mackerels with E-Nose System Using Hybrid Binary Decision Tree Structure", *Int. J. Patt. Recogn. Artif. Intell.*, c. 34, sy 03, s. 2050003, May. 2019, doi: 10.1142/S0218001420500032.
- [19] "Study of peach freshness predictive method based on electronic nose", *Food Control*, c. 28, sy 1, ss. 25-32, Kas. 2012, doi: 10.1016/j.foodcont.2012.04.025.
- [20] R. Dutta, E. L. Hines, J. W. Gardner, D. D. Udrea, ve P. Boilot, "Non-destructive egg freshness determination: an electronic nose based approach", *Meas. Sci. Technol.*, c. 14, sy 2, ss. 190-198, Oca. 2003, doi: 10.1088/0957-0233/14/2/306.
- [21] E. Ergün ve Ö. Aydemir, "Decoding of Binary Mental Arithmetic Based Near-Infrared Spectroscopy Signals", içinde 2018 3rd International Conference on Computer Science and Engineering (UBMK), Eyl. 2018, ss. 201-204. doi: 10.1109/UBMK.2018.8566462.
- [22] A. Ceccarelli vd., "Nectarine volatilome response to fresh-cutting and storage", *Postharvest Biology and Technology*, c. 159, s. 111020, Oca. 2020, doi: 10.1016/j.postharvbio.2019.111020.


# Realization of the Autonomous Driving System on the Experimental Vehicle

\*<sup>1</sup>Namig Aliyev, <sup>2</sup>Mehmet Turan Guzel, <sup>3</sup>Oguzhan Sezer


<sup>1</sup>Department of Computer Engineering, Sakarya University, Sakarya, Turkey,

[namig.aliyev@ogr.sakarya.edu.tr](mailto:namig.aliyev@ogr.sakarya.edu.tr) 

<sup>2</sup> Department of Computer Engineering, Sakarya University, Sakarya, Turkey,

[mehmet.guzel@ogr.sakarya.edu.tr](mailto:mehmet.guzel@ogr.sakarya.edu.tr) 

<sup>3</sup> Department of Computer Engineering, Sakarya University, Sakarya, Turkey,

[oguzhan.sezer1@ogr.sakarya.edu.tr](mailto:oguzhan.sezer1@ogr.sakarya.edu.tr) 

## Abstract

Running control software on limited computing resources is considered one of the toughest problems. In this study, an autonomous driving software has been developed that can safely complete the map by tracking the lanes and avoiding obstacles on a robot vehicle with limited hardware components. The data was simplified with the image processing technique and the neural network was trained. Overfitting was prevented by hyperparameter tuning and synthetic data augmentation. In order to avoid obstacles, optical flow was calculated by detecting corners every 4 seconds and was used to find the focus of expansion of the vehicle. Time-to-collision was found with the FOE and the distance between the previous position and the current position of the detected point. Optimization was made by averaging the values of close points. The balance mechanism was created according to the TTC difference calculated on the right and left parts of the vehicle.

**Keywords:** Convolutional Neural Network, Overfitting, Hyperparameter Tuning, Data augmentation, Lane tracking, Optical Flow, Focus of Expansion, Time to Collision

## 1. INTRODUCTION

The autonomous driving system is one of the most popular smart autonomous systems recently. Nowadays, it is aimed to minimize driver-related errors with autonomous driving systems. Today, we can say that autonomous driving systems have speed control with radar and distance sensors, lane tracking, and lane change after cameras on the vehicle. Autonomous vehicles are one of the most effective use cases where hardware and software work together. The hardware enables the vehicle to move and communicate with a range of cameras, sensors, while the software processes information and provides control.

Today, many automobile companies are attempting to produce cars with autonomous driving systems. We can say Tesla company as the leading company. The cars they produce have a full automation driving system. The data set is collected in real-time from approximately 1 million vehicles. 70,000 GPU's are trained per hour. It is capable of semantic segmentation, object recognition, depth estimation. There are 1000 different estimates per step each time. Some companies use the LIDAR device to model depth prediction and 3D perception. Depth prediction is a fundamental task in perceiving the 3D environment around us [1].

In this study, lane tracking, which is one of the two most important abilities in autonomous vehicles, and the ability to avoid obstacles for the robot vehicle to drive freely without hitting any obstacle are discussed. The main purpose of this research was to develop lane tracking and obstacle avoidance capabilities with different methods and solutions for an autonomous driving system on an experimental vehicle.

The robot vehicle, remote control module, and experimental map that constitute the hardware part of the project were prepared. Raspberry Pi module on the vehicle forms the brain of the vehicle. Raspberry PI communicates with the remote-control module via wireless network (RF24) and computer via embedded software. The Raspberry PI module, which plays the role of the brain of the system in the later stages of the project, was renewed with the Coral Dev Board [2] device developed by Google for artificial intelligence model's due to its inadequate performance.

Supervised Learning [3, 4] approach, which is one of the Machine Learning [5, 6, 30] techniques, was used for lane tracking. With the help of a remote-control device, the robot vehicle was moved along the track and dataset collection was carried out through the camera on the vehicle. This dataset created consists of images and action information taken at

\* Corresponding Author

the time of that image. Then, the images taken were simplified with image processing techniques, and the strip lines were brought to the fore. At this stage, Convolutional Neural Network [7] was used while creating an artificial intelligence model. CNN is a type of artificial neural network developed to solve problems such as image classification, object detection, and style transfer. Since the images are our main data source, it was decided to use CNN.

The target problem for avoiding obstacles is the calculation of contact time or time to collision. The most important feature focused here was the calculation of the time until the collision, ie the contact time. In this direction, corner detection, optical flow focus of expansion, and collision time were calculated instantaneously on the image taken from the camera [8 - 10]. The balance calculation has been made for the right and left body of the robot vehicle and a decision mechanism has been created to avoid obstacles. The robot vehicle has been provided to move without hitting any obstacle.

There are many studies on road lane tracking in the literature. Bounini and Farid [11] obtained a result by detecting corners on the data taken from the camera image. J. Han, D. KIM [12] using 2D Lidar sensors, they were able to gather information about the environment and keep the vehicle within the lane by performing road boundary extraction. There are a few studies investigating vehicle obstacle avoidance using only information extracted from the camera image. Kachluche Souhila and Achour Karim [13] measured the distance to objects using optical flow and corner detection.

## 2. PROPOSED METHOD

### 2.1. Lane Tracking

In this section, simplification of lane information with computer vision techniques, and data set collection are given initially to enable the robot vehicle to move autonomously by following the lane information on the experimental map. Next, a detailed description of the designed network architecture and training is provided. To achieve the successful model, hyper parameter tuning and data augmentation, and finally, the testing process is explained.

The dataset collection process will be performed by moving the robot vehicle over the experimental environment with the help of a remote control. The images taken from the camera correctly positioned on the vehicle will first be recorded in the filing system by simplifying the lane information with image processing techniques. At the same time, the action information of the car at the time the image is taken is recorded in the filing system simultaneously with the images.

#### 2.1.1. Simplifying Lane Information with Computer Vision Techniques

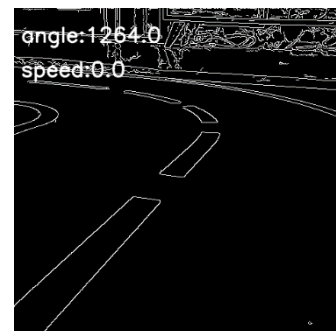
Preparing the images in the data set that we will give to the neural network in accordance with the purpose is the most influential factor in the result of the developed neural network model. If the image is messy, difficult to

understand, and the neural network is not able to distinguish the features in the image, the error values of the model will be high and the operation is nothing but a waste of time. For this purpose, the images taken from the camera on the robot vehicle were first simplified with image processing techniques. First, the color space change was performed. It is planned to increase frames per second in the future and add new capabilities using 2D Lidar and Google Coral Dev Board. It is planned to increase frames per second in the future and add new capabilities using 2D Lidar and Google Coral Dev Board. It is planned to increase frames per second in the future and add new capabilities using 2D Lidar and Google Coral Dev Board. med on the image. Many color spaces are supported in the OpenCV library and you can convert between them. In the first step, the image was converted from RGB color space to grayscale color space [14].



**Figure 1.** Change from R.G.B. color space to grayscale color space

Figure 1. shows the image obtained by converting the camera image taken on the robot vehicle from RGB colour space to grayscale colour space. In the image in the grayscale colour space obtained, the stripe lines are desired to be prominent. With the help of the Canny [15] edge detection algorithm, the strip lines required on the image were made more prominent.



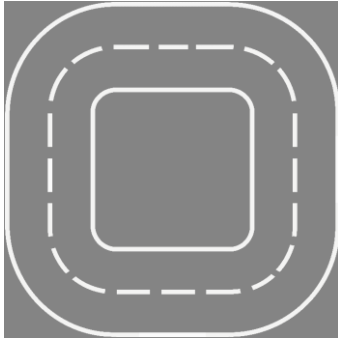
**Figure 2.** Transformation of grayscale image with Canny edge detection algorithm

The image taken from the camera has been successfully simplified and made ready for the use of the neural network. If you pay attention to the upper left corner of the figure, you can see the angle and speed, which are the action information of the vehicle at the time the image is taken from the camera.



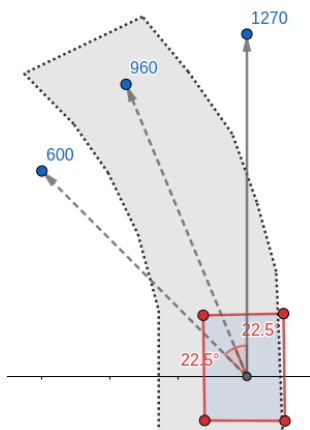
### 2.1.2. Data Set Collection and Editing

The collection of the data set will be carried out by moving the robot vehicle over the experimental environment with the remote control. In the Supervised Learning machine learning approach we will use, the data to be trained should be given to the learner as  $x$  and  $y$  outputs. While the vehicle is controlled remotely on the experimental environment we have prepared before, the image from the camera, the current servo angle and engine speed are recorded in the filing system simultaneously.



**Figure 3.** Experimental map on which the robot vehicle will be moved

In the Supervised Learning approach, the data should be given to the trainer as  $x$  and  $y$  outputs. While the robot vehicle was being moved over the experimental environment, the servo angle information, which is the current action information, was recorded in the filing system along with the camera image. According to the general structure of the experimental map, the servo  $x$  angle values in the data collected vary between 1270 and 600.



**Figure 4.** The angles the robot vehicle takes at the moment of movement.

Angle data collected at this stage has a complex structure and needs to be simplified. For this purpose, while the robot vehicle is moving on the experimental map, the angle information as *FLAT*, *MIDDLE*, and *SHARP* is updated in the filing system by simultaneously looking at its location on the map and the instant angle information from the computer. Alternatively, the angle information, which is a parameter of the data set, can be compressed between 0 and 1 for linear regression [16, 17], allowing linear estimation.

### 2.1.3. CNN Neural Network Model

While the robot vehicle is in motion, it should analyze the environmental conditions and make control predictions. Environmental conditions consist of data collected in the previous topic. The robot vehicle needs a system that can use this data and make predictions.

Artificial neural networks are Artificial Intelligence structures that are trained with the given data and can make predictions according to the information they learn.

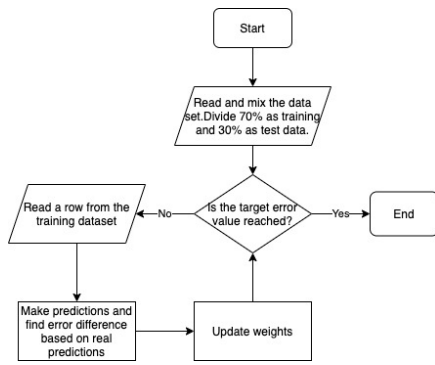
In this subsection, the design of the architecture and the training of the model presented initially. Next, the hyper parameter tuning [18-19], and data augmentation [20], and finally the testing and result are explained in detail.

### 2.1.4. Designing and Training the Model

Before creating a neural network model, the neural network structure to be used is decided by considering the data set, project conditions and properties. The main source of data consists of images. The neural network is required to be predicted according to the images and action information taken from the camera. The neural network will distinguish the features in the images taken from the camera and perform the learning and prediction processes. Therefore, it was decided to use convolutional neural networks at this stage of the project.

Data set consists of binary color pictures simplified with Canny edge detection algorithm and angle information, which is the action information at the time the picture is taken. The stored images were resized to 128 x 128 pixels before being transferred to the model. The action information, *FLAT*, *MIDDLE* and *SHARP*, are updated to correspond to 0, 1, and 2, respectively. Due to the general structure of neural networks, the complexity of the structure is directly proportional to the estimation time. Therefore, it is important that the model to be designed has a simple structure. On the other hand, the education period of the models with a simple structure is short and time saving is obtained. Another issue in neural networks is that there are no rules for establishing the best model. For this reason, until we find the model with which we have achieved high performance, the models have been designed by taking the available data into consideration.

The steps to be taken during the training of our artificial neural network model are as follows. The first step is to read and store the data set and mix it randomly. The second step is to separate 70% of the data set as training data and 30% as test data. After separating the training and test data set, an image from the training data set is given to our model and the weights are updated according to the error value. Figure 5. shows the flow chart representing the training process of the model.



**Figure 5.** The algorithm flow chart representing the training process of neural network model.

The first layer of the first model prepared is a convolution layer with 32 x 32 depth and 3 x 3 filter dimensions. Input data is 128 x 128 x 2 size simplified image with Canny edge detection algorithm. Relu, the next layer activation function, has been applied. Two Max-Pooling layers were then applied. Filter dimensions of the Max-Pooling layers are determined as 2 x 2. By applying Max-Pooling layers in succession, which yields a feature map of the 32 x 32 x 32 size. Next, the Flatten and Dense layer added.[21].

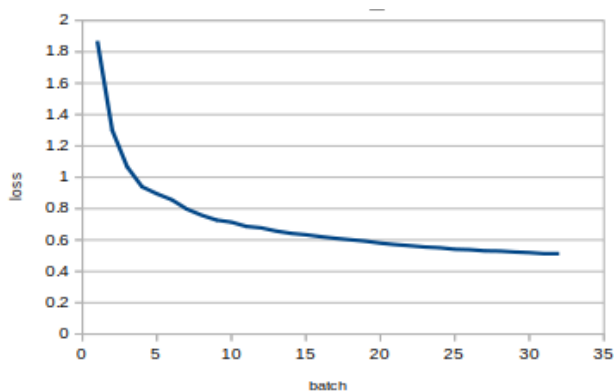
	Name	Type	Shape
0	conv2d_48	Conv2D	(None, 128, 128, 32)
1	activation_24	Activation	(None, 128, 128, 32)
2	max_pooling2d_30	MaxPooling2D	(None, 64, 64, 32)
3	max_pooling2d_31	MaxPooling2D	(None, 32, 32, 32)
4	flatten_26	Flatten	(None, 32768)
5	dense_25	Dense	(None, 3)

-----

Total params: 98915

**Figure 6.** First CNN model summary.

Considering the Raspberry PI module performance, the first model was kept simple and the total number of calculated parameters was obtained as 98915.



**Figure 7.** Loss graph of the first CNN model

When the loss graph in Figure 7. is examined, it is seen that during the batch of 32 pictures each, it goes to overfitting quickly [22, 23]. It has been observed that the loss of the neural network model rapidly approaches zero at the end of one epoch.

The result we will get here is that the dropout layer used to reduce overfitting is insufficient. In order to eliminate this

problem caused by the fact that the dataset consists of few and similar images, data diversity will be increased by data augmentation. Thus, a more general model that can respond to real-life problems will be obtained by considering parameters such as lighting conditions and noise in the image.

### 2.1.5. Hyperparameter Tuning and Data Augmentation

While designing a model in artificial neural networks, there is no rule to reach a successful model. There is no rule to be followed in line with the information obtained from the studies conducted on this subject in the world so far. The improvement of the model is done by techniques such as trial-and-error method, hyperparameter tuning [18] and data augmentation [24].

One of the hyperparameter adjustment is to prevent overfitting. The dropout layer which have been added with 0.25 value to the model is a regularization approach [21] that helps reduce dependent learning between neurons. Another hyperparameter regulation is increase the computable parameter counts. The Max-Polling layer has been removed and a new convolution layer of 16 x 16 depth and 3 x 3 filter dimensions has been added.

	Name	Type	Shape
0	conv2d_49	Conv2D	(None, 128, 128, 32)
1	conv2d_50	Conv2D	(None, 128, 128, 16)
2	max_pooling2d_32	MaxPooling2D	(None, 64, 64, 16)
3	flatten_27	Flatten	(None, 65536)
4	dropout_25	Dropout	(None, 65536)
5	dense_26	Dense	(None, 3)

-----

Total params: 201843

**Figure 8.** Third neural network with hyperparameters tuned.

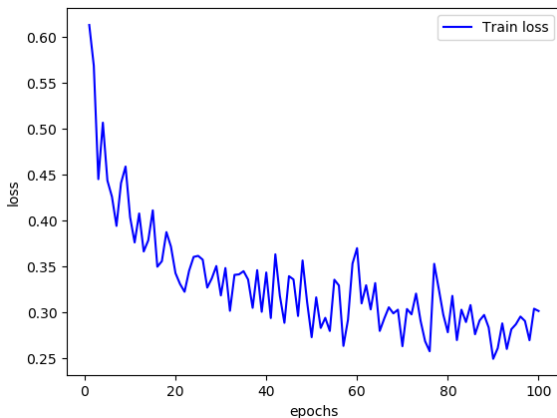
The total number of parameters calculated after tuning was obtained as 201843. The data given to the model were augmented by producing synthetic data with the data augmentation technique [24].



**Figure 9.** Synthetic image created through data augmentation.

Figure 9. shows the synthetic image obtained after applying flip, shift, and zoom to the real image. These variations in the data set enable the trained model to achieve a similar performance in different image conditions. Thus, a more

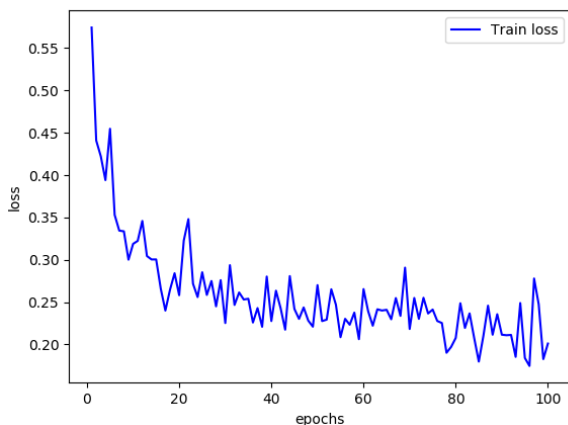
general solution will be achieved. The loss graph of the model trained after data augmentation is shown in Figure 10.



**Figure 10.** Loss graph of the model trained after data augmentation

There is an obvious improvement in overfitting [23] rate compared to previous training. It is seen that the loss ratio that converges rapidly to zero at the batch level before is now decreasing at the epoch level. At the end of 100 epoch, the lowest loss value was reached as 0.25 in the 90th epoch.

New synthetic data were generated by making changes in the augmentation parameters to reduce the loss value even smaller. Zoom ratio decreased from 0.4 to 0.2, flip angle from 40 degrees to 10 degrees. The changes applied here will make less distortion of the lane information in the image and help achieve the goal of obtaining a more general model. The highest performing neural network model has been retrained with the new dataset, and the expected reduction in loss data occurred.

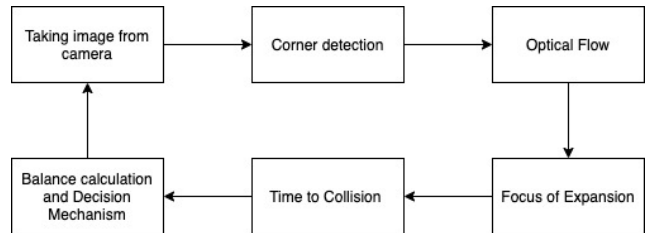


**Figure 11.** Loss graph of the most successful CNN model.

As seen in the graph, the 95th epoch has reached 0.17 loss value. Thus, it was observed that the change made in the data augmentation parameters had an effect on the decrease of the loss value. The final model was trained three times over 100 epochs with synthetic data. The number of frames per second, which represents the reaction speed of the vehicle, reached the maximum 14 frames per second which is the best result of all time.

## 2.2. Obstacle Avoidance with Optical Flow

While an autonomous robot vehicle is moving in a constant velocity, the time until the collision can be found without any knowledge of the distance to be traveled or the velocity the robot is moving [8]. Calculating the time to collision is one of the practical optical flow uses. The optical flow knowledge is extracted from the image sequence taken from the Google camera placed in the robotic vehicle, and then the time until the robot reaches a particular area is determined. Calculated collision times are considered separately as collision times on the left and right of the image. Depending on whether the difference between the collision times of the left and right side is higher or lower than a certain threshold value, the vehicle is ordered to ignore the obstacle in front of it or to take action.



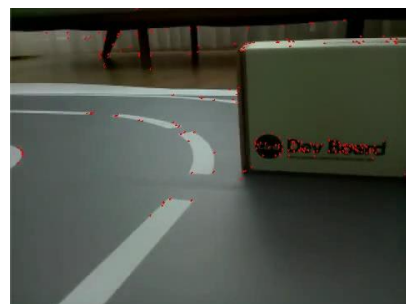
**Figure 12.** The flow char representing obstacle avoidance procedure [26].

In this section, corner detection, and calculating optical flow are introduced in the first place. After, the focus of expansion and time to collision calculation procedures are explained. Next, the balance strategy and decision mechanism are explained in detail and the movement of the vehicle is presented according to the decision produced by the mechanism.

### 2.2.1. Corner Detection with FAST (Features from Accelerated Segment Test)

It is necessary to extract the optical flow information from the image sequence taken from the camera. To find the optical flow between consecutive frames, the motion of a pixel feature set should be tracked. Features in the image are points of interest that provide rich picture content information, and these points are not affected by intensity changes in the image [27].

Using the FAST [26] algorithm, which is known for its high performance in real-time images, corner detection performed in the real-time image sequence.

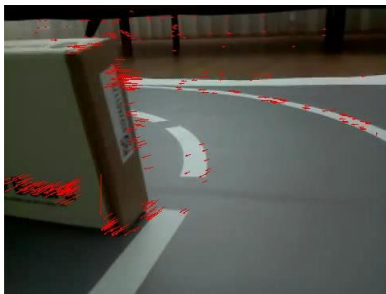


**Figure 13.** Corner detection on the image with FAST algorithm.

Figure 13. shows the image formed after applying the corner detection algorithm on the image. The corner detection process is run every 50th iteration of the runtime, which means that the corners are refreshed at approximately 3-4 second intervals. The detected corners are stored on a vector for later use in calculating the optical flow.

### 2.2.2. Calculating Optical Flow

For the optical flow to be computable, a selected point on the first image must change its location on the next image. While the selected point is moving, the shape of the light reflected on that point is constantly changing and optical flow occurs. In other words, the vehicle must be moving in order to obtain optical flow with the robot vehicle. The most widely used Lucas-Kanade [27] method was used to calculate the optical flow between consecutive frames. The vector containing the vertices detected by the FAST corner detection algorithm is given to the function and it returns two vectors containing the (x, y) coordinates of the previous and next points. Now that the changing coordinates of a corner point in the previous and ongoing frame are known, an arrow can be drawn from the previous position to the next position. In other words, an arrow is drawn in the direction of the point's movement in consecutive frames if the tracked corner point exists (detected) in the next frame. Suppose  $(x_2, y_1)$  and  $(x_2, y_2)$  are the coordinates of the point in the previous and next squares.

$$angle = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (1)$$


**Figure 14.** Arrows were drawn in the direction of movement of the points.

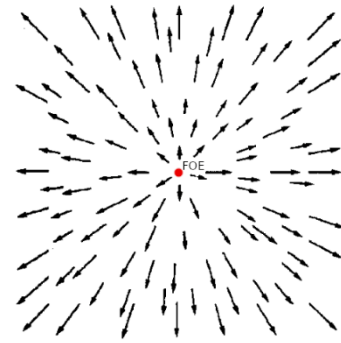
$$\begin{aligned} arrow_x &= x_2 + len * \cos\left(angle + \frac{3.14}{180}\right) \\ arrow_y &= y_2 + len * \sin\left(angle + \frac{3.14}{180}\right) \end{aligned} \quad (2)$$

### 2.2.3. Calculating Focus of Expansion

The motions of objects moving around are projected to the eyes of the observer as two fundamental motions. An optical flow field is formed as a result of the projection of the translation and rotation fundamental motions into an image plane [8]. Rotational motion can be imagine as flow vectors produced as a result of the surrounding objects shifting left or right as the robot vehicle turns left or right.

Translation motion occurs when the camera is moving forward or backward. If the camera moves backward, it creates an area called a focus of contraction (FOC) where the flow vectors converging around a point. On the contrary, if it moves forward, it creates an area called the focus of

expansion (FOE) where the flow vectors diverge around from a central point.



**Figure 15.** Diverging flow vectors and focus of expansion during forward translation motion.

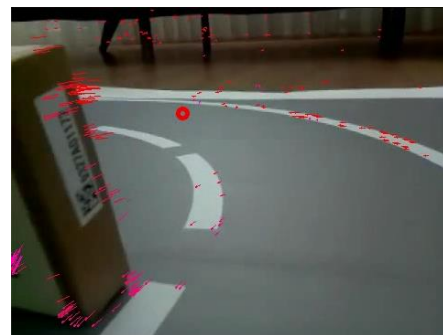
Any two vectors are needed to calculate the focus of expansion. If the place where these two vectors meet can be determined, the focus of expansion is found. The least-squares [28] solution of all available flow vectors was used to find focus of expansion. Each optical flow vector has a previous point and delta. Let  $pt = (x, y)$  be the x and y coordinates of the previous position of an optical flow vector. Let  $v = (u, v)$  be the x and y coordinate differences between the previous and current position of the optical flow vector.

$$A = \begin{bmatrix} a_{00} & a_{01} \\ \dots & \dots \\ a_{n0} & a_{n1} \end{bmatrix} \quad b = \begin{bmatrix} b_0 \\ \dots \\ b_n \end{bmatrix} \quad (3)$$

In matrix A it should be known as  $a_{i0} = -v$  and  $a_{i1} = u$ , and in matrix B, each value is obtained by  $b_i = xv - yu$ . The focus of expansion is calculated using the least-squares method and inversion of the matrices.

$$\begin{aligned} FOE &= (A^T A)^{-1} A^T b \\ &= \begin{bmatrix} \sum a_{i0} b_i \sum a_{j1}^2 - \sum a_{j1} b_i \sum a_{j0} a_{j1} \\ -\sum a_{i0} b_i \sum a_{j0} a_{j1} + \sum a_{i1} b_i \sum a_{j0}^2 \end{bmatrix} - \frac{1}{\sum a_{j0}^2 a_{j1}^2 - (\sum a_{i0} a_{i0})^2} \end{aligned} \quad (4)$$

The OpenCV library has the necessary functionality for the matrix inversion method. The function is given matrices A and b and an empty matrix of 2x1 dimensions. Additionally, the DECOMP\_QR flag was added for QR decomposition [29].



**Figure 16.** Calculation of the focus of expansion on the image taken from the camera. The FOE is shown by the red circle in the image.



### 2.2.4. Calculating Time to Collision

The most valuable information we can obtain for the robot vehicle to avoid obstacles is the determination of the contact time or the time until the collision. This information can be found without requiring any information about the distance to travel or the velocity the robot is moving [8].

The studies carried out up to this stage were to obtain information to be used in the TTC because when calculating the TTC, optical flow vectors and FOE are required. Let  $p = (x, y)$  be the x and y positions of an optical flow vector and  $FOE = (x, y)$  be the x and y positions of a focus of expansion. Let  $v = (u, v)$  be the x and y coordinate differences between the previous and current positions of the optical flow vector.

$$TTC = \sqrt{\frac{(p_x - foex)^2 + (p_y - foey)^2}{u^2 + v^2}} \quad (5)$$

The locations of the detected points on the image vary according to the position and movement of the objects captured by the robot's camera. There are situations where the detected points on the image are not equably positioned in the whole image plane. This means there are more corners in some parts of the image and fewer corners in others. This imbalance caused by the objects and movements in the frame can be avoided by using 16 x 16 dimensional matrices.



Figure 17. Drawing the matrix on the image.

After calculating the TTC of each flow vector, it is collected at one of the closest A matrix points in the image. In matrix B, the number of TTCs collected at each matrix point is stored.

$$A = \begin{bmatrix} \sum ttc_i & \dots & \sum ttc_i \\ \vdots & \ddots & \vdots \\ \sum ttc_i & \dots & \sum ttc_i \end{bmatrix}_{16 \times 16} \quad b = \begin{bmatrix} a_{00} & \dots & a_{0n} \\ \vdots & \ddots & \vdots \\ a_{n0} & \dots & a_{nn} \end{bmatrix}_{16 \times 16} \quad (6)$$

$ttc_i$  is the time until the collision of a flow vector, and  $a$  in matrix B is the number of flow vectors. Using matrices A and b, the average TTC for each matrix point can be calculated.

$$TTC[i] = \frac{A[i]}{b[i]} \quad (7)$$

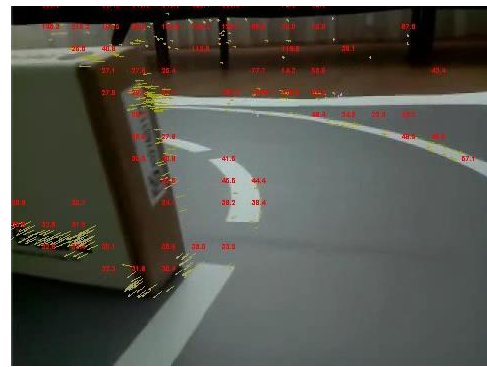


Figure 18. Shows the average collision times. It seen that vectors with the large optical flow on the image have low TTC values.

### 2.2.5. Balance Calculation and Decision Mechanism

The basic idea is that when the robot is in motion, close objects move faster than farther objects on the retina. Also, closer objects cover the field of view more, causing greater optical flows. In the region where the optical flow is greater, the collision time is low and the robot vehicle must go to the other side and avoid from the obstacle. Kachluche Souhila and Achour Karim in their article [13], they present a different perspective in which the robot car moves away from the side where there is greater optical flow.

The image taken from the camera is divided into two parts to give the vehicle balance. Balance can be achieved by minimizing the difference between collision times on the left and right side of the vehicle.

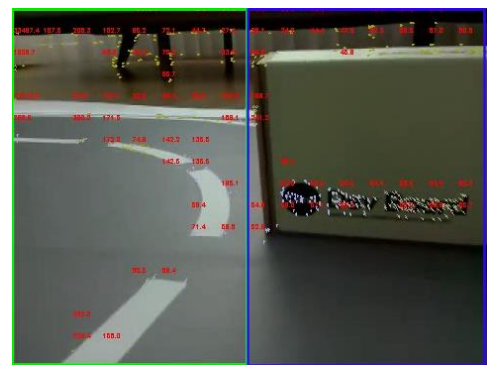


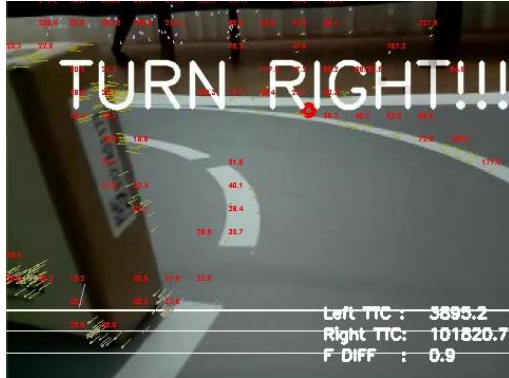
Figure 19. Dividing the image into two parts.

The following control formula is used to calculate the difference between collision times on the left and right side of the vehicle.

$$\Delta(F_L - F_R) = \frac{\sum |TTC_L| - \sum |TTC_R|}{\sum |TTC_L| + \sum |TTC_R|} \quad (8)$$

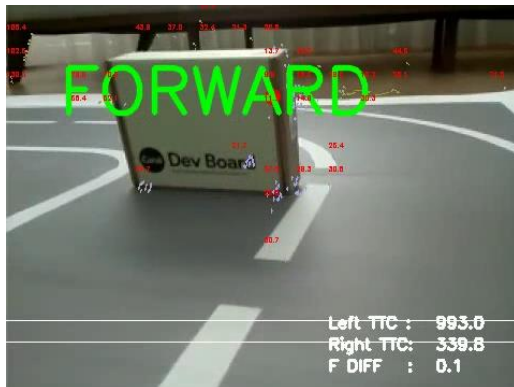
Here  $(F_L - F_R)$  is the difference between the forces on both sides of the robot body and TTC is the average of the collision time in the visual half-field on one side. The difference between the forces calculated in equation 8 is a linear number and varies between 0 and 1. The balance mechanism was applied to the robot vehicle. Due to the environmental conditions of the experimental environment, the threshold value of the difference between the left and right TTC was determined as 0.5. In cases where the

threshold value is exceeded, the robot vehicle will be given the necessary rotation order. As shown in Figure 21, the left collision time is calculated as (3895.2), the right collision time (101820.7) and the difference in forces (0.9) were found. It is decided to turn right, because the left collision time is less than the right, and the difference in forces is greater than 0.5.



**Figure 20.** Avoiding the box to the left of the robot vehicle.

In Figure 21, it can be seen that the robot vehicle is given a forward motion command. Left collision time (993.2), right collision time (339.8) and difference in forces (0.1) were found.



**Figure 21.** Robot vehicle is commanded to go forward.

### 3. CONCLUSION

This article presented develop lane tracking and obstacle avoidance capabilities with different methods and solutions for an autonomous driving system on an experimental vehicle.

Considering the Raspberry PI module performance, the first model was kept simple and fast overfitting was observed due to the small dataset. In order to develop prediction of the model and prevent overfitting, the Dropout layer was added, the dataset was enlarged by generating synthetic data, and the number of computable parameters was increased. The final model has been trained three times over 100 epochs with synthetic data.

Training of the neural network model was done with simplified images. During the test stage, the images taken from the camera should be similar to the images used in the training stage of the neural network. The similarity emphasized here is that the images are in the same colour

space and simplified. For this reason, the images taken from the camera during the test stage are instantly simplified and then transmitted to the neural network. The prepared neural network model produces 0, 1, and 2 as output. These correspond to the values for FLAT, MEDIUM, and SHARP, respectively. Since these labels are obtained by simplifying the rotation angle of the servo, the rotational motion is provided by converting the predictions back to the servo angle with the help of an algorithm.

In order to avoid obstacles, optical flow was calculated by detecting corners every 4 seconds by FAST algorithm and was used to find the focus of expansion of the vehicle. Time-to-collision was found with the FOE and the distance between the previous position and the current position of the detected point. There are situations where the detected points on the image are not equally positioned in the whole image plane. For this reason, the values of the close points are averaged and placed in the 16x16 matrix. The balance mechanism was created according to the TTC difference calculated on the right and left parts of the vehicle.

Frames per second, representing the vehicle's response speed, reached 14 frames per second when following the lane, 20 frames when avoiding obstacles, and 12 frames when both modules were working together. And the vehicle completed the map safely without hitting any obstacle.

It is planned to increase frames per second in the future and add new capabilities using 2D Lidar and Google Coral Dev Board.

**Acknowledgement:** This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK), Grant No: 1919B011903963

### REFERENCES

- [1] Casser, Vincent, et al. "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. No. 01. 2019.
- [2] Google LLC, "Get started with the Dev Board." 2020.
- [3] Caruana, Rich, and Alexandru Niculescu-Mizil. "An empirical comparison of supervised learning algorithms." Proceedings of the 23rd international conference on Machine learning. 2006.
- [4] Zhu, Xiaojin, and Andrew B. Goldberg. "Introduction to semi-supervised learning." Synthesis lectures on artificial intelligence and machine learning 3.1 (2009): 1-130.
- [5] Alpaydin, Ethem. "Introduction to machine learning." MIT press, 2020.
- [6] Schalkoff, Robert J. "Pattern recognition." Wiley Encyclopedia of Computer Science and Engineering (2007).
- [7] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." 2017 International Conference on Engineering and Technology (ICET). Ieee, 2017.
- [8] O'Donovan, Peter. "Optical flow: Techniques and applications." International Journal of Computer Vision (2005): 1-26.




- [9] Beauchemin, Steven S., and John L. Barron. "The computation of optical flow." *ACM computing surveys (CSUR)* 27.3 (1995): 433-466.
- [10] Barron, John L., and Neil A. Thacker. "Tutorial: Computing 2D and 3D optical flow." *Imaging science and biomedical engineering division, medical school, university of manchester 1* (2005).
- [11] Bounini, Farid, et al. "Autonomous vehicle and real time road lanes detection and tracking." *2015 IEE Vehicle Power and Propulsion Conference (VPPC)*. IEEE, 2015
- [12] Han, J., et al. "Road boundary detection and tracking for structured and unstructured roads using a 2D lidar sensor." *International Journal of Automotive Technology* 15.4 (2014): 611-623
- [13] Souhila, Kahlouche, and Achour Karim. "Optical flow based robot obstacle avoidance." *International Journal of Advanced Robotic Systems* 4.1 (2007): 2.
- [14] Saravanan, C. "Color image to grayscale image conversion." *2010 Second International Conference on Computer Engineering and Applications*. Vol. 2. IEEE, 2010.
- [15] Xu, Zhao, Xu Baojie, and Wu Guoxin. "Canny edge detection based on Open CV." *2017 13th IEEE international conference on electronic measurement & instruments (ICEMI)*. IEEE, 2017.
- [16] Edwards, Allen L. *An introduction to linear regression and correlation*. No. 04; QA278. 2, E3 1984.. 1984
- [17] Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.
- [18] Bergstra, James, et al. "Algorithms for hyper-parameter optimization." *25th annual conference on neural information processing systems (NIPS 2011)*. Vol. 24. Neural Information Processing Systems Foundation, 2011.
- [19] Feurer, Matthias, and Frank Hutter. "Hyperparameter optimization." *Automated Machine Learning*. Springer, Cham, 2019. 3-33.
- [20] Mikołajczyk, Agnieszka, and Michał Grochowski. "Data augmentation for improving deep learning in image classification problem." *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE, 2018.
- [21] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.
- [22] Ciliberto, Carlo, Lorenzo Rosasco, and Alessandro Rudi. "A consistent regularization approach for structured prediction." *Advances in neural information processing systems* 29 (2016): 4412-4420.
- [23] Hawkins, Douglas M. "The problem of overfitting." *Journal of chemical information and computer sciences* 44.1 (2004): 1-12.
- [24] Takahashi, Ryo, Takashi Matsubara, and Kuniaki Uehara. "Data augmentation using random image cropping and patching for deep cnns." *IEEE Transactions on Circuits and Systems for Video Technology* 30.9 (2019): 2917-2931.
- [25] Fleet, David, and Yair Weiss. "Optical flow estimation." *Handbook of mathematical models in computer vision*. Springer, Boston, MA, 2006. 237-257.
- [26] Viswanathan, Deepak Geetha. "Features from accelerated segment test (fast)." *Proceedings of the 10th workshop on Image Analysis for Multimedia Interactive Services*, London, UK. 2009.
- [27] Lucas, B. and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, Vancouver, Canada, pp. 674-679.
- [28] Levenberg, Kenneth. "A method for the solution of certain non-linear problems in least squares." *Quarterly of applied mathematics* 2.2 (1944): 164-168.
- [29] Gander, Walter. "Algorithms for the QR decomposition." *Res. Rep 80.02* (1980): 1251-1268.
- [30] Satti, Satish Kumar, et al. "A machine learning approach for detecting and tracking road boundary lanes." *ICT Express* 7.1 (2021): 99-103


# A Novel Algorithmic Similarity Measure for Collaborative Filtering: A Recommendation System Based on Rating Distances

\*<sup>1</sup>Şule Öztürk Birim, <sup>2</sup>Ayça Tümtürk

<sup>1</sup>Manisa Celal Bayar Üniversitesi, Salihli İktisadi ve İdari Bilimler Fakültesi, İşletme, Sayısal Yöntemler ABD, Manisa,

[sule.ozturk@cbu.edu.tr](mailto:sule.ozturk@cbu.edu.tr) 

<sup>2</sup>Manisa Celal Bayar Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, İşletme, Üretim Yönetimi ve Pazarlama ABD, Manisa

[ayca.tumturk@cbu.edu.tr](mailto:ayca.tumturk@cbu.edu.tr) 

## Abstract

Internet sources contain a vast amount of information about items that people desire to purchase. It is impossible to evaluate these resources and come to an informed decision. People need automated systems that evaluate previous information and propose item alternatives. Recommending items using a smart system, which is based on the previous user preferences, has growing importance since the available product data is exponentially growing. Additionally, it is difficult to find new and correct things that a user would like among this massive amount of data. To make accurate recommendations with a smart system, researchers and practitioners use collaborative filtering methods with similarity calculation based on user preferences. The crucial point in collaborative filtering is to find a valuable measure that resembles correct similarity between users. The current similarity metrics in the literature have some disadvantages in conducting accurate recommendations. To improve the recommendation performance, this study proposes a novel similarity measure that assesses the distance between the user's ratings and the median score. Considering distance from the median score is essential since some users may prefer to rate close to the median rather than the extremes. Experiments were conducted with a famous collaborative filtering dataset. Results showed that proposed similarity measure demonstrated superior performance regarding the recommendation accuracy. Implications of our results for XYZ are discussed.

**Keywords:** Collaborative filtering, recommender systems, similarity measure, prediction accuracy, classification accuracy

## 1. INTRODUCTION

In recent times, people have been exposed to a growing amount of data one-commerce as well as social media websites on several products. When dealing with such data, it is often difficult for individuals to evaluate and compare all of their product options to make an informed purchase decision. During the pandemic period, online shopping remarkably increased. Results of a recent study showed that online shopping has enlarged by almost 20% in 2020 and near 9% in 2021 [1]. Around 70% of customers who shop online do not come to a final decision before looking at the product ratings or reading the reviews [2]. These numbers indicate individuals consider others' opinions when deciding to purchase a product online. However, it is hard to examine all of the views about a product, since there may be too many. To alleviate this difficulty, automated smart systems, which make recommendations based on previous preferences, have been developed. These smart systems are called recommender systems and are widely used by commercial sites like Amazon [3] as well as non-commercial websites that possess the purpose of research such as GroupLens [4]. Recommender systems aim to make the lives of the people more comfortable by recommending the products or services that are similar to their previous preferences.

Rich (1979) provided one of the first references of recommender system modeling. In this research, a librarian known as Grundy grouped users into clusters based on their book type preference, including education, sports and romance. Then using these clusters, Grundy would recommend novels to people [5].

The methods that the recommender systems use to make suggestions can be classified as content-based methods, collaborative filtering and hybrid methods. The content-based recommender systems use content information related to the item and makes recommendations based on the user's previous item preferences [6]. The commonly used collaborative filtering methods suggest items based on the similarities between the items or the different users [7]. The collaborative filtering methods are widely used in practice as well as in academia. Lastly, the hybrid systems combine the content based and collaborative filtering methods of recommender systems [8].

Collaborative filtering systems can be classified as model-based and memory-based methods [9]. A model-based system uses an offline user database to predict a model, which in turn is utilized to make predictions. Unlike the model-based systems, a memory-based system requires the entire user database to calculate the similarities between the users. In memory-based approaches, different metrics may

be used to estimate user or item similarity. Widely used similarity measures in memory-based collaborative filtering are Pearson correlation coefficient and Vector cosine similarity metrics.

This study aims to increase the prediction performance of user-based collaborative filtering in a recommendation system. To reach this goal, we offer a novel algorithmic similarity measure that is to be used in a collaborative filtering system. Traditional similarity methods (i.e., Pearson correlation, Cosine) as well as some recently proposed similarity measures (i.e., in the literature can lead to misleading similarities, such that while ratings show that two users are similar, similarity measure demonstrate the opposite. The traditional methods also have some drawbacks in providing accurate recommendations when the data set is too sparse [10]. Additionally, measures in the literature do not consider the degree of the user tendency of rating close to the median. Some users may avoid giving ratings which are on the extremes. For example, they may prefer to rate the second highest score to an item which they liked a lot. The similarity measure proposed in this study addresses this drawback in the literature by avoiding extreme ratings. Our similarity measure was experimented on one of the most popular datasets used in the recommender systems. We compare our measure with the traditional similarity measures as well as two other similarity measures, which were recently introduced in the literature.

In the next part of this paper, we give a brief review of the literature concerning prominent collaborative filtering studies. Then, the proposed similarity method is explained by comparing it to the existing similarity methods. We then discuss the experiments that were conducted to test the proposed similarity measure. Then the results of the experiments are then discussed to highlight implications for the literatures on recommendation systems.

## 2. LITERATURE REVIEW

Several studies have used collaborative filtering methods to recommend items to their users. For example, the collaborative filtering term was first introduced in the literature by ... [11]. In their study, an experimental mail system known as Tapestry was introduced as a hybrid approach, such that it supported both the content-based and collaborative filtering methods. In similar vein, Goldberg and Roeder's [12] proposed a collaborative filtering algorithm that was named as Eigentaste. The Eigentaste algorithm employed principal component analysis to solve eigenvalues and eigenvectors matrices with 2500000 ratings, which belong to 57000 users on Jester website, an online joke recommender system. Researchers used normalized mean absolute error (NMAE) [13], [14] to compare the Eigentaste with other selected algorithms, including the algorithm proposed in [9]. Researchers concluded that the algorithm proposed in [9] provides good results by NMAE, but it ignored user differences entirely [12].

In the collaborative filtering methods, numerous similarity metrics were proposed to measure the similarity between the users or the items. When computing the similarity, the most popular traditional approach is the Pearson correlation

coefficient (PCC) [15]. PCC considers all the items or the users in the user-item matrix as equal subjects. It does not consider the commonly rated items or the common users of the identified items. In [16], the authors suggested an algorithm to overcome the disadvantages of PCC and computed the weights for distinct items depending on the scores obtained from training users. They showed that the weighted PCC system gives better performance than the traditional PCC method on two different datasets.

An essential issue in recommender systems is to calculate similarities and perform predictions when the data in the user-item matrix is very sparse. Data sparsity means users rate only a few items at a time?. This problem is known as a *cold start problem*. A cold start problem is a severe issue in collaborative filtering, and many researchers are trying to find solutions to it by offering new approaches. For example, Ahn [17] argued that PCC and Cosine (COS) [18], which are the commonly used similarity measures in collaborative filtering, are not enough when a cold-start problem occurs. To solve this problem, Ahn [17] introduced a similarity measure named PIP (proximity-impact-popularity), which considers the facets of proximity, impact, and popularity of the user ratings. The author demonstrated that in an artificial cold start problem, PIP gives better results than other measures, whereas PCC produced the best results when considering the real data set.

The study of Luo and colleagues [19] proposed a global similarity concept to calculate the similarities between users who have no commonly rated items. The authors used both local and global user similarities when calculating the overall similarity between users. Furthermore, they used the surprisal vector instead of the users' rating vector when computing local similarity. Their measure outperformed PCC based on MAE [20]. Al-Shamri and Bharadwaj [21] integrated demographic or genre information of users into the collaborative filtering. They suggested a hybrid fuzzy-genetic recommender system and their model derived better MAE values when compared to PCC. Jamali and Ester [22] proposed a model that was a mix of item-based and trust-based collaborative filtering methods. The trust-based approach uses Epinions website, which includes trust network and user ratings together. For both the users as well as the cold start users, the suggested model outperformed the traditional similarity methods.

Bobadilla et al. [23]–[28] conducted several studies about collaborative filtering between 2010–2012. In one study [24] the authors used both the MSD (mean squared difference) [29] as well as the Jaccard [30] metric to overcome the weaknesses of PCC. In Netflix and Movielens datasets, new metric provided acceptably good results, while the FilmAffinity data set did not demonstrate appropriate effects [24]. In another study, [27] the authors proposed a similarity metric considering the significance of items by weighting them while calculating PCC and COS, instead of traditional PCC and COS. They measured prediction performance with MAE, precision [31], and recall [32]. They obtained acceptable results [27]. Bobadilla et al [23] proposed a neural network learning based on the similarity metric, which was found to be faster than other measures when it came to MAE, recall, and precision. In yet another study, a framework for collaborative filtering was proposed

to assess the performance of the recommendations based on the trust on users' neighbors and the novelty of the conducted recommendations [26].

Anand and Bharadwaj [33] focused on the data sparsity problem in collaborative filtering with computing local and global similarities like [19] and used the prediction formula in [34] to conduct recommendations. Cai et al [18] derived the typicality from cognitive psychology to be used in collaborative filtering and observed improvements on MAE. Baltrunas and Ricci [31] used item splitting with matrix factorization and nearest neighbor collaborative filtering algorithms. In another study, Luo et al. [13] introduced an improving non-negative matrix factorization-based collaborative filtering method to recommend items.

In a recent paper, Chen et al. [35] used a special type of k-means clustering to focus on preventing user privacy. Recommendations were formed from the set of neighbors, which belonged to the selected cluster aiming to achieve privacy. The results indicated that the proposed system improved performance compared to the previous ones. Another study [36], aimed to address the sparsity problem in movie recommendation by forming a collaborative system using a singular value decomposition. In this paper, similarities were calculated by using the movie's content information in cosine similarity calculations.

A recent study of Afoudi et al [37] created a hybrid recommender system that combines collaborative filtering, content based similarity and a special type of neural network. The authors used the hybrid system in an unsupervised data. Their results showed that the proposed methodology outperformed the traditional methods, such as...

The study of Liu et al. [38] improved PIP similarity measure, which belongs to Ahn [17]. First, the authors showed existing measures (Pearson, cosine, etc.) and PIP's disadvantages. Then, they upgraded PIP with a more straightforward measure known as the new heuristic similarity measure (NHSM). Authors noted that the PIP measure only considers the absolute value of the rating, repeatedly penalizes on the factors, and sometimes calculates inaccurate results. To increase accuracy, they considered the proportion of the common ratings in NHSM. NHSM showed improved performance than most of the compared methods regarding precision and recall. In a recent study of [39], the authors proposed a new similarity measure based on an adaptive neighbor selection mechanism and outperformed several widely-used methods.

Wang et al. [38, 39] conducted two studies that focused on new approaches to eliminate poor prediction accuracy with collaborative filtering. In the study of [40], the authors used MAE to compare their new approach named -Fuzzy Similarity Measure-User Relevant Aggregation (FSM-URA) with other approaches. They used MovieLens 100k dataset to test the proposed approach and found that the new method performed better than others did. In the research of [41], the same authors suggested a hybrid approach named novel entropy-based similarity measure. This new approach was compared with five different approaches, including item-based Pearson correlation coefficient and user-based Pearson correlation coefficient. Their hybrid approach produced better MAE results than others did in both the

MovieLens 100K dataset as well as the SmartBizSeeker dataset. They also took advantage of the Manhattan distance model to overcome the fat-tail problem that occurs due to ratings that are far from average. We recommend that future researchers examine [41] for further inquiry about the details of the hybrid model.

Thus far, the studies in the literature have proposed improved similarity measures to heighten the recommendation performance of collaborative filtering methods. The current study also suggests a new similarity measure, which can be an alternative to the existing ones. In contrast to those measures already existing in the literature, the proposed new similarity measure is based on how the users tend to rate close or far from the median. This idea originated from the issue that people may avoid giving extreme ratings [42]. In this study, avoiding extreme ratings was called the tendency towards rating close to the median. We chose this definition because when a person avoids giving the highest rating to an item that he or she likes a lot, this also represents that person's tendency towards giving a score close to the median. It is often proposed in the literature that extreme scores would have higher impact in the similarity, such that if two users rate five (the highest score in a 5-point Likert scale) this represents a more similar behavior [17], [38], [43]. In addition to that, the current study suggests that the strength of the rating should be determined when considering the rating behavior by observing all the ratings of that user. If a user tends to give a rating close to the median, and thereby avoids extreme rating, then a four (second highest score in a 5-point Likert scale) can also represent a strong preference. Considering this, in the current study, a novel algorithmic similarity measure that considers people's tendency towards rating close to the median is proposed and the advantages of the proposed similarity measures over the existing ones were analyzed.

### 3. PROPOSED SIMILARITY METHOD

The similarity method proposed in this study aims to minimize the disadvantages of the existing similarity measures. Several similarity methods that are prominent in the literature are chosen for benchmark. The results of the chosen similarity measures were calculated on a designated user-item matrix. The sample matrix can be seen in Table 1. In this section, firstly, explanation and formalization of the existing similarity measures were demonstrated. Then the existing similarity measures were calculated using the sample matrix. Based on the calculated similarity results, drawbacks of each existing similarity method were discussed. Then, the proposed similarity method calculation steps were explained. For each user combination in the sample matrix, the new similarity measure results were calculated. Based on the obtained results, advantages of the new similarity measure over the existing ones were demonstrated.

User set was described as  $U = \{u_1, u_2, u_3, \dots, u_n\}$  and the item set was described as  $P = \{p_1, p_2, p_3, \dots, p_m\}$ . The user item matrix was represented as  $R = \{r_{ij}\}$  where  $i = 1, 2, 3, \dots, n$ ,  $j = 1, 2, 3, \dots, m$  and  $r_{ij}$  is the rating of user  $i$  to the item  $j$ .

**Table 1:** A sample user-item matrix

	Item-1	Item-2	Item-3	Item-4
User-1	2	1	2	2
User-2	5	4	5	5
User-3	4	5	4	4
User-4	1	2	1	1

### 3.1. Traditional Similarity Measures and Their Drawbacks

To observe the disadvantages of the existing similarity measures, similarities were calculated based on the sample matrix. As seen in Table 1, the sample matrix consists of four users who gave ratings to four different items. Users were resembled in rows, while items were represented in columns. The intersection of the user row and item column resembles the rating that the selected user gave for the selected item.

In collaborative filtering, widely used similarity measures are Pearson Correlation Coefficient (PCC) and Cosine similarity (COS). In addition to these traditional methods, the constrained Pearson correlation [44] [45], was used to eliminate the disadvantages of PCC. In this section, these three similarity measures and their drawbacks are discussed. The notation used in the formalization of the similarity measures are shown below:

$C$	Commonly rated items by the selected users
$u_a, u_b$	Users of a and b
$r_{u,j}$	The rating of the user $u$ to the item $j$
$\bar{r}_u$	The average rating value of the user $u$
$r_{med}$	Median score in the rating scale

#### 3.1.1. Pearson Correlation Coefficient

The Pearson correlation coefficient shows how the two users use similar patterns while giving ratings to the items. The formalization of the Pearson correlation is as follows:

$$PCC(u_a, u_b) = \frac{\sum_{j \in C} (r_{u_a, j} - \bar{r}_{u_a})(r_{u_b, j} - \bar{r}_{u_b})}{\sqrt{\sum_{j \in C} (r_{u_a, j} - \bar{r}_{u_a})^2} \sqrt{\sum_{j \in C} (r_{u_b, j} - \bar{r}_{u_b})^2}} \quad (1)$$

PCC similarity scores of all the pairs in the sample matrix are shown in Table 2. While identifying how the users have similar patterns in ratings is important, this pattern may sometimes be misleading and direct to wrong similarities. When the ratings of User1 and User2 were observed in Table 1, it is seen that User1 and User2 rated the items as (2,1,2,2) and (5,4,5,5) respectively. Based on the scores of the User1 and User2, a low similarity was expected between them because while User1 dislikes all the items, User2 likes all of them. However, Table 2 shows that PCC score of User1 and User2 is +1, representing a perfect correlation. PCC only considers the scoring behavior of the two users and does not consider the meaning of the score. A similar situation was observed for User2 and User3. User2 rated the items as (5,4,5,5), while User3 rated the items as (4,5,4,4). While high similarity is expected between User2 and User3, PCC scores for the two users were calculated as -1. This misleading PCC similarity scores were also observed for other pairs, as seen

in Table 2. Such misleading similarity scores may lead to deficiencies in collaborative filtering.

**Table 2.** Similarity scores based on sample user-item matrix

Pair	Similarity Scores				
	PCC	COS	CPC	PIP	NHSM
User1 User2	1	0.9885	-3.6689	6	0.0174
User1 User3	-1	0.9414	-3.5714	18.861	0.0078
User1 User4	-1	0.8386	3.6689	3822	0.0216
User2 User3	-1	0.9815	3.6689	3822	0.0216
User2 User4	-1	0.9112	-3.7692	6.5830	0.0087
User3 User4	1	0.9732	-3.6689	6	0.0078

#### 3.1.2. Cosine Similarity

Cosine calculates similarity by observing how magnitudes of the rating vectors differ from each other [46]. Cosine similarity (COS) is calculated as follows:

$$COS(u_a, u_b) = \frac{\sum_{j \in C} r_{u_a, j} * r_{u_b, j}}{\sqrt{\sum_{j \in C} r_{u_a, j}^2} \sqrt{\sum_{j \in C} r_{u_b, j}^2}} \quad (2)$$

COS results for the user pairs in the sample matrix is given in Table 2. COS found the most similar pairs as User1 and User2. Scores of User1 and User2 were (2,1,2,2) and (5,4,5,5) respectively, indicating different preferences. While expecting a low similarity, Cosine found the highest similarity with the score of 0.988 between User1 and User2. Similarly, a high similarity was expected between User1 and User4 with the ratings of (2,1,2,2) and (1,2,1,1) respectively. However, COS found 0.838, which is the lowest score for these two users. These results indicate that COS can lead to misleading similarities in certain conditions. Another drawback of COS is that it proposes a narrow range between the lowest and the highest score, which is 0.8386 – 0.9885. This may lead to problems in differentiating similar and dissimilar users.

#### 3.1.3. Constrained Pearson Correlation

The Constrained Pearson Correlation (CPC) was proposed as an alternative to PCC. It takes into account positive and negative attitudes of the user towards the item by choosing a threshold to decide the direction of the attitude [47]. Therefore, CPC considers the absolute value of the rating. In other words, it considers whether the user liked or disliked the item [47], [48]. In our study, we choose the median as the threshold to decide the direction of the attitudes of the users. Expression of CPC is as follows:

$$CPC(u_a, u_b) = \frac{\sum_{j \in C}(r_{u_a, j} - r_{med})(r_{u_b, j} - r_{med})}{\sqrt{\sum_{j \in C}(r_{u_a, j} - r_{med})^2} \sqrt{\sum_{j \in C}(r_{u_b, j} - r_{med})^2}} \quad (3)$$

As can be noted from Table 2, CPC seems to overcome the drawbacks of PCC. CPC found the lowest similarity score for the pair of User2 and User4. This result is acceptable because User2 and User4 show different preferences. CPC found the highest similarity score for the pairs of (User1, User4) and (User2, User3). These results are consistent with the preferences of these pairs. Prediction performance of CPC should be observed with a broader data to decide how efficient CPC is in collaborative filtering.

### 3.2. Recently proposed similarity methods

In the real-life user-item preference matrices, users give ratings to only a few items among all the items. This problem is referred to as cold start problem in collaborative filtering literature. The problems with the traditional similarity methods, as stated in Section 3, become even more severe when the data sparsity levels are high [17]. To overcome this shortcoming, the researchers proposed different similarity metrics. Two of the most prominent similarity measures proposed recently are Proximity-Impact-Popularity (PIP) [17] and New Heuristic Similarity Method (NHSM) [38]. The details of the two popular similarity methods are explained below.

#### 3.2.1. Proximity – Impact – Popularity (PIP) Similarity

The PIP similarity measure proposed by Ahn [17] embraces three factors: proximity, impact, and popularity of user ratings. PIP is computed as follows:

$$SIM_{PIP}(u_a, u_b) = \sum_{j \in C} PIP(r_{u_a, j}, r_{u_b, j}) \quad (4)$$

Where for the two ratings  $PIP(r_1, r_2)$  is calculated as:

$$PIP(r_1, r_2) = Proximity(r_1, r_2) * Impact(r_1, r_2) * Popularity(r_1 * r_2) \quad (5)$$

The *proximity* of the two ratings demonstrates the mathematical difference between two ratings. It also gives a penalty to ratings in disagreement by increasing the distance. The *impact* factor represents how strong the two users' preferences are about the item. When both users give the highest score or the lowest score to the item, this shows a stronger common preference, explaining a high impact score. Further lowering the impact score when the two ratings disagree also penalizes impact score. *Popularity* measures how the two ratings are far from the mean of all the ratings given to that item. Popularity computes a higher score when the two ratings are both further from the average in the same direction. Further information about three factors and how they are calculated can be found in [17].

PIP measure has several drawbacks. Firstly it repeatedly penalizes ratings on disagreement [38]. This extra penalization can lead to increased deviation in similarity results. PIP similarity scores demonstrate a vast range between the lowest as well as the highest score. When the results of the PIP measure on the sample matrix is observed in Table 2, lowest score is 6 while the highest score is 3822. High deviation in similarity results may lead misleading

predictions. Secondly, the PIP measure does not consider the tendency to rate close to the median. The impact factor of PIP considers how strong the user's preferences is by finding agreement on the extreme values on the rating scale. However, impact does not consider overall rating behavior of the users about how they tend to rate close to the median while deciding stronger preference level.

#### 3.2.2. The New Heuristic Similarity Measure (NHSM)

The new heuristic Similarity Method (NHSM) was proposed by Liu et al. [38] as an improvement to the PIP measure. Proximity-Significance-Singularity (PSS), a revised version of PIP, is used as a factor in NHSM. Proximity only calculates the absolute distance and does not change when there is a disagreement on the item between the two users. Significance measures how the two ratings are different from the median. Singularity measures how the two ratings deviate from the average score of the item. PSS is calculated as follows:

$$PSS(r_1, r_2) = Proximity(r_1, r_2) * Significance(r_1, r_2) * Singularity(r_1 * r_2) \quad (6)$$

After calculating the PSS measure, the proportion of common ratings is added as a factor in the NHSM similarity. In this factor, to penalize the low number of common ratings, the combination of user ratings is calculated by multiplying the number of ratings of the users. It is necessary to indicate that real value of common ratings as proposed in Jaccard measure [30] is modified in NHSM measure.

While significance considers deviation from the median, it does not reflect the effect of the user's tendency towards scoring around the median. NHSM uses another factor named URP to consider rating behavior of the users. However, when the formalization of URP is observed in [38], URP only resembles how the ratings of users differentiate from the mean value. This formalization does not include the effect of the user's tendency about scoring close to the median. For a detailed description of how NHSM is calculated, Liu et al. [38] should be observed.

#### 3.2.3. The reasons for proposing a new similarity measure

As described in the preceding section, traditional similarity measures have some drawbacks. While recently proposed measures of PIP and NHSM have been proven to be effective, alternatives to the traditional methods such as PIP and NHSM have some deficiencies. The deficiencies in these similarity measures, and how the proposed novel algorithmic similarity measure will handle the stated deficiencies as stated in the following part.

PIP measures uses common items while calculating similarity, but it does not consider the effect of the proportion of commonly rated items. This may lead to misleading similarity results such that a pair who have one common item may be treated on par with a pair who has twenty items in common.

While NHSM considers the proportion of common items, the way it calculates the proportion of these items has some drawbacks. Let User A and User B be the pair for similarity calculation. NHSM uses a revised version of Jaccard and divides the number of common items by the product of the



number of rated items of User A and User B, as shown in this equation:

$$Jaccard_{nhsm} = \frac{|n(A \cap B)|}{n(A) * n(B)} \quad (7)$$

Where  $A \cap B$  is the set of commonly rated items of User A and User B.  $n(A)$  and  $n(B)$  represent the number of items rated by User A and User B respectively. NHSM version of Jaccard calculation not only penalizes users who have less common items, but also is disadvantageous to users who have a high amount of common items. The proportion of common items for the users who have a high number of common items will lower the value of proportion since the denominator will be high. In our study, to protect the accuracy of the proportion of common ratings original Jaccard formula were used.

Adopting the idea that some people tend to avoid scoring extreme values [42], our study argues that some people do not prefer to give high ratings, although they liked the item a lot. For example, a person who loves a movie may give four to the item. They may think that the film should have extraordinary features to get a score of five. This situation may be correct in the reverse direction. A person who extremely dislikes a movie may rate the item as two instead of one. We refer to this behavior as the *tendency to rate close to the median*. We consider this tendency of the users in the novel algorithmic similarity measure calculation.

Neither PIP nor NHSM consider the tendency to rate close to the median. In NHSM, a factor named user-rating preference (URP) is considered to reflect the preference to rate high or low. However, URP does not measure users' tendency to rate close to the median. It represents how the users' ratings are different from each other. It is a product of the mean difference as well as the standard deviation difference between users, calculated as such:

$$URP_{nhsm}(u_a, u_b) = 1 - \text{sigmoid}(|\mu_a - \mu_b| * |\sigma_a * \sigma_b|) \quad (8)$$

where the Sigmoid function is calculated as follows:

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad (9)$$

While URP calculates how the mean and the standard deviation of two users differentiate from each other, URP may not be sufficient in considering how the users avoid rating the highest/lowest score or tend to rate close to the median.

In this study, while calculating proximity between the users' ratings, the *tendency to rate close to the median* is considered to reflect proximity value more accurately. *Singularity* is calculated as how each user's rating is different from the mean ratings of that item. In addition to proximity and singularity, Vector Space Similarity (VSS) is adapted to calculate user similarity. This algorithm is chosen because, in VSS, values that represent user ratings can be calculated to reflect how each user's score is different from the median. With adopting VSS in the new similarity measure, not only do the proximity value represent the *tendency to rate close to the median*, but the overall similarity measure considers how user's ratings are different from the median. Details about calculating the novel algorithmic similarity measure are explained in the next section.

### 3.3. Expression of the Novel Algorithmic Similarity Measure

#### 3.3.1. The Notation

In this section, we give a mathematical formula to calculate the new similarity method, which we named **the Novel Algorithmic Similarity (NAS)**. We implemented the sigmoid function while calculating first part of the similarity known as the Proximity –Singularity (PS). The sigmoid function not only normalizes the results, but it also rewards good similarity and punishes the bad one [38]. Initial similarity measure PS was calculated as follows:

$$PS(r_1, r_2) = \text{Proximity}(r_1, r_2) * \text{Singularity}(r_1, r_2) \quad (10)$$

PS for each rating pair is calculated by multiplying proximity and singularity of the two ratings. Proximity in PS is different from proximity calculation in PIP and NHSM. Before we calculate proximity for PS, we calculate a user factor that resembles the tendency to rate close to the median for each user. Small average absolute deviation from the median means the user has a higher tendency to rate close to the mean. Reversely, high deviation from the median shows the user has lower tendency to rate close to the mean. Therefore, to compute user factor, average absolute deviation from the median is subtracted from 1 after it is normalized by the sigmoid function. User factor (UF) is calculated as follows:

$$UF_u = 1 - \text{sigmoid}\left(\frac{\sum_{r \in A} |r - r_{med}|}{n(A)}\right) \quad (11)$$

where A is the set of user u's ratings.

Since UF resembles tendency to rate close to the median, when the user rated above the median, UF should be added to the score to reflect possible rating preference of the user. This sum will resemble the score of the user if the user does not have any tendency of rating close to the median. Similarly, if the user rated below the median, UF should be subtracted from the user's rating to make the rating free from the tendency towards the median. Calculation of the scores with adopting UF is as follows:

$$\begin{aligned} s_{u,j} &= r_{u,j} + UF_u, & \text{if } r_{u,j} > r_{med} \\ s_{u,j} &= r_{u,j} - UF_u, & \text{if } r_{u,j} < r_{med} \\ s_{u,j} &= r_{u,j}, & \text{if } r_{u,j} = r_{med} \end{aligned} \quad (12)$$

where  $s_{u,j}$  is the updated score of the user, u for the item, and j after adopting  $UF_u$ .

The proximity is calculated as the absolute value of the distance between the updated scores as follows:

$$\text{Proximity}(s_{u_a,j}, s_{u_b,j}) = 1 - \text{sigmoid}(|s_{u_a,j} - s_{u_b,j}|) \quad (13)$$

Singularity resembles how each users' rating is different from the average rating for that item. Different from [38], we consider deviation from the mean separately for each score and then multiply them. This makes the singularity value stronger by empowering the scores, which are far from the mean rating of the item. Singularity of two ratings is calculated as follows:

$$\text{Singularity}(r_{u_a,j}, r_{u_b,j}) = \text{sigmoid}(|r_{u_a,j} - \bar{r}_j| * |r_{u_b,j} - \bar{r}_j|) \quad (14)$$

If the two users rated an item as five, where the average rating on the Likert scale is three, the singularity value of the

two ratings will have a high impact on the similarity score. Similarly, if a user rates an item five, and another user rates it as one, the singularity of these two scores will have a high impact on the similarity in a negative direction because the VSS considers the direction of the relationship (as will be seen in the following section).

When a rating is far from the median and close to the extreme, this reflects a stronger preference. In a similarity measure, it is necessary to consider that if two scores are far from the median and close to the extremes, these scores should have a stronger impact on the similarity between users [17], [19]. This issue resembles the significance of the scores [27]. For two users, let the first pair of scores be (5,5) and the second pair of scores be (2,3). The first pair indicates a higher similarity between the users than the second pair. To reflect the significance of the scores, we can use the Surprisal-based Vector Similarity method (SVS). Vector similarity method uses a maximum likelihood estimator, which refers to the average attitude of the scores [19]. The score that is far from the maximum likelihood estimator has more significance on the similarity.

In Luo et al. [19], the mean score of the item was used as the maximum likelihood estimator. In our study, to calculate SVS, we use the median as the maximum likelihood estimator. With this estimation, users who rate on the extremes will represent a strong preference and as well as have a high similarity between them. SVS similarity is calculated using the surprisal vector of the user rather than the actual rating vector. Surprisal vector of a user is calculated as follows [19]:

$$S_a = [s_{a,1}, s_{a,2}, s_{a,3}, \dots, s_{a,z}]$$

$$s_a = [sign(r_{u_{a,1}} - r_{med}) * I(r_{u_{a,1}}), \dots, sign(r_{u_{a,z}} - r_{med}) * I(r_{u_{a,z}})] \quad (15)$$

where Z is the set of items user a has rated,  $sign(r_{a,1} - r_{med})$  means the sign of the notation,  $r_{a,1} - r_{med}$  is positive or negative, and  $I(r_{a,1})$  is defined as the quantity of information and calculated as follows:

$$I(r_{u_{a,j}}) = \ln(2\hat{b}_j) + \frac{|r_{u_{a,i}} - r_{med}|}{\hat{b}_j} \quad (16)$$

where  $\hat{b}_j$  is defined as a scale parameter and calculated as follows:

$$\hat{b}_j = \frac{1}{N} \sum_{j=1}^N |r_{u_{i,j}} - r_{med}| \quad (17)$$

where N is the set of all users.

After calculating the surprisal vector of users, Surprisal-based Vector Similarity (SVS) coefficient is computed to calculate the similarity between two users as shown in follows:

$$SVS(u_a, u_b) = \frac{\sum_{j \in C} s_{a,j} * s_{b,j}}{\sqrt{\sum_{j \in C} s_{a,j}^2} \sqrt{\sum_{j \in C} s_{b,j}^2}} \quad (18)$$

To calculate the proportion of common ratings we used the Jaccard formula. It is important to calculate the exact

proportion of common ratings to see how they affect the similarity score. Jaccard similarity is calculated as follows:

$$Jaccard(u_a, u_b) = \frac{n(A \cap B)}{n(A \cup B)} \quad (19)$$

Finally, all the factors in the similarity measure are represented in the Novel Algorithmic Similarity (NAS) measure, which is calculated by multiplying the equations, (10), (18) and (19) as follows:

$$NAS(u_a, u_b) = PS(u_a, u_b) * SVS(u_a, u_b) * Jaccard(u_a, u_b) \quad (20)$$

### 3.3.2. Discussion on the Novel Algorithmic Similarity Measure

The Novel algorithmic similarity measure was applied to the sample matrix on Table 1 to observe how it overcame the drawbacks of the existing similarity measures, as discussed in Section 3.1 and 3.2. As Table 3 shows, the highest similarity based on NAS was observed between users 1-4 and users 2-3, as expected based on the ratings. However, PCC and COS did not accurately identify the most similar users. According to PCC, similarity of users 1-4 and users 2-3 were negative. According to COS, users 1-4 were the least similar pair. These results indicate that NAS overcame the drawbacks of PCC and COS when it came to misleading similarity score regardless of the similar preferences.

As observed in Table 3, NAS correctly identified the least similar users as users 1-2, users 1-3, users 2-4 and users 3-4. NAS accurately determined the direction of the relationship between the least similar users as negative. However, in COS, PCC, PIP, and NHSM, direction identification was either wrong or not applied. PCC incorrectly identified the direction of the relationship between users 1-2 and users 3-4. The other similarity measures, with the exception of CPC, did not show the direction of the relationship. NHSM similarity between users 1-3 was found relatively high and it is difficult to say whether users 1-3 had a high or low similarity based on the overall scores of NHSM.

As seen in Table 3, NAS correctly identified the most and the least similar users based on their ratings. NAS also correctly determined the direction of the similarity between the users. These features make NAS an appropriate similarity measure that could be used as an alternative to other similarity measures in collaborative filtering.

**Table 3.** NAS results for the sample matrix

Pair	NAS Similarity
User1 – User2	-0.1126
User1 – User3	-0.1863
User1 – User4	0.9376
User2 – User3	0.9376
User2 – User4	-0.1064
User3 – User4	-0.1126

To summarize, the steps taken to calculate the algorithm for the NAS measure for a user pair is as follows:

- 1- The user factor, based on the tendency of scoring close to the mean, is calculated for each user.

- 2- The proximity between scores is calculated with adapting the user factor, based on equation (13).
- 3- The singularity between the scores is calculated based on equation (14).
- 4- The PS score for the user pair is calculated by summing the multiplication of the proximity and the singularity scores of the commonly rated items.
- 5- The SVS similarity score for the user pair is calculated based on the equation (18) where maximum likelihood estimator is the median.
- 6- The commonly rated items for the user pair is calculated based on the Jaccard formula in equation (19).
- 7- The NAS measure for the user pair is calculated by multiplying the PS, SVS and Jaccard scores.

## 4. EXPERIMENTS

### 4.1. Data Set

The MovieLens Dataset of ML-100K and FilmTrust datasets were used in our experiments. The MovieLens Dataset of ML-100K was prepared by the GroupLens Research at the University of Minnesota [4]. In ML-100K, there were 943 users and 1682 movies with a total of 100 000 ratings. Each user rated at least 20 movies out of 1682 movies. Ratings changed between 1 and 5, where 1 represented the lowest score for the preference and 5 depicted the highest score. This made the ML-100K dataset very sparse with %6.3 density. FilmTrust is a movie-rating website, wherein the users give ratings for the selected movies. This dataset was prepared by Guo et al. [49]. In the FilmTrust dataset, there were 1508 users and 2071 movies with a total of 35500 ratings. Ratings change between 0.5 and 4 and the sparsity level of FilmTrust was % 1.14.

To observe the performance of the proposed novel algorithmic similarity (NAS) measure, we compared the proposed measure with the traditional methods of COS, PCC, CPC and the recent methods of PIP and NHSM. Recommendations were conducted based on each similarity method, and the performance of each similarity method was compared based on selected evaluation metrics of MAERMSE, recall, and precision.

For each experiment, 70% of users in the dataset were selected as training users for similarity calculation. The remaining 30% of users were testing users for whom the recommendations were conducted. For each testing user, 70% of their rated items were selected as training ratings for similarity calculation, while 30% of the rated items were used for conducting the actual recommendations.

### 4.2. Performance metrics

After calculating the user based similarities, the predicted ratings of an item for a specific user was calculated using the the following formula [19]

$$\hat{r}_{u_t,p} = \frac{\sum_{i \in K} sim_{u_t,u_i} r_{u_i,p}}{\sum_{i \in K} |sim_{u_t,u_i}|} \quad (21)$$

where  $sim_{u_t,u_i}$  resembles the similarity between the test user  $u_t$  and his neighbor  $u_i$ .  $K$  denotes the set of most similar neighbors of the test user  $u_t$ .

With the above formula, predicted ratings were calculated for each similarity measure framework. To compare the performance of each similarity measure, prediction accuracy was calculated with MAE. As stated before, the MAE is a commonly used metric to calculate how the predicted values differ from the real values [14]. MAE is calculated as follows:

$$MAE = \frac{\sum_{j=1}^N |r_j - \hat{r}_j|}{N} \quad (22)$$

Another commonly used metric to measure how the predicted values are distant from the real values is RMSE. RMSE calculates the averaged squared distances between the real ratings and the predicted ratings[14]. RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^N (r_j - \hat{r}_j)^2}{N}} \quad (23)$$

Besides measuring prediction accuracy of the selected algorithms, we also wanted to measure the classification accuracy of the proposed frameworks. Classification accuracy metrics tolerate deviations from the actual ratings when they measure how the items are truly classified as recommended [50]. Recall and precision are the classification accuracy metrics, which were also used in our research. . Recall is defined and calculated as follows [51]:

$$Recall = \frac{\text{number of testing items liked by the testing user and assigned to recommended list}}{\text{number of testing items actually liked by the testing user}} \quad (24)$$

Precision is defined and calculated as follows [51], [52] :

$$Precision = \frac{\text{number of testing items liked by the testing user and assigned to recommended list}}{\text{number of testing items assigned to recommended list}} \quad (25)$$

For performance measures, smaller MAE values indicate better prediction accuracy while higher recall and precision scores indicate better classification accuracies.

### 4.3. Experimental Design

To compare the proposed NAS algorithm with the other selected similarity algorithms, we build several configurations of the chosen parameters. We altered the number of nearest neighbors in our experiments. The number of nearest neighbors ( $K$ ) is a fundamental parameter in collaborative filtering, and it affects the prediction performance [53]. We identified eight levels for the number of nearest neighbors and calculated the MAE, RMSE, recall and precision values for each level using the proposed similarity measure NAS as well as the other selected similarity measures of COS, CPC, PCC, PIP, and NHSM.

## 5. Results and Discussion

In this section to compare the performance of the proposed NAS measure, several experiments were conducted, using two datasets that were the MovieLens ML-100k and FilmTrust. Performance comparison were based on prediction accuracy and classification accuracy. Prediction accuracy was measured using the metrics of MAE and RMSE, while classification accuracy was measured using Recall and Precision. We compared the results of each of the

performance metric based on the pre-determined levels of the numbers of the nearest neighbor. First, we presented the results for prediction accuracy. Then, we presented the classification accuracy performance of predicted values.

**5.1. MAE and RMSE results for prediction accuracy**

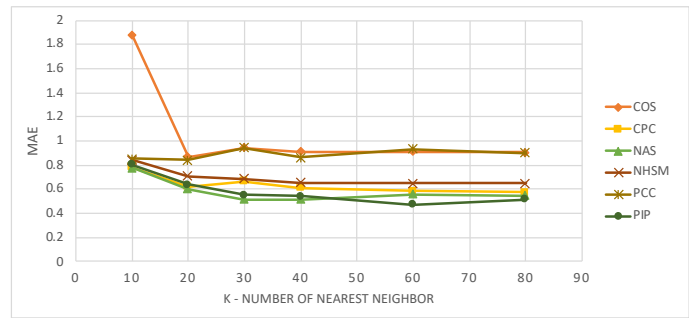
As K denotes different levels for the number of nearest neighbors, we analyzed the impacts of the different number of nearest neighbors on prediction accuracy measured with MAE and RMSE. Figure 1 shows the MAE results of the similarity measures used with six levels of the number of nearest neighbors for the MovieLens 100k (ml-100k) dataset. Figure 1a shows that except Cosine, for all similarity measures MAE decrease as K increase from 30 to 80. We observed that the novel algorithmic similarity (NAS) measure reveals lower MAE values than the most of the other similarity measures in all levels of the number of nearest neighbors. When K is 20, 30 and 40 NAS obtains the lowest MAE among all of the similarity measures. When K is 10, 60 and 80, NAS gets the second lowest MAE after NHSM measure.

Figure 1b shows the MAE results for the FilmTrust dataset. Figure 1b shows that for all similarity measures MAE decrease as K increase from 10 to 20. NAS measure produces lower MAE values than the other similarity measures except for the levels when the number of nearest neighbors are 80 and 60. When K is 10, 20, 30 and 40 NAS obtains the lowest MAE among all of the similarity measures. When K is 60 and 80, NAS gets the second lowest MAE after NHSM measure after PIP.

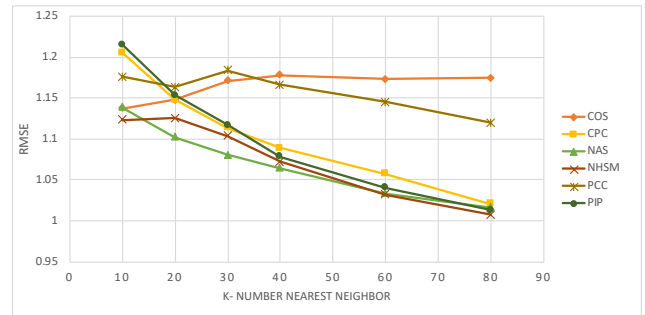
RMSE results for MovieLens 100k for all the measures with six levels of K can be observed in Figure 2a. NAS is the best similarity measure among the six similarity measures based on RMSE because NAS is the only measure that provided the best performance in the four levels out of six levels of K. When K is 20, 30, 40, and 60 NAS provides the lowest RMSE score. When K is 10, NAS shows the second lowest score with PIP after NHSM. When K is 80 NAS comes third after NHSM and PIP. RMSE results for FilmTrust can be observed in Figure 2b. NAS provided the best performance in the four levels out of six levels of K. When K is 10, 20, 30, 40 NAS produces the lowest RMSE score. When K is 60 and 80, NAS shows the second lowest score after PIP.



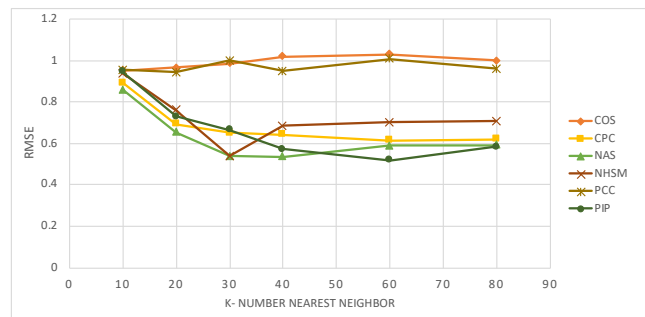
**Figure 1a.** MAE results of Similarity measures in different K nearest neighbors for MovieLens 100k



**Figure 1b.** MAE results of Similarity measures in different K nearest neighbors for FilmTrust



**Figure 2a.** RMSE results in different K nearest neighbors for MovieLens 100k



**Figure 2b.** RMSE results in different K nearest neighbors for FilmTrust

Concerning prediction accuracy, our newly proposed similarity measure NAS has outstanding performance when compared to the other similarity measures. It showed lower MAE score than all of the traditional similarity measures of COS, CPC, and PCC in all of the levels of K for both of the two datasets. For the ml-100k dataset, NAS obtained lower MAE from the recently proposed popular similarity measure PIP in five out of six levels of K. For three levels of K, NAS comes first meaning showing the best prediction accuracy while providing better results than the NHSM and PIP. In the lowest number and the two highest numbers of nearest neighbors(K), NAS comes second best after NHSM. Regarding RMSE for ml-100k, in the lowest and the highest

K, NAS comes second after NHSM. For the FilmTrust dataset, NAS brought lower MAE and RMSE levels for all number of nearest neighbors but the two highest level of number of nearest neighbors. In the two highest number of nearest neighbors PIP is the only measure that showed lower MAE and RMSE results. This means NAS showed better performance than the NHSM for all the levels.

The reason for NAS coming second in the lowest and highest K is related with the similarity calculation based on the number of nearest neighbors. When K is small like 10, some similar users may be left out to calculate accurate similarity scores between the users. When K is high like 80, some false neighbors may be identified. False neighbor is a user that is identified as a neighbor but in fact he or she is not similar to the interested user. To summarize, using low number of neighbors may result in leaving some similar users out of the group and using high number of neighbors may result in bringing false neighbors inside the group. In both situations, neighbor group may not accurately represent the similar users. This may be the reason for NAS not being the best similarity measure regarding prediction accuracy when K is the lowest and the highest. When K is between 20, 3 and 40, a more representative similar neighbor group can be established to make healthier predictions and this may lead the NAS being the best measure in the given K interval.

### 5.1. Recall and Precision results for classification accuracy

We measured classification accuracy with the metrics of recall and precision. We firstly analyzed the impacts of the different K on classification accuracy. The results of the analysis are given in Figures 3 and 4 for precision and recall respectively. As Figure 3a shows, for ml-100k regarding recall scores the newly proposed similarity measure NAS showed remarkable results among the six similarity measures. Recall of NAS was higher than the three of the metrics which are Cosine, PCC, and NHSM. The highest recall is observed when K is 20 and 30 for CPC, PIP, NAS, and NHSM. For these similarity measures recall gets lower as K increases. Only in Cosine and PCC recall increased as K increased. The two mostly used traditional similarity measures Cosine, and PCC shows the lowest recall score among all measures in all K. CPC revealed the highest recall values in all the K while PIP comes second and NAS comes third.

For the FilmTrust dataset, as Figure 3b shows, number of levels has a huge impact on the recall results. As number of nearest neighbor increases Recall decreases for all the measures. While K is 10 NAS comes third after PIP and CPC. When K is 20, NAS is the second after CPC. When K is 30, NAS comes second after NHSM. When K is 40 NAS is has the highest recall results. When K is 60 NAS is the third after NAS and NHSM. Finally when K is 80 NAS is the second after COS.

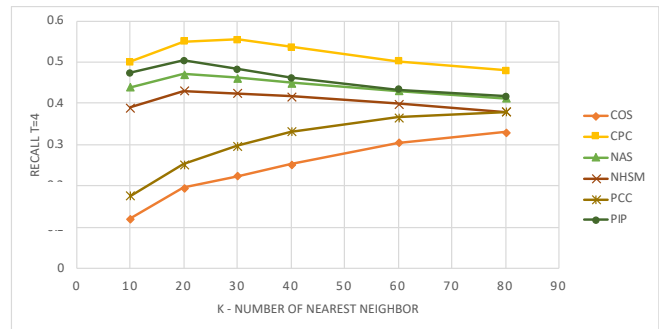


Figure 3a. Recall results for ml-100k

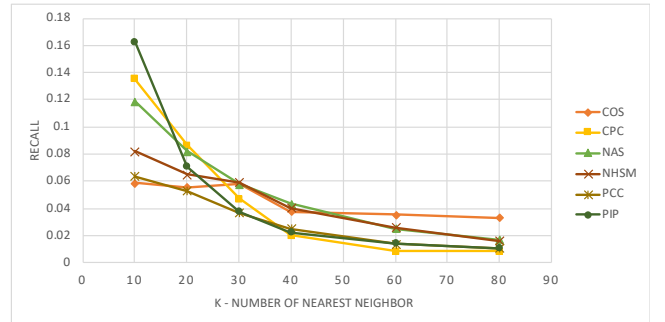
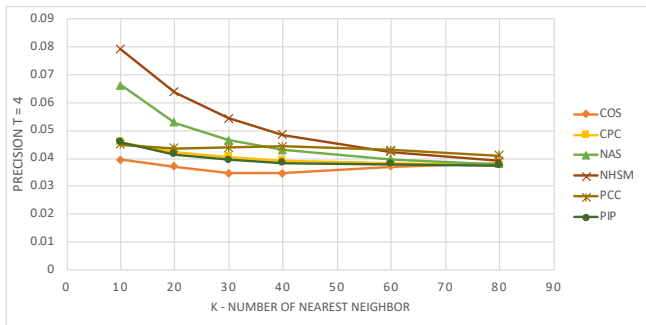


Figure 3b. Recall results for FilmTrust

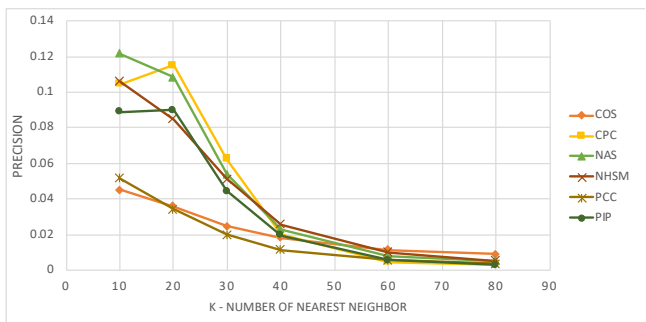
Precision values for ml-100k for all K values when the score threshold is 4 can be observed in Figure 4a and 4b. Precision of NAS shows remarkable results. As shown in Figure 4a, for ml-100k, in the three lowest K which are 10, 20 and 30 NAS showed the second-best precision score. In all levels of K, NHSM revealed the best precision score. Except for Cosine and PCC, precision values are decreased as K decreased. Cosine and PCC showed stable values in all K, while Cosine revealed the lowest precision scores in the first four levels of K and PIP obtained the lowest precision score when K is 60 and 80. Precision results also show that as K increases, precision score of all values come close to each other. Especially after 40 precision values of the similarity measures are very close to each other. This result indicates that as the number of nearest neighbors increase false neighbors increase. Increasing number of false neighbors decrease the ability of precision measure to differentiate similarity measures from each other.

For FilmTrust dataset, precision score decreases as K increases for all similarity measures. When K is 10, NAS showed highest precision results. When K is 20 and 30 NAS is the second after CPC. When K is 40 NAS is second after NHSM. When K is 60 NAS is the third after COS and NHSM. When K is 80 NAS is the second after COS. Although NAS was not the best measure for all the levels of K, it showed higher precision scores than the majority of the similarity measures for all the levels of K.





**Figure 4a.** Precision results different K nearest neighbors for ml-100k



**Figure 4b.** Precision results different K nearest neighbors for FilmTrust

To summarize the results, we can conclude that proposed Novel Algorithmic Similarity (NAS) measure can provide better prediction and classification accuracy performance than most other tested similarity measures as can be seen from Figure 1a to Figure 4b. NAS provided best results when the number of nearest neighbors is average like 20, 30 and 40. Average number of nearest neighbors result in a more accurate similar neighbor set. Regarding prediction accuracy, NAS can outdo all of the tested similarity measures in the all but the two extreme levels of the number of nearest neighbors. Concerning classification accuracy, NAS comes second or third best among all the proposed measures in the whole number of nearest neighbors. The above results demonstrate that the newly proposed NAS is an effective similarity measure in collaborative filtering and can be an excellent alternative to the similarity measures used in the literature. NAS demonstrated a remarkable performance because it distinguishes users not only how they score different from each other, it also considers how the users are different based on their tendencies towards scoring close to the median.

## 6. CONCLUSIONS

This research proposes a new similarity model that can compete with the popular similarity measures in the literature. For comparison with the proposed measure, five similarity measures were chosen from the literature. Firstly, the main drawbacks of the five similarity measures were stated using a sample matrix. The stated similarity measures had problems in genuinely identifying similar users. To overcome the drawbacks of the existing similarity measures, the novel algorithmic similarity (NAS) measure was used. NAS distinguishes similar users by considering the tendency towards scoring close to the median, which is then used as

the maximum likelihood estimator for the suppose vector similarity. To demonstrate the effectiveness of the proposed NAS measure, several configurations of the nearest neighbors and score threshold for classification accuracy were used in our experiments. Results reveal that the novel algorithmic measure yields better prediction accuracy performance than other similarity measures in almost every level of the nearest neighbors. The novel algorithmic similarity measure also showed better performance in terms of classification accuracy than most of the other similarity measures when the threshold is the score of four. These findings indicate that the proposed NAS measure can overcome the shortcomings of the existing measures and be a strong competitor to the similarity measures used for collaborative filtering.

## REFERENCES


- [1] S. Davis and L. Toney, "How Coronavirus (COVID-19) Is Impacting Ecommerce," *Roi Revolution*, 2021.
- [2] M. Z. Fisher, "Why Product Reviews are Important for Buyers and Sellers | ShipStation," *ShipStation*, 2018. [Online]. Available: <https://www.shipstation.com/blog/product-reviews-important-buyers-sellers/>. [Accessed: 12-Nov-2021].
- [3] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003, doi: 10.1109/MIC.2003.1167344.
- [4] F. M. Harper and J. A. Konstan, "The MovieLens Datasets," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 1–19, 2016, doi: 10.1145/2827872.
- [5] E. Rich, "User modeling via stereotypes," *Cognitive Science*, vol. 3, no. 4, pp. 329–354, 1979, doi: 10.1016/S0364-0213(79)80012-9.
- [6] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston: Springer, 2011.
- [7] W. W. Cohen and W. Fan, "Web-collaborative filtering: recommending music by crawling the Web," *Computer Networks*, vol. 33, no. 1, pp. 685–698, 2000, doi: 10.1016/S1389-1286(00)00057-8.
- [8] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Information Sciences*, vol. 180, no. 22, pp. 4290–4311, 2010, doi: 10.1016/j.ins.2010.07.024.
- [9] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998, doi: 10.1111/j.1553-2712.2011.01172.x.
- [10] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 39–46, doi: 10.1145/1277741.1277751.


- [11] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992, doi: 10.1145/138859.138867.
- [12] K. Y. Goldberg and T. M. Roeder, "Eigentaste: A Constant Time Collaborative Filtering Algorithm," *CEUR Workshop Proceedings*, vol. 1225, no. July, pp. 41–42, 2001, doi: 10.1023/A.
- [13] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014, doi: 10.1109/TII.2014.2308433.
- [14] M. D. Ekstrand, "Collaborative Filtering Recommender Systems," *Foundations and Trends® in Human-Computer Interaction*, vol. 4, no. 2, pp. 81–173, 2011, doi: 10.1561/1100000009.
- [15] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005, doi: 10.1109/TKDE.2005.99.
- [16] R. Jin, J. Y. Chai, and L. Si, "An automatic weighting scheme for collaborative filtering," *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*, pp. 337–344, 2004, doi: 10.1145/1008992.1009051.
- [17] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008, doi: 10.1016/j.ins.2007.07.024.
- [18] Y. C. Cai, H. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicality-Based Collaborative Filtering Recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 766–779, 2013, doi: 10.1109/TKDE.2013.7.
- [19] H. Luo, C. Niu, R. Shen, and C. Ullrich, "A collaborative filtering framework based on both local user similarity and global user similarity," *Machine Learning*, vol. 72, no. 3, pp. 231–245, 2008, doi: 10.1007/s10994-008-5068-4.
- [20] B. Zhang and B. Yuan, "Improved collaborative filtering recommendation algorithm of similarity measure," *AIP Conference Proceedings*, vol. 1839, 2017, doi: 10.1063/1.4982532.
- [21] M. Y. H. Al-Shamri and K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1386–1399, 2008, doi: 10.1016/j.eswa.2007.08.016.
- [22] M. Jamali and M. Ester, "TrustWalker: a random walk model for combining trust-based and item-based recommendation," *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 397–406, 2009, doi: citeulike-article-id:5151320.
- [23] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012, doi: 10.1016/j.knosys.2011.07.021.
- [24] J. Bobadilla, F. Serradilla, and J. Bernal, "A new collaborative filtering metric that improves the behavior of recommender systems," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 520–528, 2010, doi: 10.1016/j.knosys.2010.03.009.
- [25] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1310–1316, 2011, doi: 10.1016/j.knosys.2011.06.005.
- [26] J. Bobadilla, A. Hernando, F. Ortega, and J. Bernal, "A framework for collaborative filtering recommender systems," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14609–14623, 2011, doi: 10.1016/j.eswa.2011.05.021.
- [27] J. Bobadilla, A. Hernando, F. Ortega, and A. Gutiérrez, "Collaborative filtering based on significances," *Information Sciences*, vol. 185, no. 1, pp. 1–17, 2012, doi: 10.1016/j.ins.2011.09.014.
- [28] J. Bobadilla, F. Ortega, and A. Hernando, "A collaborative filtering similarity measure based on singularities," *Information Processing and Management*, vol. 48, no. 2, pp. 204–217, 2012, doi: 10.1016/j.ipm.2011.03.007.
- [29] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, 2nd editio., vol. 53, no. 9, 2011.
- [30] G. Koutrika, B. Bercovitz, and H. Garcia-Molina, "FlexRecs: expressing and combining flexible recommendations," *Proceedings of the 35th SIGMOD international conference on Management of data*, pp. 745–758, 2009, doi: 10.1145/1559845.1559923.
- [31] L. Baltrunas and F. Ricci, "Experimental evaluation of context-dependent collaborative filtering using item splitting," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1–2, pp. 7–34, 2014, doi: 10.1007/s11257-012-9137-9.
- [32] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model-based collaborative filtering for personalized POI recommendations," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 907–918, 2015, doi: 10.1109/TMM.2015.2417506.
- [33] D. Anand and K. K. Bharadwaj, "Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5101–5109, 2011, doi: 10.1016/j.eswa.2010.09.141.
- [34] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186, 1994, doi: 10.1145/192844.192905.
- [35] Z. Chen, Y. Wang, S. Zhang, H. Zhong, and L. Chen, "Differentially private user-based collaborative filtering recommendation based on k-means clustering," *Expert Systems with Applications*, vol. 168, no. April 2019, 2021, doi: 10.1016/j.eswa.2020.114366.
- [36] N. Bhalse and R. Thakur, "Algorithm for movie recommendation system using collaborative filtering,"

- Materials Today: Proceedings, no. xxxx, pp. 1–6, 2021, doi: 10.1016/j.matpr.2021.01.235.
- [37] Y. Afoudi, M. Lazaar, and M. Al Achhab, “Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network,” *Simulation Modelling Practice and Theory*, vol. 113, no. July, p. 102375, 2021, doi: 10.1016/j.simpat.2021.102375.
- [38] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A new user similarity model to improve the accuracy of collaborative filtering,” *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014, doi: 10.1016/j.knsys.2013.11.006.
- [39] S. Ahmadian, M. Meghdadi, and M. Afsharchi, “A social recommendation method based on an adaptive neighbor selection mechanism,” *Information Processing and Management*, vol. 0, pp. 1–19, 2017, doi: 10.1016/j.ipm.2017.03.002.
- [40] W. Wang, J. Lu, and G. Zhang, “A new similarity measure-based collaborative filtering approach for recommender systems,” in *Foundations of Intelligent Systems*, 2014, pp. 443–452.
- [41] W. Wang, G. Zhang, and J. Lu, “Collaborative Filtering with Entropy-Driven User Similarity in Recommender Systems,” *International Journal of intelligent Systems*, vol. 30, no. 8, pp. 854–870, 2015, doi: 10.1002/int.
- [42] S. Lee, “Improving Jaccard Index Using Genetic Algorithms for Collaborative Filtering,” in *Information Science and Applications 2017: ICISA 2017*, Springer, 2017, pp. 378–385.
- [43] X. Amatriain, J. M. Pujol, and N. Oliver, “I like it... i like it not: Evaluating user ratings noise in recommender systems,” in *International Conference on User Modeling, Adaptation, and Personalization*, 2009, vol. 5535 LNCS, pp. 247–258, doi: 10.1007/978-3-642-02247-0\_24.
- [44] A. Agarwal and M. Chauhan, “Similarity Measures used in Recommender Systems: A Study,” *International Journal of Engineering Technology Science and Research*, vol. 4, no. 6, pp. 2394–3386, 2017.
- [45] B. Yapriady and A. Uitdenbogerd, “Combining demographic data with collaborative filtering for automatic music recommendation,” in *Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference*, 2005, pp. 201–207, doi: 10.5772/38338.
- [46] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in *10th International Conference on World Wide Web - WWW '01*, 2001, pp. 285–295, doi: 10.1145/371920.372071.
- [47] U. Shardanand and P. Maes, “Social information filtering algorithms for automating ‘word of mouth,’” *Conference proceedings on Human factors in computing systems*, pp. 210–217, 1995.
- [48] Y. Wang, J. Deng, J. Gao, and P. Zhang, “A hybrid user similarity model for collaborative filtering,” *Information Sciences*, vol. 418–419, pp. 102–118, 2017, doi: 10.1016/j.ins.2017.08.008.
- [49] G. Guo, J. Zhang, and N. Yorke-Smith, “A Novel Evidence-Based Bayesian Similarity Measure for Recommender Systems,” *ACM Transactions on the Web*, vol. 10, no. 2, pp. 1–30, 2013, doi: 10.1145/2856037.
- [50] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004, doi: 10.1145/963770.963772.
- [51] K. Miyahara and M. J. Pazzani, “Collaborative Filtering with the Simple Bayesian Classifier,” *IPSI Journal*, vol. 43, no. 11, pp. 679–689, 2002, doi: 10.1007/3-540-44533-1\_68.
- [52] S. J. Yu, “The dynamic competitive recommendation algorithm in social network services,” *Information Sciences*, vol. 187, no. 1, pp. 1–14, 2012, doi: 10.1016/j.ins.2011.10.020.
- [53] C. C. Aggarwal, “Neighborhood-Based Collaborative Filtering,” in *Recommender Systems*, 2016, pp. 29–70.

# Use of Reflection Coefficients and Decision Tree Algorithm for Rapid Classification of Hazardous Chemical Liquids

<sup>1</sup>Ebru Efeoglu, <sup>\*2</sup>Gurkan Tuna

<sup>1</sup>Kütahya Dumlupınar University, Faculty of Engineering, Department of Software Engineering, Kütahya  
ebru.efeoglu@dpu.edu.tr, 

<sup>\*2</sup>Trakya University, Edirne Vocational School of Technical Sciences, Department of Computer Technologies, Edirne  
gurkantuna@trakya.edu.tr, 

## Abstract

The purpose of occupational health and safety studies is to protect employees from work accidents and occupational diseases and to ensure that they work in a healthy environment. Most of the work accidents happen as a result of wrong storage, transportation and use of chemicals. In order to protect employees from chemical hazards and eliminate their possible risks, a risk assessment should first be carried out. Based on the results of the risk assessment, if not done before, bottles that contain hazardous chemicals must be classified and labelled according to their risk levels. The labels of bottles that contain chemical liquids must be checked and if the labels are worn or unreadable, they must be renewed. After these have been done, hazardous chemical liquids must be classified, stored and transported according to these labels. In this study, a non-contact, liquid measurement system based on microwave data is proposed to detect hazardous liquids. In order to select the most suitable algorithm for use in this measurement system, 3 different classification algorithms have been used and the performance analysis of the algorithms has made. In the study, 3 different classification processes have been applied according to the chemical properties of the liquids. It has been observed that Random Tree algorithm has achieved the best performance while Rep Tree algorithm has done the worst performance. Using this system, hazardous chemical liquids can be detected without opening the cover of the bottles that contain the liquids. Therefore, it can be used to quickly label hazardous liquids for their safe storage and transportation.

**Keywords:** Decision Tree Algorithm, Hazardous Chemicals, Microwave, Occupational Safety, Patch Antenna

## 1. INTRODUCTION

Most work accidents occur when the hazards of a working environment are not properly managed, either because they are not perceived as risks before the work starts, or because they are not fully understood. Therefore, appropriate training should be given to new employees to recognise and understand the hazards [1]. It has been reported that young workers have a higher non-fatal accident rate than older workers [2,3]. In order to create and maintain a safe working environment, hazards that may occur at workplaces should be identified and it is necessary to effectively monitor potential risks and give priority to preventive activities [4].

Although chemicals play an important role in the daily life of people around the world [5,6], some of the work accidents occur due to the chemicals. The chemical industry produces

a wide variety of substances necessary for daily use and chemical hazards generally arise from chemical synthesis or production, processing, transportation and misuse. Compared to the other substances, explosive, flammable and poisonous ones are more dangerous when released inappropriately [7]. It is well-known that the most dangerous of work accidents are explosions and fires that occur during the storage and transportation of petroleum and petroleum products, as well as other types of fuel [8,9]. On the other hand, there are ones like chemical liquids, which are less found in daily life [10]. Chemical liquids are often used in laboratories. Laboratories are one of the places where the danger and thus risks are almost the highest in terms of chemical hazards. Since flammable, explosive and poisonous chemicals are stored in chemical laboratories, they have many potential work accident risks. In addition, there is a higher risk of fire and explosion in them due to

\* Corresponding Author

electrical equipment used in chemical experiments, high temperature and pressure.

Chemical liquids can be divided into 5 main groups as easily flammable liquids, corrosive liquids, toxic liquids, oxidising liquids and explosive liquids as given in Figure 1 [11,12]. Different measures have been proposed to prevent explosions related to the use of chemical liquids [12]. Figure 1 gives general information that indicates which class should be stored together or not. As it is known, the most hazardous chemicals encountered in major work accidents are sulphuric acid, hydrochloric acid and ammonia [11]. If oxidising chemicals come into contact with combustible materials, fire or explosion may occur. In the event of impact or spillage, these materials may be mixed with organic materials and fire and explosion may occur. These chemicals are also sensitive to heat and should be stored in a cool place. The flash point of a volatile substance is the lowest temperature at which the vapour of the substance will ignite when a source of ignition is given. In most cases, as the flash point decreases, the relative danger of a flammable liquid increases. Most of the solvents available in laboratories have flash points well below room temperature. Therefore, their proper use, storage and disposal is highly critical.














It has been suggested to use different techniques such as nuclear magnetic resonance and X-ray [13, 14], NQR method [15] for the detection of hazardous liquids. X-ray systems are the most widely used among these techniques [14]. In addition, liquid detection and identification can be performed using THz time domain spectroscopy [16]. However, these methods are generally effective in detecting peroxide-based liquids. And they cannot distinguish many types of liquids used in daily life. Therefore, there is a need for a system to distinguish these liquids [17].

The proposed system is based on microwave and decision trees. There are many microwave measurement methods. Of these methods the coaxial probe method is commonly used in microwave based systems. It has been used for different purposes including the examination of the microwave absorbing properties of ionic liquids at room temperature [18], electromagnetic properties of water and selected fruits and vegetables [19] and dielectric measurement of biological tissues [20]. In order to make measurements with this technique, the probe must be touched on the solid material in solids and the probe must be immersed in the liquid for liquid measurements. Metamaterial-based compact microwave liquid sensor was proposed for dielectric characterization of liquids [21].

In this study, a non-contact, a liquid classification system that can quickly detect hazardous liquids is presented. The classification system can classify liquids contained in bottles without the need of opening the caps of the bottles and without the possibility of being exposed to any breathing and skin contact with the liquids. It is a microwave-based system and based on the use of well-known Decision Tree algorithm family for classification. In the study, three different classification processes were applied to liquids according to

their chemical content and intended use. In the classification studies, 3 different decision tree algorithms were used and their performances were compared. As a result of the performance analysis study, it was understood that Random Tree algorithm was the most successful algorithm.

The rest of this paper is as follows. Experimental setup, methodology used in this paper and Decision Tree algorithms are given in Section 2. Section 3 presents experimental study and its results. Finally, the paper is concluded in Section 4.

					
Flammable	 +	-	-	-	+
Explosive	-	 +	-	-	-
Toxic	-	-	 +	-	+
Oxidising	-	-	-	 +	 ○
Irritant harmful	 +	-	 +	 ○	+

- Can not be stored  
+ Storable  
○ Can be stored with special precautions

Figure 1. Chemical substance storage matrix.

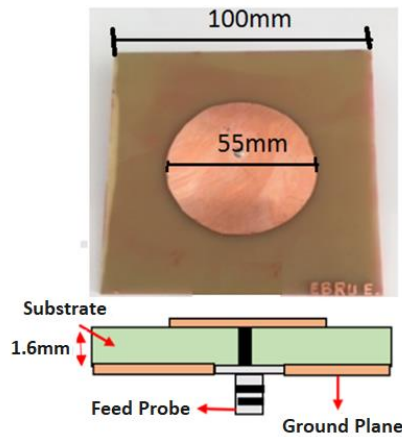
## 2. MATERIALS AND METHODS

The experimental setup used in this study for liquid classification using microwave patch antenna is shown in Figure 2. It consists of a microwave circular patch antenna design connected to a vector network analyser in order to measure of the reflection coefficient of electromagnetic wave. To build the experimental setup, an antenna with a resonant frequency of 1.5 GHz was designed. The design was constructed on a FR4 based dielectric substrate with 1.6 mm height, 4.4 relative permittivity and 10x10 cm<sup>2</sup> ground plane beneath it. The antenna is feed by 50 Ohm SMA (Sub Miniature Version A) feed probe. The design and geometry of the antenna is illustrated in Figure 3. The electromagnetic wave reflection coefficient of the liquids was measured by placing the bottle on the antenna. For each liquid measurement, the reflection coefficient was recorded at 40 measurement points between 1.52-1.42GHz frequency band. The dataset, consisting of 31 liquid measurements, formed a 31x40 matrix in total. Then, this data matrix was given as input to classification algorithms.



Figure 2. Experimental setup for liquid classification.





**Figure 3.** Design and geometry of the antenna.

Diameter of the antenna is calculated using (1) and (2).

$$F = \frac{8,791 \times 10^9}{f_r \sqrt{\epsilon_r}} \quad (1)$$

$$a = \frac{F}{\left\{ 1 + \frac{2h}{\pi \epsilon_r F \left[ \ln \left( \frac{\pi F}{2h} \right) + 1,7726 \right]^{1/2}} \right\}} \quad (2)$$

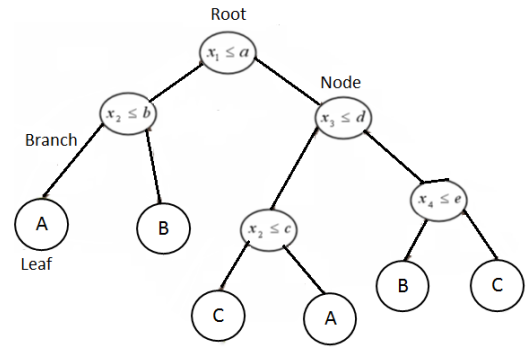
where  $\epsilon_r$  is relative permittivity of the substrate,  $f_r$  is the resonant frequency,  $h$  is the height of the substrate, and  $a$  is the radius of the patch.

## 2.1. Decision Trees

A decision tree structure consists of a root node containing data, inner nodes (branches) and end nodes (leaves). When decision tree algorithm is used, first, a decision tree is created, then the rules produced from the decision tree is used to classify the records in the database. By applying this created decision tree class on unknown data, the classes of this data are determined. In this tree structure, each node represents an attribute [22]. In Figure 4, a tree structure consisting of four dimensional attribute values belonging to three classes are shown. In this figure,  $x_i$  represents attribute values; The values  $a$ ,  $b$ ,  $c$ ,  $d$  and  $e$  indicate the threshold values used for branching, and  $A$ ,  $B$  and  $C$  indicate the class labels. There are univariate or multivariate decision tree structures according to the number of variables used in each stage of tree formation [23].

The most important step in creating decision trees is the selection of criteria that the branch of the tree will be made. Some of the approaches that can be used for this purpose are knowledge gain and knowledge gain rate [24], Gini index [25], Towing rule [24] and Chi Square probability table statistics [26]. The entropy method is used to determine which branch of the decision tree will be used in the use of information gain and information gain rate. Let  $C_1, C_2, \dots, C_n$  be a dataset consisting of several classes. If  $T$  shows class values, the probability of a class can be computed using (3).

$$P_i = (C_i / |T|) \quad (3)$$



**Figure 4.** A decision tree structure consisting of three classes with four dimensional property spaces.

The entropy value of the classes is computed using (4).

$$Entropi(T) = - \sum_{i=1}^n p_i \log_2(p_i) \quad (4)$$

$T$  class values according to  $B$  attribute in the data set  $T_1, T_2, \dots, T_n$ . Considering that it is divided into sub-sets, the gain to be obtained by using  $B$  attribute values is computed using (5).

$$Gain(B, T) = Entropi(T) - \sum_T \frac{|T_i|}{|T|} Entropi(T_i) \quad (5)$$

Segmentation information for determining the value of  $B$  attribute for the  $T$  set is computed using (6).

$$B = - \sum_{i=1}^k \frac{|T_i|}{|T|} \log_2 \left( \frac{|T_i|}{|T|} \right) \quad (6)$$

Then, the gain rate can be computed using (7).

$$Gain\ rate = \frac{Gain(B, T)}{B} \quad (7)$$

Using this criterion, the  $T$  training set is separated repeatedly, with the maximum rate of gains at each node of the tree. This process is repeated until each leaf node contains observation values of only one class. The decision tree classification algorithm divides the training data into subsets containing only one class, resulting in a very large and complex tree structure. For this reason, it is possible to replace a subtree with a leaf. This process is called pruning and removes the parts of the decision tree that do not affect or contribute to the classification accuracy in order to obtain a more understandable tree structure [27]. Pre-pruning is done while creating the tree structure. On the other hand, post-pruning is done after the tree structure is created [24].

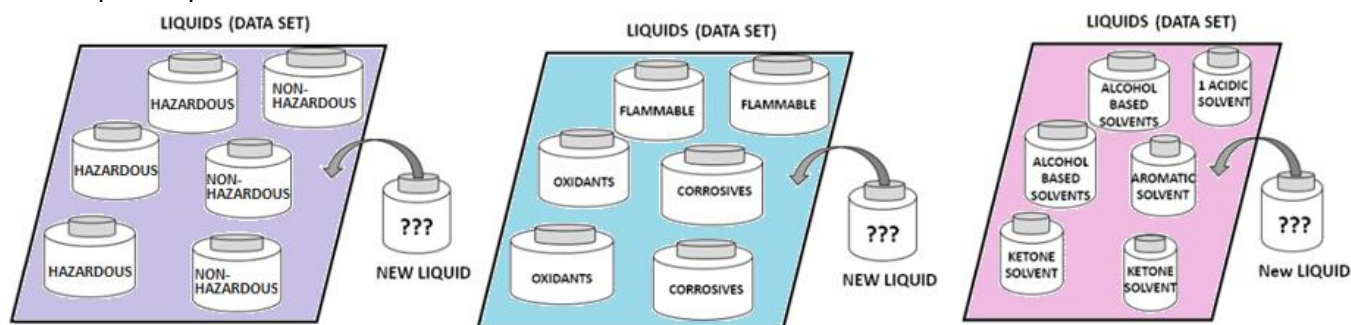
In this study, Random Tree, Rep Tree and Extra Trees algorithms are used for the classification process. Random Tree is a classification algorithm that creates a tree by taking a certain number of randomly selected properties in each node. There is no pruning. It also has an option that allows estimation of class possibilities based on the data set held.



Rep Tree algorithm was first proposed by Quinlan [27]. It is fast machine learning that creates a decision tree by pruning to reduce the effects of noise in the training examples [28]. With the regression tree logic, the best one is selected from the decision trees created after creating more than one tree in different iterations. The tree is pruned by reduced error pruning from the bottom up [29]. Because, the pruned tree reduces complexity in the classification process. The leaves of the decision tree always contain the exit values. The outputs at each branch point in the decision tree are selected based on the reduction of the "special threshold value" variance. To sum up, Rep Tree algorithm is based on the principle of minimizing the error caused by variance and the principle of gaining knowledge with entropy [29] and only works with numerical data. Extra Trees algorithm creates a community of decision trees that have not been pruned to the classic top-down procedure. The two main differences with

other tree-based community methods are that they select the breakpoints completely randomly, separating nodes and using the entire learning example to grow trees [30].

Decision tree algorithms are supervised learning algorithms. In this study, the purpose of using them is to predict which class the new liquid belongs to, using the model created from the data in the database when data from a new liquid arrives. Schematic representation of the classification processes held in this study is shown in Figure 5. As shown in this figure, in this study, 3 different classification experiments were carried out. In the first experiment, a total of 30 liquids listed in Table 1 were classified into two categories: hazardous or non-hazardous. Extra Trees, Random Tree and Rep Tree algorithms were used to classify the liquids given in Table 1.



**Figure 5.** Classification of hazardous liquids.

### 3. RESULTS AND DISCUSSION

For the first classification experiment, a database was created from  $S_{11}$  parameter (reflection coefficient) measurements and two different methods were used to test the success of Extra Trees, Random Tree and Rep Tree algorithms. In the first method, all the data in the database was used as the training. In the second method cross validation was used. The purpose of doing this was to test the success of the algorithms in classifying liquids that were not in the database. Frequency-dependent reflection coefficient measurements of the liquids used in the classification experiment with the whole data set are given in Figure 6.

When Figures 7, 8 and 9 are considered, it can be seen that Extra Trees and Random Tree algorithms correctly classified all the liquids in the classification without cross validation but Rep Tree algorithm misclassified 3 liquids. When cross-validated, Extra Trees algorithm misclassified 3 of the liquids, Random Tree algorithm misclassified 2 of the liquids and Rep Tree algorithm did 6 of the liquids, respectively.

Other classification metrics of the algorithms are presented in Figure 10. As it can be seen, high accuracy, precision and recall values are obtained when Random Tree algorithm was used.

Hazardous chemical liquids can be generally divided into 3 groups as shown in Table 2. These are flammable (ethanol, methanol, acetone, 1-propanol, 2 propanol, butanol), corrosives (sulphuric acid, nitric acid, acetic acid), oxidants (hydrogen peroxide). In the second classification experiment, hazardous chemical liquids were divided based on the abovementioned groups. Cross validation was not used in the remaining experimental studies since the number of liquids was limited. In other words, the entire dataset was used in the classification process.

Frequency-dependent reflection coefficient measurements of the liquids used in the second classification experiment are given in Figure 11. Hazardous liquids were divided into 3 classes using Extra Trees, Random Tree and Rep Tree algorithms. The confusion matrices of the algorithms are presented in Figure 12. Other classification metrics of the algorithms are presented in Figure 13.

**Table 1.** Liquids used in the classification experiments.

Hazardous liquids			Non-Hazardous liquids		
Ethanol	1-propanol	Water	Turnip juice	Raki	Peach juice
Toluene	Methanol	Soda	Milk	Champagne	Ice-tea(Peach)
Butanol	Acetone	Beer	Liquid hand soap	Tequila	Apricot juice
Nitric acid	Acetic acid	Mineral water	Buttermilk	Whiskey	Vinegar
Sulphuric acid	Isopropanol	Rose juice	Gin	Vodka	Liqueur

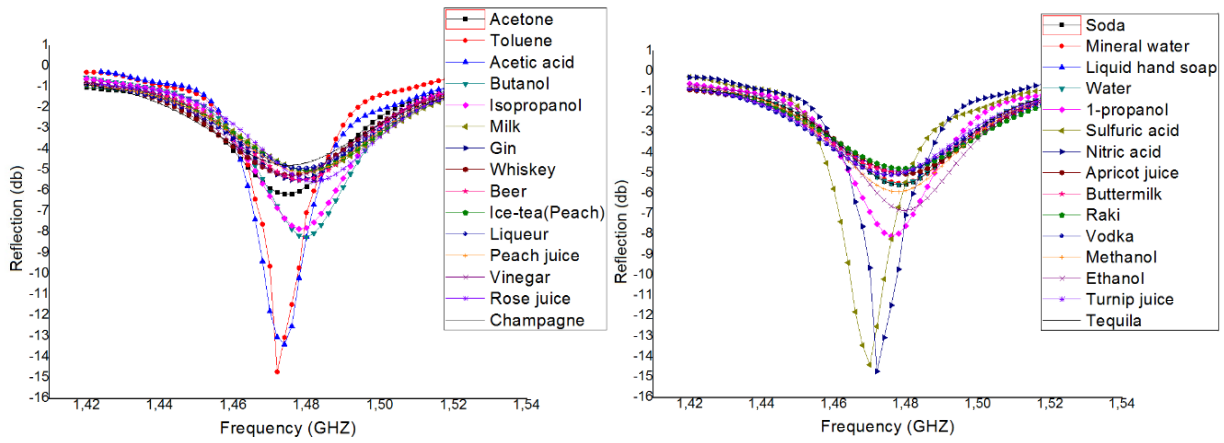


Figure 6. Reflection coefficient measurements of the liquids used in the first classification experiment.

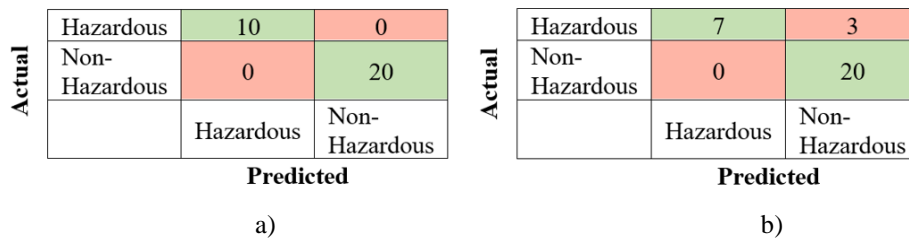


Figure 7. Confusion matrix of Extra Trees a) without cross validation b) after cross validation.

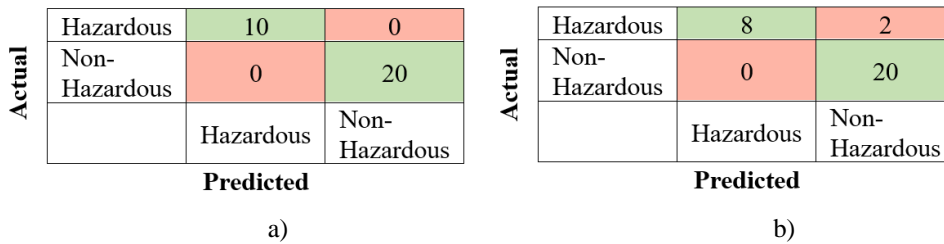


Figure 8. Confusion matrix of Random Trees a) without cross validation b) after cross validation.

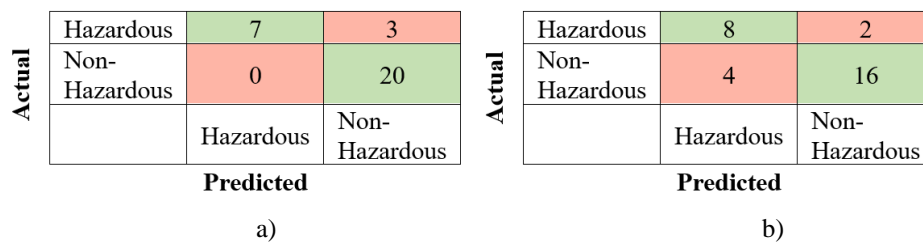


Figure 9. Confusion matrix of Rep Trees a) without cross validation b) after cross validation.

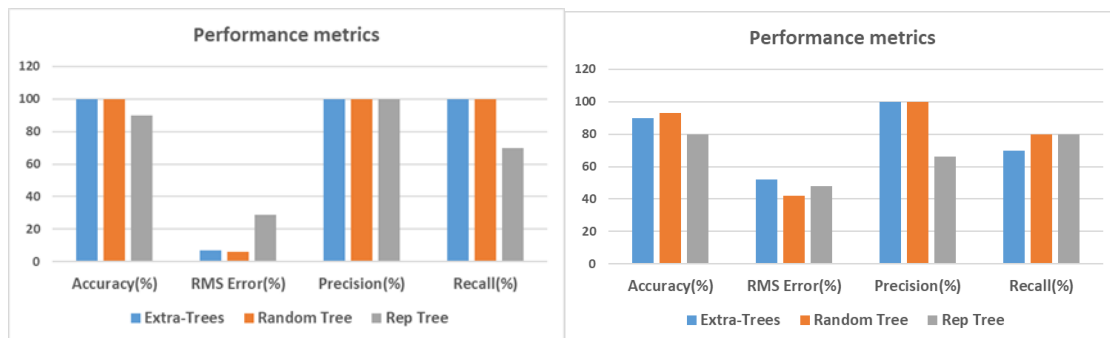
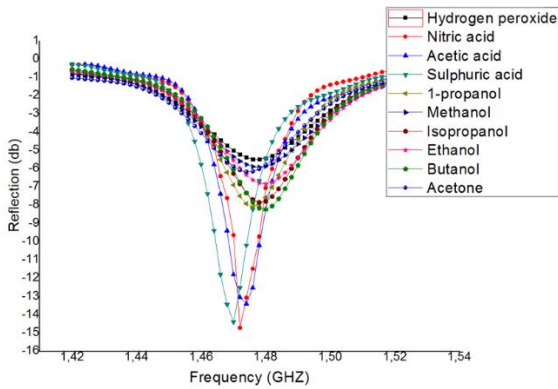


Figure 10. Performance metrics a) without cross validation b) after cross validation.

**Table 2.** Hazardous chemical liquids divided into 3 groups.

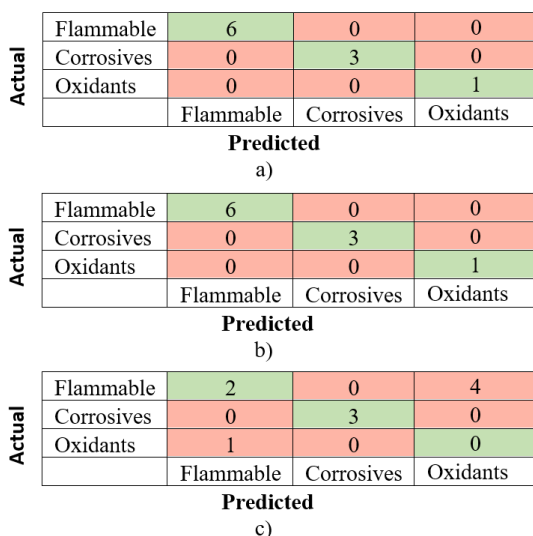
Flammable	Corrosives	Oxidants
Ethanol	Sulphuric acid	Hydrogen peroxide
Butanol	Acetic acid	
1-propanol	Nitric acid	
Methanol		
Acetone		
Isopropanol		



**Figure 11.** Reflection coefficient measurements of the liquids used in separating hazardous chemical liquids into 3 groups.

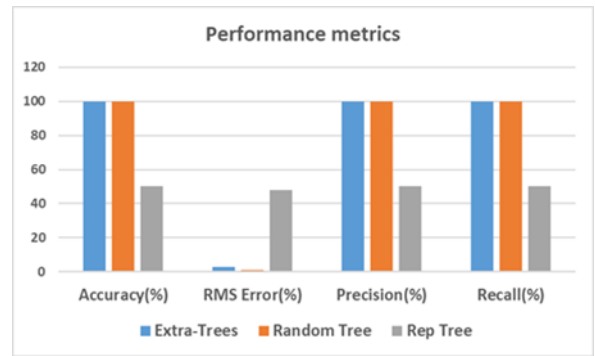
**Table 3.** Solvents used in the classification process.

Alcohol-Based Solvents	Ketone Solvents	Aromatic Solvents	Acidic Solvents
Ethanol	Acetone	Toluene	Acetic
Isopropanol			
Butanol			
1-propanol			
Methanol			



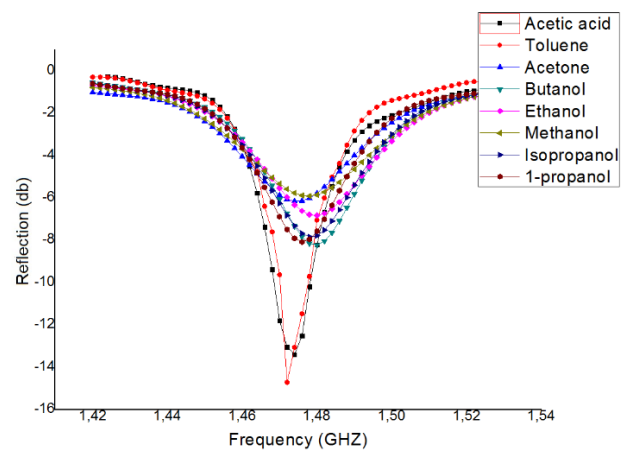
**Figure 12.** Confusion matrix a) Extra tree b) Random tree

c) Rep tree.

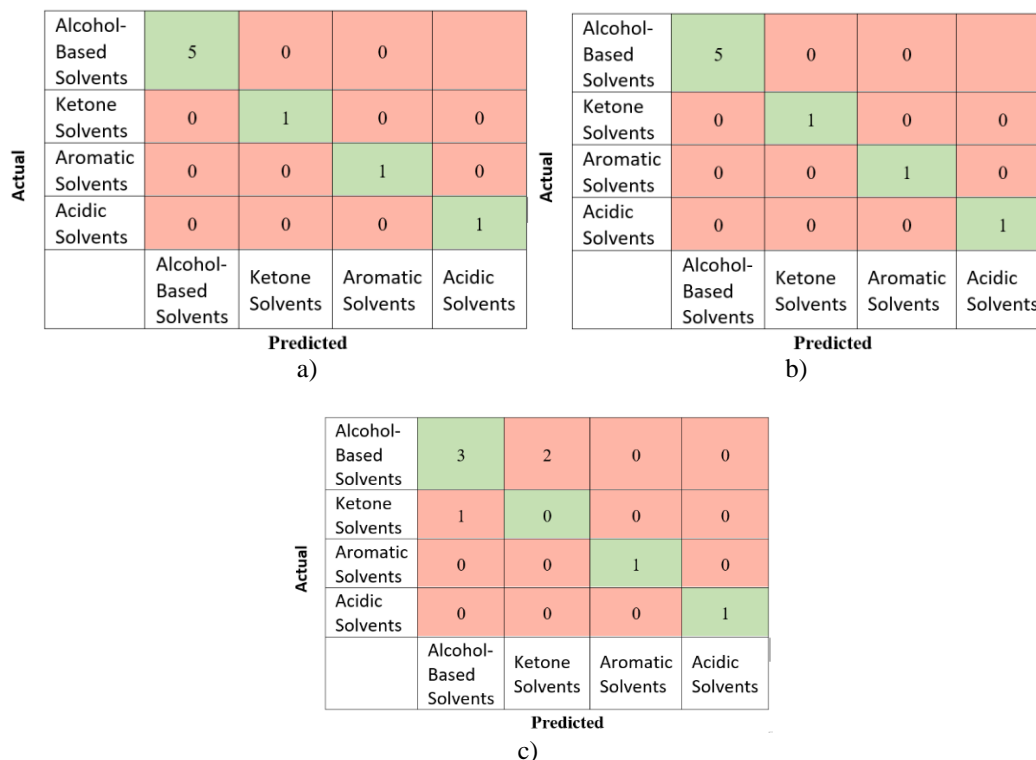


**Figure 13.** Performance metrics.

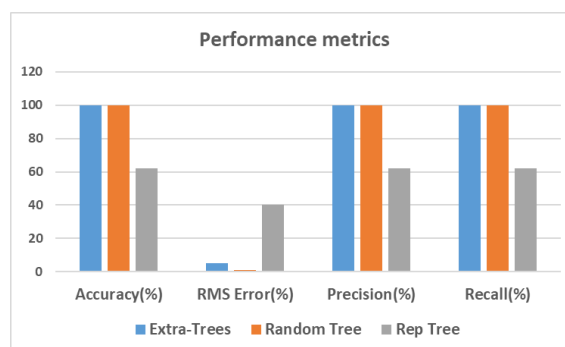
In this classification experiment, Extra Trees algorithm provided 100% accuracy, Random Tree did 100% accuracy, and Rep Tree algorithm did 50% accuracy. Random Tree had a better RMS error value compared to Extra Trees algorithm. Frequently used organic solvents both create risks for human and environmental health during their use and cause an environmental problem that is difficult to dispose when it becomes waste. However, there are situations where the use of solvents is mandatory. In these cases, it is necessary to choose the safest alternatives when choosing the solvent. As listed in Table 3, in the third classification experiment a total of 8, 5 of which are alcohol based solvents (Methanol, Ethanol, 1 propanol, 2-propanol, Butanol), 1 ketone solvent (Acetone), 1 aromatic solvent (Toluene) and 1 acidic solvent (acetic acid), were used and the solvents were divided into 4 main groups among themselves. Frequency-dependent reflection coefficient measurements of the liquids used in the third classification experiment are given in Figure 14. Confusion matrices and other performance metrics are presented in Figure 15 and Figure 16. In this classification experiment, Extra Trees algorithm achieved 100% accuracy, Random Tree achieved 100% accuracy, and Rep Tree algorithm did 62% accuracy. Random Tree had a better RMS value compared to Extra Trees algorithm. In this classification, Rep Tree algorithm had the highest error value and the worst performance metrics.



**Figure 14.** Reflection coefficient measurements of liquids used in the classification of solvent liquids



**Figure 15.** Confusion matrices a) Extra Trees b) Random Tree c) Rep Tree.



**Figure 16.** Performance metrics.

#### 4. CONCLUSION

Thousands of people are at the risk of exposure to hazardous chemicals in their workplaces. If the potential risks associated with hazardous chemicals are not identified beforehand and not controlled by appropriate methods, serious work accidents, occupational diseases and even deaths can occur. On the other hand, if appropriate precautions are taken, both the health and safety of the workers can be secured and the loss of working days due to temporary and permanent incapacity or death can be prevented. In this study, a non-contact, liquid classification system that can detect hazardous liquids was presented. The classification system can quickly classify liquids contained in bottles without the need of opening the caps of the bottles. The classification system is based on a group of algorithms that are members of the well-known, Decision Tree algorithm family. The success of the algorithms was tested with a group of liquids and satisfactory results were obtained. Three classification experiments were carried out

and in each of them three different classification algorithms were used. In the first classification experiment, hazardous liquids and non-hazardous liquids were distinguished from

each other. In this classification, Extra Trees algorithm and Random Tree algorithm achieved 100% accuracy in the classification process when the whole data set was used. For liquids not found in the database, Random Tree algorithm performed better than Extra Trees algorithm with an accuracy rate of 80%. The worst result in this classification was obtained by Rep Tree algorithm. This algorithm obtained 90% accuracy in the classification when the whole dataset was used and 60% in the classification of liquids that are not in the database. In the second classification experiment, hazardous chemical liquids were divided into three groups: flammable, corrosive and oxidants. In this classification, Extra Trees and Random Tree algorithms classified the liquids with 100%.

The proposed system in the study is still under development. In the future study, the classification process will be applied by increasing the number of liquids in the database. After increasing the number of measurement data for liquids and applying many tests, the system can be used to ensure occupational safety.

**Author contributions:** Concept – E.E., G.T.; Data Collection and Processing - E.E.; Literature Search – E.E.; Discussion and Writing – E.E., G.T.

**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] R. H. Hill Jr, "Recognizing and understanding hazards—The key first step to safety," *Journal of Chemical Health and Safety*, vol. 26, no. 3, pp. 5-10, 2019.
- [2] S. Salminen, "Have young workers more injuries than older ones? An international literature review," *Journal of safety research*, vol. 35, no. 5, pp. 513-521, 2004.
- [3] A. Parent-Thirion et al., *Sixth European Working Conditions Survey: Overview Report*. Eurofound (European Foundation for the Improvement of Living and Working ...), 2016.
- [4] A. Byzov, A. Telegina, I. Korotkiy, and J. Veber, "Consequence assessment of explosions for fuel-air mixtures at hazardous production facilities," in *E3S Web of Conferences*, vol. 140: EDP Sciences, p. 08014, (2019).
- [5] J. Casal, *Evaluation of the effects and consequences of major accidents in industrial plants*. Elsevier, 2017.
- [6] A. Z. Mendiburu, J. A. de Carvalho Jr, and C. R. Coronado, "Method for determination of flammability limits of gaseous compounds diluted with N<sub>2</sub> and CO<sub>2</sub> in air," *Fuel*, vol. 226, pp. 65-80, 2018.
- [7] D. K. Horton, Z. Berkowitz, G. S. Haugh, M. F. Orr, and W. E. Kaye, "Acute public health consequences associated with hazardous substances released during transit, 1993–2000," *Journal of hazardous materials*, vol. 98, no. 1-3, pp. 161-175, 2003.
- [8] K. Kempna et al., "Fire Safety Protection Assessment of Industrial Technologies," in *Journal of Physics: Conference Series*, 2018, vol. 1107, no. 4: IOP Publishing, p. 042036.
- [9] A. Nikulin and A. Y. Nikulina, "Assessment of occupational health and safety effectiveness at a mining company," *Ecology, Environment and Conservation*, no. 23, p. 1.
- [10] C. Wei, W. J. Rogers, and M. S. Mannan, "Application of screening tools in the prevention of reactive chemical incidents," *Journal of Loss Prevention in the Process Industries*, vol. 17, no. 4, pp. 261-269, 2004.
- [11] T. Hoppe, N. Jaeger, and J. Terry, "Safe handling of combustible powders during transportation, charging, discharging and storage," *Journal of Loss Prevention in the Process Industries*, vol. 13, no. 3-5, pp. 253-263, 2000.
- [12] W. L. Welles, R. E. Wilburn, J. K. Ehrlich, and C. M. Florida, "New York hazardous substances emergency events surveillance: learning from hazardous substances releases to improve safety," *Journal of hazardous materials*, vol. 115, no. 1-3, pp. 39-49, 2004.
- [13] S. Kumar, "Liquid-contents verification for explosives, other hazards, and contraband by magnetic resonance," *Appl. Magn. Reson.*, 2004, vol. 25, nos. 3–4, pp. 585–597.
- [14] S. Singh and M. Singh, "Explosives detection systems (EDS) for aviation security," *Signal Process.*, vol. 83, no. 1, pp. 31–55, 2003.
- [15] L. Cardona, J. Jiménez and N. Vanegas, "Nuclear quadrupole resonance for explosive detection," *Ingeniare Revista chilena de ingeniería*, vol. 23, no. 3, pp. 458–472, 2015.
- [16] K. Choi, T. Hong, K. I. Sim, T. Ha, B.C. Park, J.H. Chung, et al. "Reflection terahertz time-domain spectroscopy of RDX and HMX explosives," *J. Appl. Phys.*, vol. 115, no. 2, p. 023105, 2014.
- [17] Z.Z. Abidin, F.N. Omar, P. Yogarajah, D.R.A. Biak, and Y.B.C. Man, "Dielectric characterization of liquid containing low alcoholic content for potential halal authentication in the 0.5-50 GHz range," *Am. J. Appl. Sci.*, vol. 11, no. 7, pp. 1104–1112, 2014.
- [18] F. Yang , J. Gong , E. Yang , Y. Guan , X. He , S. Liu, X. Zhang , Y. Deng, " Microwave-absorbing properties of room-temperature ionic liquids, " *Journal of Physics D: Applied Physics*, 52(15):155302, 2019.
- [19] A. La Gioia et al., "Open-ended coaxial probe technique for dielectric measurement of biological tissues: Challenges and common practices," *Diagnostics*, vol. 8, no. 2, p. 40, 2018.
- [20] T. Karpisz, B. Salski, P. Kopyt, J. Krupka, "Measurement of Electromagnetic Properties of Food Products and Liquids," In: 2018 12th International Conference on Electromagnetic Wave Interaction with Water and Moist Substances (ISEMA): 2018: IEEE: 1-9, (2018).
- [21] S. Kayal, T. Shaw, D. Mitra, "Design of metamaterial based compact and highly sensitive microwave liquid sensor," *Applied Physics A* ,126:13, 2020.
- [22] M. Pal and P. M. Mather, "An assessment of the effectiveness of decision tree methods for land cover classification," *Remote sensing of environment*, vol. 86, no. 4, pp. 554-565, 2003.
- [23] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote sensing of environment*, vol. 61, no. 3, pp. 399-409, 1997.
- [24] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.
- [25] W. Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 1, pp. 14-23, 2011.
- [26] J. Mingers, "An empirical comparison of pruning methods for decision tree induction," *Machine learning*, vol. 4, no. 2, pp. 227-243, 1989.
- [27] J. R. Quinlan, "Simplifying decision trees," 1986.
- [28] J. Li, S. Zhang, Y. Lu, and J. Yan, "Real-time P2P traffic identification," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, 2008: IEEE, pp. 1-5.
- [29] M. F. Amasyali and O. Ersoy, "Evaluation of regression ensembles on drug design datasets," 2009.
- [30] Y. Freund and L. Mason, "The alternating decision tree learning algorithm," in *icml*, 1999, vol. 99, pp. 124-133.