e-ISSN: 2980-177X

JCEES

Volume: 1 Issue: 2 Year: 2023

Journal of Computer & Electrical and Electronics Engineering Sciences

MHA
MEDIHEALTH ACADEMY

Journal of
Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

Journal of
Computer & Electrical and Electronics Engineering Sciences

EDITORIAL BOARD

EDITORIAL BOARD

Journal of
Computer & Electrical and Electronics Engineering Sciences

EDITORIAL

Dear Colleagues,

Academic journals are one of the most essential elements of the academy, where researchers announce their findings to the scientific world. Therefore, an academic journal should have continuity, stability, and quality characteristics.

I am honored and happy to greet you with the first issue of our journal, “Journal of Computer & Electrical and Electronics Engineering Sciences,” which we set out to contribute to the scientific world with the great sense of responsibility brought by academic publishing and quality publishing.

Our journal, which aims to publish original and high-quality studies in Computer Engineering and Electrical-Electronics Engineering, started its publication to announce new studies and original research in the field to the scientific world.

Our journal, which evaluates the principle of blind refereeing, is also sensitive to protecting personal data and ethical concerns.

Our journal, which we plan to be published as two issues a year, will climb the steps of quality in academic publishing one by one. My belief comes from our publishing team, who work voluntarily for this job. As the editor-in-chief, I would like to thank this beautiful team dedicated to science.

We know that with each subsequent issue, our intensity will increase; However, with our commitment to science, we will continue to offer you the latest studies as a team without compromising our principles. I want to express my sincere gratitude and respect to you, our readers, writers, referees, and our publishing team working with great devotion; I sincerely look forward to your comments and suggestions for improving our journal.

Assist.Prof.Dr. Fuat Türk, PhD
Editor-in-Chief

Journal of
Computer & Electrical and Electronics Engineering Sciences

CONTENTS

Volume: 1 **Issue: 2** **Year: 2023**

ORIGINAL ARTICLES

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0006

Personalized daily-weekly workout arrangement application "cardio fit"

Zülal Ataman[1] Mehmet Emin Heybet[1] Fatih Sarısoy[2] Büşra Yılmaz[2] Mehmet Günay[3] Ömer Şenel[4]

[1]*Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey*
[2]*Faculty of Sport Sciences,Gazi University, Ankara, Turkey*
[3]*Department of Physical Education and Sports Education, Faculty of Sport Sciences, Gazi University, Ankara, Turkey*
[4]*Department of Coaching Education, Faculty of Sport Sciences, Gazi University, Ankara, Turkey*

Cite this article: Ataman Z, Heybet ME, Sarısoy F, Yılmaz B, Günay M, Şenel Ö. Personalized daily-weekly workout arrangement application "cardio fit" *J Comp Electr Electron Eng Sci.* 2023;1(2):25-28.

Corresponding Author: Zülal Ataman, zulalatamaaan@gmail.com

Received: 13/07/2023 ◆ **Accepted**: 21/08/2023 ◆ **Published**: 31/10/2023

ABSTRACT

Having mobile applications has become an indispensable part of our lives in recent years. There are many mobile applications that are already being used in the field of fitness. By using these apps, people can lead a more organized and planned lifestyle. However, many apps fail to offer personalized programs for individuals. This makes mobile applications an inefficient method for users to reach their goals. Our mobile application aims to enable users to reach their fitness level effectively and quickly with a program suitable for their body type, physical characteristics, training goals and exercise preferences. Body mass indexes are calculated based on the body measurements that users enter and their maximum oxygen capacity is determined according to their age, the user chooses their fitness goals, selects the equipment they will train with, and indicates whether they have heart rate monitors. Based on this, the appropriate program is presented to the user. This app is not suitable for users over 50 years old. Users have two goal options for the app: "Feeling Good – Breaking a Sweat" and "Achieving Fitness Level (Firming-Fat Burning)." In the "Reach Fitness Level" option, a tightening program is recommended if the user's weight is above their ideal weight, while a fat burning program is recommended for users who are significantly above their ideal weight. The Kotlin programming language was used to develop this application, and the Firebase database was used to speed up the development process to ensure the effectiveness of fitness programs, the cardio program was provided by a fitness trainer who is an expert in the field. This app retrieves the necessary user information from the database and provides them with the most suitable fitness program. As a result, users can achieve their goals without needing a fitness trainer or going to the gym.

Keywords: Mobile app, kotlin, firebase, fitness, fitness app

INTRODUCTION

With the widespread use of technology, having mobile applications is an indispensable part of our lives. Also, people can save time by using mobile apps to accomplish many tasks. Many people equally use mobile apps to perform their daily activities. Fitness applications are also among the mobile applications that are frequently used in daily life (Azar, Lesser, Laing et al. 2013; Fanning 2012).

Through fitness apps, users can create personalized workouts, track their progress, receive social support, and find motivation to achieve their goals (Direito, 2017). Fitness apps are mobile app services that offer programs such as exercise, nutritional guidance, and monitoring children's health. It is possible to exemplify some of the most used applications according to the number of downloads and popularity as follows (Azar et al, 2013).

- Nike Training Club: A workout app that provides users with features such as a variety of fitness exercises, training programs, video guides, and progress tracking (MyFitnessPal: Calorie counter and nutrition tracking app, 2023).
- MyFitnessPal: A health and nutrition system with features such as nutrition tracking, calorie calculation, food diarying, meal planning, and a macronutrient breakdown app (Wang et al, 2016).
- Strava: It is a fitness tracking application that tracks users' outdoor sports such as Running, Walking, Horseback Riding, Swimming, Cycling using GPS data. You can record your activities and track data such as speed, distance and time, as well as interact with other athletes
- Seven-7 Minute Workout: High-intensity 7-minute workout routines. It is an application for users. Users can

practice exercises at home, in the office, or while traveling and survive in a short time by having an effective training experience (Seven: Fitness and Workout App, 2023).

- Fitbit: An app and accessory that offers a variety of features for a combination of fitness and health tracking. Users can track steps, calories burned, sleep quality, and heart rate. It can track its speed. You can also set activity goals and work within the community. Can compete with other users (Google Fit: Health & Fitness Tracking App, 2023).
- Google Fit: Allows users to track their daily activities, count steps, calories It is a fitness app that offers features such as calculating your mileage and setting goals. It also has the ability to automatically recognize different sports activities (Fitbit: Activity and Health Trackers, 2023).

These apps often help users set their health and fitness goals and provide preventative training programs and diet plans to achieve them (Peterson et al, 2017). In addition, some fitness apps offer rewards, achievement badges, and motivational features to achieve social share (Azar et al, 2013).

In addition, Fitness apps allow users to exercise with more motivation and enjoyment according to their interests (Wang et al, 2016).

Research shows that the use of fitness apps increases users' physical activity levels, increases their motivation and engagement, and helps them achieve their health goals (Wang et al, 2016; Fanning et al, 2012).

The aim of this study is to develop an application that is easily accessible to everyone and allows to achieve a fitness level in a short time with a professional program that can be applied in the gym or to provide an exercise routine without the need for fitness trainers. Another purpose is to create an application that is different from other applications in the current market, apart from being more useful. This application primarily stores the necessary information about individuals in its database. For convenience, the user has been advised to use a simple and visually appealing interface during data collection. The data collected is information such as whether there is a smart watch, body measurements, age, gender, fitness goals, equipment such as exercise information, and heart rate monitors, and in the light of this information, it is aimed to offer the most suitable fitness programs to the users. Research studies show that the fitness market is growing every year and the usage rate is increasing (Farrokhi et al, 2021).

A study conducted by Azar, which focuses on the theoretical study analysis of content mobile applications, is based on weight management. This research showed that; These apps are effective in helping users set weight loss goals, create meal plans, and track their exercise activity (Azar et al, 2013). Another study by Wang aimed to increase the physical activity levels of overweight and obese adults. used a wearable device called; These results show "Fitbit One and SMS" text messages. This combination led to a significant physical increase in activity level (Wang et al, 2016).

Another lesson in the study was to increase user motivation and have a positive impact on exercise frequency The researchers find that rewards, achievement badges, and social sharing features expand with fitness apps provided for exercise motivation, encouraging users to exercise regularly (Direito et al, 2017). Another study shows the effectiveness of smart fitness apps in improving exercise performance Researchers have determined that smart fitness apps provide feedback to users, which helps improve exercise techniques, optimize workout times, and achieve target heart rates (Martin et al, 2015).

Studies by Smith et al. have shown that the use of a smart bracelet and a compatible mobile app increases users' physical activity levels and increases the frequency of exercise (Smith et al, 2016).

In another study by Direito et al.; It showed that the use of a smartphone-based fitness app increased physical activity levels and helped users achieve their health goals (Direito et al, 2017).

METHODS

Firebase was used as the database in the work. Firebase Authentication is used for user authentication, and Firebase Database uses the programming language using Firestore, the Android Studio runtime tool, and Kotlin using GitHub, dividing tasks among team members. Firebase Authentication, a cloud-based authentication service, simplifies user authentication and authorization (Getting Started with Firebase Authentication: A Comprehensive Guide, 2023). Firestore is a scalable, reliable, and flexible database that supports real-time data synchronization (Firebase Realtime Database, 2023). Android Studio supports Java or Kotlin programming languages and provides rich documentation, error analysis, and autocomplete features (Comparison Java to Kotlin, 2023). Kotlin is a modern programming language that can run on an advanced Jet Brains Java Virtual Machine (JVM) (Trust in Collaborative Automation in High Stakes Software Engineering Work: A Case Study at NASA, 2023). IT offers the ease of hybrid migration to Kotlin and Java projects and the ability to convert existing Java code to Kotlin (Comparison Java to Kotlin, 2023).

GitHub is a web-based platform. It allows to manage, share, and collaborate on software development projects (Dabbish et al, 2012). IT promotes open source development projects and enables millions of developers to collaborate on projects. (Kalliamvakou et al, 2014). Developers can store their projects on GitHub, store their code, track the release date, and collaborate with other developers (Trust in Collaborative Automation in High Stakes Software Engineering Work: A Case Study at NASA, 2023). A fitness trainer provided provides its users' age range, body mass index, goals, use of equipment to exercise, and heart rate monitors monitoring schedules (Nike Training Club: NTC App, 2023).

Figure 1 provides a summary of the firming program for individuals with a body mass index above and below ideal weight.

1.-2. Weeks	3.-4. Weeks	5.-6. Weeks	7.-8. Weeks
10 dk Warm-Up Run 5 dk Stretch Vo2 Max =%55-65 3 Days in A Week 600 Kcal /3 Target Time= 30dk Cooling Down 5 dk	10 dk Warm-Up Run 5 dk Stretch Vo2 Max = %60-65 4 Days in A Week 1200 Kcal /4 Target Time= 40 dk Cooling Down 5 dk	10 dk Warm-Up Run 5 dk Stretch Vo2 Max = %70-75 4 Days in A Week 1000 Kcal /4 Target Time= 30 dk Cooling Down 5 dk	10 dk Warm-Up Run 5 dk Stretch Vo2 Max = %70-75 4 Days in A Week 1600 Kcal /4 Target Time= 50 dk Cooling Down 5 dk

Figure 1. Sample exercise protocol for 8 weeks

As you can see, the application starts the page with a greeting, it welcomes the user If the user has registered, he can log in from the application. If this page is not registered, they can create a new page. It is a program prepared according to the information entered by the users who will enter the application for the first time, on the registration page for this account, on the gender selection page. needs are created. For the first time, they were redirected directly to the homepage, if it is not theirs, then they can view the page of their current workout. By clicking on their profile page, they can view their personal information or create a new program.

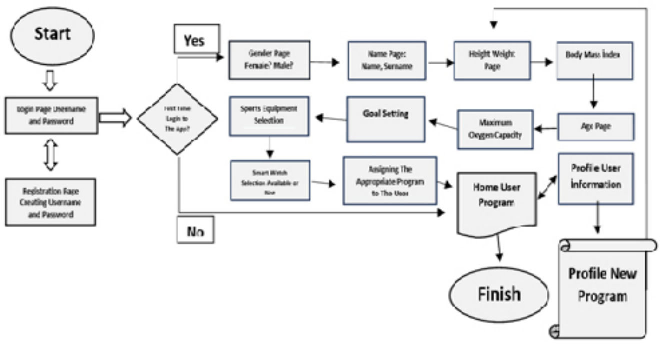

Figure 2. Module steps and personalized working principle of the application

When the application was first opened, the user greeted the screen with a splash After the splash screen, the user was presented with the login screen. On the login screen, the user can enter their email and password to log in to the application If the user is not registered, they can create a new account by entering their email and password of their choice and pressing the "Sign Up" button. If the user forgets their password, they can click on the "Forgot" option. To reset your passwords? From the password recovery screen, type "Password" and send it.

After logging in, the user is greeted with a motivation screen. This screen contains motivational quotes to inspire the user. First, the user is asked for gender information, addresses the user in the application, and first and last name information is collected. Next, the user enters their height (in cm) and weight (in kg). Once this information is entered, the reference calculates the user's body mass index (BMI) and displays it to the user according to the user's age, the maximum oxygen capacity is calculated, and the user is displayed.

On the next page, the user is asked to choose an exercise objective that is suitable for their fitness level to reduce stress and sweating by feeling good. Next, the user is asked to choose the equipment they will use to perform the exercise. The user's heart rate monitor indicates whether they have a wristwatch or not. Based on the information provided by the user reference, it displays the appropriate workouts on the home screen Clicking on an exercise with the user is directed to the exercise screen.

On the training screen, the user can start the timer by pressing the "Start" button to start the training session. In the profile section, users can view their information, as well as log out. Reference by pressing the "Log Out" button. By pressing the "New Program" button, the user can create a new program for himself.

RESULTS AND DISCUSSION

The purpose of this beginning was to provide. This app was to provide people with a useful fitness app, and this was the goal that many people achieved with a more efficient approach of personalized exercise trainers in the gym, but many users may not have time, may prefer to exercise at home.

Figure 3. Application steps and program interface

For this reason, fitness apps are widely used by many people: it is worth working to improve the app, which can be used both at home and in the gym. Many fitness apps don't provide personalized programs for users fitness apps that offer personalized programs usually only offer a few options. These applications often rely on several features to form the basis of the programs. These obstacles prevent users from making progress in following an effective program and make it more difficult to achieve their goals, plus fitness apps are often developed without the cooperation of knowledgeable fitness professionals, ultimately making them less effective apps.

In this study, a fitness trainer was collaborated to ensure that it is the most beneficial exercise program for users. When preparing programs for users in gyms, there are several user-specific factors that are taken into account. Similarly, in this study, these factors were used to offer the most suitable program to the users.

CONCLUSION

There are many areas in this regard, there are still incomplete studies, this work is not completed in the current state and there are many areas for further improvement.

With the sudden advancement of technology, there are planned developments for the future, and the proliferation of artificial intelligence can also be used to create various areas of intelligence. In order for the programs to train the users more efficiently, the most appropriate exercises for the users can be obtained by using "artificial intelligence", and the ability to communicate with a fitness trainer when there is a need for users to go to the gym without the necessary facilitator in achieving their fitness goals. The need to go to the gym can be made easier with these features.

ETHICAL DECLARATIONS

Peer Review Process: Externally refereed.

Conflict of Interest Statement: The authors do NOT have a conflict of interest statement.

Financial Disclosure: The authors declare that this study has received NO financial support.

Author Contributions: All authors declare that they agree with the design, implementation, and analysis of the article and approve the final version.

REFERENCES

1. Farrokhi, A., Farahbakhsh, R., Rezazadeh, J., & Minerva, R. (2021). Application of Internet of Things and artificial intelligence for smart fitness: A survey. *Computer Networks,* 189, 107859.
2. Azar, K.M., Lesser, L.I., Laing, B.Y., Stephens, J., Aurora, M.S., Burke, L.E., and Palaniappan, L.P. (2013). Weight management for mobile apps: theory-based content analysis. *American journal of preventive medicine*, 45(5), 583-589.
3. Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social in GitHub coding: transparency and open collaboration software repository. Computer-Aided ACM 2012 conference Proceedings in Cooperative Work (pp. 1277-1286).
4. Direito, A., Carraça, E., Rawstorn, J., Whittaker, R., Maddison, R., and Olds, T. (2017). Technologies influencing mHealth physical activity and sedentary behaviors: behavior modification techniques, systematic review and randomized meta-analysis of controlled trials *Behavioral Chronicles Medicine*, 51(2), 226-239-348.
5. Fanning, J., Mullen, S.P., McAuley, E. (2012). Increased physical activity with mobile devices: A meta-analysis. *Journal of Medical Internet Research*, 14(6), e161.
6. Fitbit: Activity and Health Trackers. (Accessed 19 May 2023). Access Address: https://www.fitbit.com/global/us/home
7. Google Fit: Health & Fitness Tracking App (Accessed May 19, 2023). Access Address: https://www.google.com/fit/
8. Comparison Java to Kotlin (Accessed 19 May 2023). https://kotlinlang.org/docs/reference/comparison-to-java.html
9. Firebase Realtime Database (Accessed 19 May 2023). https://firebase.google.com/products/realtime-database
10. Getting Started with Firebase Authentication: A Comprehensive Guide (Accessed 19 May 2023) https://dev.to/shahabmalikk/getting-started-with-firebase-authentication-a comprehensive-guide-1j81
11. Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D.M., and Damian, D. (2014). these promises and the dangers of GitHub mining. Proceedings of the 11th Mining Software Pools Working Conference (pp. 92-101).
12. Martin, S.S., Feldman, D.I., Blumenthal, R.S., Jones, S.R., Post, W.S., & McKibben, R.A. (2015). mActive: A clinical trial that randomized automation interfered with mHealth physical activity promotion. Journal American Heart Association, 4(11), e002239.
13. MyFitnessPal: Calorie counter and nutrition tracking app. (Accessed 19 May 2023). Access Address: https://www.myfitnesspal.com/en
14. Nike Training Club: NTC App. (Accessed May 19, 2023). Access Address: https://www.nike.com/en/ntc-app
15. Peterson, D., & Duncan, D.P. (2017). Development and availability of evidence-based testing to improve decision-making aid information for tablet devices and promote positive health behavior change. JMIR research protocols, 6(11), e225.
16. Seven: Fitness and Workout App. (Accessed May 19, 2023). Access Address: https://seven.app/
17. Smith, J.J., Morgan, PJ, Plotnikoff, R.C., Dally, KA, Salmon, J., Okely, A.D., and Lubans, D.M. (2016). Adolescent low-income boys' populations for smartphone obesity prevention trial: ATLAS RCT. Paediatrics, 137(3), e20153144.
18. Strava: Sports Tracking and Sharing App. (Accessed 19 May 2023). Access Address: https://www.strava.com/mobile
19. Trust in Collaborative Automation in High Stakes Software Engineering Work: A Case Study at NASA (Accessed 19 May 2023) https://dl.acm.org/doi/10.1145/3411764.3445650
20. Wang, J.B., Cadmus-Bertram, L.A., Natarajan, L., White, M.M., Madanat, H., Nichols, J.F., & Ayala, GX (2016). Wearable sensor/device (Fitbit One) and SMS text messaging seek to boost physical overweight activity and obese adults: randomized controlled. *Trial of Telemedicine and eHealth*, 22(5), 345-352.

Zülal Ataman

Zülal Ataman, received the BS degree in computer engineering from Gazi University in 2023. She is currently working as sofware developer. Her research interests include mobile programming and Software.

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0007

Cryptocurrency analysis using machine learning and deep learning approaches

Farah Muhammed Sakran

Department of Electronic and Computer Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Turkey

Cite this article: Muhammed Sakran F. Cryptocurrency analysis using machine learning and deep learning approaches. *J Comp Electr Electron Eng Sci*. 2023;1(2):29-33.

Corresponding Author: Farah Muhammed Sakran, farahalameri338@gmail.com

Received: 18/07/2023 ◆ **Accepted**: 25/09/2023 ◆ **Published**: 31/10/2023

ABSTRACT

Since cryptocurrencies are becoming more widely used and accepted in the financial system, precise price forecasting is essential for optimizing bitcoin investments. In this research study, we evaluated various machine learning models, including linear regression (LR), decision tree regression (DT), random forest regression (RF), support vector regression (SVR), gradient boosting regression (GB), adaboost regression, extreme gradient boosting regression (XGR), light gradientboosting regression (LGBM), k-nearest neighbors regression (KNN), ridge, andlasso. Additionally, we incorporated two deep learning (DL) models, namely artificial neural networks (ANN) and convolutional neural networks (CNN), to forecast daily bitcoin prices (BP). The initial data was obtained from Kaggle, a well-known platform for data science projects, and we applied the min-max scaler technique for consistent scaling during preprocessing. To assess the predictive capabilities of the models, we utilized regression metrics such as root mean square error (RMSE), mean absolute error (MAE), and correlation coefficient (R). Based on our findings, the CNN model demonstrated the highest effectiveness in predicting BPs among the DL models, with an RMSE of 0.0543, MAE of 0.0324, and an R value of 0.960. In the case of machine learning models, the RF model outperformed others, achieving an RMSE of 0.0246 and MAE of 0.0561.

Investors, scholars, and decision-makers may all gain from these findings' insightful revelations about BP forecasting. Developing these models further, investigating different preprocessing methods, and expanding the analysis to other cryptocurrencies might be the main goals of future research.

Keywords: Blockchain, cryptocurrency, machine learning

INTRODUCTION

Within the context of this study, previous research provides several insights that can be expanded upon (Sureshbhai, 2020). Due to the complexity of bitcoin systems, individuals often have misconceptions about their technological functioning, making them challenging to comprehend. Previous studies have extensively addressed the issue of key management as a significant problem for users (Sureshbhai, 2020). Custodial wallets, which eliminate the need for users to worry about key management, offer a means to interact with cryptocurrencies but require trust in the intermediary. However, limited information is available regarding the challenges faced by clients using custodial wallets (Mell, 2017). A study focusing on novice users' perceptions of Bitcoin's usability sheds light on the necessity for further research in this area (Begum, 2020). This study also examines the difficulties encountered by new bitcoin users and proposes potential solutions (Begum, 2020). The current article draws upon various findings from prior research. The complexity of cryptocurrencies often leads to discrepancies between users' mental models and the actual technological processes. Key management has primarily been addressed in previous research, highlighting its complexity for users (Wood, 2016). Custodial wallets, which alleviate key management concerns, are commonly used, yet there is limited research on the issues they pose for users. The first study (Financial Platform and News Website, 2008) investigating novice users' perceptions of Bitcoin's usability emphasizes the need for additional research. This work (Business Insider India, 2008) focuses on the challenges faced by new Bitcoin users and provides solutions to address them, addressing these unresolved issues.

Digital currency's roots trace back to Chaum's untraceable payment mechanism from the 1980s, along with blind signature technology (Mikhaylov, 2019). The 1990s saw various developments in digital money payments, such as fair offline e-cash and untraceable offline cash (Mikhaylov, 2019). Still, these methods depended on trusted parties to prevent double-spending attacks. Strategies like B-Money (Chuen, 2017) and Bit Gold (Chuen, 2017) later aimed to remove these intermediaries, but the adoption of decentralized consensus faced significant hurdles. A breakthrough came with Hal Finney's "Reusable Proofs of Work" in 2004 (Corbet, 2021), using trusted computing as a backend. More recent efforts involve forecasting cryptocurrency price movements, like

Bitcoin prices (Aggarwal, 2019). Rather than handling all cryptocurrency data collectively, the proposed approach analyzes each cryptocurrency's data separately. It employs ANN, CNN, and other ML models to leverage their unique advantages, aiming to create a comprehensive, accurate method for Bitcoin price prediction.

Literature Review

Numerous studies have applied diverse machine learning (ML) and deep learning (DL) algorithms to predict Bitcoin prices and identify influencing factors. Notably, LSTM models have shown strong performance (Jiang, 2019; Chen, 2020). For example, a 2019 study (Jiang, 2019) reported an RMSE of 47.91, while a 2021 study (Carbó and Gorjón, 2022) proposed an ensemble LSTM model yielding an RMSE of 37.24. However, LSTM models have shown some limitations, such as failing to identify a positive link between gold prices and Bitcoin prices (Jiang, 2019). Other research has utilized advanced hybrid models such as WT-CATCN (Saadah, 2020), achieving a 25% improvement in accuracy with an RMSE of 19.020, and a combined GRU and 1DCNN model (Wardak and Rasheed, 2022) that outperformed alternatives with an RMSE of 43.933, 3.511, and 0.00128. Twitter sentiment analysis was also found to correlate positively with Bitcoin price (Jiang, 2019). Another study (Chen, 2020) achieved 95.7% accuracy and a 0.05 RMSE using LSTM. Furthermore, research has shown that various ML algorithms including decision tree and regression models (Rathan, 2019), XGBoost and SDA (Borges and Neves, 2020), as well as ensemble learning (Mallqui and Fernandes, 2018), can be effective in predicting Bitcoin prices. These studies underline the potential of ML and DL models in cryptocurrency forecasting, with room for further exploration and optimization.

METHODS

In this research study, we aimed to optimize Bitcoin investments by developing a precise price forecasting model with existing artificial intelligence algorithms. We evaluated a range of ML models. To incorporate DL techniques, we also utilized ANN and CNN. The initial dataset was obtained from Kaggle, a renowned platform for data science projects, and we applied the min-max scaler technique for consistent scaling during the data preprocessing stage. Throughout the study, we assessed the predictive capabilities of these models using regression metrics. The model used a multi-stage methodology, which included preparing the data to standardize it for use with ML/DL algorithms. It made use of a variety of ML/DL approaches to build a forecasting framework that was precise and tuned for the dataset's specific problems and characteristics.

Datasets Description

The dataset utilized in this research study provides valuable information on the price movements and trading activity of various cryptocurrencies. This rich dataset allows us to create reliable prediction models by examining the potential relationships or correlations between different digital assets. By combining multiple cryptocurrencies, we gain a deeper understanding of the Bitcoin market, adding complexity and generating a more comprehensive picture. To ensure the dataset is suitable for analysis, appropriate preprocessing procedures are implemented. One crucial

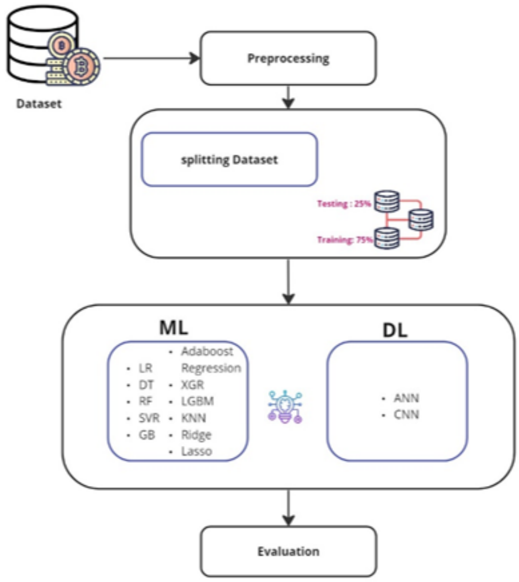

Figure 1. Proposed model

preprocessing technique employed is the min-max scaler. This technique scales the data to a specific range, typically between 0 and 1. By normalizing the data, it eliminates biases caused by variations in magnitudes between different features, enabling fair comparisons. The application of the min-max scaler ensures that the dataset is uniformized and ready for the subsequent modeling phase. The dataset used in this study presents a diverse and intriguing collection of historical price and volume data for four significant cryptocurrencies. By incorporating multiple digital assets and employing preprocessing techniques, we lay the foundation for developing precise and trustworthy prediction models for BP forecasting. This approach allows us to capture the complexities of the cryptocurrency market and enhance the ACC of our forecasting models.

	Date	Adj Close (BNB)	Volume (BNB)	Adj Close (BTC)	Volume (BTC)	Adj Close (USDT)	Volume (USDT)	Adj Close (ETH)	Volume (ETH)
0	09/11/2017	1.99077	19192200	7143.580078	3226249984	1.00818	358188000	320.884003	893249984
1	10/11/2017	1.79684	11155000	6618.140137	5208249856	1.00601	756446016	299.252991	885985984
2	11/11/2017	1.67047	8178150	6357.600098	4908680192	1.00899	746227968	314.681000	842300992
3	12/11/2017	1.51969	15298700	5950.069824	8957349888	1.01247	1466060032	307.907990	1613479936
4	13/11/2017	1.68662	12238800	6559.490234	6263249920	1.00935	767884032	316.716003	1041889984
...	...	...	...	...	...	...	...	...	...
1748	23/08/2022	299.03000	1034959891	21528.090000	31878280659	1.00000	52978066914	1662.770000	18322041914
1749	24/08/2022	296.45000	935405757	21395.020000	31962253368	1.00000	48708536003	1657.060000	16780932907
1750	25/08/2022	301.58000	973529233	21600.900000	31028679593	1.00010	45854093190	1696.460000	14818795695
1751	26/08/2022	279.60000	1210077085	20260.020000	42326789564	1.00000	62406193046	1507.780000	26713710143
1752	27/08/2022	277.30000	1023169984	20029.790000	32739299328	1.00000	48439021568	1470.760000	20430931968

Figure 2. BitCoin dataset

The next step in the data preparation process is to split the dataset into training and testing sets. This division is crucial to evaluate the effectiveness and generalization capabilities of the prediction models. In this study, a 25:75 split ratio is employed, where 25% of the dataset is allocated to testing and the remaining 75% is used for training the models. By splitting the data in this manner, a significant portion is dedicated to training the models, allowing them to learn patterns and relationships from a substantial amount of historical data. This enhances the models' ability to make accurate predictions. The testing set serves as an independent dataset to assess how well the models perform when applied to new data. It provides an unbiased evaluation of the models' prediction capabilities and their adaptability to novel situations. The 25:75 split ratio strikes a balance between having sufficient training data to adequately train the models and reserving a reasonable amount for testing. This ensures that the models can generalize effectively to new scenarios without being overfitted to the training data.

During the training phase, the models learn from the training set and adjust their internal parameters to minimize prediction errors. Their performance is then evaluated by comparing their predictions to the actual values using the testing set. This assessment process helps determine the precision, robustness, and applicability of the models in real-world scenarios. Moving on to preprocessing, it is the subsequent step in the data preparation procedure. Preprocessing involves transforming the data into a standardized format suitable for prediction models. One commonly used preprocessing technique is the min-max scaler, which normalizes the data. The min-max scaler rescales the values within a specific range, typically between 0 and 1. This normalization ensures that all features are on an equal scale, preventing any single feature from dominating the model's learning process. By applying the min-max scaler, the values of the dataset are proportionally adjusted to fit within the defined range. This process ensures consistent scaling across all features while preserving the relative relationships between data points. Normalized data enables the models to effectively learn from the data and generate accurate predictions. Additionally, the min-max scaler can handle outliers and extreme values in the dataset. By compressing the data into a narrow range, the influence of outliers is reduced, preventing them from negatively impacting the model's performance. The preprocessing phase involving the min-max scaler is crucial for preparing the data before feeding it into the prediction models. It allows for efficient normalization of the dataset, enabling the models to learn from patterns and correlations in the data, leading to more accurate and reliable predictions.

Predictive Methods

The next stage in our methodology involves applying various ML and DL models to forecast daily Bitcoin prices. We evaluate a range of ML models, including LR, DT, RF, SVR, GB, Adaboost Regression, XGR, LGBM, KNN, Ridge, and Lasso. Additionally, we incorporate two DL models, namely ANN and CNN. Linear Regression is a simple regression method used to model the relationship between a dependent variable and one or more independent variables through a linear equation. Decision tree regression represents data using a tree structure and makes predictions of the dependent variable's value by traversing the tree based on input features. Random forest regression improves prediction accuracy by using multiple decision trees and reducing overfitting. Support vector regression uses support vector machines to model linear or nonlinear relationships between data points. Gradient boosting regression enhances prediction accuracy by combining weak learners (usually decision trees) to make more accurate predictions. Adaboost regression is an ensemble method that boosts the performance of weak regression models by combining them. XGBoost is an optimized version of gradient boosting that offers faster and more efficient learning. LightGBM is a gradient boosting method optimized for speed and distributed learning. K-Nearest neighbors regression predicts values based on the average of the nearest data points in the feature space. Ridge regression is a linear regression method that adds L2 regularization to resist overfitting. Lasso Regression adds L1 regularization to linear regression, reducing unnecessary features and increasing model simplicity. Artificial neural networks are deep learning methods that model complex relationships inspired by biological neural systems. Convolutional neural networks are a type of artificial neural network primarily used for image processing and pattern recognition, employing convolution operations on input data. These models are selected based on their specific capabilities and suitability for addressing the challenges and characteristics of the dataset. By leveraging a diverse range of ML and DL techniques, we aim to optimize the ACC and predictive capabilities of our forecasting model for Bitcoin prices. This set of metrics evaluates the performance of encryption algorithms. These metrics include MSE, which measures the square root of the average "error" between the original and encrypted pixel values in images. The MSE equation (1) calculates the discrepancy for each pixel and averages them across the entire image. Another metric is RMSE, which estimates the size of the error between the expected and actual values. The RMSE equation (2) takes the square root of the MSE to provide a more interpretable measure of the error.

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\ [f(i,j)-f`(i,j)]^2}{MN} \quad (1)$$

$$RMSE(\hat{\theta}) = \sqrt{MSE\ (\hat{\theta})} \quad (2)$$

RESULTS

Utilizing the approach, we addressed the study goals mentioned in the first chapter in this work. After providing a general overview of the Cryptocurrency Prices dataset, the needs of the study were thoroughly explained. The experiment's findings were then carefully examined and evaluated. After comparing several ML regressors, it appears that the RF provides the best performance with an MAE of 0.024609, RMSE of 0.056137, and R2 of 0.958287. The Random Forest Regressor therefore seems to be the most accurate model for this task, offering the lowest loss. In addition, it's worth mentioning that DT and XGB also achieved commendable results with relatively low MAE and RMSE, and high R2 scores. This indicates a strong fit to the data and an ability to predict Bitcoin prices reliably. On the other end of the spectrum, the Lasso regression model did not perform well in comparison to other models, with a significantly higher MAE and RMSE, and a negative R2 score, which indicates a poor fit to the data. These results demonstrate the power of ensemble learning methods like RF and XGBoost in predicting volatile cryptocurrency prices such as Bitcoin, and underscores the importance of choosing the right model and tuning it correctly for predictive tasks in financial contexts.

Table 1. Loss values of different models for predicting cryptocurrency prices

Regressor	MAE	RMSE	R2
Linear Regression	0.065311	0.107985	0.845651
Decision Tree Regressor	0.026540	0.069409	0.936232
Random Forest Regressor	**0.024609**	**0.056137**	**0.958287**
SVR	0.062093	0.083781	0.907090
Gradient Boosting Regressor	0.035017	0.066422	0.941602
AdaBoost Regressor	0.055780	0.080166	0.914934
XGB	0.026591	0.058331	0.954962
LGBM	0.029611	0.060188	0.952049
KNN	0.028625	0.059169	0.953658
Ridge	0.065359	0.107970	0.845694
Lasso	0.233261	0.274884	-0.000177

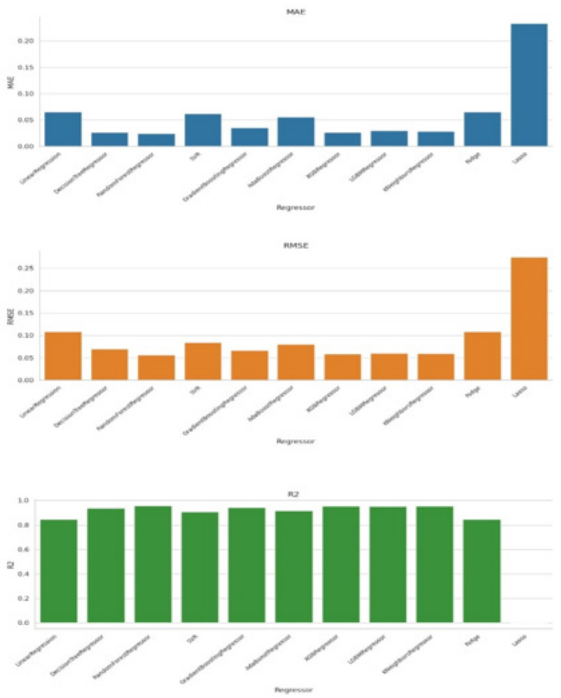

Figure 3. ML comparison

The results provided are evaluation metrics of two different models, ANN and CNN which were presumably used for predicting Bitcoin prices. Starting with the ANN model, it obtained an RMSE of 0.09349, an MAE of 0.0612, and a correlation coefficient of 0.8842. These figures suggest a decent model performance, but there seems to be room for improvement. In comparison, the CNN model outperforms the ANN model on all three metrics. It achieved an RMSE of 0.05437, which is considerably lower than the ANN model's RMSE, suggesting a more accurate prediction. Similarly, the MAE for the CNN model is 0.03247, also significantly lower than the ANN model's MAE, suggesting less absolute error in the CNN model's predictions. The correlation coefficient of the CNN model is 0.960, closer to 1 than the ANN model's correlation coefficient. This suggests that the CNN model's predictions are more strongly correlated with the actual values.

Table 2. Comparative results ML/ DL

Model (Type)	MAE	RMSE	R
Random Forest (ML)	**0.024609**	**0.056137**	**0.958287**
ANN (DL)	0.061200	0.093494	0.884297
CNN (DL)	0.032474	0.054376	0.960863

The graph depicted in Figure 4 compares the true values versus the predicted values of Bitcoin prices as predicted by the DL models, namely ANN and CNN. The figure shows two distinct curves, the red curve representing the predicted values from the models and the blue curve indicating the actual or true Bitcoin prices. From the visualization, it is evident that the red curve, which signifies the predicted values, consistently lies above the blue curve, indicating the actual values. This pattern reveals that both the ANN and CNN models, on average, predicted a higher Bitcoin price than the actual observed price. However, the fact that the predicted curve closely follows the trend of the actual curve indicates that these models have done a commendable job in capturing the general pattern and movements of Bitcoin prices. The models were able to predict the direction of price changes accurately, even if the predicted prices were slightly higher than the actual prices.

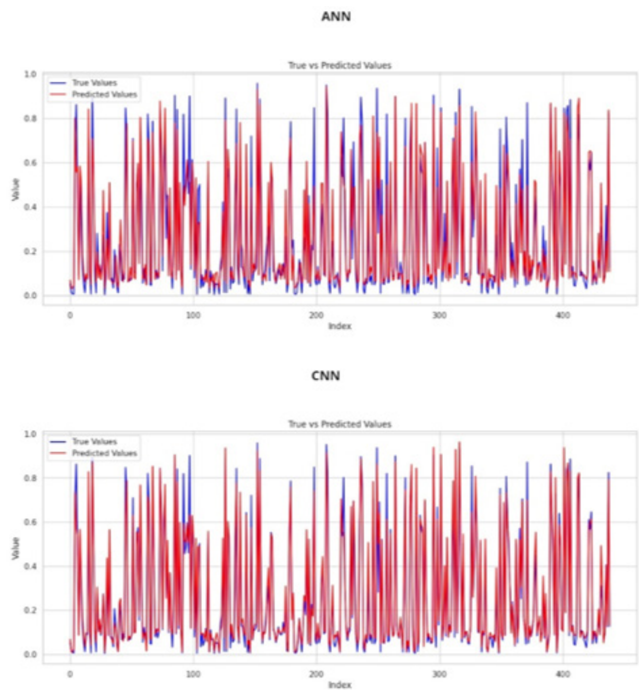

Figure 4. DL true vs prediction

DISCUSSION

In this study, traditional ML and DL models were used to predict Bitcoin prices. The Random Forest (RF) emerged as the best ML model, exhibiting robust performance and accuracy. However, CNN, a DL model, outperformed all with its ability to capture complex time-series patterns. Despite these promising results, there were limitations. The models were restrained by the historic data, which might not reflect future events or trends. Also, important external factors affecting Bitcoin prices weren't considered. The performance of DL models could be enhanced with extensive hyperparameter tuning, not fully explored in this study due to computational limits. Future work should focus on model optimization, exploring diverse neural network architectures, and employing larger datasets for better prediction accuracy.

CONCLUSION

In summary, this study highlighted the effectiveness of various machine learning and DL models for Bitcoin price prediction. Among traditional machine learning models, random forest regressor stood out, while in the realm of DL, the CNN model outperformed the ANN model. The results underscore the potential of advanced analytics and predictive modeling techniques in navigating the complex and volatile landscape of cryptocurrency markets. However, a degree of caution is needed when interpreting the results due to the limitations of the study. Future research should focus on addressing these limitations, incorporating more external factors that affect cryptocurrency prices, and refining the models to better capture the nuances of cryptocurrency price fluctuations. The pursuit of more accurate and reliable predictive models for Bitcoin prices remains a challenging yet important task for researchers, investors, and policy makers alike.

ETHICAL DECLARATIONS

Referee Evaluation Process: Externally peer-reviewed.

Conflict of Interest Statement: The authors have no conflicts of interest to declare.

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

Author Contributions: All of the authors declare that they have all participated in the design, execution, and analysis of the paper, and that they have approved the final version.

REFERENCES

1. Sureshbhai, P. N., Bhattacharya, P., and Tanwar, S. (2020 June). 'KaRuNa: A blockchainbased sentiment analysis framework for fraud cryptocurrency schemes. In 2020 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.
2. Mell, P., Kelsey, J., & Shook, J. (2017). Cryptocurrency smart contracts for distributed consensus of public randomness. In Stabilization, Safety, and Security of Distributed Systems: 19th International Symposium, SSS 2017, Boston, MA, USA, November 5-8, 2017, Proceedings 19 (pp. 410-425). Springer International Publishing.
3. Begum, A., Tareq, A., Sultana, M., Sohel, M., Rahman, T., & Sarwar, A. (2020). Blockchain attacks analysis and a model to solve double spending attack. *International Journal of Machine Learning and Computing*, 10(2),352-357.
4. G. Wood. Ethereum: A secure Decentralised Generalised Transaction Ledger. Accessed: 2016. [Online]. Available: https://scholar.google. Com/scholar?hl=en&as_sdt=0%2C5&q=ETHEREUM%
5. Financial Platform and News Website. Accessed: 2008. [Online]. Available: https://www.investing.com/
6. Business Insider India. Accessed: 2008. [Online]. Available: https:// www. businessinsider.in/
7. A. Mikhaylov. Asset Allocation in Equity, Fixed-Income and Cryptocurrency on the Base of Individual Risk Sentiment. Accessed: 2019. [Online]. Available: https://pdfs.semanticscholar.org/df78/ f60a84c2a17f47bce27578746c6313251%a88.pdf
8. D. L. K. Chuen, L. Guo, and Y. Wang, "Cryptocurrency: A new investment opportunity?" *J. Alternative Investments,* vol. 20, no. 3, pp. 16-40, 2017.
9. S. Corbet, Y. Hou, Y. Hu, C. Larkin, B. Lucey, and L. Oxley, "Cryptocurrency liquidity and volatility interrelationships during the COVID-19 pandemic," Finance Res. Lett., early access, May 2021, Art. no. 102137.
10. A. Aggarwal, I. Gupta, N. Garg, and A. Goel, "Deep learning approach to determine the impact of socio-economic factors on BP prediction," in 2019 Twelfth International Conference on Contemporary Computing (IC3), pp. 1-5, 2019.
11. X. Jiang, "Bitcoin price prediction based on deep learning methods," *J. Math. Finance.*, Vol. 10, pp. 132-139, 2019.
12. Saadah, S.; Whafa, A.A. Monitoring Financial Stability Based on Prediction of Cryptocurrencies Price Using Intelligent Algorithm. In Proceedings of the 2020 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 5-6 August 2020; pp. 1-10.
13. Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., Alazab, M. Stochastic neural networks for cryptocurrency price prediction. IEEE Access 2020, 8, 82804–82818.
14. A. Politis, K. Doka, and N. Koziris, "Ether price prediction using advanced deep learning models,". In Proceedings of the 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3-6 May 2021; pp. 1-3.
15. S. MyungJae, D. Mohaisen, and J. Kim, "Bitcoin Price Forecasting via Ensemble-based LSTM Deep Learning Networks". Paper presented at 2021 International Conference on Information Networking (ICOIN), Jeju Island, Republic of Korea, January 13-16; pp. 603-608, 2021.
16. J. Carbó and S. Gorjón, "Application of Machine Learning Models and Interpretability Techniques to Identify the Determinants of the Price of Bitcoin," Banco de Espana Working Paper No. 2215, 2022.
17. C. Kang, C. Lee, K. Lim, "Cryptocurrency Price Prediction with Convolutional Neural Network and Stacked Gated Recurrent Unit," Data, Vol. 7, No. 149, 2022.
18. A. Wardak and J. Rasheed. (2022). "Bitcoin Cryptocurrency Price Prediction Using Long Short-Term Memory Recurrent Neural Network," *European Journal of Science and Technology,* No: 38, pp. 47-53.
19. Chen, Z., Li, C., and Sun. W. (2022). "Journal of Computational and Applied Bitcoin price prediction using machine learning: An approach to sample dimension engineering," *J. Comput. Appl. Math.*, vol. 365, p. 112395, doi: 10.1016/j.cam.2019.112395.
20. Rathan, K., Sai, S.V., and Manikanta, T.S. (2019). "CryptoCurrency price prediction using Decision Tree and Regression techniques," 3rd *Int. Conf. Trends Electron. Informatics*, no. Icoei, pp. 190-194.
21. Borges, T. A., and Neves, R.F. (2020). "Ensemble of machine learning algorithms for cryptocurrency investment with different data resampling methods," *Appl. Soft Comput. J.,* vol. 90, p. 106187, doi:10.1016/j.asoc.2020.106187.
22. Mallqui, D.C.A., and Fernandes, R.A.S. (2018). "Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques," *Appl. Soft Comput. J.* doi: 10.1016/j.asoc.2018.11.038.

Farah Muhammed Sakran

Farah Sakran is doing her master's degree in Electronic-Computer Engineering Department of Çankırı Karatekin University, Institute of Science.

Journal of
Computer & Electrical and Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0008

Controlling the operation of the dc motor by using pid with metaheuristic technology

Khalaf Abdullah Khalaf KHALAF Mustafa Teke

Department of Electrical and Electronics Engineering, Faculty of Engineering, Çankırı Karatekin University, Çankırı, Turkey

Cite this article: Khalaf KAK, Teke M. Controlling the operation of the dc motor by using pid with metaheuristic technology. *J Comp Electr Electron Eng Sci.* 2023;1(2):34-40.

Corresponding Author: Khalaf Abdullah Khalaf KHALAF, khalafabd1994@gmail.com

Received: 29/07/2023 ◆ **Accepted**: 04/10/2023 ◆ **Published**: 31/10/2023

ABSTRACT

This paper presents an investigation into precise trajectory tracking and synchronization of two-axes direct current (DC) motor control, with an emphasis on a cascade proportional-plus-integral (P-PI) controller to regulate the speed and position of a single-axis permanent magnet DC (PMDC) motor. Various methods were explored for the controller's design process, including classical methods (CM) and three optimization strategies: genetic algorithm (GA), dandelion optimization algorithm (DOA), and butterfly optimization algorithm (BOA), with the latter found to be the most effective. Simulation was a crucial component in assessing the efficiency of these methods. A comparative analysis of four tuning strategies (CM, GA, BOA, and DOA) was conducted to ascertain optimal settings for the P-PI cascade controller. The DOA outperformed the others, providing accurate tracking with no deviation from the reference location or speed overshoot. Moreover, DOA ensured safety by limiting voltage and current to prevent potential damage to the motor. The findings thus suggest that the proposed P-PI controller with DOA can serve as a reliable solution for speed and position control in single-axis PMDC motors.

Keywords: PMDC, fine-tuning the PI controller, BOA, DOA, PID controllers.

INTRODUCTION

In its most basic form, an electric motor is a mechanism that uses electricity to create motion (Bigelow, 2020; Franchi and Claiton, 2019). This generated motion can be exploited in various ways for a variety of applications. This study was done by using one of the types of DC motors, which is PMDC. This motor is used in many applications and is similar in structure and operation to the shunt-connected DC motor (Momoh and James, 2018; Krause et al., 2002). On the other hand, to control the most important functions of PMDC, namely velocity and position, a cascade control system consisting of three controllers were used to control the current, speed, and position (Son and Young 2014; Cankurtaran et al., 2019). This controller provides the required response to the system, minimizes error, and returns to a steady state when a specified load is applied. The main reason for using such controllers and other controllers is to use them to control many motor functions such as speed, current, and position that humans cannot control manually (Wang and Liuping, 2020; Raja et al., 2017). In reference (Cuong, 2013), the model was developed by connecting two (dual) motors instead of one DC motor to take advantage of them in many applications, such as sharing a specific load. This model can also be used in many applications, some of which require precise speed control, such as paper and textile mills, and others require high and accurate position control, such as metal cutting machines, CNC machines, and robotics (Tang, 2001; Romero et al., 2006).This work has been implemented in a real-time Matlab environment, this study demonstrates a remarkable degree of precision in guiding the motor to its target location at a constant speed, irrespective of the presence or absence of a load. Simulation results obtained from Matlab verified that the PMDC motor could be controlled with exceptional accuracy, fulfilling the desired position, while maintaining an accurate tracking trajectory. A comprehensive review of related literature confirms that a variety of controller types, including PID, fuzzy logic, and 2-SMC controllers, can manage the PMDC motor effectively. In this study, a cascade control system, characterized by its flexibility and robustness, is adopted for this study. The cascade controller employed in this investigation comprises three individual controllers: the position, speed, and current controllers.

Literature Review

Tang (2001) proposed a PID controller architecture for managing the speed and position of a DC motor, which employs a low-cost digital signal processor, the TMS320C31 suite. Real-time values of speed and position controller parameters were set in a direct online manner during the operation of the DC motor to ensure its continuous operation without shutdown. Romero and Concha (2006) presented a system for controlling the velocity and position of a moving robot using a brushless DC motor, with a motor connected to each wheel. A three-phase bridge was designed using N- Mosfet, along with an electronic circuit to drive the motors. This paper references several studies on motor control methodologies. Tang (2001) implemented a PID controller architecture utilizing a low-cost digital signal processor, the TMS320C31 suite, to manage the DC motor's speed and position. Outputs from Hall sensor and optical encoder were also harnessed in conjunction with pulse width modulation to form closed-loop control for velocity and position. Pisano et al. (2008) developed a cascade control system for managing the speed and location of the permanent magnet DC motor. Despite uncertainties in motor and load parameters, the application of a second-order sliding mode controller (2-SMC) ensured excellent performance and precise tracking. Talavera et al. (2014) presented a bidirectional DC motor speed and position control system constructed using ATMEGA32 microcontrollers and LabVIEW. An optical encoder attached to the motor shaft collected information on rotational speed and angular position, and the motor was powered using the PWM technique. A graphical user interface in LabVIEW software was used to input the reference signal, be it a velocity signal or a position angle. Taha et al. (2015) reported a cascade PI(D) control system for the speed and position of the permanent magnet DC motor. PID gains were adjusted using three distinct methods: a traditional manner and two ideal ones. These strategies were then compared to identify the one providing the most reliable velocity and position regulation. Taut and Marius (2018) introduced a closed-loop DC motor speed and position control system comprising a DC actuator motor, a position sensor, and the mechanical load to be moved by the motor. This system's control methodology was realized using Simulink and embedded code. Syh-Shiuh and Pau (2000) proposed an integrated movement system to enhance the performance of a CNC machine in terms of path tracking, which included feedforward and feedback with multi-axis cross-coupled control (CCC). This controller was efficiently designed through a comprehensive system analysis and a novel formulation of the contouring error transfer function (CETF) for multi-axis systems. Ishizaki et al. (2014) implemented a speed and position control system for two PMDC motors used in a gantry-style machine tool. Each PMDC motor in this setup was managed by two separate controllers, a position (P) controller and a speed (PI) controller. A cross-coupling approach was employed to interconnect the motors, and a methodology was created to determine the values of the system parameters for precise synchronization of motor movement along the axes. Panlong and Wang (2019) proposed an autonomous guided vehicle powered by two separate DC motors. Two control methods were used: parallel PID control and coupled PID control of deviation. These methods were evaluated using MATLAB simulations and KINECT sensor experiments. The coupled PID controller of deviation performed marginally better than the parallel PID controller in a transport scenario. Lastly, (Ali et al., 2019) demonstrated that two DC motors could be controlled in parallel using a Hall sensor. Precise control of the motors' speeds and positions was necessary to achieve the desired parallel positioning. Position and speed were estimated using the back EMF compensator and the current model speed observer, respectively. Position discrepancies between the motors could be adjusted using the instantaneous position compensator, and the speed could be approximated using the average speed between the speed observer and the actual speed.

METHODS

The PMDC (Permanent Magnet Direct Current) motor is a significant component in many applications due to its simple construction and reliable performance. Understanding its construction and the principles behind its operation is essential for those involved in its application and further development. A typical PMDC motor consists of two main parts: the stator and the rotor (also known as the armature). The stator forms the stationary part of the motor, providing the magnetic field. It's a steel cylinder with permanent magnets attached to its inner surface. These magnets are oriented so that their poles are facing inward toward the rotor. The magnetic poles of the stator are fixed in such a way that they face the armature, providing the necessary magnetic field for operation. The rotor or armature, the moving part of the motor, is composed of a winding of wire placed around an iron core. This armature winding is connected to a commutator, which is a device that switches the direction of current flow through the windings as the rotor turns. This switching action ensures that the rotor continues to turn in the same direction. The commutator brushes, usually made of carbon, maintain the electrical contact with the rotating commutator.

PMDC Mathematical Model

The circuit is powered by a voltage source linked to the armature coil. The electrical elements of this circuit encompass the inductance (La) and armature resistance (Ra), wired in series. Additionally, a counter-electromotive force or back emf (Ea), which emerges once the coil spins and intersects the flux lines produced by the permanent magnet, opposes the voltage source's direction. The mechanical aspects of this circuit are made up of the moment of inertia (Jm) and the viscous friction coefficient (Bm). Other integral parameters include the torque constant (Kt) and the back emf constant (Kv). The corresponding circuit for a DC motor with a permanent magnet is shown in Figure 1.

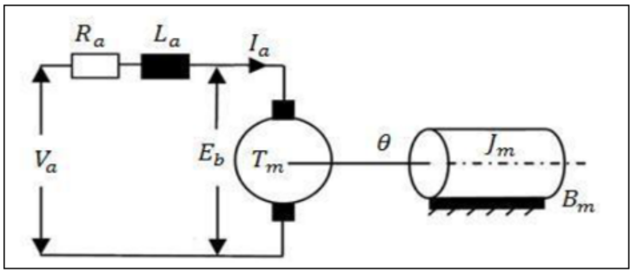

Figure 1. Equivalent circuit of PMDC (Adel et al., 2018)

The Permanent magnet DC motor block diagram is shown in Figure 2.

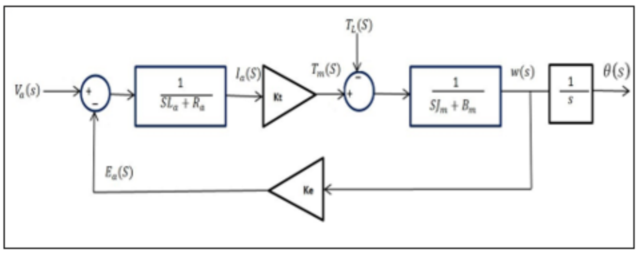

Figure 2. Block diagram of PMDC (Adel et al., 2018)

The mathematical representation for the function shown in Figure 2 can be encapsulated in the Equations (1,2,3 and 4) provided below, as per references (Adel et al. 2018).

$$v_a(t) = e_a(t) + R_a i_a(t) + L_a \frac{d}{dt} ia(t) \quad (1)$$

$$e_a(t) = k_e \omega_m(t) \quad (2)$$

$$T_m(t) - T_L = J_m \frac{a}{dt}\omega_m(t) + B_m \omega_m(t) \quad (3)$$

$$T_m(t) = K_t i_a(t) \quad (4)$$

By using Laplace transformation for the Equations (1-4), Equations (5-8) will be:

$$V_a(s) = E_a(s) + R_a I_a(s) + SL_a I_a(s) \quad (5)$$

$$E_a(s) = k_e \ \omega_m(s) \quad (6)$$

$$T_m(s) - T_L = SJ_m \omega_m(s) + B_m \omega_m(s) \quad (7)$$

$$T_m(s) = K_t I_a(s) \quad (8)$$

Equations (9) and (10) define the global transfer functions for speed and position control of the PMDC motor.

$$\frac{\omega_m(s)}{V_a(s)} = \frac{k_e}{JL_a S^2 + (JR_a + BL_a)S + BR_a + k_e{}^2} \quad (9)$$

$$\frac{\theta(s)}{V_a(s)} = \frac{k_e}{JL_a S^3 + (JR_a + BL_a)S^2 + (BR_a + k_e^2)S} \quad (10)$$

Optimization Tuning Methods

This paper employs the classic method (CM), in conjunction with three optimal approaches for determining and extracting PID cascade parameters. These include the genetic algorithm (GA), the dandelion optimizer algorithm (DOA), and the butterfly optimization algorithm (BOA). A comparative analysis will be undertaken among these methods to discern the most beneficial outcomes. Further details about these techniques will be elucidated in the ensuing section.

Classical Method (CM)

The classical method computes the P and PI parameters of the cascade control system by employing block diagram reduction, simplifying each loop independently, and calculating its respective parameters. However, this method isn't feasible for real-time applications due to certain assumptions it makes, such as disregarding the load and negating the back EMF effect (Biswas and Anupam 2013). Initial calculations focus on the parameters of the inner loop or the current loop. Figure 3 illustrates the initial control loop. In a PMDC motor, the torque and current exhibit proportionality, thereby allowing the current to be treated as a control variable.

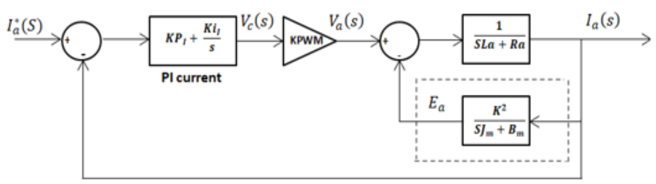

Figure 3. Current loop

For the purpose of clarity, this method simplifies things by making assumptions that lead to important circuit details being ignored. In particular, because of the large magnitude of the motor's moment of inertia (Jm), the effects of the torque load and the counter-electromotive force (Ea) are disregarded. Therefore, these elements have been removed from the current control loop and a simpler form is shown in Figure 4.

The transfer function of The PI current controller as in Equation (11)

$$\frac{V_c(s)}{E(s)} = Kp + \frac{K_i}{s} = \frac{K_i}{s}\left(1 + \frac{s}{Ki/_{Kp}}\right) \quad (11)$$

Figure 3 shows a simplification of the current loop, and its open-loop transfer function designated G_I o_l) is provided by Equation (12). This function characterizes the system's operation in the absence of any kind of feedback.

$$G_{I\,ol} = \frac{k_{iI}}{s}\left(1 + \frac{s}{k_{iI}/_{k_{pI}}}\right) * KPWM * \left(\frac{1/_{R_a}}{1+s\tau_e}\right) \quad (12)$$

Where k_{pI} is the proportional gain of current controller k_{iI} is the integral gain of current controller τ_e is the electrical time constant $\frac{L_a}{R_a}$

By cancellation of the pole in the motor transfer function renders Equation (13) will be:

$$G_{Iol} = \frac{k_{iI} * KPWM}{sR_a} \quad (13)$$

he range or crossover frequency for the current loop, denoted by ci in Equation (14), may be calculated from the open-loop transfer function. At this frequency (ωci), the system's power reduces to half its peak value, denoting the boundary between the passband and the stopband in the frequency response.

$$\omega ci = \frac{KPWM * k_{iI}}{R_a}, \text{or} \quad k_{iI} = \frac{\omega ci * R_a}{KPWM} \quad (14)$$

The DC-DC converter's switching frequency, fci, may be used in the equation ωci=2πfci to get the bandwidth frequency, fci. In most cases, ci will have a value that is 10 times less than fci. We assume a switching frequency of 2 kHz for the purposes of this work. We assume the closed current loop operates optimally for the purpose of determining the speed loop parameters. Figure 4 is a graphic representation of this idea, which represents harmony.

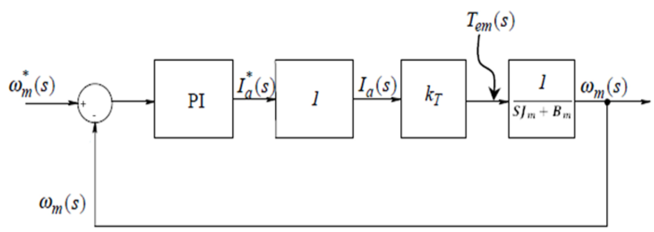

Figure 4. Block diagram of the speed loop

The pole associated with the motor's mechanical component will be cancelled, as described in Equation (15), in a manner analogous to the inner current control loop.

K_{is} represents the integral gain of the controller, whereas K_{ps} specifies the proportional gain. These two variables play crucial roles in how the control system reacts to perturbations or mistakes.

$$\frac{k_{iS}}{k_{pS}} = \frac{1}{\tau_m} \tag{15}$$

The current control loop's bandwidth, written by ci, is often chosen to be an order of magnitude larger than the speed control loop's bandwidth, denoted by cs. To illustrate this connection, consider Equation (16). This setup guarantees a consistent and proportional response from the system's dual control loops.

$$\omega cs = \frac{k_{is} \cdot k_t}{B_m} \tag{16}$$

It is assumed that the velocity loop is perfect, and is therefore represented as unity, in order to calculate the kpP, parameter of the position loop. Equation (17) represents the open-loop transfer function for position control.

$$G_{Pol}(s) = \frac{k_{pP}}{s} \tag{17}$$

Using Equation (18), we may determine the value of the position parameter, kpP, by choosing a bandwidth frequency, denoted by ps, that is ten times lower than the bandwidth frequency, ωps, of the speed control loop. This choice guarantees a consistent hierarchy of control loop frequencies, which boosts the robustness and efficiency of the overall system.

$$\omega pS = k_{pP} \tag{18}$$

Genetic Algorithm Optimization (GA)

The current generation produces the next by way of the evolutionary process. It is anticipated that this next generation would provide a superior outcome. This process is iterated over many generations to get the best possible outcome for the system.

Table 1 represents for GA parameters.

Table 1. GA parameters

GA Parameters	Value
Generation	100
Population size	20
No. dimension	5
Crossover	0.6
LU, UB	0.300

Butterfly Optimization Algorithm (BOA)

The BOA is a metaheuristic algorithm that takes its inspiration from the behavior of butterflies, particularly their use of natural cues for food searching and mating to produce offspring. Chemical receptor nerve cells on the butterfly's body disperse the scent of nectar, which allows other butterflies to detect its presence. The intensity of this fragrance is directly proportional to the butterfly's fitness. The foundational principles of sensing and odor processing can be grouped into three main categories: sensory modality (c), stimulus level (I), and power exponent (a). The natural behaviors exhibited by butterflies inform the formulation of the fragrance function and the variations in scent intensity. The fragrance function can be represented mathematically as a function of physical density, as shown in Equation (19). This equation encapsulates the underlying principles of the algorithm, leveraging the natural behaviors of butterflies to guide the search for optimal solutions..

$$f = cI^a \tag{19}$$

The value of f in the aforementioned equation indicates the intensity with which other butterflies detect the aroma. Perceived scent intensity depends on a number of factors, including the previously discussed metrics *c*, *I*, and *a*.

The butterfly moves in a global search direction when it detects another butterfly's odor, represented by Equation (20). In contrast, when it cannot detect another butterfly's odor, it moves in a random direction, represented by Equation (21).

$$X_i^{t+1} = X_i^t + \left(r^2 * g^* - X_i^t \right) * f_i \tag{20}$$

$$X_i^{t+1} = X_i^t + \left(r^2 * X_j^t - X_k^t \right) * f_i \tag{21}$$

Where X_i^{t+1} represents to new solution vector, X_i^t represents to solution vector, r represents to rondom number (n) between [0, 1], g represents to the current best solution, t is the iteration number, i is the butterfly and X_j^t & X_k^t are *jth* and *kth* butterflies from the solution space and fi the perceived magnitude of the frequence.

Butterflies may either look close to home or far and wide for food and companionship. A butterfly may choose between the two different kinds of search methods. It may choose to either stick with the butterfly that is doing the best overall search, or it can wander aimlessly. The'switch probability, represented by the symbol, is the method by which the butterfly may toggle between local and global searches. The BOA parameters are shown in Table 2.

Table 2. BOA parameters

Parameters	Value
Max iteration	5
No. search agents	20
No. dimension	5
Switch probability ρ	0.8
Power exponent a Sensory modality LB, UB	0.1 0.01 0.300

The butterfly optimization technique is able to be depicted in the form of an approximate flowchart, which may be found in Figure 8.

Dandelion Optimizer Algorithm (DOA)

This segment introduces the mathematical formulas utilized in the Dandelion Optimization Algorithm (DOA). It commences with an explanation of two distinct types of conditions and their corresponding mathematical expressions. After this, it examines mathematical simulations for the last phases of the flight, including descent and landing. Equation (22) shows the population array.

$$population = \begin{bmatrix} x_1^1 & \cdots & x_1^{Dim} \\ \vdots & \ddots & \vdots \\ x_{pop}^1 & \cdots & x_{pop}^{Dim} \end{bmatrix} \tag{22}$$

Here '*pop*' stands for population size, whereas 'Dim' refers to the dimension of the variable in question. The problem's upper limit (UB) and lower bound (LB) are used as constraints to construct a set of randomly generated candidate solutions. According to Equation (23), the ith person's representation (Xi) is generated at random.

$$Xi = rand \times (UB - LB) + LB \tag{23}$$

In the Equation, "i" is an integer that ranges from 1 to "pop", and "rand" represents a randomly generated number between 0 and 1. The lower bound (LB) and upper bound (UB) values are expressed as follows in Equation (24).

$$LB = [lb_1, \ldots\ldots\ldots\ldots, lb_{\text{Dim}} \tag{24}$$

$$UB = [ub_1, \ldots\ldots\ldots\ldots, ub_{\text{Dim}} \tag{25}$$

The initial elite is determined by DOA's ideal fitness value during the initiation phase. This particular person is said to be the best environment for the dandelion seed to grow. Equations (26) and (27) reveal the mathematical equation for the initial elite *Xelite* if the smallest value is used:

$$f_{best} = min(X_i) \tag{26}$$

$$X_{elite} = X(find(f_{best} == (X_i))) \tag{27}$$

The expression "*find*()" refers to two indices that have the same value. Table 3 presents the parameters of the Dandelion Optimization Algorithm (DOA) used.

Table 3. DOA parameters

Parameters	Value
Max iteration	10
No. search agents	30
No. dimension	5
Switch probability ρ	0.7
Power exponent a Sensory modality LB, UB	0.1 0.01 0.500

Objective Functions

In optimal control theory and the construction of estimators employing linear state variable feedback, the notion of an objective function is fundamental. The goal of the system in such situations is to maximize some measure of performance under strict limits. Functions that rely on error and time are often used as performance indices. The greatest potential performance of the system is ensured by this optimization procedure. This is crucial for improving control systems' efficacy and efficiency.

Integral of absolute error (IAE), integral of squared error (ISE), integral time absolute error (ITAE), and integral time squared error (ITSE) and etc.

In this study the integral time absolute error (ITAE) is used. The ITAE criterion generally produces a smaller overshoot and oscillation than ISE and IAE criteria. In addition, it is the most sensitive of the three, and sometimes too sensitive-slight parameter variation degrades system performance.

ITAE can be represented by the following equation:

$$\text{ITAE} = \int_0^{T_s} t\ |e_\theta(t)|dt + \int_0^{T_s} t\ |e_w(t)|dt + \int_0^{T_s} t\ |\, e_{Ia}(t \tag{28}$$

Simulation Results and Discussions

The purpose of this research is to identify the best values for the controller's parameters in a cascade P-PI setup, so that a single-axis PMDC motor may be controlled with precision in terms of speed and position. We achieve this by comparing the results of four different tuning strategies: GA, CM, BOA, and the DOA. The findings will provide light on which of these approaches yields the most precise tuning for the controller, and hence better motor control. According to Figures 5, 6, 7 and 8 analysis of the Classical Method (CM) revealed that it did not yield satisfactory results. The system became unstable when parameters extracted from this method were implemented. When the Genetic Algorithm (GA) was applied, a noticeable deviation of approximately 27° from the reference position was observed. Furthermore, the speed appeared irregular, and the motor rotated in the opposite direction before stopping at the desired position. The application of the Butterfly Optimization Algorithm (BOA) resulted in an 18° deviation from the reference position. This also led to irregular speed and the motor reversing its rotation before settling at the desired positionIn contrast, the Dandelion Optimization Algorithm (DOA) showed no deviation from the reference position or speed overshoot.

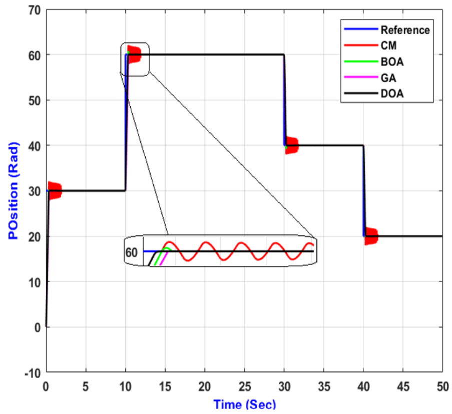

Figure 5. Position based on CM, BOA, GA, and DOA strategies at 17 Nm as load

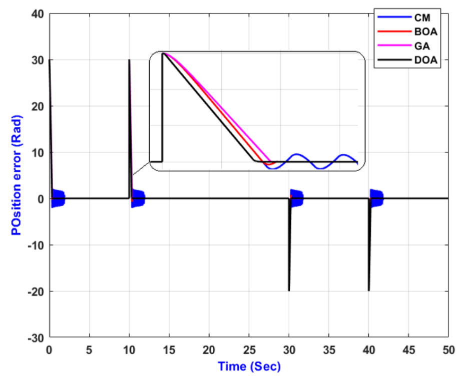

Figure 6. Position error based on CM, BOA, GA, and DOA strategies at 17 Nm as load

The system was rigorously tested under all conditions, including load scenarios, with single or multiple applied reference positions.

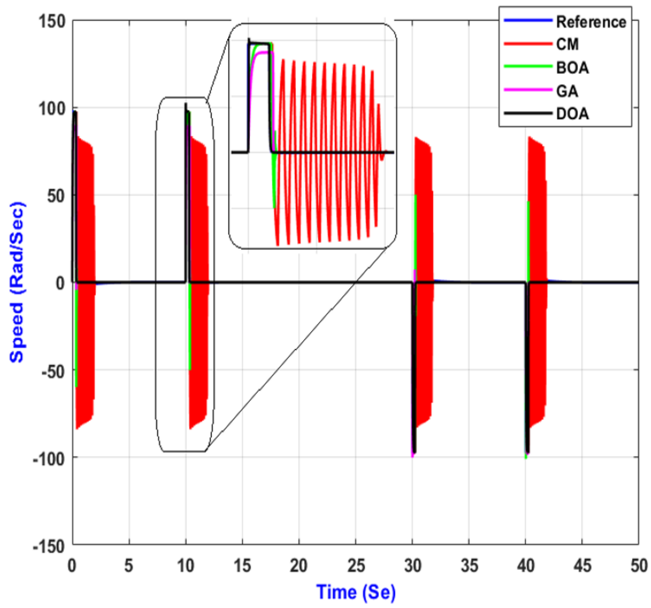

Figure 7. Speed based on CM, BOA, GA, and DOA strategies at 17 Nm as load

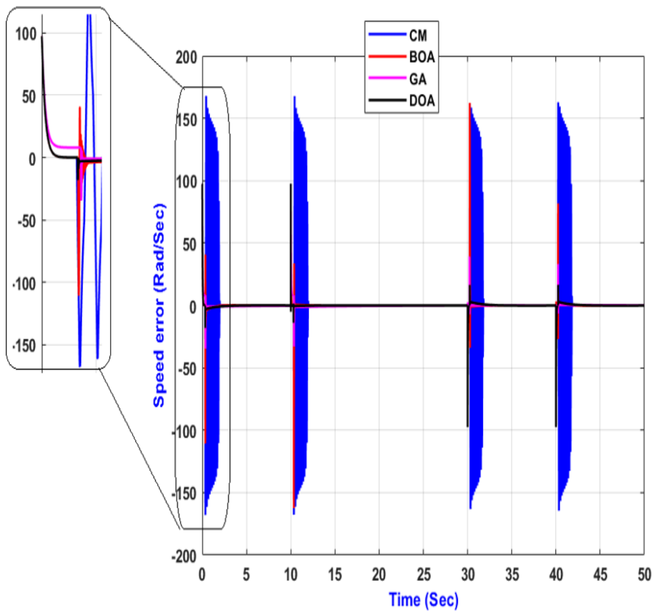

Figure 8. Speed error result based on CM, BOA, GA, and DOA strategies at 17 Nm as load

The system was rigorously tested under all conditions, including load scenarios, with single or multiple applied reference positions. The DOA algorithm consistently demonstrated accurate trajectory tracking to reach the desired position at a steady speed, Notably, the voltage and current values remained within the safe limits, preventing potential damage to the motor. Table 4 provides a detailed comparison of the performance criteria values for the position simulation results under full load conditions for the three optimal algorithms (BOA, DOA, and GA).

Table 4: Performance criteria

Performance criteria	GA	DOA	BOA
Rising time (sec)	0.35	0.32	0.6
Settling time (sec)	0.5	0.4	0.4
Overshoot %	8.1128	4.796	2

The Table 5 explains the values of P-PI parameters that we obtaineded from the application of different optimization methods.

Table 5: Cascade P-PI parameters

Method	KP position	KP speed	KI speed	KP current	KI current
CM	62.8319	2.0944	0.9425	0.3817	21.153
GA	186.4578	133.669	11.887	55.6998	2.365
BOA	24.2032	26.985	38.9883	8.79824	46.0113
DOA	44.0937	63.3902	19.4847	70.639	3.21979

CONCLUSION

The results of this study significantly contribute to the field of motor control systems by introducing a robust cascade P-PI controller for effective speed and position control of single-axis PMDC motors. The controller, primarily based on simulation design, was developed and tested using both classical methods (CM) and three advanced optimization techniques: Genetic Algorithm (GA), Dandelion Optimization Algorithm (DOA), and Butterfly Optimization Algorithm (BOA). The performance evaluation conducted under different operating conditions, including load and no-load scenarios, further corroborated the superiority of the DOA technique. In particular, the DOA excelled by delivering no deviation from the reference position or speed overshoot and keeping voltage and current values within permissible limits. This enhanced safety feature signifies the potential of the proposed controller to prevent possible motor damage. The comparative analysis performed in this study emphasized the drawbacks of the CM and the inconsistency of the GA and BOA. Meanwhile, it spotlighted the effectiveness and resilience of the DOA in tuning the controller's parameters, consistently achieving accurate trajectory tracking and the desired position at a steady speed. So, the proposed cascade P-PI controller, combined with the Dandelion Optimization Algorithm, offers a robust and efficient solution for controlling the speed and position of single-axis PMDC motors. It opens new avenues for further research in the area of motor control systems, particularly those that focus on enhancing accuracy, resilience, and safety.

REFERENCES

1. Latif, A. (2020). Optimum Synthesis of a BOA optimized novel dual-stage PI-(1+ ID) controller for frequency response of a microgrid. Energies, 13(13), 3446.
2. Ali, W. H., Sadiku, M. N. O., & Abood, S. (2019). Fundamentals of Electric Machines: A Primer with MATLAB: A Primer with MATLAB. CRC Press.
3. Bigelow, T. A. (2020). Electric Circuits, Systems, and Motors. Springer.
4. Cankurtaran, M. F., & Kocamis, A. E. (2019). Sensorless Speed Control of PMDC Motor with Cascade PI Controller. International Symposium ELMAR. IEEE.
5. Chau, K. T., & Wang, Z. (2011). Chaos in electric drive systems: analysis, control and application. John Wiley & Sons.
6. Cuong, N. D., & Puta, H. (2013). Design of MRAS based control systems for load sharing of two DC motors with a common stiff shaft. 2013 International Conference on Control, Automation and Information Sciences (ICCAIS). IEEE.
7. Dorf, R. C., & Bishop, R. H. (2011). Modern control systems. Pearson.
8. Franchi, C. M. (2019). Electrical Machine Drives: Fundamental Basics and Practice. CRC Press.
9. Ishizaki, K., Sencer, B., & Shamoto, E. (2013). Cross Coupling Controller for Accurate Motion Synchronization of Dual Servo Systems.
10. Krause, P. C., Wasynczuk, O., Sudhoff, S. D., & Pekarek, S. (2002). Analysis of electric machinery and drive systems (3rd ed., Vol. 2). Wiley Online Library.
11. Mehta, V. K., & Mehta, R. (2008). Principles of electrical machines. S. Chand Publishing.
12. Mohan, N. (2003). Electric drives.

13. Momoh, J. A. (2018). Energy Processing and Smart Grid. John Wiley & Sons.
14. Namazov, M., & Basturk, O. (2010). DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods.
15. Panlong, Z., & Wang, Z. (2019). Improvements of direct current motor control and motion trajectory algorithm development for automated guided vehicle. *Advances in Mechanical Engineering*, 11(2), 1687814018824937.
16. Pisano, A. (2008). Cascade control of PM DC drives via second-order sliding-mode technique. IEEE *Transactions on Industrial Electronics*, 55(11), 3846-3854.
17. Raja, G. L., & Ali, A. (2017). Series cascade control: An outline survey. 2017 Indian Control Conference (ICC). IEEE.
18. Romero, L., & Antonio, C. (2006). Control of position/velocity in a mobile robot using dc brushless motors. Electronics, Robotics and Automotive Mechanics Conference. IEEE.
19. Son, Y. I. (2014). Robust cascade control of electric motor drives using dual reduced-order PI observer. IEEE *Transactions on Industrial Electronics*, 62(6), 3672-3682.
20. Taha, N., & Gücin. (2015). Tuning cascade PI (D) controllers in PMDC motor drives: A performance comparison for different types of tuning methods. 2015 9th International Conference on Electrical and Electronics Engineering (ELECO), IEEE.
21. Talavaru, U., & Naik, R. N. (2014). Microcontroller based closed loop speed and position control of DC motor.
22. Tang, J. (2001). PID controller using the TMS320C31 DSK with online parameter adjustment for real-time DC motor speed and position control. ISIE. IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570), 2, IEEE.
23. Taut, E., & Marius, A. (2018). Model-in-the-Loop for Determining the Speed and Position of a DC Motor. 41st International Spring Seminar on Electronics Technology (ISSE). IEEE.
24. Wang, L. (2020). PID control system design and automatic tuning using MATLAB/Simulink. John Wiley & Sons.

Khalaf Abdullah Khalaf KHALAF

Khalaf Abdullah Khalaf Khalaf is a dedicated. Master's student specializing in power electronics, optimization and control systems. Khalaf's academic journey started with a Bachelor's degree in Electrical Engineering. His innovative work revolves around creating advanced algorithms and control mechanisms to improve energy utilization and system performance. He is driven by the goal to contribute substantially to the field and enhance the efficiency and reliability of power electronic systems.

Journal of
Computer & Electrical and Electronics Engineering Sciences

Original Article

DOI: 10.51271/JCEEES-0009

Classification of heart disease dataset with k-NN optimized by pso and gwo algorithms

Murat Lüy[1] Nuri Alper Metin[2]

[1]Department of Electrical and Electronics Engineering, Faculty of Engineering, Kırıkkale Univercity, Kırıkkale, Turkey
[2]Department of Electronic Communication Program, Kırıkkale Vocational School, Kırıkkale Univercity, Kırıkkale, Turkey

Cite this article: Lüy M, Metin NA. Classification of heart disease dataset with k-NN optimized by pso and gwo algorithms. *J Comp Electr Electron Eng Sci.* 2023;1(2):41-45

Corresponding Author: Nuri Alper Metin, nurialpermetin@kku.edu.tr

Received: 12/08/2023 ◆ **Accepted:** 06/10/2023 ◆ **Published:** 31/10/2023

ABSTRACT

Lifestyle changes worldwide are increasing chronic diseases (CD). This research concluded that cardiovascular diseases (CVDs) cause 46% of global mortality, excluding communicable diseases and accidents, and heart attacks are 7.4 million of the 17.5 million CVD deaths. 2030 will see 22.2 million cardiovascular deaths. Avoiding and treating most HD reduces cardiovascular disease fatalities. In this study, the data set for heart disease is optimized with PSO and GWO, and classification is performed with k-NN.

Keywords: PSO, GWO, k-NN

INTRODUCTION

Today, viral cardiovascular disease creates heart issues quickly. If the heart fails, all bodily functions will be impacted. One-third of the world dies of heart failure, according to WHO figures. 2016 marks 18 million CVD sufferers. Heart failure or attack killed almost 84%. Down nations had three-quarters of cardiovascular disease mortality. 2015 survey. He has non-infectious heart failure from 82 to 20. From these statistics, CVD caused her heart illness (Dulhare, 2018; Hasanova et al., 2022; Naga & Asst, 2023; Sengur, 2008; Seslier & Karakuş, 2022; Tharwat et al., 2018). Health risks, including smoking, poor diet, and excessive alcohol usage, often cause cardiovascular disease deaths. Advanced cardiovascular disease or CVD patients, diabetes, hypertension, hyperlipidemia, and other risk factors should be discussed early on. Adipose-stored CVD puffs have formed inside atherosclerosis and blood clusters. It harms the eyes, kidneys, heart, and mind. UK/US CVD has killed severely—cardiac stroke. Episodes are severe events that stop heart or blood circulation. CVD development targets fat buildup and venous consequences. Obesity, cigarette usage, and poor nutrition contribute to CVD strokes and failures. The heart circulates blood via blood vessels. Blood flow eliminates metabolic waste from the heart and provides nutrients and oxygen to various areas. Poor blood flow may damage organs and worsen heart failure. This study presents PSO and GWO optimization and k-NN classification for heart disease (Dulhare, 2018; Hasanova et al., 2022; Naga & Asst, 2023; Sengur, 2008; Seslier & Karakuş, 2022; Tharwat et al., 2018).

Literature Review

Khourdifi et al., Machine learning predict heart disease. Optimization strategies may handle complex non-linear problems. FCBF filtered redundant characteristics to classify heart disease in this article. k-NN, Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), and a PSO-ACO-optimized Multilayer Perception, Artificial Neural Network Classification. A heart disease dataset tests the hybrid heart disease classification method. Its efficacy and robustness were shown. This study examines machine learning approaches using accuracy, precision, recall, f1-score, etc. Improved FCBF, PSO, and ACO models classify 99.65% accurately. The suggested technique surpasses classification (Khourdifi & Bahaj, 2019).

Sandhiya et al., Heart disease kills all ages since people don't know their severity. Heart disease kinds and monitoring are crucial in our fast-paced world. IoT and Deep Learning create a patient-safe heart disease monitoring system. Feature selection helps deep learning classification. Our proposed system monitors heart disease using IoT device inputs. It classifies individuals by heart illness kind and severity. Finally, heart disease kind and inputs notify patients. The recommended system predicted better testing (UPalani Teaching Fellow Professor, 2022).

Tama et al., Coronary heart disease (CHD) are common and severe. Poor lives kill the most people globally. Since cardiac attacks are symptomless, sophisticated detection is needed. This article introduces classifier ensembles for CHD detection. Two-tier ensemble classifiers are ensemble-specific. A stacked design uses random forest class label prediction, gradient boosting, and extreme slope boosting. Z-Alizadeh Sani, Statlog, Cleveland, Ohio, and Hungarian heart disease datasets validate the detection model. Particle swarm optimization chooses each dataset's most significant properties. Finally, a two-fold statistical study disproves classifier performance disparities based on assumptions. Our 10-fold cross-validation outperforms ensemble base classifiers. Our detection system surpassed classifier groups and classifiers in accuracy and AUC. This study reveals that our model contributes more than earlier publications (Tama et al., 2020).

Khourdifi et al., The hybrid Machine Learning model for the proposed PA-RF, a classification based on the Random Forest model, was optimized by PSO, ACO, and FCBF to filter redundant and irrelevant characteristics to improve heart disease classification. Heart disease data is mixed. The hybrid method is effective and durable in classifying heart disease using different data sources. Thus, this study evaluates autonomous learning algorithms using Accuracy, Precision, Recall, F1-Score, etc. This study used UCI's autonomous learning repository's "Heart Disease" data set. PA-RF excels (Khourdifi & Bahaj, 2019).

The study conducted by Muthukaruppan et al. focuses on developing a Particle Swarm Optimization (PSO)-based fuzzy expert system for Computer-Aided Design (CAD) diagnosis. The system was influenced by the Cleveland and Hungarian Heart Disease databases. Using a decision tree (DT) enabled the identification of diagnostic characteristics within datasets encompassing many input qualities. The DT output was enhanced by transforming it into crisp if-then rules and incorporating a fuzzy rule base, which was further optimized using PSO to tune the Fuzzy Membership Functions (FMFs). The fuzzy expert system achieved a classification accuracy of 93.27% using improved membership functions. This methodology demonstrates enhanced interpretability of fuzzy expert system selections (Muthukaruppan & Er, 2012).

Syafi et al., The heart pumps blood. Data mining from a massive data warehouse aids decision-making. Data structure precedes data mining. Then, Information Gain Ratio and Particle Swarm Optimization choose the best features. Adaboost maximized accuracy. Data classification follows. C4.5 classes. The C4.5 approach uses Information Gain Ratio (IGR) and Particle Swarm Optimization, then applying the Adaboost ensemble has a 96.68% accuracy rate, whereas the one without feature selection has 95.87%. Information Gain Ratio and Particle Swarm Optimization utilizing the Adaboost ensemble may improve C4.5 classification algorithm performance (Qois Syafi, 2022).

Roostaee et al., Large heart disease data sets challenge analysis. All features disappoint. Thus, qualities need improvement. Binary cuckoo optimization reduces property. SVM classifies heart disease best. This article simplifies illness diagnosis. Accuracy, sensitivity, and specificity. 14 attributes for 303 ML Repository users. Current approaches are costly, accurate, and time-consuming. The suggested technique is faster, more accurate, and more efficient (Roostaee & Ghaffary, 2016).

Li et al., Electrocardiograms (ECGs) objectively measure heart function and physiological condition, making them helpful in diagnosing cardiac illness. Feature extraction determines AECG judgment accuracy. Environmental noise makes ECG detection challenging. Different AECG species exist. ECG findings from a long time ago cannot be utilized to identify or diagnose sickness. Thus, AECG detection requires an innovative classification method with exact feature extraction. This study used ECG feature extraction to create an AECG detection and cardiac disease diagnostic algorithm (Li et al., 2021).

Dubey et al., Cardiovascular diseases are global health issues. Early diagnosis may improve survival for cardiovascular illnesses, which have the highest worldwide mortality rate and increase with age. ML and optimization predict heart diseases in this paper. ML approaches were used to classify data and IACPSO to choose the best features. UCI ML evaluated Cleveland, Statlog, and Hungarian heart disease datasets. Model performance was assessed. LR and SVMGS outperformed on Statlog, Cleveland, and Hungarian datasets. IACPSO-ML improved performance indicators by 3–33% (Dubey et al., 2022).

Jabbar et al., Knowledgeable information has been mined extensively using data mining methods from medical databases. Classification is a kind of supervised learning used in data mining, and it is put to use by building models that characterize various classes of data. The nearest neighbor (k-NN) method is the simplest, most widely used, and most successful pattern recognition algorithm. Classifying samples according to the category of their closest neighbor, k-NN is a simple classifier. The amount of data in healthcare databases is quite large. Classification performance may suffer if the data set includes superfluous or unnecessary characteristics. In INDIA, heart disease is number one among all causes of mortality. Heart disease is the top cause of death in Andhra Pradesh, accounting for 32% of all fatalities. It aligns with the 35% seen in Canada and the USA. Creating a decision support system is crucial to help physicians decide whether to take preventive steps. This research combines KNN with a genetic algorithm to provide a fresh approach to accurate categorization. A global search conducted by genetic algorithms over vast, multimodal landscapes may find the optimal solution. The experimental data indicate that our method enhances the diagnostic accuracy of heart illness (Jabbar et al., 2013).

This study involved the utilization of PSO and GWO algorithms for the optimization of the data set. Additionally, the classification task was carried out using the k-NN algorithm. The accuracy and processing time of both algorithms are compared.

Table 1. Literature review

Literature Articles	Optimization Methods	Classification Methods
Khourdifi et al (2018)	ANN-PSO, ANN-ACO	k-NN, SVM, RF, NB, MLP
Sandhiya et al. (2022)	GWO	DBN
Tama et al. (2020)	Two-Tier PSO	RF, GBM, XGBoost
Khourdifi et al (2019)	RF-PSO, RF ACO	k-NN, SVM, RF, NB, MLP, PA-RF
Muthukaruppan et al. (2012)	-	PSO-Fuzzy
Syafi et al. (2022)	PSO, IGR	C4.5
Roostaee et al. (2016)	BCOA	SVM
Li et al. (2021)	PSO-BPNN	PCA
Dubey et al. (2022)	IAPSO	LR, DT, RF, SVM, SVMGS k-NN, NB
Jabbar et al. (2013)	GA	k-NN

This research aims to enhance classification accuracy using an optimized k-NN algorithm to diagnose heart disease. PSO and GWO algorithms are used to optimize the heart disease dataset.

METHODS

In this study, PSO and GWO are used for data set optimization. The PSO algorithm is an example of a heuristic algorithm that works by repeatedly attempting to improve upon an existing solution to a problem. The animals' interactions with one another provided the basis for this program. A parameter value is specified in the PSO method to maximize the size of the particles scattered across the search space and the particles wandering the search space. The direction of these particles in the search space is determined not only by their flight route but also by the collective flight path of the flock, just as it is in a flock of birds (Wadhawan & Maini, 2022). The Block Diagram of PSO is shown in Figure 1.

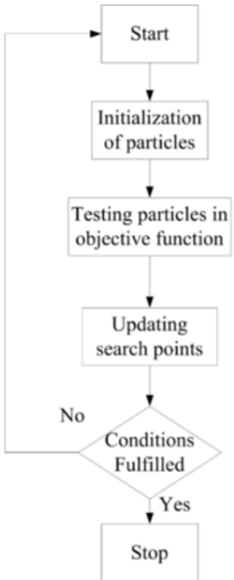

Figure 1. Block diagram of PSO

The grey wolf optimization algorithm, often known as the GWO, is a technique that mathematically models and imitates the behavior of a pack of grey wolves' behavior when hunting. In the hierarchy of wolf packs, grey wolves are referred to as the alpha (∝), beta (β), delta (δ), and omega (ω) positions, respectively. Because it plays by the same rules as the other species, the alpha wolf pack is considered the head of the pack. The deputy leader is referred to as the beta wolf, and they assist the alpha wolf throughout the decision-making process. The grey wolves designated as omega are positioned at the bottom of this hierarchy. The wolf is known as a delta if it does not belong to any species stated in the pack. An intriguing illustration of how grey wolves communicate with one another is shown in the behavior of group hunting. The behavior of GWOs may be broken down into four distinct stages: searching, surrounding, assaulting, and hunting (Al-Tashi et al., 2019; Mirjalili et al., 2014). The Block Diagram of GWO is shown in Figure 2.

K-NN classifier is a well-known classification method that is also relatively straightforward. Fix and Hodges initially presented it as a non-parametric approach, meaning it does not make any assumptions about the input data distribution. As a result, it has found widespread usage in various applications since its beginning.

The K-Nearest Neighbors (k-NN) classifier assigns an unidentified sample to a specific category by evaluating the distance between the sample and all labeled instances.

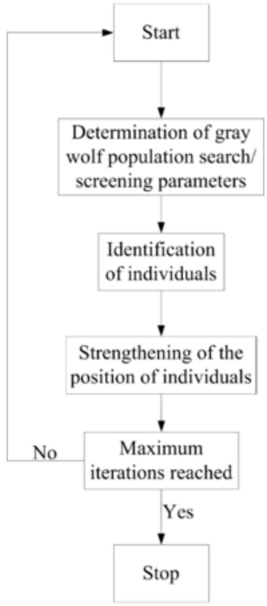

Figure 2. Block diagram of GWO

The unidentifiable sample is allocated to the class with the most samples within its k-nearest neighbors. The k-NN classifier algorithm relies on three essential components. Choosing a precise value for the integer k is crucial to deciding how many closest neighbors should be considered throughout the analysis. Additionally, it is imperative to have training data that includes labeled samples, as these can be adjusted by adding or eliminating samples. Finally, a distance metric is utilized to compute the closeness or proximity between different data points. The utilization of the Euclidean distance metrics to calculate the distance between samples is exemplified in Equation (1) within the k-NN algorithm. The k-NN classifier is advantageous due to its traceability and ease of construction. Nevertheless, storing every sample used for training in memory during runtime requires categorizing it as a memory-based classification approach (Karabulut et al., 2019; Tharwat et al., 2018; Prasath et al., 2017).

$$d(x_i, x_j) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

Datasets Description

The dataset under consideration originates from 1988 and comprises four distinct databases: Cleveland, Hungary, Switzerland, and Long Beach V. The dataset comprises 76 attributes, encompassing the predicted attribute. However, all experiments published thus far exclusively utilize a subset of 14 attributes. The term "target" pertains to the existence of cardiovascular disease in the individual. The variable is assigned integer values, where 0 represents the absence of disease, and 1 represents the presence of disease.

The heart disease dataset is defined by gender, age, pain in the chest type (4 values), laying down level of blood pressure, cholesterol levels in mg/dl fasting blood, sugar>120 mg/dl, maximum heart rate accomplished, exercise-induced chest discomfort, old maximum=ST depression caused by physical activity compared to a resting state, the slope of the highest point during exercise, the segment of the ST with three or more significant vessels (0-3) colored by fluoroscopy that, and other variables are defined in the database. Typical is 0, fixed defects are 1, and reversible defects are 2 (Kaggle, A.D: 02/08/2023).

Predictive Methods

Early detection of the disease and appropriate treatment are essential in preventing the increasing number of male and female patients with heart disease and reducing the risk of death. Today, the use of optimization and classification methods for the causes of heart disease is increasing. With the optimization and classification methods used, it is ensured that accurate results are obtained, the time required for disease detection is minimized, and human errors are prevented. This study used particle swarm optimization (PSO) and grey wolf optimization (GWO) algorithms to optimize the data set. The k-Nearest Neighbors (k-NN) algorithm also classified the data set. The accuracy and processing time of both algorithms are compared.

RESULTS

PSO parameters are given in Table 2. PSO consists of parameters lb (lower bound), ub (upper bound), coefficients c1, c2, and w (weight).

Table 2. PSO parameters

Parameter Name	Parameter Value
lb	0
ub	1
c1	2
c2	2
w	0.9
Number of solutions	10
Iteration number	1000

GWO parameters are given in Table 3. GWO consists of parameters lb (lower bound), ub (upper bound), coefficients c1, c2, and c3.

Table 3. GWO parameters

Parameter Name	Parameter Value
lb	0
ub	1
c1	Between 0 and 2
c2	Between 0 and 2
c3	Between 0 and 2
Number of solutions	10
Iteration number	1000

This paper optimizes the heart disease dataset with PSO and GWO. Fitness values according to iteration are shown in Figure 3.

The processing time of PSO and GWO optimization algorithms is given in Table 4.

Table 4. The processing time of PSO and GWO

Optimization Type	Processing Time (sec)
GWO	60.120612
PSO	60.768583

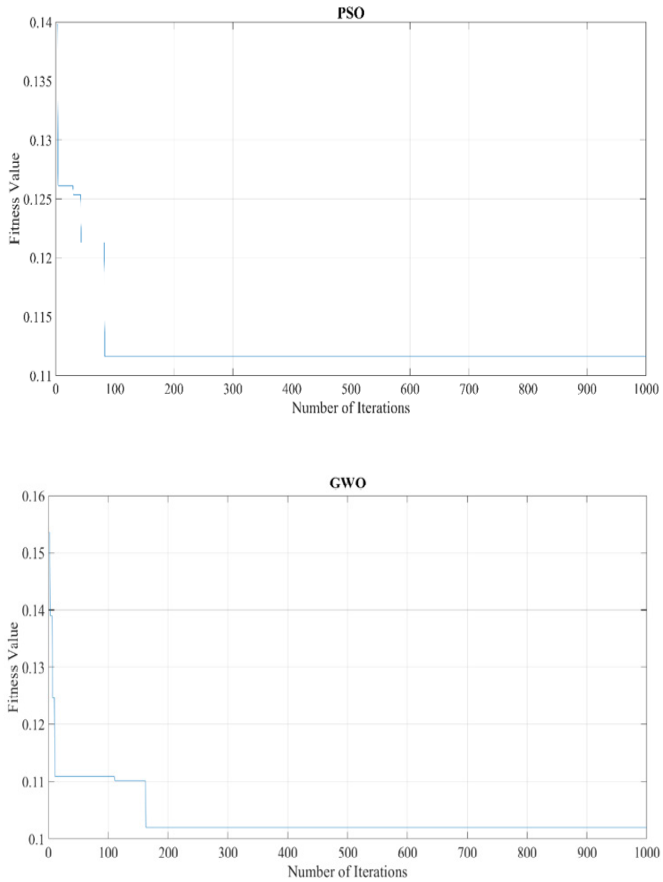

Figure 3. Fitness values according to the number of iterations of GWO and PSO

For k-NN classification on the GWO and PSO optimized heart disease dataset, k is chosen to be 5. accuracy is shown in Figure 4.

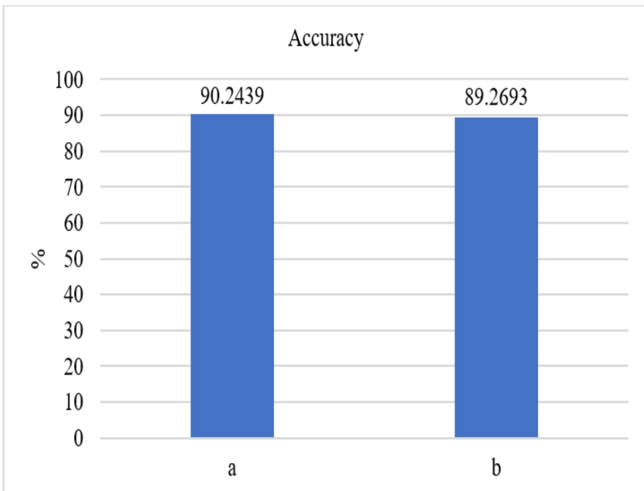

Figure 4. a) Accuracy calculated using GWO in heart disease dataset optimization b) Accuracy calculated using PSO in heart disease dataset optimization

DISCUSSION

This study optimized the heart disease dataset with GWO and PSO, and classification was performed with k-NN. The comparison process was performed in a Matlab environment. GWO proved to be better than PSO in both processing time and accuracy. The proposed system was compared with studies in the literature in terms of accuracy. This situation is given in Table 5.

Table 5. A comparison of accuracy

Study	Feature Selection	Classify Techniques	Disease	Accuracy(%)
Khourdifi et al. (2018)	ANN-PSO, ANN-ACO	k-NN	Heart Disease	99.65
Sandhiya et al. (2022)	NGWO	DBN	Heart Disease	85.4
Tama et al. (2020)	Two-Tier PSO	RF, GBM, XGBoost	Heart Disease	85.71
Khourdifi et al. (2019)	PSO, ACO	k-NN	Heart Disease	84.13
Muthukaruppan et al. (2012)	-	PSO-Fuzzy	Heart Disease	93.27
Syafi et al. (2022)	PSO-IGR	C 4.5	Heart Disease	96.68
Roostaee et al. (2016)	BCOA	SVM	Heart Disease	84.44
Li et al. (2021)	PSO-BPNN	PCA	ECG	96
Dubey et al. (2022)	IACPSO	k-NN	Heart Disease	92
Jabbar	GA	k-NN	Heart Disease	95.73
Proposed System	PSO, GWO	k-NN	Heart Disease	90.2439 (GWO) 89.2693 (PSO)

For future studies, the system's accuracy can be increased by changing the optimization algorithm and classifier in the system.

CONCLUSION

This study uses PSO and GWO algorithms for heart disease dataset optimization. The k-NN algorithm classifies the optimized datasets. These algorithms are run for 1000 iterations. When the heart disease dataset is optimized with the PSO algorithm and classified with k-NN, it has 60.768583 processing time and 89.2683% accuracy rate, while when it is optimized with GWO and classified with k-NN, it has 60.120612 processing time and 90.2439% accuracy rate. As a result of the simulation analysis shows that the GWO algorithm has a shorter processing time and higher accuracy rate than PSO when the heart disease dataset is optimized and classified with k-NN.

REFERENCES

1. Al-Tashi, Q., Rais, H., & Jadid, S. (2019). Feature selection method based on grey wolf optimization for coronary artery disease classification. Advances in Intelligent Systems and Computing, 843, 257-266. https://doi.org/10.1007/978-3-319-99007-1_25
2. Dubey, A. K., Sinhal, A. K., & Sharma, R. (2022). An Improved Auto Categorical PSO with ML for Heart Disease Prediction. *Engineering, Technology & Applied Science Research*, 12(3), 8567-8573. https://doi.org/10.48084/etasr.4854
3. Dulhare, U. N. (2018). Prediction system for heart disease using Naive Bayes and particle swarm optimization. *Biomedical Research*, 29(12), 2646-2649. https://doi.org/10.4066/biomedicalresearch.29-18-620
4. Hasanova, H., Tufail, M., Baek, U. J., Park, J. T., & Kim, M. S. (2022). A novel blockchain-enabled heart disease prediction mechanism using machine learning. *Computers and Electrical Engineering*, 101(May), 108086. https://doi.org/10.1016/j.compeleceng.2022.108086
5. Jabbar, M. A., Deekshatulu, B. L., & Chandra, P. (2013). Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm. *Procedia Technology*, 10, 85-94. https://doi.org/10.1016/j.protcy.2013.12.340
6. Karabulut, B., Arslan, G., & Ünver, H. M. (2019). A Weighted Similarity Measure for k-Nearest Neighbors Algorithm. Celal Bayar Üniversitesi Fen Bilimleri Dergisi, 15(4), 393-400. https://doi.org/10.18466/cbayarfbe.618964
7. Khourdifi, Y., & Bahaj, M. (2019). Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization. *International Journal of Intelligent Engineering and Systems*, 12(1), 242-252. https://doi.org/10.22266/ijies2019.0228.24
8. Khourdifi, Y., & Bahaj, M. (2019). The Hybrid Machine Learning Model Based on Random Forest Optimized by PSO and ACO for Predicting Heart Disease. https://doi.org/10.4108/eai.24-4-2019.2284088
9. Li, G., Tan, Z., Xu, W., Xu, F., Wang, L., Chen, J., & Wu, K. (2021). A particle swarm optimization improved BP neural network intelligent model for electrocardiogram classification. BMC Medical Informatics and Decision Making, 21(Suppl 2), 1-15. https://doi.org/10.1186/s12911-021-01453-6
10. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007
11. Muthukaruppan, S., & Er, M. J. (2012). A hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease. *Expert Systems with Applications*, 39(14), 11657-11665. https://doi.org/10.1016/j.eswa.2012.04.036
12. Naga, M., & Asst, S. (2023). Detection of Cardiovascular Disease using Machine Learning, Genetic Algorithms and Particle Swarm Optimization. *IJERT*, 12(03), 120-127.
13. Prasath, V. B. S., Alfeilat, H. A. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., & Salman, H. S. E. (2017). Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier -- A Review. 1-39. https://doi.org/10.1089/big.2018.0175
14. Qois Syafi, M. (2022). Increasing Accuracy of Heart Disease Classification on C4.5 Algorithm based on information gain ratio and particle swarm optimization using adaboost ensemble. *Journal of Advances in Information Systems and Technology*, 4(1), 100-112. https://journal.unnes.ac.id/sju/index.php/jaist
15. Roostaee, S., & Ghaffary, H. R. (2016). Diagnosis of heart disease based on meta heuristic algorithms and clustering methods. *Journal of Electrical and Computer Engineering Innovations JECEI*, 4(2), 105-110. https://doi.org/10.22061/jecei.2016.570
16. Sengur, A. (2008). An expert system based on principal component analysis, artificial immune system and fuzzy k-NN for diagnosis of valvular heart diseases. *Computers in Biology and Medicine*, 38(3), 329-338. https://doi.org/10.1016/j.compbiomed.2007.11.004
17. Seslier, T., & Karakuş, M. Ö. (2022). In healthcare applications of machine learning algorithms for prediction of heart ATTACKS. *Journal of Scientific Reports-A*, 051, 358-370.
18. Tama, B. A., Im, S., & Lee, S. (2020). Improving an Intelligent Detection System for Coronary Heart Disease Using a Two-Tier Classifier Ensemble. BioMed Research International, 2020. https://doi.org/10.1155/2020/9816142
19. Tharwat, A., Mahdi, H., Elhoseny, M., & Hassanien, A. E. (2018). Recognizing human activity in mobile crowdsensing environment using optimized k-NN algorithm. *Expert Systems with Applications*, 107, 32-44. https://doi.org/10.1016/j.eswa.2018.04.017
20. UPalani Teaching Fellow Professor, Ss. (2022). An IoT Enabled Heart Disease Monitoring System Using Grey Wolf Optimization and Deep Belief Network. https://doi.org/10.21203/rs.3.rs-1058279/v1
21. Wadhawan, S., & Maini, R. (2022). EBPSO: Enhanced binary particle swarm optimization for cardiac disease classification with feature selection. *Expert Systems*, 39(8), 1-20. https://doi.org/10.1111/exsy.13002
22. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset AD: 02/08/2023

Murat Lüy

Murat Lüy continues as associate professor in Kırıkkale University, Faculty of Engineering and Architecture, Electrical and Electronics Engineering.

Journal of
Computer & Electrical and
Electronics Engineering Sciences
Original Article
DOI: 10.51271/JCEEES-0010

Machine learning methods on quantized vectors

Ufuk Tanyeri[1] Taner Dindar[2] Yunus Kökver[3] Nuri Furkan Koçak[2]

[1]Department of Computer Technologies, Nallıhan Vocational School, Ankara University, Ankara, Turkey
[2]Department of Electronic Technologies, Nallıhan Vocational School, Ankara University, Ankara, Turkey
[3]Department of Computer Technologies, Elmadağ Vocational School, Ankara University, Ankara, Turkey

Cite this article: Tanyeri U, Dindar T, Kökver Y, Koçak NF. Machine learning methods on quantized vectors. *J Comp Electr Electron Eng Sci.* 2023;1(2):46-49.

Corresponding Author: Ufuk Tanyeri, ufuktanyeri@ankara.edu.tr

Received: 15/09/2023 ◆ **Accepted**: 19/10/2023 ◆ **Published**: 31/10/2023

ABSTRACT

Vector quantization is one of the important issues in digital images. There are many studies conducted on quantized vectors or images. On the other hand, machine-learning approaches are a popular issue today. In this study, the classification performances of machine learning approaches on reduced image vectors are examined. Firstly, Corel 1K data set were reduced to 64 colors with octree and histogram feature vectors extracted. Classification was carried out using various machine learning approaches on the relevant vectors. As a result of the classification, the success of the methods was examined.

Keywords*:* Vector quantization, machine learning, corel 1k, classification

INTRODUCTION

Vector quantization (VN) is a technique used in signal processing and data compression approaches. While analysis, modification, or improvement of analog or digital signals is carried out with signal processing, data size is reduced while preserving important features with data compression. Lloyd (1957) laid the scalar-based foundation of VN in the field of signal processing (Lloyd, 1957), while Forgy used it in vector form (Forgy, 1965). In fact, the issue of optimal quantization of a vector space is called Dirichlet tessellation in two- and three-dimensional spaces and Voronoi tessellation in arbitrary-dimensional spaces (Lejeune Dirichlet, 1850; Voronoi, 1908). VN is a standard in digital signal processing today. With this method, a large set of data points is represented by a smaller set of reference points known as centroids or codebook vectors. Each data point is then assigned to the nearest center. This way, the data is effectively quantified. If the input data is finite, the bulk calculation method called Linde-Buzo-Gray (LBG) is used (Linde et al, 1980). There is also a k-means classification that is widely used in the literature (Gersho, 1982; Gray, 1984; Makhoul et al, 1985). Octree is another method used to divide a three-dimensional space into smaller and more manageable parts (Meagher, 1982). It is particularly used in computer graphics, 3D modeling, geographic information systems (GIS), game development, and similar fields.

Today, the increase in the number of digital images brings with it the problem of storing them. One solution is to quantize the pixel values in the image into a smaller set of values using VN. VN can greatly reduce the storage requirements of images while preserving essential features. Additionally, the size of voice and many other types of data can also be reduced. Pattern recognition is another area where VN is used. Pattern recognition is a versatile method that helps understand, classify, group data, and predict future events (Li et al, 2021; Serey et al, 2023; Mantel, 1974). In this field, since the performance of VN depends on the design of the code book, the selected distance measure, and the specific application area, it is preferred as an alternative to other methods such as neural networks.

Another important issue, such as quantization of data, is classification. Classification is based on machine learning algorithms and is used to separate input data points into predefined classes or categories. These techniques leverage input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. This way, it can find the same pattern (similar words or emotions, number sequences, etc.) in future datasets.

The field of classification studies is very extensive, and depending on the dataset you are working with, you can employ a variety of techniques. Logistic regression, naive bayes, random forest, gradient boosting, k-nearest neighbors (KNN), decision trees and support vector machines (SVM) are common approaches. Another alternative classification method logistic regression (LR) is used for binary classification. Here, the probability of a certain entry point belonging to a certain class is modeled using the logistic function (Kononenko, 1989). Naive bayes (NB) is based on the bayes theorem and assumes that the presence of a particular feature in a class is not correlated with the presence of other features (Palmer et al, 1979). The random forest (RF) model uses decision trees. RF builds each tree on a bootstrapped subset of the data and combines the predictions to improve accuracy and control overfitting (Friedman, 2001). Gradient boosting (GB) is a frequently used ensemble learning method in the field of machine learning. The related method aims to create a stronger learner by combining weak learners (usually decision trees) (Fukunaga et al, 1975). Variations of the GB approach include XGBoost (extreme gradient boosting), LightGBM (light gradient there are libraries such as boosting machine) and CatBoost. K-nearest neighbors (KNN) is a machine learning method used as a classification or regression algorithm. The basic idea is to use most data points around a data point to classify a new data point or make a prediction. K-NN is among the supervised learning algorithms and is used in data mining, pattern recognition and classification problems (Magee, 1964). Decision trees (KA) are another classification and regression algorithm used in data mining and machine learning fields. The main purpose is to classify new data samples or make predictions by analyzing the dataset. Decision trees use a tree structure containing a set of decision nodes and leaf nodes (Vapnik et al, 1996). Finally,s upport vector machines (SVM), one of the classification algorithms, aims to find a hyperplane that best separates different classes in the input feature field and maximizes the margin between them.

In this study, feature vectors were first extracted from the corel 1K data set by reducing it to 64 colors using the octree method. Then, the relevant vectors were classified with machine learning-based approaches. The motivation of the study is to examine the classification success of machine learning approaches of images reduced to 64 colors with the octree reduction technique.

This paper is structured as follows. Information on vector quantization is provided in section 2. Following that, section 3 assesses how effective the machine learning techniques are when applied to quantized vectors. Section 4 concludes with giving the final decision.

Vector Quantization

Vector quantization is one of the important techniques used in image processing. Carrying out this process in color reduction, image compression and segmentation makes quantization important. Octree, one of the most used quantization approaches, reduces the image to powers of 8. It is one of the image-dependent methods classified as a hierarchical clustering method. Even though its performance is low, its speed is an important advantage. Another method developed in 1980 is the LBG approach. The LBG method first determines random starting points on the input set. Then, the distances of the cluster elements to the starting points are calculated and the process continues until the optimum situation is achieved. Additionally, LBG It has an iterative structure. There is a stopping threshold constant for the number of iterations. Figure 1 shows the original pepper image, reduced to 8 colors using the octree and LBG methods.

(a) (b) (c)

Figure 1. Pepper image: a) original b) octree 8 colors c) LBG 8 colors

The loss of information in reduced images is an expected situation. The information loss of pepper images reduced to 64 colors with octree and LBG approaches is shown in Table 1. Signal noise is shown in ratio (PSNR).

Table 1. 64 color pepper image PSNR results

Method	PSNR (dB)
Octree	28.59
LBG	24.42

The PSNR is frequently used to evaluate how well an image has been reconstructed. As seen in Table 1, according to the PSNR result, octree is approximately 14.58% more successful than LBG. For this reason, while classification performances are examined in the study, only octree is used as vector quantization.

METHODS

In this study, machine learning-based classification is performed on 64 color features obtained from the Corel 1K data set using the octree method. In classification, TensorFlow 2.0 and above library and Python programming language are used. As a result of the classification, detailed findings for each approach were presented and evaluated.

At the beginning of the study, the data set was divided into training and test data. 20% of the data set is used for testing. Additionally, the standard scaling process was applied to the training data and the features were scaled to have a mean value of 0 and a standard deviation of 1. This process ensures that the data is suitable for model training.

Success and performance of classification processes, confusion. It is evaluated with the matrix technique. Indicators and mathematical models obtained from the relevant method; accuracy (1), precision (2), recall (3) and F-1 score (4) are given below:

$$Accuracy = \frac{TN+TP}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1-Score = \frac{2x(RecallxPrecision)}{Recall+Precision} \quad (4)$$

Figure 2 shows how success and performance evaluation are done in the classification process.

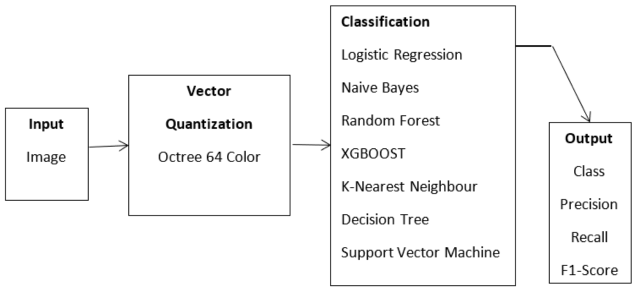

Figure 2. Block diagram of machine learning methods on quantized vectors

The average accuracy rate resulting from the classification made with the Logistic Regression algorithm was calculated at 82.50%. Figure 3 shows the confusion matrix values of the algorithm.

```
[[12  1  0  0  0  1  0  0  0  1]
 [ 1  8  1  0  0  0  0  0  1  0]
 [ 5  2 17  1  0  2  0  0  0  0]
 [ 0  1  1 20  0  0  0  0  0  0]
 [ 0  0  0  0 23  0  0  0  0  0]
 [ 0  3  0  0  0 13  0  1  1  0]
 [ 0  0  0  1  0  0 18  0  0  1]
 [ 0  1  0  0  0  1  0 22  0  0]
 [ 0  1  3  1  0  1  0  0  9  0]
 [ 1  0  0  1  0  0  0  0  0 23]]
```

Figure 3. Confusion matrix of the logistic regression algorithm

Table 2 shows the recallprecision and F1-Score indicators obtained according to the logistic regression algorithm.

Table 2. Logistic regression algorithm recall, precision and F1-Score values

Class	Precision	Recall	F1-Score
0	0.63	0.80	0.71
1	0.47	0.73	0.57
2	0.77	0.63	0.69
3	0.83	0.91	0.87
4	1.00	1.00	1.00
5	0.72	0.72	0.72
6	1.00	0.90	0.95
7	0.96	0.92	0.94
8	0.82	0.60	0.69
9	0.92	0.92	0.92

The average accuracy rate resulting from the classification made with the naive bayes algorithm was calculated at 71%. Figure 4 shows the confusion matrix values of the algorithm.

```
[[ 5  0  2  4  0  2  0  1  0  1]
 [ 1  2  5  0  0  0  0  0  2  1]
 [ 0  0 23  1  0  3  0  0  0  0]
 [ 1  1  0 19  0  0  0  1  0  0]
 [ 0  0  3  0 19  0  0  0  0  1]
 [ 0  1  2  0  0 13  0  1  1  0]
 [ 0  0  0  2  0  0 13  0  0  5]
 [ 0  0  0  0  0  0  0 24  0  0]
 [ 0  0  6  2  0  2  0  0  5  0]
 [ 4  1  0  1  0  0  0  0  0 19]]
```

Figure 4. Naive confusion about the bayes algorithm matrix

Table 3 shows the recall, precision and F1-Score indicators obtained according to the naive bayes algorithm**.**

Table 3. Naive bayes algorithm recall, precision and F1-Score values

Class	precision	Recall	F1-Score
0	0.45	0.33	0.38
1	0.40	0.18	0.25
2	0.56	0.85	0.68
3	0.66	0.86	0.75
4	1.00	0.83	0.90
5	0.65	0.72	0.68
6	1.00	0.65	0.79
7	0.89	1.00	0.94
8	0.62	0.33	0.43
9	0.70	0.76	0.73

The average accuracy rate resulting from the classification made with the random forest algorithm was calculated at 81.5%. Figure 5 shows the confusion matrix values of the algorithm.

```
[[11  0  0  1  0  1  0  0  0  2]
 [ 1  5  1  1  0  2  0  0  1  0]
 [ 0  2 19  1  0  3  0  0  2  0]
 [ 1  0  0 19  0  1  0  0  1  0]
 [ 0  0  0  0 23  0  0  0  0  0]
 [ 0  0  0  0  0 18  0  0  0  0]
 [ 0  0  0  0  0  0 19  0  0  1]
 [ 1  0  0  0  0  0  0 23  0  0]
 [ 0  4  1  2  0  2  0  0  6  0]
 [ 3  0  0  1  0  0  0  0  1 20]]
```

Figure 5. Confusion matrix of the random forest algorithm

Table 4 shows the recall, precision and F1-Score indicators obtained according to the random forest algorithm.

Table 4. Random forest algorithm recall, precision and F1-Score values

Class	precision	Recall	F1-Score
0	0.65	0.73	0.69
1	0.45	0.45	0.45
2	0.90	0.70	0.79
3	0.76	0.86	0.81
4	1.00	1.00	1.00
5	0.67	1.00	0.80
6	1.00	0.95	0.97
7	1.00	0.96	0.98
8	0.55	0.40	0.46
9	0.87	0.80	0.83

The average accuracy rate of the classification process has also been applied to other machine learning approaches and the results obtained are given in Table 5. Additionally, Figure 6 shows the accuracy rate graph of the applied models.

Table 5. Accuracy rates of classifiers

Class	Model	Accuracy
0	Logistic Regression	82.5
1	Naive Bayes	71.0
2	Random Forest	81.5
3	**XGBOOST**	**85.5**
4	K-Nearest Neighbour	70.5
5	Decision Tree	71.5
6	Support Vector Machine	79.0
7	stacking classification	83.0

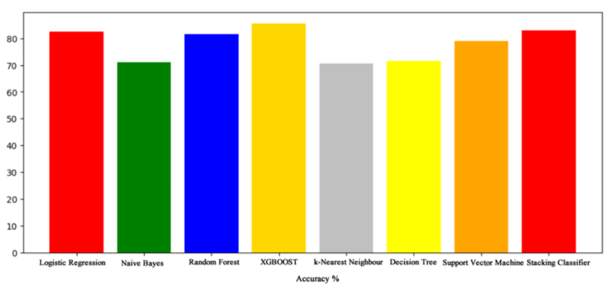

Figure 6. Accuracy rates graph of applied models

As can be seen in Table 5 and Figure 6, the XGBOOST method has the highest accuracy at 85.5%. Ranking success, from high to low, are XGBOOST, stacking classifier, logistic regression, random forest, support vector machine, decision tree, naive bayes and K-nearest neighbor.

CONCLUSION

Histograms are an important feature of images. Its simple calculation makes it a widely used identifier. However, the histogram of an image defined in the RGB color space consists of 3 different channels. Additionally, each color channel has 256 components. Performing operations on 3 different color channels and combining them is an important issue. For this reason, Corel's octree and LBG color reduction approaches images in the 1K dataset were reduced to 64 colors and a one-dimensional color histogram was obtained. According to the PSNR results, octree gave better results in color reduction. Finally, the XGBOOST method showed better performance according to the accuracy rate metric in classifying histograms with various machine learning methods.

REFERENCES

1. Lloyd, S. P. (1957). Least square quantization in PCM. Bell Telephone Laboratories Paper. Published in journal much later: Lloyd, SP: Least squares quantization in PCM. *IEEE Trans. Inform. Theor.* (1957/1982), 18(11).
2. Forgy, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21, 768-769.
3. Lejeune Dirichlet, G. (1850). Over the reduction of the positive square shapes with three indefinite entire numbers . *Journal for the pure and applied Mathematics (Crelles Journal),* 1850(40), 209-227.
4. Voronoi, G. (1908). News apps of the settings continue with the theory of the shapes quadratic. Second memory. Research on _ parallelohedrons primitives . Newspaper for die queen und angewandte Mathematik *(Crelles Journal)*, 1908(134), 198-287.
5. Linde, Y., Buzo, A., & Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Transactions on communications*, 28(1), 84-95.
6. Gersho, A. (1982). On the structure of vector quantizers. *IEEE Transactions on Information Theory*, 28(2), 157-166.
7. Gray, R. (1984). Vector quantization. *IEEE Assp Magazine,* 1(2), 4-29.
8. Makhoul, J., Roucos, S., & Gish, H. (1985). Vector quantization in speech coding. Proceedings of the IEEE, 73(11), 1551-1588.
9. Meagher, D. (1982). Geometric modeling using octree encoding. Computer graphics and image processing, 19(2), 129-147.
10. Uhr, L. (1963). " Pattern recognition" computers as models for form perception. Psychological Bulletin, 60(1), 40.
11. Li, D., & Wu, M. (2021). Pattern recognition receptors in health and diseases. Signal transduction and targeted therapy, 6(1), 291.
12. Serey, J., Alfaro, M., Fuertes, G., Vargas, M., Durán, C., Ternero, R., ... & Sabattin, J. (2023). Pattern recognition and deep learning technologies, enablers of industry 4.0, and their role in engineering research. *Symmetry*, 15(2), 535.
13. Mantel, N., & Brown, C. (1974). Alternative tests for comparing normal distribution parameters based on logistic regression. *Biometrics*, 485-497.
14. Kononenko, I. (1989). ID3, sequential Bayes, naive Bayes and Bayesian neural networks. Proc. of European Working Session on Learning EWSL, 4-6.
15. Palmer, E. M., & Schwenk, A. J. (1979). On the number of trees in a random forest. *Journal of Combinatorial Theory, Series B,* 27(2), 109-121.
16. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.
17. Fukunaga, K., & Narendra, P. M. (1975). A branch and bound algorithm for computing k-nearest neighbors. IEEE transactions on computers, 100(7), 750-753.
18. Magee, J. F. (1964). Decision trees for decision making (pp. 35-48). Brighton, MA, USA: Harvard Business Review.
19. Vapnik, V., Golowich, S., & Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. Advances in neural information processing systems, 9.

Ufuk Tanyeri
Lecturer Dr. Ufuk Tanyeri is a full-time academic staff in the department of Computer Technologies at Ankara University. He received BA (2001) degrees of Computer Education, MA (2009) degrees of Computer Education in Institute of Information at Gazi University, and PhD (2015) degree of Industrial Technological Education in Natural and Applied Sciences at Gazi University. His research interests include Distance Education, Mobil Area Adhoc Network, and Image Processing.