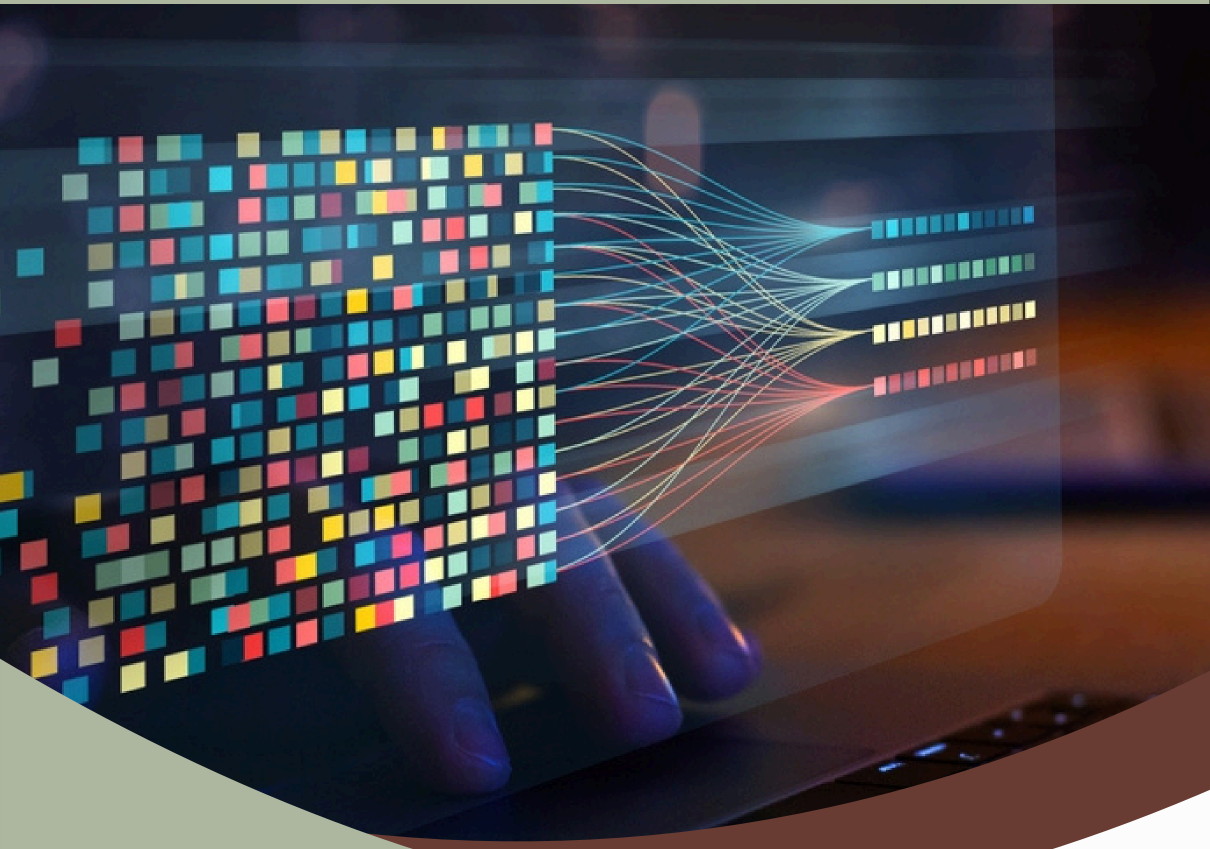
NEW VISIONS IN ENGINEERING:

CONCEPTS - THEORIES - APPLICATIONS



Editor: Prof. Dr. Meltem SARIOĞLU CEBECİ

DUVAR

NEW VISIONS IN ENGINEERING:

CONCEPTS - THEORIES - APPLICATIONS

Editor

Prof. Dr. Meltem SARIOĞLU CEBECİ

New Visions In Engineering: Concepts - Theories - Applications
Editor: Prof. Dr. Meltem SARIOĞLU CEBECİ

Editor in chief: Berkan Balpetek
Cover and Page Design: Duvar Design
Printing : October -2024
Publisher Certificate No: 49837
ISBN: 978-625-6183-31-5

853 Sokak No:13 P.10 Kemeraltı-Konak/İzmir
Tel: 0 232 484 88 68

www.duvaryayinlari.com
duvarkitabevi@gmail.com

TABLE OF CONTENTS

Chapter 1

A Sample Calculation Regarding the Design of Jointed and Unreinforced Concrete Pavement

Abdulrezzak BAKIŞ [1]

[1] Assoc. Dr.; Batman University, Faculty of Engineering and Architecture, Department of Civil Engineering. abdulrezzak .bakis@batman.edu.tr ORCID No: 0000-0002-7487-884X

INTRODUCTION

It has been observed that concrete road applications have been increasing in our country and many other countries in the world in recent years. As a result of the rapid decrease in oil resources in the world, high increases in oil prices have started to accelerate rigid pavement applications in road applications. (Vandenbossche et al. 2011; Sadek et al, 2014; Johanneck et al, 2011).Developed countries built a national road system by rapidly expanding their road networks in the post-second world war period and in the 1970s. Along with this process, many new technologies, design methods, and alternative engineering solutions were introduced to our country, and it also confirmed the leading and key position of the General Directorate of Highways for transportation infrastructure. In this context, the General Directorate of Highways has brought new technologies and technical developments to our country in many fields such as materials, projects, construction and inspection, and has realized its pioneering role by making trial roads on different routes by coming together with sector representatives on concrete roads as an alternative to asphalt pavements (Komut et al., 2019). (Romero et al. 2016; Li et al, 2010; Johanneck et al, 2011)

In our country, where heavy vehicle traffic and seasonal changes are high, concrete roads stand out as a very important option with their durability and longevity. For example, in America, a street that is over 125 years old has a concrete pavement, and some of the concrete roads made by the Germans before and after World War II are still in use. In addition, concrete roads increase road safety with reduced braking distances and road holding performance in rainy weather, while offering a human health and environmentally friendly solution with their fire resistance and emission-free structure. (Vandenbossche et al. 2011; Sadek et al, 2014; Romero et al. 2016; Li et al, 2010; Johanneck et al, 2011). Apart from this, concrete roads provide energy efficiency in terms of lighting due to their high road reflection coefficients. Lighting expenses are a constant economic expense for highways and require continuous payment of electricity bills after the initial operating cost. In this respect, the use of concrete roads, which have a higher reflection coefficient than asphalt, provides more lighting with less energy. (Kocaman and Rüstemli, 2019;Cengiz, 2024;Rüstemli and Cengiz, 2015;Cengiz et al;2018).

This situation shows that the costs will be much higher when considering the 24-hour energy consumption in tunnel lighting throughout the day. Because energy consumption in tunnel lighting continues day and night. In this respect, the use of concrete roads should be encouraged both on normal roads and tunnel roads. (Kocaman and Rüstemli, 2019;Rüstemli and Cengiz, 2015;Cengiz et al, 2018)

Since concrete road applications are a new study in our country, applications and calculations related to road design are very few. In this study, a sample calculation related to the design of a jointed and unreinforced concrete highway pavement has been made. It is expected that this study will help concrete pavement engineers.

In the example calculation, the length of a ring road is taken as 5 km, and it is planned to construct this ring road in 2023 as a non-reinforced concrete pavement with joints. Concrete road is a 2x3= 6-lane state road. The number of directions is taken as i= 2, the number of lanes in the same direction as j=3, the platform width in one direction is taken as W=12 m.

Project start year 2023, as traffic values; 2021 General Directorate of Highways State Roads Volume Map Annual Average Daily Traffic (AADT) values are considered as Truck + Trailer, Tractor + Semi-Trailer 679, Truck 1078, Bus 79, Medium load commercial vehicle 2221 and automobile 14851.

The projecting analysis period for the determined road was determined as 30 years and j= 2.7. The compressive strength class of the concrete material to be used for the superstructure was chosen as C 35/45.

CONCRETE ROAD PAVEMENT THICKNESS CALCULATIONS WITHOUT JOINT REINFORCEMENT

Annual increase percentages (g) were calculated using General Directorate of Highways-AADT values for 2021 and 2020. Annual increase percentages are shown in Table 1.

Table 1: Vehicle Annual Increase Percentages

VEHICLE CLASS	2020 AADT	2021 AADT	Percentage of Increase (g) (%)
Truck + Trailer	627	679	8
Truck	1077	1078	0.1
Bus	75	79	4.5
Medium load commercial vehicle	2173	2221	2.2
Automobile	14221	14851	4.5
TOTAL AADT	18173	18908	

The estimated traffic model covering the “Project Planning Analysis Period” for the route to be primarily designed can be obtained from the relevant department, including the vehicle types specified in Table 1. In other cases, the traffic increase coefficients (g) specified in Table 1 are calculated and included in the calculations.

The final service capability for highways is chosen as $P_t = 2.5$ from Table 2 (Komut *et al.*, 2019).

Table 2: Final Service Capabilities of Roads

Road Class	**P_t**
Highway, State Highway	2.5
Provincial Road	2.0

The daily traffic for the first year and the daily traffic for the end of the project are calculated with Equation (1) (Komut *et al.*, 2019):

$$t_t = t_0 \times (1+g)^t \quad (1)$$

Here;

t: Projected traffic analysis (or performance) period

t_t: Traffic analysis (or performance) period end year for traffic value (each one vehicle for different types calculated)

t_o: Daily traffic value in the first year of the traffic analysis period

g: Annual increase rate of traffic

Traffic for 2023;

Trailer for; $t_{2023} = 679*(1+0.08)^2 = 792$

Truck for; $t_{2023} = 1078*(1+0.001)^2 = 1080$

Bus for; $t_{2023} = 79*(1+0.045)^2 = 86$

For medium-load commercial vehicles for; $t_{2023} = 2221*(1+0.022)^2 = 2320$

For automobile; $t_{2023} = 14851*(1+0.045)^2 = 16218$

Daily traffic for the year 2053;

For trailer; $t_{2053} = 679*(1+0.08)^{32} = 7969$

For truck; $t_{2053} = 1078*(1+0.001)^{32} = 1113$

For the bus; $t_{2053} = 79*(1+0.045)^{32} = 323$

For medium-load commercial vehicles; $t_{2053} = 2221*(1+0.022)^{32} = 4456$

Automobile; $t_{2053} = 14851*(1+0.045)^{32} = 60740$

It is calculated as.

Project vehicle traffic (t_p) for each vehicle group is calculated with the help of Equation 2 (Komut *et al.*, 2019):

$$t_p = 0.4343*[(t_t - t_0) / \log(t_t - t_0)] \quad (2)$$

t_p with the help of Equation (2);

Trailer for; 3109

For truck; 1096

For the bus; 179

For medium-load commercial vehicles; 3273

For automobile; 33717

Values are obtained. Vehicle equivalence factors (TEF) are shown in Table 3 (Komut *et al.*, 2019):

Table 3: Vehicle Equivalence Factors

Traffic Group	Vehicle Equivalence Factor
Trailer	4.10
Truck	2.90
Bus	3.20
Medium Duty Commercial Vehicle	0.60
Automobile	0.0006

Lane distribution factors are shown in Table 4 (Komut *et al.*, 2019):

Table 4: Lane Distribution Factors

Number of Lanes in Two Directions	Lane Spread Factor (η)
2	1.00
4	0.90
6	0.80
8 or more	0.70

Since the number of lanes in both directions is 6, the lane distribution factor is taken as 0.80 from Table 4 (η= 0.80).

The number of repetitions (W_g) of the daily standard axle load falling on the calculation strip is calculated for each vehicle with the help of Equation 3 (Komut *et al.*, 2019):

$$W_g = t_p * TEF * \eta * 0.5 \quad (3)$$

Then $\sum W_g$ is found as the grand total.

Trailer: 3109 * 4.10 * 0.8 * 0.5= 5099
Truck: 1096 * 2.90 * 0.8 * 0.5= 1275
Bus: 179 * 3.20 * 0.8 * 0.5= 229
Medium-load commercial vehicle: 3273 * 0.6 * 0.8 * 0.5= 785
Automobile: 33717 * 0.0006 * 0.8 * 0.5= 8
$\sum W_g$ (Daily Total) = 7396

Total Number of Standard Axle Repetitions is calculated with the help of Equation 4 (Komut *et al.*, 2019):

$$T_{8.2} = \sum W_g * 365 * t \quad (4)$$

From Equation (4); T is calculated as $T_{8.2}$ = 7396*365*30= 80986200. Since the compressive strength class of the concrete material to be used for the pavement is C 35/45, the Elastic Modulus and bending tensile strength of the concrete are determined with the help of Table 5. From Table 5, Modulus of Elasticity (E_c) is 4786244 psi and flexural tensile strength (S'c) is 660 psi.

Table 5: S'c and E_c Values Depending on Concrete Classes

Concrete Grade	Characteristic Cylinder compressive strength, (fck), (MPa)	Characteristic Cube compressive strength, (fck) (MPa)	Tensile strength at average bending according to TS 500 (MPa)	Approx. S'c to TS500 (psi)	Approx. S'c according to ASHTO (psi)	Average S'c (psi)	E_c (MPa)	E_c (psi)
C30/37	30	37	3.83	556	688	622	32000	4641206
C35/45	35	45	4.20	610	709	660	33000	4786244
C40/50	40	50	4.43	643	731	687	4000	4931281

The road is a highway; the Confidence rate (R) is 95% from Table 6 (Komut *et al.*, 2019).

Table 6: AASHTO 93 Guidelines Recommended Confidence Level (R), % Ratios

Road Type	**(R), %**
Highways and State Roads	80–99.9
Provincial Roads	75–95
Collector Paths	75–95
Local Roads or Secondary Roads	50–80

The (R) values used in practice are 95% for highways and state roads, and 85% for provincial roads. The standard normal deviation value is found as -1,645 from Table 7 (Komut *et al.*, 2019).

Table 7: Standard Deviation Values Based on Confidence Ratios in AASHTO 93 Guidelines

Confidence Ratios (R) %	**Standard Normal Deviation (Z_R)**
50	0.000
60	-0.253
70	-0.524
75	-0.674
80	-0.841
85	-1.037
90	-1.282
91	-1.340

92	-1.405
93	-1.476
94	-1.555
95	-1.645
96	-1.751
97	-1.881
98	-2.054
99	-2.327
99.9	-3.090
99.99	-3.750

Total standard deviation (S_o) value of 0.35 on average for concrete pavements is taken.

P_o values from the AASHTO trial path were 4.5 for rigid pavements. The final service capability value is taken from Table 1.

Serviceability change (ΔPSI) is calculated with the help of Equation (5) (Komut *et al.*, 2019):

$$\Delta PSI = P_o - P_t \quad (5)$$

$\Delta PSI = P_o - P_t = 4.5 - 2.5 = 2.0$

The load transfer coefficient is taken as J= 2.7. The load transfer coefficients according to the design conditions are shown in Table 7 (Komut *et al.*, 2019).

Table 7: Load Transfer Coefficients According to Project Design Conditions

Clearway	**Asphalt**		**Concrete**	
Load transfer	Available	Not available	Available	Not available
Pavement Type				
Jointed and Reinforced Jointed	3.2	3.8–4.4	2.5–3.1	3.6–4.2
Continuously Reinforced Concrete	2.9–3.2	-	2.3–2.9	-

Recommended drainage coefficient for the concrete pavement (C_d) is shown in Table 8 (Komut *et al.*, 2019).

Table 8: Recommended Drainage Coefficient for Concrete Pavement

Drainage Quality	Percentage of Time the Pavement is Exposed to Moisture Levels Approaching Saturation			
	<1%	1–5%	5–25%	>25%
Very good	1.25 – 1.20	1.20 – 1.15	1.15 – 1.10	1.10
Good	1.20 – 1.15	1.15 – 1.10	1.10 – 1.00	1.00
Middle	1.15 – 1.10	1.10 – 1.00	1.00 – 0.90	0.90
Bad	1.10 – 1.00	1.00 – 0.90	0.90 – 0.80	0.80
Too bad	1.00 – 0.90	0.90 – 0.80	0.80 – 0.70	0.70

C_d= 1 value recommended in adhering to AASHTO 93. Considering the loss of support that may occur in the material under the concrete slab layer, it is recommended to take the Subgrade reaction module (k) value of 350 pci in slab calculations.

Subsoil reaction modulus values are shown in Table 9 (Komut *et al.*, 2019).

Table 9: Subsoil Reaction Module Values

Floor Type	k (pci)
Plastic clay	50–100
Silt and Silty Clay,	100–200
Sand, clay gravel	200–300
Gravel	300–350
Crushed Stone (Base and Sub base)	350–450

Concrete pavement thickness was found to be 33 cm from Table 10.

Table 10: Recommended concrete pavement thicknesses for R= 95% and Concrete Class C 35/45 (Komut *et al.*, 2019)

Traffic Categories (x1000)		Concrete Pavement Thickness (cm)
0	4 000	20
4 000	5 500	21
5 500	7 500	22
7 500	9 500	23
9 500	12 000	24
12 000	15 500	25
15 500	20 000	26
20 000	25 000	27
25 000	31 000	28

31 000	39 500	29
39 500	49 000	30
49 000	60 000	31
60 000	74 000	32
74 000	90 000	33
90 000	110 000	34
110 000	132 000	35
132 000	159 000	36

SLIP AND CONNECTION IRONS CALCULATION

Slip Iron Calculations

Widespread in applications aspect 50 cm (18–20 inch) in length and 32–38 mm (1 .25 – 1.5 inch) diameter dowel bars are used. The transverse distance between the dowel bars is 30 cm. For coating thicknesses up to 28 cm, dowel iron with a diameter of 32 mm, dowel iron with a diameter of 38 mm for coating thicknesses of 28–34 cm, and dowel iron with a diameter of 44 mm for pavement thicknesses of 35–40 cm are preferred. Dowel bars should be flat iron painted with anti-corrosion paint and lubricated. 30 mm diameter dowel bars are preferred for low pavement thicknesses (Komut *et al.*, 2019).

- L= 50 cm long, (Ø 38) 38 mm diameter flat iron should be placed at 30 cm intervals.
- Non-reinforced concrete pavement, the joint spacing (in feet) should not exceed 2 times the pavement thickness (in inches). For example; for 8-inch (20.32 cm) thick pavement, the maximum joint spacing should be 16 feet (488 cm). Also, as a general guideline, the ratio of pavement length to width should not exceed 1.25. Transverse joint spacing is preferred 4.6–6 m on concrete roads with joints.
- In high slab thicknesses, up to 6 meters of transverse joint spacing can be given depending on the longitudinal joint spacing.
- Up to 15 meters between transverse joints on reinforced concrete roads can be given. Transverse joint ranges it varies between 4.6–6 m. In general, it can be given as 4.6–5 m. Generally, longitudinal joints are used at intervals of approximately 3.0–4.2 m.
- Transverse joint spacing, longitudinal joint spacing; Considering that there are three strips in one direction, considering the clearway, inner clearway and strip widths for the longitudinal joint intervals, a longitudinal joint at 4 m and a transverse joint at 5 m can be left.

Connection irons

Today, the most common use is to use ribbed ties with a length of 80 cm (31.5 inches) and a diameter of 12–14 mm (0.5 inches). For pavement thicknesses up to 30 cm, 12–14 mm diameter tie bars are preferred, for pavement thicknesses over 30 cm, 16–20 mm diameter tie bars are preferred. The distance between the tie bars should be adjusted to be 80 cm. L= 80 cm long, (Ø16) 16 mm diameter ribbed iron should be placed at 80 cm intervals (Komut *et al.*, 2019).

CONCLUSION

Since concrete road applications are a new study, applications and calculations related to road design are very few. In this study, a sample calculation related to the design of a jointed and unreinforced concrete highway pavement has been made. In this study, brief information is given about how the concrete road pavement thickness calculation is made, and how the slip and connection iron applications are made. It is expected that this study will help concrete pavement engineers.

REFERENCES

AASHTO Guide for Design of Pavement Structures, (1993). https://habib00ugm.wordpress.com/wp-content/uploads/2010/05/aashto1993.pdf

Beton Yol Kaplamaları Teknik Şartnamesi, Karayolları Genel Müdürlüğü Araştırma ve Geliştirme Dairesi Başkanlığı, (2016).

Bustos M., Cordo O., Girardi P., Pereyra M., Calibration of Distress Models from the Mechanistic–Empirical Pavement Design Guide for Rigid Pavement Design in Argentina, Transportation Research Record: Journal of Transportation Research Board, 2226 (1), 3-12, 2011.

Cengiz, C. (2024). The Relationship Between Electricity Consumption From Outdoor Lighting and Economic Growth, Light & Engineering Vol. 32, No. 4, pp. 14–21, pp. 61–67, https://doi.org/10.33383/2023-032

Cengiz M.S., Cengiz, C. (2018). Numerical analysis of tunnel led lighting maintenance factor, IIUM Engineering Journal, 19 (2018), pp. 154-163, doi:10.31436/iiumej.v19i2.1007

Johanneck L., Tompkins D., Clyne T., Khazanovich L., 2011. Minnesota Road Research Data for Evaluation and Local Calibration of the Mechanistic-Empirical Pavement Design Guide's Enhanced Integrated Climatic Model, Transportation Research Record: Journal of Transportation Research Board, 2226 (4), 30-40.

Kaşak, S., Komut, M., (2019). Karayolları Beton Yol Üstyapılar Projelendirme Rehberi, Karayolları Genel Müdürlüğü Araştırma ve Geliştirme Dairesi Başkanlığı.

Kocaman B., Rustemli S., (2019). Comparison of LED and HPS Luminaires in Terms of Energy Savings at Tunnel Illumination, Light & Engineering Vol. 27, No. 3, pp: 67–74, 2019

Komut, M., Ünal, E. N., Altıok, Ş., (2019). Karayolları Genel Müdürlüğü Ağındaki Beton Deneme Yolların Performans Değerlendirmesi, 1.Beton Yollar Kongresi ve Sergisi, 13–14 Kasım 2019 KGM Halil Rıfat Paşa Konferans Salonu, Ankara.

Li J., Uhlmeyer J., Mahoney J., Muench S., 2010). Updating the Pavement Design Catalog for the Washington State Department of Transportation: Using 1993 AASHTO Guide, Mechanistic-Empirical Pavement Design Guide, and Historical Performance, Transportation Research Record: Journal of Transportation Research Board, 2154 (12), 124-129.

Romero M.J.M.E., Garro N.M.T., Zevallos G.G., 2016). Implementation of the mechanistic–empirical pavement design in northern Peru using a calibration coefficient for the International Roughness Index, Constr Build Mater, 102 (1), 270-280.

Rüstemli S., Cengiz, M.S., 2014). The relationship between height and efficiency and solution offerings in tunnel and subsea tunnels, Light and Engineering, vol 22, No.2, pp. 76- 83.

Sadek H.A., Masad E.A., Sirin O., Al-Khalid H., Sadeq M.A., Little D., 2014. Implementation of mechanistic-empirical pavement analysis in the State of Qatar, Int J Pavement Eng, 15 (6), 495-511.

Vandenbossche J.M., Mu F., Burnham T.R., 2011). Comparison of measured vs. predicted performance of jointed plain concrete pavements using the Mechanistic–Empirical Pavement Design Guideline. Int J Pavement Eng, 12 (3), 239-251.

Chapter 2

Multi-Storey Highway Transportation Model

Abdulrezzak BAKIŞ[1]

[1] Assoc. Dr.; Batman University, Faculty of Engineering and Architecture, Department of Civil Engineering. abdulrezzak .bakis@batman.edu.tr ORCID No: 0000-0002-7487-884X

INTRODUCTION

Rapid and unplanned urbanization, population explosions and the increase in the number of fast vehicles bring along important traffic problems. The most important of them are; increase in vehicle fuel consumption, environmental pollution, accidents, traffic jam and transportation difficulties.

Increasing population in metropolitan areas; along with housing, education, transportation, etc. gives rise to many need parameters. With the developed mathematical-predictive models, road networks should be created in urban-scale settlement areas, solutions to traffic problems should be proposed, traffic flow should be ensured and safe transportation of pedestrians should be ensured (Cengiz, 2019;Karakaş, 2023).Motor vehicle ownership is accelerating as a result of factors such as developments in the automotive industry, sales methods that facilitate automobile acquisition, and urban population growth. As a result, urban transportation difficulties and traffic congestion reduce the quality of life. Effective mobility in the urban area is an inevitable need (Erçetin, 2023).

One of the important factors in the country's economy is uninterrupted transportation. Traffic congestion creates economic losses for the region and the country. Traffic congestion increases economic losses and maximizes energy losses, negatively affecting the country's level of development (Cengiz,2024). In addition, due to the transportation infrastructure that cannot meet the increasing number of vehicles, negative factors such as traffic accidents and environmental pollution affect the country's economy negatively. With the advancement of technology and the rapid increase in the population, the existing infrastructure becomes insufficient and this issue necessitates new investments (Kaşka, 2019;Sony et al, 2021;Zamboroni et al, 2020).

Pedestrian transportation is a form of transportation that is considered more flexible in terms of security compared to other transportation systems. The main purpose of pedestrian transportation is to be safe. While providing security; it is very important to separate pedestrian and motor vehicle traffic, and to take precautions at the intersections of pedestrian and motorized transportation axes (Paşalı, 2023;Tran et al, 2020).

In urban transportation, it should be essential to make and implement a planning that has low cost to the country and its individuals, has a high contribution to economic and social development, and that will positively affect the urban development process. Traffic congestion is similar to unwanted disturbing effects (harmonics) in electrical systems. Harmonics in electrical systems represent the effect of traffic congestion instead of a normal flow and cause losses and malfunctions (such as traffic accidents). While harmonics in electrical systems can be eliminated with passive and active filtering, traffic congestion can be eliminated with new road alternatives. (Rüstemli et al. 2015; Rüstemli et al. 2016). First of all, it is important to prepare transportation plans that value people, not vehicles, that allow efficient and effective use of own resources, that use the existing transportation capacity in the most effective way and that do not harm urban historical textures (Bakış and Işık, 2012).

In this study, suggestions for the solution of the road transportation problem in Batman province are presented and the traffic flows in the future and the measures to be taken to eliminate the transportation problems that will arise are considered. In the study, an alternative method called “Multi-Storey Highway Model” is presented for the elimination of urban traffic congestion in the city center of Batman province.

The increase in the population of Batman city center with the migration from the village to the city and at the same time the D370 and D955 state highways passing through the city center create density in vehicle traffic. Crossing routes of D370 and D955 state highways on Batman Province are shown in Figure 1.

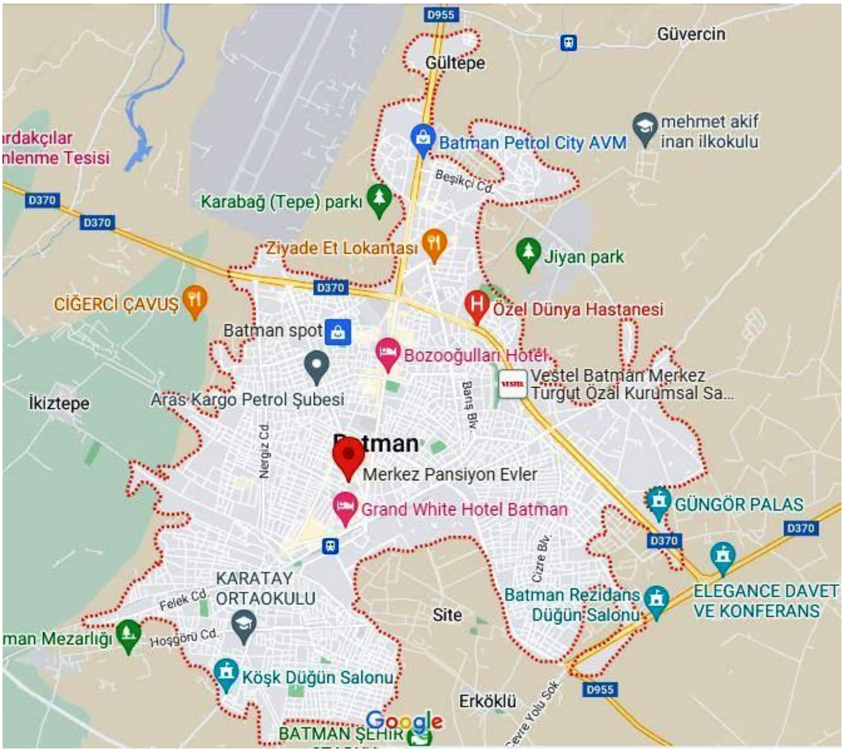


Figure 1. Transit routes of D370 and D955 state highways on Batman Province

Difficulties are encountered in opening alternative roads in Batman city center. This situation affects the city traffic negatively. As a result of not establishing new parking lots with the capacity required by the rapidly increasing number of motor vehicles in the province, random vehicle settlements on the roadsides prevent traffic flow. Infrastructures such as roads, intersection facilities and parking areas for transportation are insufficient in the province. In many places, indiscriminate on-street parking disrupts the continuity of the traffic flow. Traffic jams are also dangerous for pedestrians.

In Batman province causes traffic problems in the central region and it is difficult to solve the traffic flow and parking problems. As a result of the increase in small-capacity transportation vehicles, significant increases occur in traffic congestion. Urbanization in the province is generally close to the D370 and D955 state highways. The mountainous and rugged terrain towards Diyarbakır has reduced the development of urbanization in this direction. For this reason, urbanization has mainly developed towards Batman-Kozluk, Batman-Hasankeyf and Batman-Beşiri district directions. After the divided road works carried out throughout the country in recent years, both the connection of the region with neighboring provinces and the transportation connection within itself have become more qualified. Although Batman city center has a divided road, the traffic congestion caused by the increase in the number of vehicles continues. In the study, an alternative method named "Storey Highway Model" is presented for the elimination of urban traffic congestion in the city center of Batman province. This application is based on the principle of opening new roads to the upper floors of existing roads instead of opening new roads. Traffic congestion can be eliminated in two simple ways:

- Horizontally opening new lane addition roads (road widening) or new roads built on completely different infrastructures.
- Construction of multi-storey highways on existing roads.

In case 1, new expropriation costs are required. In the second case, there is no expropriation cost. In case 1, new public spaces will need to be created for new roads. In this way, people's natural habitats will be destroyed. In the second case, this is not the case. In case 1, new road traffic signaling costs, level crossing costs and similar costs will arise. In the second case, such expenses will not be incurred. When comparisons are made, it should be expected that the roads to be built in the second case will benefit the country's economy.

FLOOR HIGHWAY MODEL COST CALCULATION

The multi-storey highway is designed as the existing asphalt pavement on the lower floor and as a concrete road pavement on the upper floor. Calculations are based on 50 m length. For concrete flooring, when the length is 50 m, the width is 17 m and the concrete slab thickness is 20 cm, a concrete volume of 170 m^3 is obtained. A total of 40 columns will be placed, 10 each to the right of the slab, 10 to the far left, and 10 each to the right and left of the middle pavement (refuge). Calculations were made so that the column cross-sections were 0.5x0.5 m and the length was 5 m. As a result of the 0.5x0.5x5x40 process, 50 m^3 concrete volumes are formed for the columns. As a result of 17x0.4x0.3x10 treatment for 10 short beams, a concrete volume of 20.4 m^3 is formed. For 4 long beams, a total of 24 m^3 concrete volume of 50x0.4x0.3x4 is formed. With the 1x1x0.5x40 process, 20 m^3 of concrete is required for column footings. A total of 284.4 m^3 of concrete will be needed. Considering the construction unit prices of the Ministry of Environment and Urbanization in 2023, the unit price of concrete mortar (min. C30/37 at the desired compressive strength) for concrete pavements compressed with a cylinder numbered 10.130.1571 is 1,250.00 TL per m^3 concrete. The concrete cost for the concrete pavement to be installed on the upper floor in 50 m length is 355,500.00 TL.

The reinforcement to be used is Ø10/20. The approximate amount of reinforcement is 16 tons. The unit price of the Ministry of Environment and Urbanization is 12.35 TL/kg for the year 2023, the Concrete steel bar with Item No. 10.130.1704, the Ø8–12 mm ribbed reinforcement. The total reinforcement cost for the concrete pavement to be installed on the upper floor of 50 m length is 197,600.00 TL.

Approximately 850 m^2 of floors, 400 m^2 of columns, 187 m^2 of short beams, 220 m^2 of long beams, 80 m^2 for footwells; A total of 1737 m^2 of formwork is required. The mold unit price of the Ministry of Environment and Urbanization for the year 2023, Item No. 15.180.1002 (making flat surface concrete and reinforced concrete formwork from wood) is 238.03 TL/m^2. The cost of 1737 m^2 formwork for the concrete pavement to be installed on the upper floor of 50 m length is 413,458.00 TL in total.

The pier to be built is 670 m^3 in total. The unit price of the scaffolding of the Ministry of Environment and Urbanization for the year 2023, Item No. 15.185.1006 (construction of formwork scaffolding from steel pipes-between 4.01–6.00 m) is 41.10 TL/m^3. The cost of 670 m^3 scaffolding for a 50 m long concrete pavement to be installed on the upper floor is 27.537,00 TL in total.

Concrete cost is 355,500.00 TL, reinforcement cost is 197,600.00 TL, formwork cost is 413,458.00 TL and scaffolding cost is 27,537.00 TL; When all cost items are added together, the total cost of the multi-storey highway for the 50 m long concrete pavement to be installed on the upper floor is calculated as 994.095,00 TL.

MULTI-STOREY HIGHWAY TRANSPORTATION MODEL VIEW

In Figure 2, the top view of the multi-storey road transportation model is given.

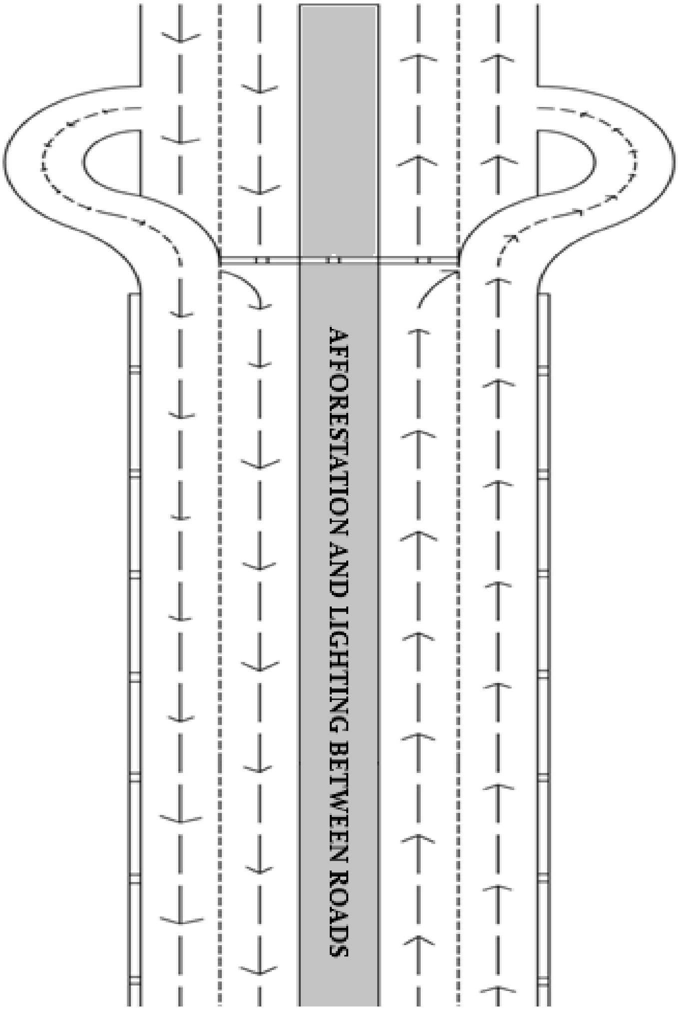


Figure 2. Multi-Storey Highway Transportation Model Top View

As seen in Figure 2, a 2nd floor duplex road was built on an existing duplex road. Without the need for any expropriation process, uninterrupted vehicle passage is provided on the upper floor. The number of floors can be increased according to the traffic density.

RESULTS

The following results will be achieved with the implementation of this project:

- With this project, no other land expropriation process will be carried out for road construction. A second concrete road pavement will be built on the existing asphalt road pavement.
- With the use of the upper floor of the road, traffic congestion will not occur and continuity will be ensured in transportation.
- Traffic flow will be safer and faster.
- Vehicle-Vehicle, Vehicle-Pedestrian accident risk will be reduced.
- Transportation infrastructure costs will decrease.
- No new roads will be built on new lands. In this way, the natural habitats of living things will not be destroyed.
- Since no new roads will be built on the new lands, there will be no traffic signaling costs, level crossing costs and similar costs.
- Since new roads will not be built on new lands, it will contribute to the country's economy.

REFERENCES

Bakış, A., Işık, E., (2012). Bitlis ulaşım analizi ve ulaşım sorunları çözüm önerileri. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 1(1), 19–33.

Cengiz, C. (2019). Nonparametric estimation of a renewal function in the case of censored sample, Bitlis Eren University Journal of Science and Technology, vol. 9, no. 2, pp. 54–57, 2019, doi: 10.17678/beuscitech.556451.

Cengiz, C. (2024). The Relationship Between Electricity Consumption From Outdoor Lighting and Economic Growth, Light & Engineering Vol. 32, No. 4, pp. 14–21, pp. 61–67, https://doi.org/10.33383/2023-032

Erçetin, C. (2023). Çevresel Sürdürülebilirlik ve Ulaşım Aracı Olarak Bisiklet: İyi Uygulama Örnekleri . Çevre Şehir ve İklim Dergisi , 2 (3) , 238–269 . Retrieved from https://dergipark.org.tr/en/pub/csid/issue/75639/1206512

Karakaş, A. S. (2023). Üniversite Yerleşkesi Çevresinde Trafik Durumunun İncelenmesi ve Alternatif Çözüm Önerileri: İstanbul Üniversitesi Cerrahpaşa Avcılar Kampüsü Örneği. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 14(1), 175–189.

Kaşka, M. S. (2019). Yap işlet devret modeli ile yapılan köprü ve tüneller (Master's thesis, İstanbul Gelişim Üniversitesi Fen Bilimleri Enstitüsü).

Paşalı, E. (2023). Edirne tarihi kent merkezinin yaya konforu açısından irdelenmesi (Master's thesis, Trakya Üniversitesi Fen Bilimleri Enstitüsü).

Rüstemli, S., Cengiz, M. S. (2016). Passive filter solutions and simulation performance in industrial plants, Bitlis Eren Univ. J. Sci. Technol., 6 (1) pp. 39-43. https://doi.org/10.17678/beujst.94574

Rüstemli, S., Cengiz, M. S. (2015). Active filter solutions in energy systems, Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 23: No. 6, Article 6. https://doi.org/10.3906/elk-1402-212

Sony, B., Rao, C. H., (2021). Review and analysis of roadway crash prediction studies on urban roads under heterogeneous traffic conditions. In IOP Conference Series: Materials Science and Engineering, 1197, 1.

Tran, M. T., Nguyen, T. V., Hoang, T. H., Le, T. N., Nguyen, K. T., Dinh, D. T., Do, M. N., (2020). iTASK-Intelligent traffic analysis software kit. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 612-613.

Zambrano-Martinez, J. L., Calafate, C. T., Soler, D., Cano, J. C., & Manzoni, P. (2020). Modeling and characterization of traffic flows in urban environments. Sensors, 18, 7.

Chapter 3

Quality Metrics and Analysis for Successful Software Project Management

Akın ÖZÇİFT[1]
Fatih YÜCALAR[2]

[1] Prof. Dr.; Manisa Celal Bayar University, Hasan Ferdi Turgutlu Technology Faculty, Department of Software Engineering. akin.ozcift@cbu.edu.tr; ORCID No: 0000-0002-5317-5678
[2] Assoc. Prof. Dr.; Manisa Celal Bayar University, Hasan Ferdi Turgutlu Technology Faculty, Department of Software Engineering. fatih.yucalar@cbu.edu.tr; ORCID No: 0000-0002-1006-2227

Abstract

The primary goal of software engineering is to produce high-quality software that meets customer requirements, is completed on time, within the specified budget, is suitable for the intended use, and adheres to standards, all while maintaining an acceptable level of errors. Software quality determines how well a software product meets customer requirements and fulfills expected performance criteria. The evaluation of software quality is conducted using various criteria, such as functionality, reliability, usability, performance, and maintainability. Functionality measures how well the software adheres to specified requirements and functional specifications. It includes the ability to operate correctly and completely and evaluates aspects like producing accurate results, error handling, and data processing. Reliability is defined as the software's fault tolerance and the system's uptime without failure. High reliability ensures that the software operates securely and stably over a long period. Reliability tests measure how stable the software is under various conditions. Usability assesses whether the software is user-friendly, easy to learn, and easy to use. It encompasses the effort and learning time required for users to effectively use the software. Performance measures the efficiency and response times of the software. Performance criteria evaluate the effective use of system resources, processing speed, and data handling capacity. Maintainability refers to the software's upgradability and sustainability. Lastly, maintainability includes the flexibility of the software in the process of fixing bugs, making improvements, and adding new features. Each of these criteria is an important factor affecting the quality of software products, and they ensure the suitability of the products for user requirements by being considered in software engineering processes.

Keywords: Software Quality Metrics, Functionality, Reliability, Usability, Performance, Maintenance.

INTRODUCTION

In software engineering, quality is one of the fundamental elements of a successful product development process. The evaluation of software quality includes not only meeting the functional requirements of the product but also important factors such as user satisfaction and system reliability. Software quality metrics are the criteria used to determine how effectively software products fulfill these requirements. These metrics play a critical role in improving quality in software development processes and providing more satisfying experiences for end-users.

In 1990, the IEEE provided a definition related to software quality metrics. In this definition, software quality metrics are described as the quantitative measurement of the degree to which a software component possesses a certain quality attribute. The basic objectives concerning software quality metrics for software quality and other software engineering areas have been clearly and explicitly defined. By achieving these basic objectives, management control over the software project is ensured, and measurement data aimed at improving development and maintenance processes are obtained. In this section, the quality metrics necessary for tracking and measuring the success of a software project will be addressed and analyzed.

SOFTWARE QUALITY METRICS AND ANALYSIS

In modern software projects, quality assurance is as important as the functionality of the software itself. In this regard, software quality metrics are tools that help assess the software's compliance with specific quality standards. These metrics contribute to optimizing both the internal and external quality of the software. In Figure 1, five key metrics that impact software quality are shown. These metrics are functionality, reliability, usability, performance, and maintainability.

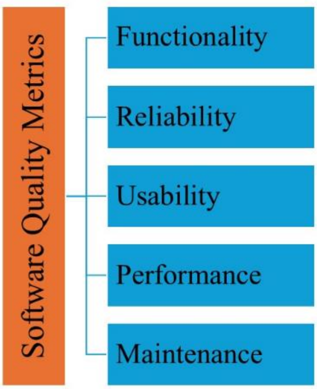


Figure 1. Software Quality Metrics

Functionality is defined as the software's ability to meet specific requirements. This metric evaluates how accurately and completely the software performs the designated functions (IEEE, 1998). Reliability refers to the software's uptime without errors and its fault tolerance. The reliability of software is assessed based on the frequency and impact of system failures (Pressman & Maxim, 2014). Usability indicates how easily and efficiently the software can be used by users, directly affecting the quality of the user experience (Nielsen, 1993). Performance encompasses performance characteristics such as speed, response time, and efficiency, measuring the overall efficiency of the system (Boehm, 1981). Maintainability refers to the software's responsiveness to future changes, updates, and bug fixes. The maintenance process is critical to ensuring that the software remains long-lasting and sustainable (Boehm, 1984).

In this section, a detailed examination of these five key metrics will be conducted, focusing on the impact of each metric on software quality and how these metrics are assessed.

Functionality

Software quality is a frequently discussed and carefully studied topic in the field of software engineering. Quality encompasses many aspects of software, one of which is known as functionality. Functionality refers to the software's success in meeting specified requirements, and it is one of the fundamental components of software quality (ISO/IEC, 2011). Understanding the impact of functionality on software quality plays a critical role in developing effective and reliable software products.

Functionality refers to the software's capacity to perform specific functions and typically evaluates how successfully the software meets its functional requirements. This is directly related to the degree to which the software satisfies user needs and is a key metric of software quality (Pressman & Maxim, 2014). Functionality is of great importance during the design and development phases to ensure that the software operates correctly and completely.

Software functionality determines its ability to meet user expectations and requirements. Software that provides functions aligned with user needs is generally associated with higher user satisfaction and lower error rates (Sommerville, 2011). In this context, functionality is considered a fundamental component of software quality because meeting functional requirements directly impacts the software's overall quality.

The impact of functionality on software quality can be examined from various perspectives. Firstly, functional deficiencies can make the software difficult to use and lead to user dissatisfaction. Functionality ensures that all functions necessary for users to effectively use the software are provided (ISO/IEC, 2011). This includes ensuring that the software operates without errors and performs the specified functions. Another important factor is the impact of functionality on software testing processes. Functional testing is conducted to determine whether the software meets the specified functional requirements. These tests assess the software's functional accuracy and reliability and help identify functional defects (Pfleeger, 2010). Through functional testing, existing errors in the software are detected and corrected early, which enhances the overall quality of the software.

Managing and improving functionality plays a significant role in software development processes. Software engineers must accurately gather user requirements and design solutions that meet these requirements to ensure functionality (Pressman & Maxim, 2014). Additionally, continuous feedback and testing processes are necessary to improve functionality.

In conclusion, functionality is a critical component of software quality and determines the software's success in meeting user requirements. Meeting functional requirements enhances the overall quality of the software and user satisfaction, while deficiencies and errors can negatively impact the software's success. Therefore, managing and improving functionality should be a primary focus in software development processes.

Reliability

Software quality encompasses many factors that determine the success of a software product, and reliability is one of these factors. Reliability refers to the software's capacity to operate without errors over a specific period and under certain conditions (ISO/IEC, 2011). Reliability plays a critical role in ensuring the software's dependable and stable performance and is a significant component of overall software quality.

Reliability indicates the software's ability to operate error-free under a range of conditions and is often assessed through error rates and system outages. A reliable software is designed to meet user expectations and maintain business processes without interruptions (Pressman & Maxim, 2014). In this context, reliability is considered a determining factor of software quality because the software's reliability directly impacts user satisfaction and system performance.

Software reliability is an important criterion for evaluating the long-term performance of a system. Reliability deficiencies can lead to frequent software failures, system crashes, and negative impacts on user experience (Sommerville, 2011). Therefore, ensuring and improving reliability should be a primary goal in software development processes.

The impact of reliability on software quality can be examined from various perspectives. Firstly, the reliability of software determines how much users can trust the system. Reliable software provides users with assurance that the software will operate smoothly and without interruptions (Pfleeger, 2010). This increases user satisfaction and positively affects the overall success of the system.

Ensuring reliability requires careful testing and quality control methods during the software development process. Reliability testing evaluates how durable the software is under various conditions and helps in the early detection of potential defects (ISO/IEC, 2011). Such tests are necessary to enhance the software’s reliability and optimize its long-term performance.

Managing and improving reliability requires continuous effort in software development processes. To ensure reliability, software engineers should conduct risk analysis and identify potential failure points. Additionally, it is important to perform reliability tests regularly and monitor the software's performance (Pressman & Maxim, 2014).

Reliability affects not only the technical aspects of software but also the user experience. Using reliable software ensures that business processes continue without interruption and enhances the overall quality of the software. Therefore, managing and improving reliability should be at the core of software development processes.

In conclusion, reliability is a critical component of software quality and refers to the software's capacity to operate error-free and without interruption. Deficiencies in reliability can negatively impact user satisfaction and reduce system performance. Therefore, ensuring and improving reliability should be a primary goal in software development processes.

Usability

There are many criteria used to assess software quality, and usability holds an important place among these criteria. Usability refers to how easily and efficiently a software can be used by its users (Nielsen, 1994). The usability of software not only enhances user satisfaction but also stands out as a factor that determines the overall quality of the software.

Usability determines how effectively and efficiently a software can be used by its users. This encompasses many elements, from user interface design to user experience (Dix et al., 2004). A usable software allows users to achieve their goals quickly and effectively, which positively impacts the overall quality of the software.

Usability affects how software is perceived and used by users. Usability issues can lead to difficulties for users when interacting with the software and may decrease productivity (Shneiderman & Plaisant, 2010). Therefore, paying attention to usability in software development processes is crucial for enhancing software quality and increasing user satisfaction.

Usability impacts many aspects of software quality. A user-friendly interface makes the software easier and more efficient to use, which, in turn, boosts user satisfaction (Gould et al., 1987). Additionally, good usability design shortens the learning curve for users and reduces the need for support services.

Usability is directly related to the functionality of the software. No matter how high the functionality of the software, if the software is not usable, users may not fully benefit from these functions. This situation negatively impacts the overall quality of the software (Nielsen & Budiu, 2012). Usability tests evaluate the user experience of the software and provide the opportunity to identify and correct potential issues early on.

Ensuring and improving usability requires continuous effort throughout the software development process. User interface design, user feedback, and usability testing are key components of this process (Norman, 2013). Regularly collecting and analyzing user feedback helps identify the necessary changes to enhance the software’s usability.

Usability improvements enhance the user experience and overall quality of software. A good usability design enables the software to be used more effectively by users, which positively impacts the success of the software. Therefore, focusing on usability in software development processes is key to increasing software quality.

In conclusion, usability is a critical component of software quality and directly affects the user experience. Improving usability enhances the overall quality of the software and ensures user satisfaction. Hence, paying attention to usability and making necessary improvements in software development processes is of great importance.

Performance

Software quality refers to the ability of software systems to meet user needs and adhere to performance standards. Performance is a critical dimension of software quality and is directly related to the efficiency, response time, resource usage, and scalability of software systems. Understanding the impact of software performance on quality can help optimize software development processes.

Performance is a characteristic that measures how quickly and effectively a software performs a specific task or function. It is typically evaluated based on factors such as processing time, response time, system resource usage, and processing capacity (Cunningham & Gil, 2019). High-performance software addresses user needs more quickly and improves the overall user experience. This increases user satisfaction and positively affects the overall quality of the software.

The performance of software directly affects the experience users have while using it. Slow response times or high resource consumption can negatively impact users' experience with the software, which can, in turn, decrease the software's quality (Smith & Williams, 2002). Performance issues can affect the software's reliability and efficiency, making performance improvements a critical part of the software development process.

Performance influences software quality in many ways. Performance metrics such as processing time and response time determine how users perceive the speed and effectiveness of the software (Jiang et al., 2017). High-performance software allows users to complete their tasks more quickly, which enhances user satisfaction. Conversely, low-performance software can lead to users spending more time and potentially causing disruptions in business processes.

Performance is also related to the scalability of software. Software systems must be able to handle an increasing number of users or data loads effectively. Performance issues can affect the scalability of the software, leading to problems when working with large-scale users or data (Kerr & Arnold, 2018). Therefore, optimizing the software's performance is crucial for its long-term success.

Managing and improving performance requires ongoing effort in the software development process. Performance testing and analysis are used to evaluate the software's performance and identify potential issues (Eich et al., 2019). Performance tests determine how the software performs under various conditions and help implement the necessary changes for performance improvements.

Performance improvements enhance the overall quality of software and improve the user experience. Effective performance management increases the efficiency of the software and enables users to utilize the software more effectively. Therefore, focusing on performance in the software development process is key to improving software quality.

In conclusion, performance is a critical component of software quality and directly affects the user experience. Improving performance boosts the overall quality of the software and ensures user satisfaction. Thus, paying attention to performance in software development processes and making necessary improvements is of great importance.

Maintenance

Maintenance in the software development process is a critical stage in the software's life cycle. Maintenance involves processes carried out to preserve the existing functionality of the software, optimize its performance, and adapt to changing requirements. The impact of maintenance on software quality is of great importance to ensure that the software is long-lasting and sustainable.

Software maintenance encompasses various activities carried out for purposes such as bug fixes, performance improvements, and adapting to new requirements in software systems. There are three main categories of maintenance: corrective, perfective, and adaptive maintenance (Van Vliet, 2019). However, the IEEE 14764 standard has defined a fourth category, preventive maintenance, as seen in Figure 2, in addition to these three categories.

Table 1. Software Maintenance Categories

	Corrective	**Enhancement**
Proactive	Preventive Maintenance	Perfective Maintenance
Reactive	Corrective Maintenance	Adaptive Maintenance

Corrective maintenance is performed to fix errors and deficiencies within the software, while adaptive maintenance is carried out to ensure the software adapts to new technologies. Perfective maintenance involves updating the software with new features. Preventive maintenance is conducted after the software product is delivered to detect and prevent potential errors within the product.

The impact of maintenance on software quality ensures the software's longevity and its continued suitability to meet user needs (Booth & Elbaum, 2018). Regular maintenance improves the software's performance, enhances its security, and increases user satisfaction. Additionally, maintenance activities are necessary to ensure the sustainability of the software.

The effects of software maintenance on quality include improving the software's reliability, performance, and compatibility. Maintenance enhances the reliability of the software by correcting errors and failures, which allows users to use the software more dependably (Perry & Wolf, 2020). Additionally, regular maintenance provides performance improvements, increasing the software's efficiency. These performance enhancements enable the software to operate faster and more effectively.

Maintenance also helps the software adapt to changing requirements. Technological advancements and evolving business needs necessitate updating the software and adding new features (Tilley et al., 2021). This type of developmental maintenance ensures that the software remains competitive and continues to meet user needs.

Effectively managing the maintenance process is crucial for enhancing software quality. Maintenance processes require effective planning, monitoring, and evaluation (Gotel & Finkelstein, 2019). Maintenance strategies should be determined based on the type of maintenance and the software's requirements. Regular maintenance reports and monitoring tools help assess the effectiveness of the maintenance process.

Managing the maintenance process improves software quality and increases user satisfaction. A good maintenance strategy ensures that the software remains of consistently high quality and helps users utilize the software effectively. Therefore, effectively managing maintenance processes is a critical factor in maintaining software quality.

In conclusion, maintenance is a key component of software quality, ensuring that the software remains long-lasting, reliable, and suited to user needs. Effectively managing the maintenance process enhances software quality and ensures user satisfaction.

CONCLUSION

The various aspects of software quality are critically important in the software development process. Functionality, reliability, usability, performance, and maintenance are key components that determine the quality of a software product. Each of these elements directly impacts the success of the software and user satisfaction.

Functionality refers to the software's ability to perform specific tasks. Meeting functional requirements ensures that the software operates as intended and plays a fundamental role in addressing user needs. However, focusing solely on functional requirements does not encompass the overall quality of the software; other quality components are equally important.

Reliability refers to the software's ability to operate without errors over a specific period of time. Having reliable software is essential for minimizing system failures and ensuring a seamless user experience. A lack of reliability can undermine user trust and lead to costly issues.

Usability refers to the software's user-friendliness and the ability of users to effectively use the software. Usability plays a crucial role in the design of software and directly impacts the user experience. Usability issues can lead to user dissatisfaction and low user adoption.

Performance defines the efficiency and speed characteristics of software. Good performance ensures that the software operates quickly and effectively, enabling users to utilize it more efficiently. Performance shortcomings can lead to system slowdowns and a degraded user experience.

Maintenance encompasses the ongoing support and update processes throughout the software's lifecycle. Effective software maintenance ensures the system remains long-lasting and continuously up to date. When maintenance processes are inadequate, outdated versions of the software can lead to usability and security issues.

Establishing an effective balance among these quality components enhances the success of software development processes and ensures the creation of high-quality software products. Considering and improving each quality attribute positively impacts user satisfaction and the overall success of the software.

REFERENCES

Boehm, B. W. (1981). *Software Engineering Economics*. Prentice-Hall.

Boehm, B. W. (1984). *Verifying and Validating Software Requirements and Design Specifications*. IEEE Transactions on Software Engineering, SE-10(3), 255-273.

IEEE. (1998). *IEEE Standard for Software Quality Assurance Processes*. IEEE.

Nielsen, J. (1993). *Usability Engineering*. Academic Press.

Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

Pfleeger, S. L., & Atlee, J. M. (2010). *Software Engineering: Theory and Practice* (4th ed.). Prentice Hall.

ISO/IEC. (2011). *ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. International Organization for Standardization.

Pressman, R. S., & Maxim, B. R. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

Sommerville, I. (2011). *Software Engineering* (9th ed.). Addison-Wesley.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (2004). *Human-computer interaction* (3rd ed.). Pearson Education.

Gould, J. D., Lewis, C., & Boies, S. (1987). *Designing for usability: Key principles and what designers think*. Communications of the ACM, 30(3), 118-122.

Nielsen, J. (1994). *Usability engineering*. Academic Press.

Nielsen, J., & Budiu, R. (2012). *Mobile usability*. New Riders.

Norman, D. A. (2013). *The design of everyday things: Revised and expanded edition*. Basic Books.

Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: Strategies for effective human-computer interaction* (5th ed.). Addison-Wesley.

Cunningham, P., & Gil, J. (2019). *Performance Engineering for Cloud Computing: Applications and Techniques*. CRC Press.

Eich, C., Filman, R. E., & Roman, J. (2019). *Performance Evaluation and Benchmarking for Modern Computing Systems*. Springer.

Jiang, Y., Chen, L., & Zhang, Y. (2017). *High-Performance Computing for Data Analysis: Applications and Techniques*. Wiley.

Kerr, M., & Arnold, R. (2018). *Scalable and High-Performance Computing for Engineering and Science*. Elsevier.

Smith, C., & Williams, L. (2002). *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software*. Addison-Wesley.

Booth, J., & Elbaum, S. (2018). *Software Maintenance and Evolution: A Roadmap*. Springer.

Gotel, O. C., & Finkelstein, A. C. W. (2019). *Software Maintenance: Concepts and Practices*. Wiley.

Perry, D. E., & Wolf, A. L. (2020). *Foundations of Software Engineering: Software Maintenance and Evolution*. ACM Press.

Tilley, S., & Finkelstein, A. (2021). *Managing Software Maintenance and Evolution*. Elsevier.

Van Vliet, H. (2019). *Software Engineering: Principles and Practice*. Wiley.

Chapter 4

Estimation of Discharge Energy of Lithium-ion Battery Under HPPC Drive Cycle Conditions Using Deep Learning Methods

Göksu TAŞ[1] *

1

[1] Manisa Celal Bayar University, Mechatronics Engineering, 45400, Manisa, TURKEY
*Corresponding Author Author mail: Göksu Taş (goksu.tas@cbu.edu.tr)

Abstract:

In this study, the discharge energy of a lithium-ion battery under Hybrid Pulse Power Characterization (HPPC) drive cycle conditions was estimated by deep learning methods. Thanks to the accurate estimation of discharge energy, the usage time of a lithium battery will be more accurate. In this way, a healthier charging opportunity will be obtained. In this study, voltage (V), current (A), and discharge capacity (mAh) were estimated as input data and discharge energy (Wh) as output data using deep learning methods. The learning methods of this value are CNN, LSTM, RNN, GRU, and CNN-LSTM methods. All models were made according to the most commonly used estimation metrics in regression estimation problems in the literature. MAE, MSE, RMSE, and R2 methods were used for evaluation metrics. The success of the proposed method is given with different metrics and graphs. Among all deep learning methods, the RNN method became the most successful method by achieving 99.9948% estimation success according to the R^2 metric.

Keywords: *Lithium-ion battery, HPPC test, Deep Learning methods, Discharge energy estimation*

1. INTRODUCTION

Countries began to use renewable energy resources to meet the rapidly rising load demand. However, the grid may experience instability and issues with power quality due to the irregularity of these renewable energy resources (RESs). Energy storage technologies, such as battery storage systems, were presented as solutions to these problems. As of right now, the most widely used batteries for stationary applications such as load tracking, area control, and energy management by addition or subtraction of power from the grid are lithium-ion (Li-ion) and lead-acid [1]. When used in off-grid or microgrid situations, BESSs in conjunction with renewable energy sources can be more affordable than diesel generators alone. The International Energy Agency projected that 17 GW of BESSs will be built worldwide in 2020. The consensus trend in the great majority of surveys and studies is substantially upwards, despite estimates and predictions for the presently installed capacity and future expansion being widely different. Since lithium-ion cells are currently the most common battery technology used in new installations, technological advancements and cost reductions have contributed to the rise of battery energy storage systems. Even with declining cell costs, batteries continue to be the principal expense of BESSs. Lithium-ion batteries degrade over time due to a variety of internal aging factors, which can lead to capacity loss, an increase in cell resistance, and safety concerns. The most notable of these degrading impacts, capacity reduction, can significantly affect a BESS's profitability. Lithium-ion cell aging mechanisms have drawn a lot of attention in the scientific literature, and there are several reviews on the topic. The rate of deterioration is known to be dependent on external stress variables, including temperature (T), depth of cycle (DOC), state of charge (SOC), and charge-discharge rate (Crate), which has been confirmed by cell aging research [2].

The chemistry of the battery, its operating environment, and its abuse tolerance all have a significant impact on battery safety. The instability of the electrochemical system is what leads to a LIB's internal failure. As a result, assessing battery safety necessitates knowledge of the material properties, adverse reactions, and electrochemical processes involved in LIBs. The two factors that control battery responses are voltage and temperature. Battery burst and the ignition of combustible objects are caused by the continuous heat and gas generation resulting from safety accidents. External forces that control temperature, voltage, and electrochemical reactions are the primary source of internal disturbances in batteries. As a result, the working environment has a big impact on battery safety [3]. Discharge energy is of critical importance for Battery Management Systems (BMS). And the energy management center of the electric vehicle is BMS. A lot of study has been done on the BMS working conditions for electric vehicle systems. The field of BMS research garners greater interest and expands the reach of studies in academia and industry.

Modeling efforts were used to study BMSs by Shen and Gao [4]. BMS hardware ideas were evaluated by Lelie et al. [5]. State-of-charge estimates and battery modeling are discussed in [6], [7]. See, K.W. et al. examined safety concerns on BMSs on a wide scale LIB [8], while Lin, Jiayuan, et al. studied battery heat management systems LIB [9]. Cloud-based smart BMSs for LIB were evaluated by Tran, Manh-Kien, et al. [10]. Many other studies have been conducted on accurate estimation of battery parameters. Cell capacity (Q) [11], resistance (R) [12], open circuit voltage (OCV) [13], coulomb efficiency (CE) [14], self-discharge rate (SDR) [15], and so on are some of the variables that are inconsistent. The state of charge (SOC) [16], state of energy (SOE) [17], remaining discharge time (RDT) [18], and other factors are all included in the system state inconsistency. Ageing, heat, and electricity can be used to characterize the inconsistent behavior of the outside world. State of power (SOP) [19], voltage (V) [20], and current (I) all exhibit inconsistent electrical behavior. Heat generation and temperature dispersion are included in the thermal aspect. State of health (SOH) [21], decay rate (DR) [22], and remaining usable life (RUL) [23] are examples of inconsistent aging behavior.

It has been observed that there is no sufficient study in the literature for the discharge energy estimation for lithium-ion batteries, and this situation constitutes the main focus of the study. In this study, this problem has been tried to be solved by estimating the discharge energy, which is of critical importance in battery energy management and battery management systems, by using deep learning methods. The discharge energy has been tried to be estimated with high accuracy by using five different deep learning algorithms. The experimental results of the obtained deep learning methods have been compared with four different regression estimation evaluation methods. In addition, a hybrid deep learning method was among the methods whose performance was evaluated, and the performance of this method was investigated. As a result of this study, important results have been obtained by using deep learning methods that can form the basis for the accurate and reliable estimation of the discharge energy of lithium-ion batteries.

2. MATHERIAL and METHODS

In this study, the discharge energy of the lithium-ion battery was estimated using deep learning methods. The learning rate, batch size value and units' values are the same in all models. In addition, Python programming language libraries such as matplotlib, pandas and tensorflow were used for deep learning studies experiments.

2.1. Experimental Data

This study was conducted using a publicly available dataset [24]. LG Chem INR21700-M50T type lithium-ion battery was used for the relevant dataset. The nominal capacity of the battery was 4.85 Ah, the nominal voltage was 3.63 V, and the cutoff current value was 50 mA, while the size of the battery was 21.44 mm x 70.80 mm. Only a portion of this dataset was used in this study. The characteristics of the data used in this study are given in Figure 1.

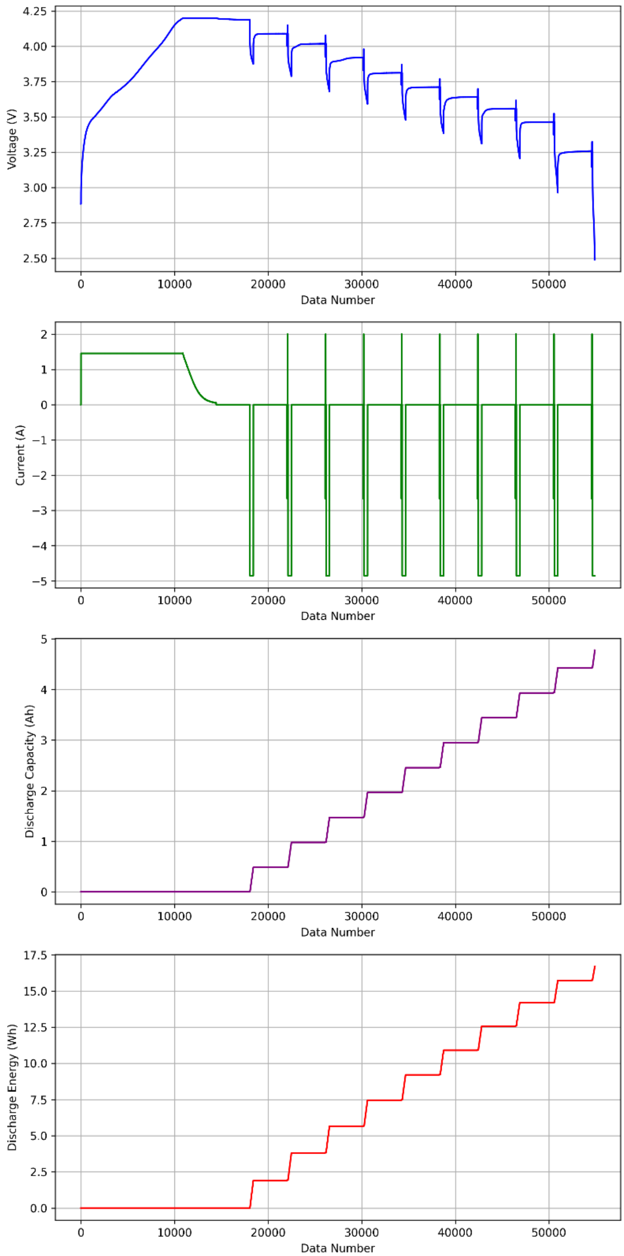


Figure 1. Data features

2.2. Deep Learning methods

Deep learning is a subfield of machine learning, and its use is increasing day by day. Deep learning is used to solve many problems, such as object recognition, natural language processing, weather forecasting, disease detection in agricultural areas, and real-time object classification. The problems solved by deep learning are divided into subfields such as regression and classification problems. The algorithms that form the basis of many deep learning models were used in this study.

Convolutional Neural Network

Using convolution structures, CNN is a type of feedforward neural network that can extract features from data. CNN does not need manual feature extraction, in contrast to conventional feature extraction techniques. CNN's architecture draws inspiration from visual perception. A CNN kernel is a variety of sensors that may respond to a wide range of variables; an artificial neuron is similar to a real neuron in that activation functions replicate the mechanism by which neural electric impulses that cross a particular threshold are sent to the next neuron. With the development of loss functions and optimizers, the CNN structure as a whole was able to understand our expectations. In contrast to the fully connected (FC) networks shown in Figure 2, CNN offers several benefits: 1) Local connections: This reduces parameters and speeds up convergence by connecting each neuron to a limited set of neurons rather than all of the neurons in the preceding layer. 2) Weight sharing: This further decreases parameters by allowing a set of links to share the same weights. 3) Downsampling dimension reduction: By using the idea of image local correlation, a pooling layer may downsample a picture and save relevant information while doing so. Eliminating insignificant characteristics might also help to lower the total number of parameters. CNN is one of the most representative algorithms in the field of deep learning because of these three compelling features [25].

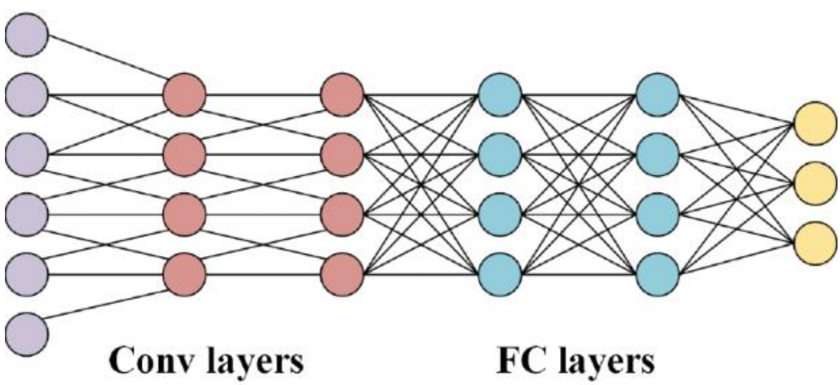


Figure 2. Convolution block and FC block [25].

Recurrent Neural Network

Finding several layers of representation for incoming data is the goal of machine learning algorithms, such as deep learning algorithms. Multilayer RNNs are among the most widely used models for deep learning; they were developed in the 1980s. These networks come in a variety of forms and each one has a memory that stores the data they have seen thus far. Furthermore, RNNs use the prior output to predict the next, making them effective models for sequential data (time series). The networks themselves in this instance contain recurring loops. These loops, found in the hidden neurons, enable the system to predict future outputs by temporarily storing prior input data. Only once the predetermined number of iterations is reached does the output of a recursive neuron go on to the next layer. Here, the output is more extensive, and the prior data is preserved for a longer period of time. Lastly, the weights are updated by going backwards with the mistakes [26].

Long-short Term Memory

The LSTM model is a powerful recurrent neural system designed for tackling the exploding/vanishing gradient difficulties that typically occur while learning long-term dependencies, even in situations where the smallest time delays are extraordinarily lengthy. In general, this may be avoided by employing a constant error carousel (CEC), which keeps the error signal contained in the cell of each unit. In actuality, these cells are recurrent networks in and of themselves. Their intriguing design comes from the way the input and output gates, which together create the memory cell, are added to the CEC. One time step lag is indicated by the self-recurrent connections. An input gate, an output gate, a forget gate, and a cell make up a vanilla LSTM unit. The forget gate was suggested by Gers et al. (2000) as a way to enable the network to reset its state, although it was not originally a component of the LSTM network. The three gates control the information flow related to the cell, and the cell retains data for arbitrarily long periods of time [27].

Gated Recurrent Unit

An enhanced version of RNNs, the GRU can recall prior knowledge and modify it to determine the present states. In the realm of machine learning, gradient vanishing or exploding (GRU) has become a prominent and efficient approach due to its strong capacity in tackling these problems. The update gate and reset gate are the two gates that provide access to the GRU's output. A typical GRU construction is shown in Fig. 3. Based on the information shown in Fig. 3, the GRU's transmission mechanism may be described as,

$$Z_t = \sigma(w_{xz}x_t + w_{hz}h_{t-1} + b_z) \quad (1)$$

where Z_t is the update gate that controls how much of the past is brought into the present state. The related weight matrices and bias vector are w_{xz}, w_{hz}, and b_z. σ represents the sigmoid function, andx_t stands for the t_{th} input vector [28].

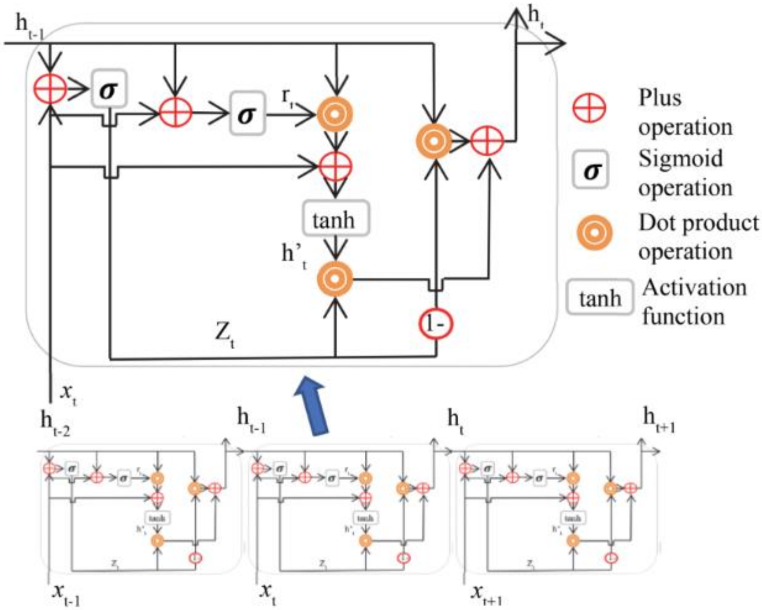


Figure 3. A block of GRU [28].

3. RESULTS

By correctly estimating the energy of the lithium-ion battery, the system is provided with more efficient operation by managing the energy appropriately. In this study, the discharge energy of the lithium-ion battery operated under HPPC drive cycle conditions was successfully estimated using deep learning methods. Table 1 shows the results of the deep learning experiments.

Table 1. Deep learning training results

Model	MSE	RMSE	MAE	R^2
CNN	0.191066	0.437111	0.224731	0.993951
LSTM	0.026525	0.162866	0.096833	0.99916
GRU	0.005327	0.072987	0.046794	0.999831
RNN	0.001645	0.040553	0.014728	0.999948
CNN-LSTM	0.041324	0.203283	0.118419	0.998692

Figure 4 shows the values estimated by the RNN method, which achieved the most successful result in this study, and the real discharge energy value. Then, an accuracy line is determined between these two values and how successful the prediction result is shown.

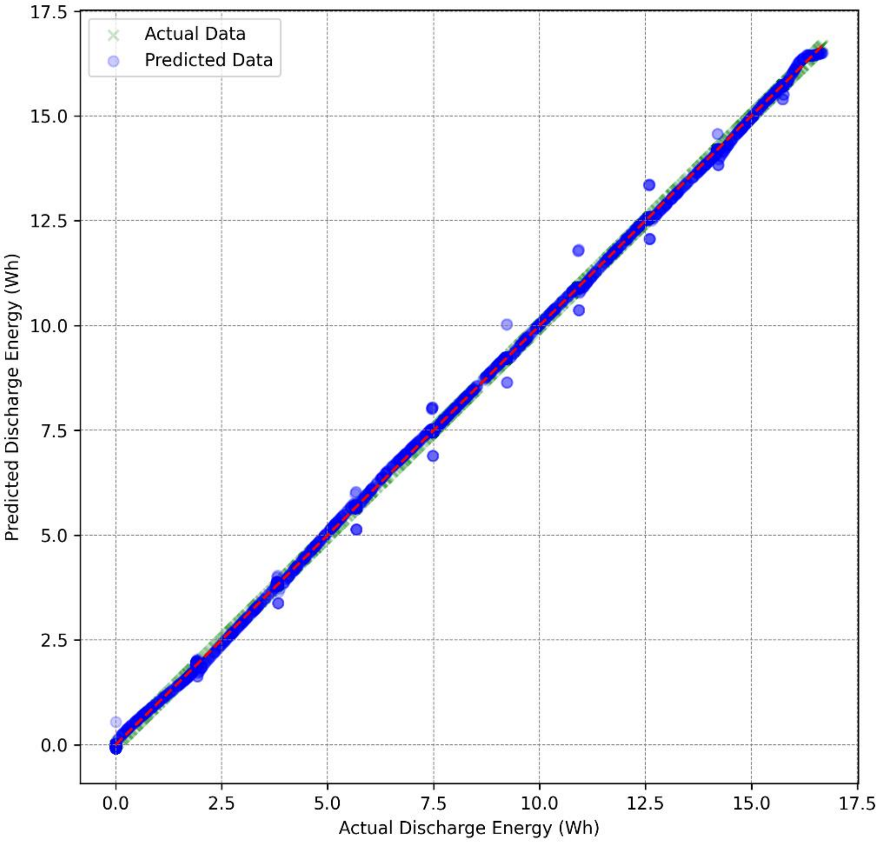


Figure 4. Prediction results of the RNN method.

In Figure 5, the prediction error that occurred during the deep learning training run in this study and the test results of the models obtained as a result of training are given. Using the MAE and MSE metrics, an attempt was made to determine the method that made the least prediction error according to the regression prediction error values. The RNN method became the method that reached the least error in prediction by making an error of 0.014728 according to the MAE metric.

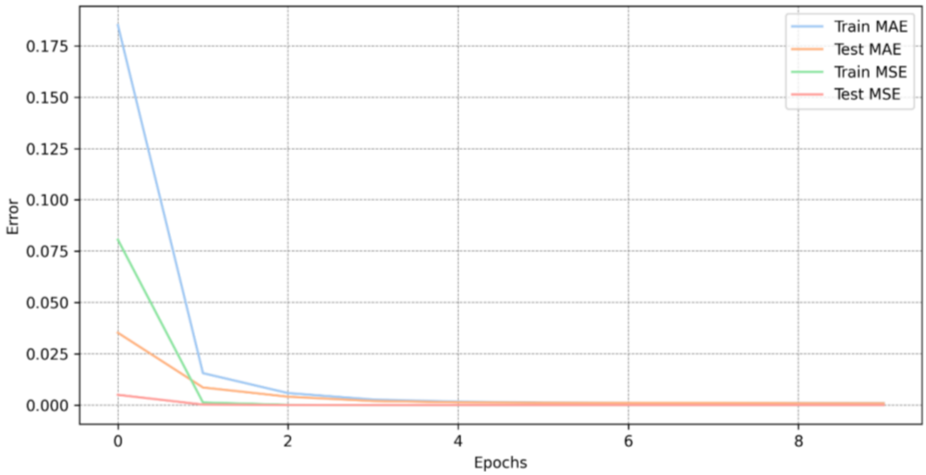


Figure 5. Prediction error results of deep learning methods.

4. CONCLUSION

According to the possibility of depletion of non-renewable energy sources, renewable energy sources are gaining importance day by day in order to prevent an energy crisis. Lithium-ion batteries, which can store electrical energy chemically, are used in many devices we use in our daily lives, and their usage is being optimized day by day. In this study, the discharge energy of a lithium-ion battery under HPPC drive cycle conditions has been successfully estimated using deep learning methods. By accurately estimating the discharge energy, battery management systems can create more accurate usage plans for efficient energy use. Using basic deep learning methods such as CNN, LSTM, RNN, GRU, and CNN-LSTM, the discharge energy of a lithium-ion battery under an HPPC test has been successfully estimated. For the proposed method, the success of MSE, RMSE, MAE, and R^2 metrics for lithium battery discharge energy estimation was 0.001645, 0.040553, 0.014728, and 0.999948, respectively. While the RNN method is the most successful method, it is open to further studies after this study because the model is a model that forms the basis for other models. The discharge energy estimation model developed is considered an important method that can contribute to the energy, safety, and performance of lithium-ion batteries. In battery management systems, it can be a basic model for the proper use of the battery and parameter estimation. In their future studies, the authors plan to conduct parameter determination studies for the performance and safe use of lithium batteries using different electronic cards.

REFERENCES

[1] A. A. Kebede *et al.*, “Techno-economic analysis of lithium-ion and lead-acid batteries in stationary energy storage application,” *Journal of Energy Storage*, vol. 40, p. 102748, Aug. 2021, doi: 10.1016/j.est.2021.102748.

[2] N. Collath, B. Tepe, S. Englberger, A. Jossen, and H. Hesse, “Aging aware operation of lithium-ion battery energy storage systems: A review,” *Journal of Energy Storage*, vol. 55, p. 105634, Nov. 2022, doi: 10.1016/j.est.2022.105634.

[3] Y. Chen *et al.*, “A review of lithium-ion battery safety concerns: The issues, strategies, and testing standards,” *Journal of Energy Chemistry*, vol. 59, pp. 83–99, Aug. 2021, doi: 10.1016/j.jechem.2020.10.017.

[4] M. Shen and Q. Gao, “A review on battery management system from the modeling efforts to its multiapplication and integration,” *International Journal of Energy Research*, vol. 43, no. 10, pp. 5042–5075, 2019, doi: 10.1002/er.4433.

[5] M. Lelie *et al.*, “Battery Management System Hardware Concepts: An Overview,” *Applied Sciences*, vol. 8, no. 4, Art. no. 4, Apr. 2018, doi: 10.3390/app8040534.

[6] M. U. Ali, A. Zafar, S. H. Nengroo, S. Hussain, M. Junaid Alvi, and H.-J. Kim, “Towards a Smarter Battery Management System for Electric Vehicle Applications: A Critical Review of Lithium-Ion Battery State of Charge Estimation,” *Energies*, vol. 12, no. 3, Art. no. 3, Jan. 2019, doi: 10.3390/en12030446.

[7] Y. Wang *et al.*, “A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems,” *Renewable and Sustainable Energy Reviews*, vol. 131, p. 110015, Oct. 2020, doi: 10.1016/j.rser.2020.110015.

[8] K. W. See *et al.*, “Critical review and functional safety of a battery management system for large-scale lithium-ion battery pack technologies,” *Int J Coal Sci Technol*, vol. 9, no. 1, p. 36, May 2022, doi: 10.1007/s40789-022-00494-0.

[9] J. Lin, X. Liu, S. Li, C. Zhang, and S. Yang, “A review on recent progress, challenges and perspective of battery thermal management system,” *International Journal of Heat and Mass Transfer*, vol. 167, p. 120834, Mar. 2021, doi: 10.1016/j.ijheatmasstransfer.2020.120834.

[10] M.-K. Tran, S. Panchal, T. D. Khang, K. Panchal, R. Fraser, and M. Fowler, “Concept Review of a Cloud-Based Smart Battery Management System for Lithium-Ion Batteries: Feasibility, Logistics, and Functionality,”

Batteries, vol. 8, no. 2, Art. no. 2, Feb. 2022, doi: 10.3390/batteries8020019.

[11] Y. Jiang, J. Jiang, C. Zhang, W. Zhang, Y. Gao, and Q. Guo, "Recognition of battery aging variations for LiFePO4 batteries in 2nd use applications combining incremental capacity analysis and statistical approaches," *Journal of Power Sources*, vol. 360, pp. 180–188, Aug. 2017, doi: 10.1016/j.jpowsour.2017.06.007.

[12] Z. He, D. Guo, X. Liu, and G. Yang, "An evaluation method of battery DC resistance consistency caused by temperature variation," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2017, pp. 7623–7628. doi: 10.1109/IECON.2017.8217336.

[13] S. A. Svatek *et al.*, "High open-circuit voltage in transition metal dichalcogenide solar cells," *Nano Energy*, vol. 79, p. 105427, Jan. 2021, doi: 10.1016/j.nanoen.2020.105427.

[14] L. Zhou, Y. Zheng, M. Ouyang, and L. Lu, "A study on parameter variation effects on battery packs for electric vehicles," *Journal of Power Sources*, vol. 364, pp. 242–252, Oct. 2017, doi: 10.1016/j.jpowsour.2017.08.033.

[15] H. Liao, B. Huang, Y. Cui, H. Qin, X. Liu, and H. Xu, "Research on a fast detection method of self-discharge of lithium battery," *Journal of Energy Storage*, vol. 55, p. 105431, Nov. 2022, doi: 10.1016/j.est.2022.105431.

[16] Y. Zheng, W. Gao, M. Ouyang, L. Lu, L. Zhou, and X. Han, "State-of-charge inconsistency estimation of lithium-ion battery pack using mean-difference model and extended Kalman filter," *Journal of Power Sources*, vol. 383, pp. 50–58, Apr. 2018, doi: 10.1016/j.jpowsour.2018.02.058.

[17] Y. Zhang, R. Xiong, H. He, and W. Shen, "Lithium-Ion Battery Pack State of Charge and State of Energy Estimation Algorithms Using a Hardware-in-the-Loop Validation," *IEEE Transactions on Power Electronics*, vol. 32, no. 6, pp. 4421–4431, Jun. 2017, doi: 10.1109/TPEL.2016.2603229.

[18] Y. Wang and Z. Chen, "A framework for state-of-charge and remaining discharge time prediction using unscented particle filter," *Applied Energy*, vol. 260, p. 114324, Feb. 2020, doi: 10.1016/j.apenergy.2019.114324.

[19] X. Zhang, Y. Wang, J. Wu, and Z. Chen, "A novel method for lithium-ion battery state of energy and state of power estimation based on multi-time-scale filter," *Applied Energy*, vol. 216, pp. 442–451, Apr. 2018, doi: 10.1016/j.apenergy.2018.02.117.

[20] C. Zhang, Y. Jiang, J. Jiang, G. Cheng, W. Diao, and W. Zhang, "Study on battery pack consistency evolutions and equilibrium diagnosis for serial-connected lithium-ion batteries," *Applied Energy*, vol. 207, pp. 510–519, Dec. 2017, doi: 10.1016/j.apenergy.2017.05.176.

[21] J. Tian, S. Li, X. Liu, and P. Wang, "Long-short term memory neural network based life prediction of lithium-ion battery considering internal parameters," *Energy Reports*, vol. 8, pp. 81–89, Nov. 2022, doi: 10.1016/j.egyr.2022.05.127.

[22] C. Chang *et al.*, "Prognostics of the state of health for lithium-ion battery packs in energy storage applications," *Energy*, vol. 239, p. 122189, Jan. 2022, doi: 10.1016/j.energy.2021.122189.

[23] F. Wang, Z. Zhao, J. Ren, Z. Zhai, S. Wang, and X. Chen, "A transferable lithium-ion battery remaining useful life prediction method from cycle-consistency of degradation trend," *Journal of Power Sources*, vol. 521, p. 230975, Feb. 2022, doi: 10.1016/j.jpowsour.2022.230975.

[24] G. Pozzato, A. Allam, and S. Onori, "Lithium-ion battery aging dataset based on electric vehicle real-driving profiles," *Data in Brief*, vol. 41, p. 107995, Apr. 2022, doi: 10.1016/j.dib.2022.107995.

[25] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[26] H. Apaydin, H. Feizi, M. T. Sattari, M. S. Colak, S. Shamshirband, and K.-W. Chau, "Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting," *Water*, vol. 12, no. 5, Art. no. 5, May 2020, doi: 10.3390/w12051500.

[27] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[28] Q. Ni, J. C. Ji, and K. Feng, "Data-Driven Prognostic Scheme for Bearings Based on a Novel Health Indicator and Gated Recurrent Unit Network," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1301–1311, Feb. 2023, doi: 10.1109/TII.2022.3169465.

Chapter 5

Evaluating the Effects of Reference Pressure Variation in Hardening Soil Model: From Soil Parameters to Finite Element Predictions

Hakan Alper KAMILOGLU[1]

1

Abstract

This study investigates the critical role of reference pressure (p^{ref}) in the Hardening Soil (HS) model and its impact on finite element (FE) analyses of geotechnical problems. The research examines the influence of p^{ref} on key soil parameters, stress-strain behavior, and FE analysis results. A series of parametric studies were conducted to evaluate the effects of different p^{ref} values on the secant stiffness (E_{50}) and initial stiffness (E_i) of soil under varying depth and lateral pressure conditions. The study also compares the HS model with the Mohr-Coulomb (MC) model in a case study of an excavation supported by a diaphragm wall. FE analyses were performed using Plaxis 8.6 software, considering three different p^{ref} values (50 kN/m^2, 100 kN/m^2, and 150 kN/m^2). Results demonstrate that p^{ref} significantly influences the stress-strain behavior of soil, particularly at higher strain levels. In the FE analyses, variations in p^{ref} led to substantial differences in predicted horizontal and total displacements of the diaphragm wall. While p^{ref} had minimal impact on axial and shear forces, it significantly affected moment distribution in the wall. The study highlights the importance of properly selecting and applying pref in HS model applications and underscores the potential errors that can arise from incorrect usage.

Keyword: Hardening Soil Model, Reference Pressure, Finite Element Analysis, Excavation Support, Diaphragm Wall

INTRODUCTION

Finite element (FE) approach is one of the widely used method to solve geotechnical engineering problems [1]. Nevertheless, the accuracy of the method varies based on the assumptions made in solving the problem. It is vital that the assumptions made to simplify the problem are as close as possible to the real case. Thus, the appropriateness of the model used in the study is effective on the accuracy of any computation [2,3]. The soil model is crucial for FE analysis for several key reasons including accurate representation of soil behavior, prediction of ground movements, structural interaction, safety and risk assessment, and advanced geotechnical analysis. The selection and implementation of an appropriate soil model are vital for achieving accurate and reliable results in geotechnical engineering projects. There are many material models such as Mohr-Coulomb, Soft Soil, Hardening Soil, Hardening Soil with Small Strain, Soft Soil Creep, or Modified Cam-Clay suitable for simulating soil behavior.

The selection and calibration of soil model parameters significantly impact the outcomes of finite element analysis. The Hardening Soil (HS) model, a comprehensive constitutive model, plays a crucial role in capturing the behavior of soils under varying loading conditions. This model considers stress-dependent stiffness for loading and unloading/reloading scenarios, accounts for irreversible strains due to primary deviatoric loading (shear hardening), and incorporates irreversible plastic strains caused by loading [4]**.** The use of the Hardening Soil model allows for a more accurate representation of soil behavior, especially in scenarios involving complex loading conditions such as slope stability analysis [5]. Moreover, studies have highlighted the importance of selecting appropriate soil models for different geotechnical problems. For instance, the choice between the Mohr-Coulomb model and the Hardening Soil model depends on the specific characteristics of the soil and the nature of the analysis being conducted [6]. In many study, the performance of these models have compared for various scenarios like deep excavations, bridge foundations, and slope stability analyses, emphasizing the need for model selection based on the soil's behavior and loading conditions [4,7,8]. The calibration and optimization of parameters within the Hardening Soil model are essential for achieving accurate results in finite element analysis. Studies have described methodologies for optimizing model parameters using Plaxis FE software, highlighting the importance of aligning modeling results with laboratory tests to ensure the reliability of the models [9–13]**.**

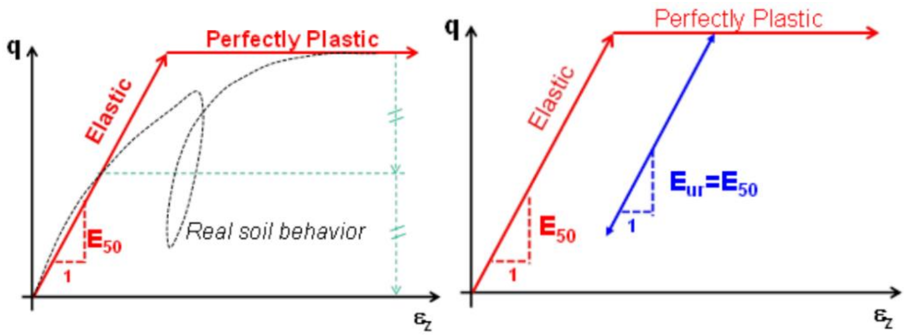
The reference pressure (p^{ref}) is a critical parameter in HS model due to several aspects. Soil stiffness varies depending on the applied stress. The p^{ref} value creates a reference point to determine this change, allowing for the prediction of soil behavior under different pressure conditions. The HS model uses stress-dependent stiffness modulus to reflect changes in soil stiffness depending on the stress level by taking into account p^{ref} value. The reference pressure plays vital role in several parameters such as E_{50}^{ref}, E_{50}, E_{oed}^{ref}, and, E_{ur}^{ref} considered in the FE analysis. The reference pressure is not a geotechnical feature, but it is useful for considering the hyperbolic stress-strain relation in the FE analyses. In the HS model, it may be perceived as the p^{ref} value is a defoult value and should not be changed. One of the common mistakes is to create a FE model using stiffness modulus which is determined considering another horizontal stress ($p^{ref} \neq \sigma'3$) or vertical stress ($p^{ref} \neq \sigma'1$) instead of the reference pressure entered into the software.

In this study, it is intended to investigate the effect of p^{ref} value on the results of FE analysis. The study comprises two phases. In the first phase variation of secant modulus and initial stiffness with soil depth and confining pressure were examined for three p^{ref} values. The stress-strain relations of a soil media for a constant depth were examined for various p^{ref} values. In the second phase an excavation problem supported by diaphram walls and anchors was taken into account using Plaxis 8.6 FE software. The FE analyses were performed for three p^{ref} values, and the effect of the p^{ref} value on the estimation displacement, axial force, shear force, and moment acting on the diaphragm wall was examined.

METHODS

Hardening Soil Model (HS)

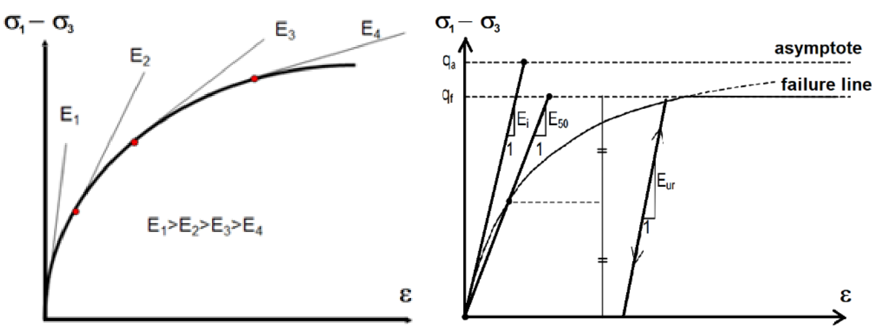
Mohr-Coulomb soil model is the one of the basic soil models used in the geotechnical engineering field. In this model non-linear soil behavior was represented with bilinear lines in stress-strain graph shown in Fig.1. The stress-strain behavior comprises linear elastic and perfectly plastic phases. In this model, soil stiffness is considered a constant value (E_{50}) throughout elastic phase. The point of criticism of the model is that, contrary to constant stiffness modulus assumptions, in reality, soil behaves nonlinearly under monotonic load, and the stiffness of the soil changes depending on the stress level. The MC model may lead to over predict or under predict the soil behavior based on the stress level (Fig.1a). Another shortcoming of the model is supposing equal stiffness modulus in loading and unloading or reloading cases is it is seen in Fig. 1b ($E_{ur}=E_{50}$) [14].



(a) **(b)**

Fig. 1. Stress- strain relation of soil represented with Mohr-Coulomb Model [14].

The hardening soil model (HS) is useful for predicting realistic soil behavior under load. Its main difference from the MC model is the stiffness approach. When the stress-strain behavior of the soil subjected to deviatoric loading is examined, it is seen that stiffness of the soil decreases and, irreversible plastic strain develops with increasing strain levels (Fig.2a). This non-linear stress-strain behavior under monotonic load can be represented with a hyperbola. There are several preliminary studies aiming to formulate this relationship. The Hyperbolic Model is a well-known model among these studies [15,16]. In Plaxis (FE) software, a revised form of the hyperbolic model is used as the Hardening Soil Model (HS). The HS model is an improved version of these models for reasons such as being developed based on plasticity theory instead of elasticity theory, taking into account the dilatancy parameter and the yield cap. HS models considers hyperbolic stress-strain relation until Mohr-Coulomb failure point as it is given in Fig.2b.



(a) (b)

Fig. 2. Stress- strain relation of soil represented with Hardening Soil Model [14].

Strength parameters, (□, c, □), power (m), loading stiffness (E_{50}), unloading-reloading modulus (E_{ur}) and oedometer modulus (E_{oed}) are the fundamental parameters for the HS model. The prominent feature of the HS model can be asserted as stress dependency of the soil stiffness and hyperbolic relationship between deviatoric stress (q) and vertical strain ($□_1$). In this context following equations are taken into account.

$$-\varepsilon_1 = \frac{1}{E_i}\frac{q}{\left(1-{q}/{q_a}\right)} \quad \Rightarrow \text{ for } q<q_f \tag{1}$$

$$q_f = \left(c\cot\phi - \sigma_3^{\prime}\right)\frac{2\sin\phi}{\left(1-\sin\phi\right)} \tag{2}$$

$$q_a = \frac{q_f}{R_f} \tag{3}$$

$$E_i = \frac{2E_{50}}{2-R_f} \tag{4}$$

$$E_{50} = E_{50}^{ref}\left(\frac{c\cos\phi - \sigma_3^{\prime}\sin\phi}{c\cos\phi + p^{ref}\sin\phi}\right)^m \tag{5}$$

Where q_a is the asymptotic value of the shear strength. E_i is initial stress. q_f is ultimate deviatoric stress derived from the Mohr-Coulomb failure criteria and strength parameters. Eq.(1) is not valid in case of $q=q_f$, as the failure criterion is staisfied. R_f is the failure ratio gemerally supposed as 0.9, $□_3$ is confining pressure in triaxial test (take negative value for compression). m is power for stress dependency of stiffness and used for simulating stress dependency. p^{ref} is reference lateral earth pressure or refenece confining pressure value from which E_{50}^{ref} is determined. E_{50}^{ref} is secant modulus, which governs shear deformations, is derived from consolidated-drained or consolidated-undrained triaxial compression tests. E_i is the initial tangent modulus obtained at the beginning of the loading curve at very small deformations. From the equations it is seen that E_{50} is derived based on strength parameters, referance secant modulus, referance lateral erath pressure, and power parameter. Thus, E_{50} and E_i play a vital role in the hyperbolic stress-strain relationship.

Another stress dependent unloading-reloading modulus (E_{ur}) is used for unloading and reloading stress paths. Following equation is taken into account for this case. E_{ue}^{ref} is reference Young's modulus for unloading and reloading. . E_{ue}^{ref} is esed to obtain elastic deformations during unloading.p^{ref} : reference lateral earth pressure or refenece confining pressure value from which E_{ur}^{ref} is determined. In pratical applications it is recommended to use $E_{ur}^{ref}=3E_{50}^{ref}$.

$$E_{ur} = E_{ur}^{ref}\left(\frac{c\cos\phi - \sigma_3' \sin\phi}{c\cos\phi + p^{ref}\sin\phi}\right)^m \tag{6}$$

Oedometer modulus (E_{oed}) obtained from an oedometer test determines how soils behave under one-dimensional compression. It represents the consolidation behavior of soils and how they respond to changes in void ratio under such conditions. E_{oed} value can be determined with following formulas.

$$E_{oed} = E_{oed}^{ref}\left(\frac{c\cos\phi - \sigma_1 \sin\phi}{c\cos\phi + p^{ref}\sin\phi}\right)^m \tag{7}$$

It is worth to mention that major principal stress (σ_1) was used in Eq.(7). E_{oed}^{ref} is initial stiffness determined with one dimensional consolidation test. Due to considering one dimensional stress in the test, principle stress (σ_1) is taken into account rather than confining pressure (σ_3). Therefore, reference stress is supposed as $p^{ref}=-\sigma_1$.

Reference Stress (p^{ref})

The reference pressure in the Hardening Soil (HS) model helps account for the nonlinear dependence of soil stiffness on stresses, such as increasing stiffness with depth. This reference pressure, denoted as p^{ref}, connects two types of stress state tests: triaxial and virgin compression. By using data from these tests, the HS model can describe both volumetric (virgin) compression and shear (deviatorial loading) behaviors of soil. To integrate these behaviors into a single model, p^{ref} is used as the standard for processing laboratory test data. In triaxial tests, the horizontal stress (σ'_3) is used as p^{ref}, while in compression tests, the vertical stress (σ'_1) is used as p^{ref} [17]. As it is seen from the above-mentioned equations, p^{ref} plays a vital role in loading stiffness (E_{50}), unloading-reloading modulus (E_{ur}), and oedometer modulus (E_{oed}).

It is not appropriate to evaluate the p^{ref} value as the stress type acting on the soil (σ_1 or σ_3) at a certain depth. This is because the reference stress is used in the model in the form of both lateral stress ($p^{ref}=\sigma'_3$) and axial stress ($p^{ref}=\sigma'_1$). The p^{ref} value should not considered as only horizontal stress or vertical stress. It is obvious that this value is not a soil feature, it is used to derive mathematical modelling in HS model. It is recommended to assume $p^{ref}=100$ kN/m^2 in the analyses [17, 18]. However, in practical applications, it is possible to perceive this value as the default value. In the FE applications using HS model it is crucial to used stiffness modulus obtained by considering reference pressure. One of the common mistakes is to create a FE model using stiffness modulus which is determined considering another horizontal stress ($p^{ref}\neq\sigma'_3$) or vertical stress ($p^{ref}\neq\sigma'_1$) instead of the reference pressure entered into the software.

The situations that may arise if the E_{50} value is determined without considering the p^{ref} value, as mentioned above, are examined in Figure 3. In the figures, the soil depth and lateral earth pressure through the depth (at rest) parameters are taken into account. Fig.3a and 3b show the variation of E_{50} and E_i values with increased horizontal pressure, respectively. Fig. 3c and 3d presents the variation of E_{50} and E_i values with soil depth. The figures examine three different cases. Variation of E_{50} and E_i values under varying lateral pressure was examined by taking into account three different reference pressures ($p^{ref}=50$ kN/m^2, 100 kN/m^2, and 150 kN/m^2). E_{50} and E_i values were determined with Eqs. (4) and (5), considering the parameters given in Table 1.

Table 1. Parameters taken into account the calculation of E_{50} and E_i

E_{50}^{ref}	**c**	ϕ	γ_{soil}	**m**	**Rf**
75000 kN/m^2	20 kN/m^2	30°	20 kN/m^3	0.8	0.9

Nonlinear relationship between the modulus and soil depth can be seen from the figures. In addition, as a result of the E_{50} and E_i value calculations made taking into account different reference pressure values and the data in Table 1, it is seen that the differences in the reference pressure value greatly affect the E_{50} and E_i values. The p^{ref} value causes more significant differences on the E_{50} value with increasing depth, or in other words, increasing lateral pressure.

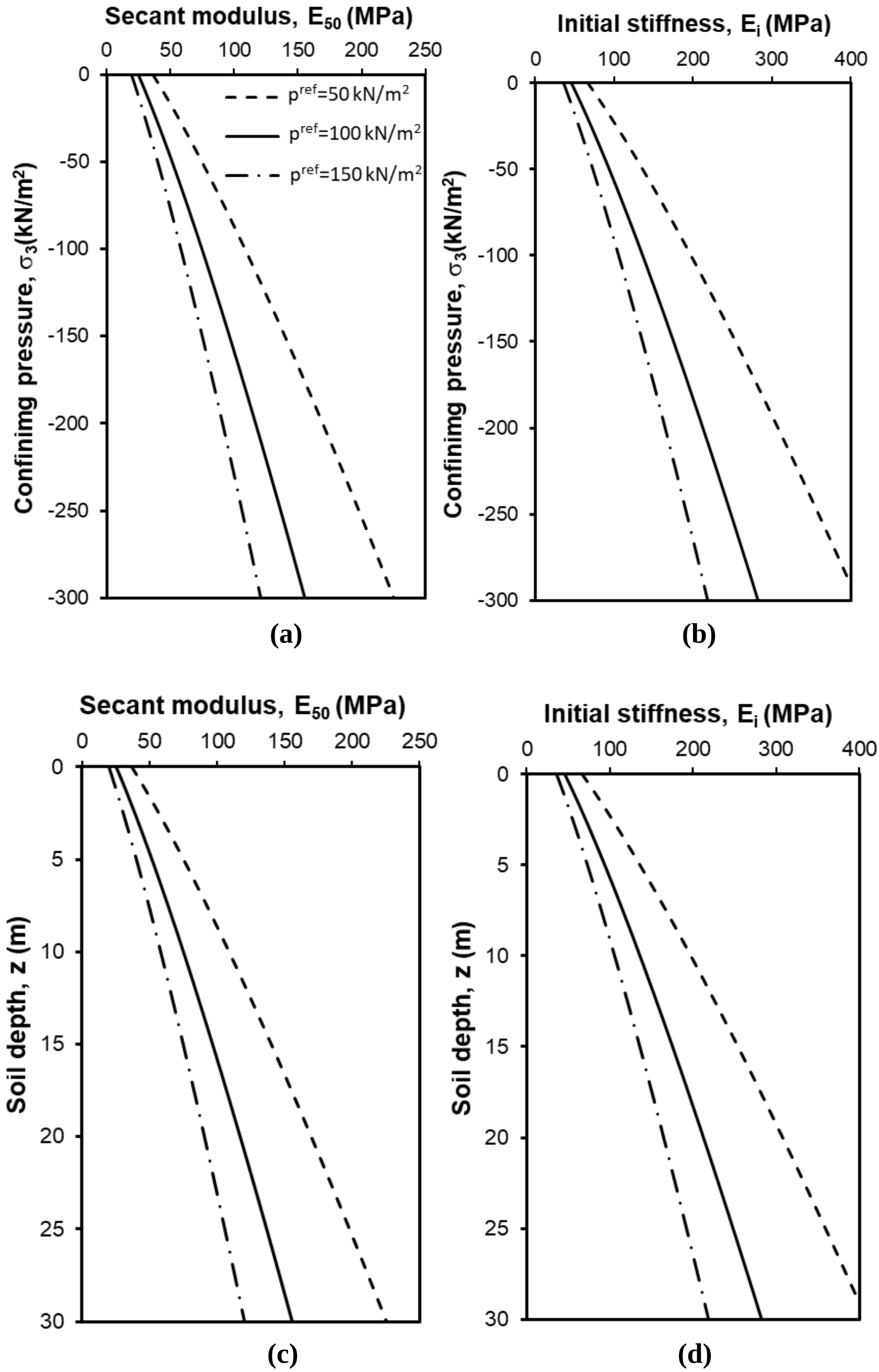


Fig. 3. The effect of pref value on retaion between **(a)** Confining pressure- E_{50} **(b)** Confining pressure- E_i **(c)** Confining pressure- E_{50} **(b)** Soil depth- E_i

In Fig.4, the effect of p^{ref} on the stress-strain relationship of a soil media in which E_{50} value was determined as 75000 kN/m^2 was examined. The strain value (□) was calculated by taking into account the Eqs.(1-4) and paramerers given in tne Table 2.

Table 2. Parameters taken into account the calculation of □

	p^{ref} = 50 kN/m^2	**p^{ref} = 100 kN/m^2**	**p^{ref} = 150 kN/m^2**
q_a	188.1 kN/m^2	299.2 kN/m^2	410.3 kN/m^2
q_f	169.3 kN/m^2	269.3 kN/m^2	369.3 kN/m^2
E_{50}	75000 kN/m^2	75000 kN/m^2	75000 kN/m^2
E_i	136363 kN/m^2	136363 kN/m^2	136363 kN/m^2
R_f	0.9	0.9	0.9
c	20 kN/m^2	20 kN/m^2	20 kN/m^2
□	30°	30°	30°

Fig. 4 was created for three different p^{ref} values (50 kN/m^2, 100 kN/m^2,150 kN/m^2), considering the monotonically increasing deviatoric stress value (q). Hyperbolic stress-strain relationship can be seen from the graph for all reference pressure values. Besides, it is seen that the difference in reference pressure has a remarkable effect on the stress-strain relationship, especially at high strain levels. As a result of examining Fig. 3 and Fig. 4, it is seen that the p^{ref} value is a very effective parameter on the stress-strain behavior of the material under pressure.

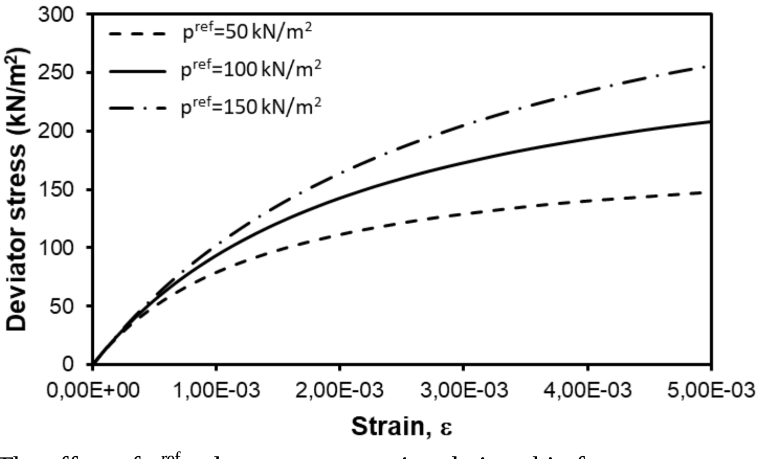


Fig. 4. The effect of p^{ref} value on stress-strain relation ship for a constant soil depth

The Effect of Reference pressure (p^{ref}) on the results of FE analysis

As a result of examining Fig. 3 and Fig. 4, it is seen that the p^{ref} value is a very effective parameter on the stress-strain behavior of the material under pressure. On the other hand, another critical point is how effective the p^{ref} value is on FE analysis results. In this context, a geotechnical problem was modeled using the HS model via Plaxis 8.6 FE software. The FE analyses were performed for three different reference pressures (p^{ref}=50 kN/m^2, 100 kN/m^2, and 150 kN/m^2), and FE analyses results were compared.

The excavation supported by concreate diaphragm wall and anchors was investigated as the geotechnical problem (Fig.6). Detailed information about the modeling, material properties, mesh generation, initial conditions, and calculation phases of the model can be found in Plaxis V8 Tutorial manual under section 6 [19]. For the sake of convenience in the modeling process, single-layer sand was taken into account instead of layered soil.

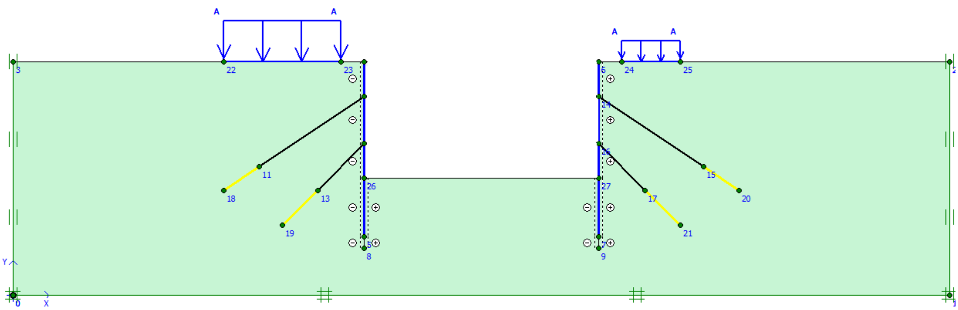


Fig. 5. Geometry of the model

A set of FE analyses were perfomed by employing MC model and HS model. FE analyzes using the HS soil model were repeated for three different reference pressure values (p^{ref}=50 kN/m^2, 100 kN/m^2, and 150 kN/m^2). As a result of analyses, horizontal displacement and total displacement of the diaphragm wall through the wall depth were presented in Fig.6a and 6b, respectively. From the figures, it is seen that there are significant differences between horizontal and lateral displacement estimations of the MC model and the HS model. From the FE analysis results using the HS model, it is seen that different reference pressure values cause large differences in results of both horizontal displacement and total displacement values. In Fig.6(c), maximum lateral and total displacement values through the wall depth, determined with MC and HS models, were compared. It can be seen from the figure that the displacements determined via the MC model are higher than the HS model. In the displacement calculations performed with

HS model, it is seen that maximum and horizontal displacements estimations increases with increasing reference pressure value.

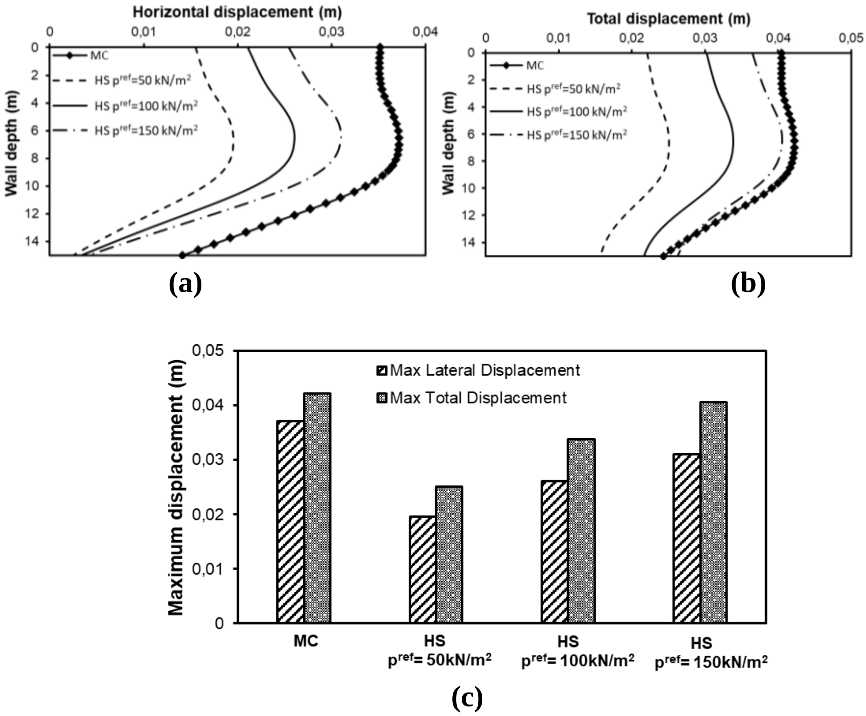


Fig. 6. Comparison of horizontal and total displacement estimations of MC model and HS model for three different reference pressures (a) Variation of horizontal displacement through wall depth (b) Variation of total displacement through wall depth (c) Maximum lateral and total displacemet

Internal forces acting on the diaphragm wall due to excavation were determined via the FE analyses. The effect of the material model on internal forces was investigated by comparing the results of the MC model and HS model. The effect of reference pressure on internal forces was investigated by assuming three different reference pressure value in the analysis using HS model. Additionally, in order to examine the effect of reference pressure on internal forces, solutions in which three different reference pressures were taken into account in the HS model were compared (Fig.7). Figures 7a and 7b show that neither the differences in the material model nor the variations of the reference pressure value cause a considerable change in the axial force and maximum axial force. Figures 7c shows the change of shear force through the wall and Figure 7d compares the maximum shear forces. Accordingly, maximum shear force obtained from the FE analysis using MC model gives the highest value with compared to HS model results. On the other hand, the reference pressure variation did not have a significant effect on the maximum shear force. As a result of examining the moment diagrams along the wall depth in Figure 7e, it is seen that there are notable differences between the MC model and the HS model. In addition, the change in reference pressure causes a change in the moment magnitude acting on the wall along the depth of the wall. The maximum moment value acting on the wall varies greatly depending on the reference pressure value (Fig.7f).

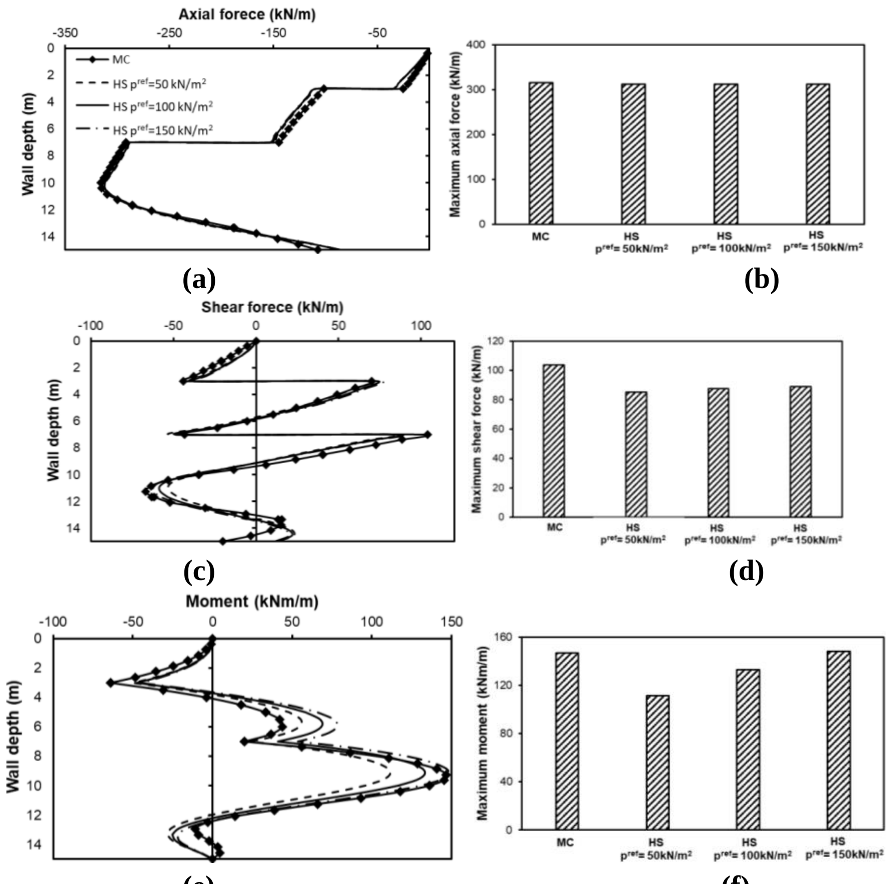


Fig. 7. Comparison of axial forces, sheaer forces and moment estimations of MC model and HS model for three different reference pressures (a) Variations of axial force along the wall depth, (b) Comparison of maximum axial force estimations, (c) Variations of shear force along the wall depth, (d) Comparison of maximum shera force estimations, (e) Variations of moment along the wall depth, (f) Comparison of maximum momentestimations.

CONCLUSIONS

This study has demonstrated the critical role of the reference pressure (p^{ref}) in the Hardening Soil (HS) model and its significant impact on finite element (FE) analyses of geotechnical problems. Several key findings emerge from this investigation:

- The reference pressure is a crucial parameter in the HS model, influencing the loading stiffness (E_{50}), unloading-reloading modulus (E_{ur}), and oedometer modulus (E_{oed}). It is not a soil property but rather a mathematical modeling tool that should be carefully considered in analyses.
- The reference pressure is a crucial parameter in the HS model, influencing the loading stiffness (E_{50}), unloading-reloading modulus (E_{ur}), and oedometer modulus (E_{oed}). It is not a soil property but rather a mathematical modeling tool that should be carefully considered in analyses.
- The reference pressure is a crucial parameter in the HS model, influencing the loading stiffness (E_{50}), unloading-reloading modulus (E_{ur}), and oedometer modulus (E_{oed}). It is not a soil property but rather a mathematical modeling tool that should be carefully considered in analyses.
- The reference pressure is a crucial parameter in the HS model, influencing the loading stiffness (E_{50}), unloading-reloading modulus (E_{ur}), and oedometer modulus (E_{oed}). It is not a soil property but rather a mathematical modeling tool that should be carefully considered in analyses.

In light of the above evaluations, it is seen that if E_{ref} and E_{oed} values obtained under different stress levels are used without paying attention to the reference pressure value in the analyses performed using the HS soil model, significant differences may occur in the analysis results.

REFERENCES

[1] Ikeagwuani CC, Nwonu DC. Emerging trends in expansive soil stabilisation: A review. J Rock Mech Geotech Eng 2019;11:423–40. https://doi.org/10.1016/j.jrmge.2018.08.013.

[2] Sabatakakis N, Tsiambaos G, Ktena S, Bouboukas S. The effect of microstructure on mi strength parameter variation of common rock types. Bull Eng Geol Environ 2018;77:1673–88. https://doi.org/10.1007/s10064-017-1059-7.

[3] Shekhawat P, Sharma G, Singh RM. Morphology and microstructure of waste material-based geopolymer with flyash, eggshell powder, and soft soil. Mater Lett 2023;334:133621. https://doi.org/10.1016/j.matlet.2022.133621.

[4] Rebolledo JFR, León RFP, Camapum de Carvalho J. Obtaining the Mechanical Parameters for the Hardening Soil Model of Tropical Soils in the City of Brasília. Soils and Rocks 2019;42:61–74. https://doi.org/10.28927/SR.421061.

[5] Tsige D, Senadheera S, Talema A. Stability Analysis of Plant-Root-Reinforced Shallow Slopes along Mountainous Road Corridors Based on Numerical Modeling. Geosciences 2019;10:19. https://doi.org/10.3390/geosciences10010019.

[6] Jennings K, Naughton PJ. Similitude Conditions Modeling Geosynthetic-Reinforced Piled Embankments Using FEM and FDM Techniques. ISRN Civ Eng 2012;2012:1–16. https://doi.org/10.5402/2012/251726.

[7] Ahmad Ishak AS, Jelani J, Wong SY, Suif Z, Ahmad Mazuki AL. Experımental and numerıcal study on the behavıour of sılıca sand slope model subjected to surcharge load. J Teknol 2023;86:165–73. https://doi.org/10.11113/jurnalteknologi.v86.20463.

[8] Bowers JT, Webb MC, Beaver JL. Soil Parameters for Design with the 3D PLAXIS Hardening Soil Model. Transp Res Rec J Transp Res Board 2019;2673:708–13. https://doi.org/10.1177/0361198119851723.

[9] Yarkin V, Kukhar A. Determination of non-uniform settlements caused by decompression of soil in the excavation. MATEC Web Conf 2018;245:08002. https://doi.org/10.1051/matecconf/201824508002.

[10] Xuan F, Xia X, Wang J. The application of a small strain model in excavations. J Shanghai Jiaotong Univ 2009;14:418–22. https://doi.org/10.1007/s12204-009-0418-3.

[11] Gowthaman S, Nasvi MCM, Krishnya S. Numerical Study and Comparison of the Settlement Behaviours of Axially Loaded Piles using Different Material Models. Eng J Inst Eng Sri Lanka 2017;50:1. https://doi.org/10.4038/engineer.v50i2.7247.

[12] Aktas T, Calisan O, Cokca E. Finite Element Analysis of Deep Excavation: A Case Study. Proc. China-Europe Conf. Geotech. Eng. Springer Ser. Geomech. Geoengin., Cham: Springer; 2018, p. 885–8. https://doi.org/10.1007/978-3-319-97115-5_1.

[13] Ter-Martirosyan A, Sidorov V, Ermoshina L. Features of optimization of model parameters for solving geotechnical problems. MATEC Web Conf 2018;251:02035. https://doi.org/10.1051/matecconf/201825102035.

[14] Gouw T-L. Common Mistakes on the Application of Plaxis 2D in Analyzing Excavation Problems. Int J Appl Eng Res 2014;9:8291–311.

[15] Duncan JM, Chang C-Y. Nonlinear Analysis of Stress and Strain in Soils. J Soil Mech Found Div 1970;96:1629–53. https://doi.org/10.1061/JSFEAQ.0001458.

[16] Konder RL. A hyperbolic stressstrain formulation for sands. 2nd Pan-American Conf. SMFE, 1963, p. 289–324.

[17] Konyushkov V, Penkov D, Fedorenko E. Initial data for Hardening Soil model. E3S Web Conf 2023;371:02019. https://doi.org/10.1051/e3sconf/202337102019.

[18] PLAXIS 2D Reference Manual. 2020.

[19] Plaxis Version 8 Tutorial Manual. 2004.

Chapter 6

Miniaturization Of Microstrip Patch Antenna

İremnur DURU[1]
Timuçin Emre TABARU[2]

[1] Sivas University of Science and Technology, Sivas, Türkiye, Res. Asst., 0000-0001-5492-803X
[2] Sivas University of Science and Technology, Sivas, Türkiye, Assoc. Prof. Dr., 0000-0002-1373-3620

Abstract

Microstrip patch antennas are classified as planar antennas. They have been studied since the late 1950s. Microstrip antennas are often preferred due to their lightweight, planar structure, and ease of production. The primary application areas of microstrip antennas, which are widely used in both civilian and military fields, include RFID, GPS, and MIMO systems. The advantages of patch antennas operating in the frequency range of 100 MHz to 50 GHz include their small volume, which makes them suitable for the aerodynamic structures of spacecraft, and their low scattering cross-section, which makes them preferred in rockets and satellites. While microstrip antennas have advantages, they also have disadvantages, such as being narrowband, low-gain, and low-efficiency. The size of a microstrip patch antenna should be at least half a wavelength. Miniaturization techniques include material loading, reshaping the antenna, creating new designs in the slot and ground plane, and shorting and holding techniques. The aim of this book chapter is to examine the techniques used in the literature to reduce the size of microstrip patch antennas. The majority of the studies in the book cover significant designs carried out in the last 20 years.

1. The Structure Of Microstrip Antennas

Wireless communication devices have become increasingly prevalent in various fields today. They are used in many areas such as AM and FM radios, mobile phones, tablets, and global positioning systems (GPS). Microstrip patch antennas, proposed by Deschamps [1,2], were initially used for circuit components and transmission lines in printed circuit technology, but they are also known to be used as electronic system radiators [3]. Although there was little interest in microstrip antennas before the 1970s, they later became a focus of interest. This class of antennas, which is the subject of intense research and development, has been the topic of many studies. A list of books, review studies, articles, conference papers, and handbook sections has been provided [4-9].

The basic structure of a microstrip antenna is shown in Figure 1. A microstrip patch antenna consists of a substrate with the same footprint on the ground plane and a radiation area (conductor) on top. The patch height is a piece of metal much smaller than the wavelength in free space. A low-cost patch is added on the substrate. On the other side of the substrate, a ground plane is placed to achieve the microstrip antenna structure. The substrate height can vary between 0.01 and 0.05 times the wavelength in free space. The purpose of using the substrate is to provide mechanical support by leaving a gap between the patch and the ground plane. The material used for the substrate should have low insertion loss. A loss tangent of 0.005 or lower values is chosen.

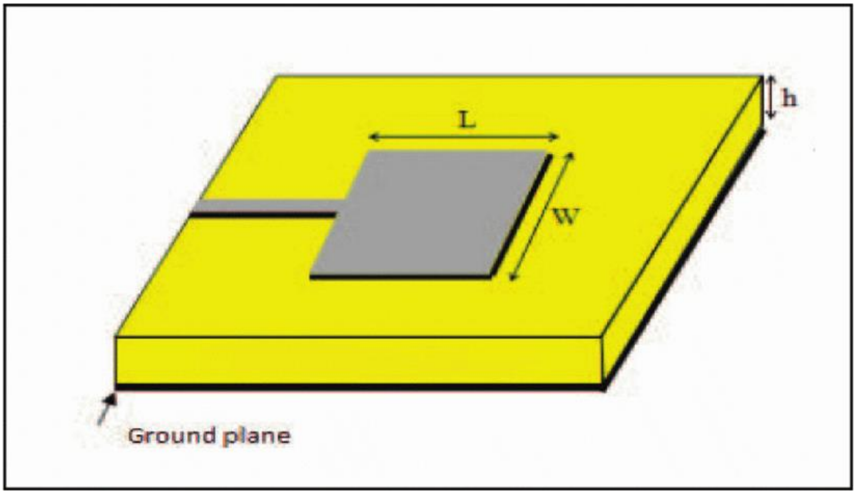


Figure 1. The general form of the microstrip patch antenna [38]

The substrate is categorized into three groups based on the dielectric constant, which varies between 1 and 10. Materials with a dielectric constant between 1 and 2 include air and polystyrene foam, those between 2 and 4 include fiberglass and reinforced Teflon, and those between 4 and 10 include ceramics and quartz.

When designing microstrip patch antennas, various geometric structures are preferred. These vary according to the requirements of the system to be designed. The geometric representations of microstrip antennas are shown in Figure 2. While antenna shapes vary, the feeding methods can also differ. Feeding methods are classified as coaxial probe feeding, microstrip line feeding, aperture-coupled feed, and proximity feed. Electromagnetic energy is first transferred to the area beneath the patch. This area creates a resonance cavity by forming open circuits at its edges. A portion of the electromagnetic energy radiates through the cavity into space, resulting in the formation of an antenna.

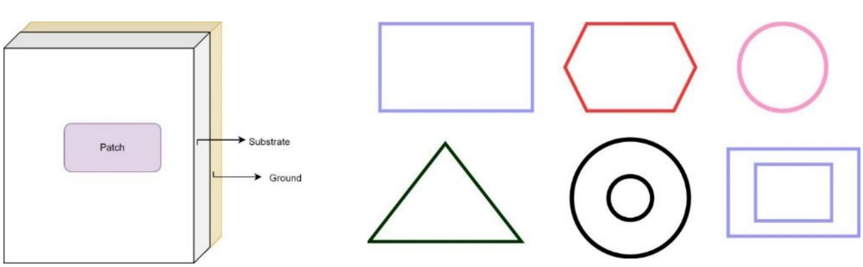


Figure 2. Patch Representations with Different Geometric Shapes

The reasons for the preference of microstrip patch antennas include their flat and thin structure, low profile, and lightweight characteristics, making them suitable for satellite and mobile applications. Patch antennas produced using printed circuit technology are quite cost-effective. Due to their production on flat surfaces, they easily integrate with electronic components and integrated circuits. The feeding methods used in microstrip antennas vary, including coaxial feeding, microstrip line feeding, and aperture feeding. Microstrip antennas actively play roles in mobile communication systems, satellite and GPS systems, and radar systems.

Future trends and expected advancements in microstrip patch antennas include new materials and technologies, miniaturization applications, manufacturing techniques, and flexibility for wearable designs. This study focuses on techniques to reduce the size of antennas. Some fundamental techniques present in the literature will be discussed. Miniaturization techniques include material loading, reshaping the antenna, the use of metamaterials, short-circuiting and folding, and adding slots and defects to the ground plane. The fundamental characteristics and disadvantages of each of these techniques will be discussed along with their effects on antenna performance.

2. Miniaturisation of Patch Antennas

The miniaturization of patch antennas has been a topic of discussion for some time. Wheeler was the first to analyze the fundamental limits of electrically small antennas. He concluded that reducing the size of the antenna decreases the bandwidth while also reducing the gain [10]. This result was later confirmed in subsequent studies [11,12]. In traditional antennas, sizes have typically been reduced to a maximum of half the wavelength. However, in recent years, studies have continued on antenna size due to the preference for compact wireless devices. The techniques for miniaturization include material loading, reshaping the antenna, shorting and folding, using metamaterials, and introducing defects in the ground plane [13]. It is known that there is a lower bound for Q to limit the antenna size. The minimum antenna size that is planned to be placed within a sphere of radius a for a lossless antenna is given by the lower bound of the Q factor in the equality.

$$Q = \frac{1}{ka} + \frac{1}{(ka)^3} \qquad (1)$$

k : wavenumber

Increasing the radius of the sphere, or miniaturizing the antenna size, results in an increase in the Q value. The ratio of the Q factor to the antenna efficiency (n_{rad}) is an important factor. Changes in the Q factor will consequently affect n_{rad}, or antenna efficiency, and therefore the gain as well. In the design of small antennas, it is crucial to establish a significant balance between antenna size, gain, and bandwidth. To miniaturize patch antennas, one approach is to change the substrate material properties or alter the geometry by widening the current path. The headings detailing these methods are presented below.

2.1. Material loading

One of the most commonly used methods to reduce the size of a patch antenna is selecting a substrate with high relative permittivity. The length and width (i.e., size) of the patch are inversely proportional to the square root of $\mathcal{E}_r$. A substrate with high permittivity increases surface wave excitation, which consequently reduces bandwidth and radiation efficiency. For this reason, researchers have turned to using different materials and sized structures. In this study [14], experimental investigations were conducted on microstrip patch antennas with substrates having a relative permittivity of 10 to 13 and dimensions of (0.02-0.03 lambda). While the input impedances remained unchanged, significant differences in resonance resistance were observed in probe-fed and microstrip-fed antennas [14]. In another study [15], the electromagnetic bandgap was tested with integrated microstrip antennas. The dielectric constant has been tested by changing it between 2.2 and 10.2. Surface waves were suppressed, and the coupling directions of the electric and magnetic fields were parametrically examined for different substrates and thicknesses. For thick and highly permeable substrates, an electromagnetic bandgap was introduced due to the significant mutual coupling caused by surface waves. In another academic study [16], experiments were conducted on patch antennas with different thicknesses and permeabilities using three different feeding types. The resonance frequency and resonance resistance were compared with theoretical calculations, indicating that it might be invalid for substrates thicker than (0.02 lambda / free space wavelength) and that this situation stemmed from coaxial feeding. Although valid results were obtained for the resonance frequency compared to theoretical models, it was concluded that the impedance results were not at a reasonable level. This study [16] emphasized the lack of acceptable work for impedance results and that more valid results need to be produced. In another study [17], the validity of two different miniaturization techniques was questioned using a social spider optimization algorithm. In this study focused on impedance improvement, the antenna's resonance frequency was experimentally reduced from 2.45 GHz to 730 MHz to achieve miniaturization with 4.4 permittivity and 1.5 mm thickness. This study [18] examined the formation of unwanted sidelobes and changes in surface waves when antennas were produced using substrates with high dielectric constants. They emphasized that removing the substrate improved the radiation pattern. In another research [19], attempts were made to achieve miniaturization using a low-type mushroom-type meta-substrate. In this study, where theoretical and experimental calculations matched, the radiation efficiency was found to be 85%. In this work [20], the central frequency for a MIMO antenna was chosen as 2 GHz, and the study was conducted over two bands. In this study, miniaturization of 68% was achieved using a semi-loop meandered driven element and a small ground plane. This study, with a radiation efficiency of 85%, measured gains of more than 5 dBi for a single element. In another existing study in the literature [21], miniaturization was achieved using modified ceramics. Another study emphasizing the importance of using a perforated substrate [22] highlighted a significant reduction in patch size. However, the bandwidth reductions accompanying miniaturization are also a critical issue to consider.

2.2. Shorting and Folding

The literature shows that the size of microstrip antennas can be reduced through folding and the use of shorting posts [23-27]. In a patch antenna using a rectangular patch, the electric field is highest at the radiating edges and zero in the center. When an electric wall is placed in the middle of the patch and the other half is removed, the resonance frequency of the resulting quarter-wavelength patch antenna will remain unchanged. The Q factor will remain the same for both the half-wavelength and quarter-wavelength patches. A reduction in aperture will decrease the antenna's directivity, thus affecting its gain. Instead of placing a conductive plate between the patch and the ground plane, short circuits represented by pins are placed at the edges of the patch. However, this method can significantly adversely affect the antenna's gain and directivity due to miniaturization. This method complicates antenna design, making the geometry complex and analysis more difficult. Nevertheless, it has been reported in the literature that this method does not significantly affect antenna efficiency [13]. In this study [23], the antenna size was reduced by performing a simple folding operation. This resulted in a short-circuited patch structure where the resonance frequency could be controlled. The theoretical analyses were compared with experimental results, showing good agreement. A 10 dB return loss of 4% was achieved for the 2.4 GHz ISM band. In another study [24], size reduction was examined in comparison to impedance, bandwidth, high gain, and polarization states to compare antenna performances. In this study [25], a dual-band patch antenna with low cross-polarization and independent band control was presented, achieved through miniaturization, short-circuiting, and folding. The resonance frequency was shifted using interleaved slots, resulting in a three-dimensional element of $0.1\lambda_0$ or smaller. The antenna's impedance matching, radiation pattern, and broad gain has been measured.

2.3. Reshaping or Introducing Slots

The miniaturization of microstrip patch antennas can be achieved by changing their shape or introducing slots. Genetic algorithms can be used to optimize the miniaturized patch shape [26]. In this study [27], the shape of a rectangular patch was modified to change the resonance frequency while keeping the physical volume of the antenna constant. The resonance frequency was shifted from 4.9 GHz to 2.16 GHz, achieving a miniaturization rate of 82%. The results recorded in the study using FR-4 were found to be consistent with simulations. In another study [28], pixel antennas with hundreds of connections were divided into density variables using a gradient-based optimization method. In this study, supported by production results, four antennas with 220 connections were optimized to achieve the expected polarization and radiation results. In this scientific research work [29], a microstrip patch antenna with a single high directivity reached a directivity of 12 dBi and a 4% fractional impedance bandwidth using genetic algorithms. It was observed that the genetic MPA resulted in higher bandwidth. In another study [30], a study was conducted to create pioneering data on antenna geometry and performance by combining machine learning methods, including convolutional neural networks and Gaussian regression. By optimizing the antenna performance and geometry, the findings can be easily transferred to future studies.

2.4. Modifications of the Ground Plane

Antenna miniaturization can be achieved through modifications made to the ground plane. In traditional patch antennas, an infinite ground plane is assumed. However, in reality, the ground plane is not infinite. During the miniaturization process, the ground plane size can be reduced to just slightly larger than the patch size. Modified patch antennas have become an option for miniaturization. When this antenna, which has poor polarization purity, is miniaturized, its input impedance changes. Additionally, due to edge diffraction, it generates back lobe radiation, resulting in a decreased front-to-back ratio. In this study [31-33], several numerical measurements and results were shared to assess the accuracy of the effects of a finite ground plane on the antenna. The modification of the ground plane involves not only reducing its size but also adding various slots. The addition of slots increases the current path, thereby lowering the resonance frequency. In this study [34], a single-layer rectangular patch antenna design was realized, and it was experimentally shown that the resonance frequency was significantly reduced using slots. The antenna size was miniaturized by approximately 90%. It was explicitly stated that further increasing the slot length would lead to an additional decrease in the resonance frequency. In addition to these studies, it has been observed that miniaturization was also achieved using defected ground structure (DGS) configurations. These structures can appear as both complex and simpler designs. The DGS improves both the radiation characteristics and the isolation between antenna elements [35].

While these studies have focused on the miniaturization of the antenna, the effects of antenna efficiency, bandwidth, and other parameters on overall antenna performance have not been thoroughly examined.

2.5. Use of metamaterials

Metamaterials are artificially designed materials aimed at providing material properties that are not easily found in the market. These materials are designed to have near-zero relative permeability or negative permeability, or a combination of both. In recent years, many structures resembling these have been developed and researched. Numerous studies have been conducted on miniaturizing antennas and simultaneously improving antenna performance using these structures. Some of these structures are physically difficult to produce, making their fabrication challenging or nearly impossible. In this study [36], simulation studies based on metamaterials were conducted. Miniaturization was achieved through derived equations and the exploration of different conditions. In a more detailed study [37], a cavity model was examined on a heterogeneous substrate. It was found that the use of metamaterials reduced the size of the antenna and improved its performance [36, 37].

3. CONCLUSION

In this study, existing literature on approaches for the miniaturization of patch antennas has been reviewed, aiming to provide detailed information about these approaches. This book chapter begins by defining patch antennas and discussing the Q factor. The Q factor is a definition that helps us understand the impact of reducing antenna size on its properties. The methods used in the literature for miniaturizing patch antennas include material loading, the use of metamaterials, modification of the ground plane, reshaping or introducing slots, and material loading. Each of these methods has its own advantages and disadvantages. While one method may achieve miniaturization at the cost of bandwidth reduction or efficiency loss, another method may allow for less miniaturization while maintaining a balance in other features. Differences among the methods also include ease of manufacturing, cost, and the materials used. The choice of miniaturization technique can depend on the objectives of the intended application. It has been observed that an approach involving theoretical methods combined with various artificial intelligence techniques yields more comprehensive and accurate results. The challenges faced in studies aimed at miniaturizing microstrip antennas can be overcome by integrating different methods. Therefore, as researchers focus on this topic, achieving more miniaturized structures will become feasible.

4. REFERENCES

[1] Sanchez-Hernandez, D., & Robertson, I. D. (1996). A survey of broadband microstrip patch antennas. Microwave journal, 39(9), 60-71.

[2] AL-Amoudi, M. A. (2021). Study, design, and simulation for microstrip patch antenna. International Journal of Applied Science and Engineering Review (IJASER), 2(2), 1-29.

[3]K. Lee and K. Luk, "Microstrip patch antennas," 2011, Accessed: Sep. 05, 2024. [Online]. Available: https://books.google.com/books?hl=tr&lr=&id=Ped4pt-vNUYC&oi=fnd&pg=PR5&dq=Microstrip+Patch+Antennas+(Second+Edition)&ots=POOXZRVDql&sig=UC_-Ilm36bzthkZN_ugKmEo3A6E

[4] Singh, I., & Tripathi, V. S. (2011). Micro strip patch antenna and its applications: a survey. Int. J. Comp. Tech. Appl, 2(5), 1595-1599.

[5] Waterhouse, R. (2018). Microstrip patch antennas. In Handbook of Antennas in Wireless Communications (pp. 6-1). CRC Press.

[6] Mohammed, A. S., Kamal, S., Ain, M. F., Ahmad, Z. A., Ullah, U., Othman, M., ... & Rahman, M. F. A. (2019). Microstrip patch antenna: A review and the current state of the art. Journal of Advanced Research in Dynamical and Control Systems, 11(7), 510-524.

[7] Christina, G. (2021). A review on microstrip patch antenna performance improvement techniques on various applications. Journal of Trends in Computer Science and Smart Technology, 3(3), 175-189.

[8] Ezzulddin, S. K., Hasan, S. O., & Ameen, M. M. (2022). Microstrip patch antenna design, simulation and fabrication for 5G applications. Simulation Modelling Practice and Theory, 116, 102497.

[9] Rana, M. S., & Rahman, M. M. (2022, January). Study of microstrip patch antenna for wireless communication system. In 2022 International Conference for Advancement in Technology (ICONAT) (pp. 1-4). IEEE.

[10] Wheeler, H. A. (1947). Fundamental limitations of small antennas. Proceedings of the IRE, 35(12), 1479-1484.

[11] Sievenpiper, D. F., Dawson, D. C., Jacob, M. M., Kanar, T., Kim, S., Long, J., & Quarfoth, R. G. (2011). Experimental validation of performance limits and design guidelines for small antennas. IEEE Transactions on Antennas and Propagation, 60(1), 8-19.

[12] Volakis, J. L., Chen, C. C., & Fujimoto, K. (2010). Small antennas: miniaturization techniques & applications. (No Title).

[13] Khan, M. U., Sharawi, M. S., & Mittra, R. (2015). Microstrip patch antenna miniaturisation techniques: a review. IET Microwaves, Antennas & Propagation, 9(9), 913-922.

[14] Schaubert, D., & Yngvesson, K. (1986). Experimental study of a microstrip array on high permittivity substrate. IEEE Transactions on Antennas and Propagation, 34(1), 92-97.
[15] Yang, F., & Rahmat-Samii, Y. (2003). Microstrip antennas integrated with electromagnetic band-gap (EBG) structures: A low mutual coupling design for array applications. IEEE transactions on antennas and propagation, 51(10), 2936-2946.
[16] Schaubert, D. H., Pozar, D. M., & Adrian, A. (1989). Effect of microstrip antenna substrate thickness and permittivity: comparison of theories with experiment. IEEE Transactions on Antennas and Propagation, 37(6), 677-682.
[17] Souza, E. A., Oliveira, P. S., D’Assunção, A. G., Mendonça, L. M., & Peixeiro, C. (2019). Miniaturization of a microstrip patch antenna with a koch fractal contour using a social spider algorithm to optimize shorting post position and inset feeding. International Journal of Antennas and Propagation, 2019(1), 6284830.
[18] Vaughan, M. J., Hur, K. Y., & Compton, R. C. (1994). Improvement of microstrip patch antenna radiation patterns. IEEE transactions on antennas and propagation, 42(6), 882-885.
[19] Xu, X., & Wei, J. (2018). Miniaturisation design of patch antenna using a low-profile mushroom type meta-substrate tailored with high permittivity. IET Microwaves, Antennas & Propagation, 12(7), 1216-1221.
[20] Jehangir, S. S., Sharawi, M. S., & Shamim, A. (2018). Highly miniaturised semi-loop meandered dual-band MIMO antenna system. IET Microwaves, Antennas & Propagation, 12(6), 864-871.
[21] Psychoudakis, D., Koh, Y. H., Volakis, J. L., & Halloran, J. H. (2004). Design method for aperture-coupled microstrip patch antennas on textured dielectric substrates. IEEE Transactions on Antennas and Propagation, 52(10), 2763-2766.
[22] Colburn, J. S., & Rahmat-Samii, Y. (1999). Patch antennas on externally perforated high dielectric constant substrates. IEEE Transactions on Antennas and Propagation, 47(12), 1785-1794.
[23] Li, R., DeJean, G., Tentzeris, M. M., & Laskar, J. (2004). Development and analysis of a folded shorted-patch antenna with reduced size. IEEE Transactions on Antennas and Propagation, 52(2), 555-562.
[24] Mohammed, A. S., Kamal, S., Ain, M. F., Ahmad, Z. A., Ullah, U., Othman, M., ... & Rahman, M. F. A. (2019). Microstrip patch antenna: A review and the current state of the art. Journal of Advanced Research in Dynamical and Control Systems, 11(7), 510-524.

[25] Brocker, D. E., Jiang, Z. H., Gregory, M. D., & Werner, D. H. (2016). Miniaturized dual-band folded patch antenna with independent band control utilizing an interdigitated slot loading. IEEE Transactions on Antennas and Propagation, 65(1), 380-384.
[26] Herscovici, N., Osorio, M. F., & Peixeiro, C. (2002). Miniaturization of rectangular microstrip patches using genetic algorithms. IEEE Antennas and wireless propagation letters, 1, 94-97.
[27] Lamsalli, M., El Hamichi, A., Boussouis, M., Touhami, N. A., & Elhamadi, T. (2016). Genetic algorithm optimization for microstrip patch antenna miniaturization. Progress In Electromagnetics Research Letters, 60, 113-120.
[28] Wang, J., Yang, X. S., & Wang, B. Z. (2018). Efficient gradient-based optimisation of pixel antenna with large-scale connections. IET Microwaves, Antennas & Propagation, 12(3), 385-389.
[29] Ayasinghe, J., Anguera Pros, J., & Uduwawala, D. (2013). Genetic algorithm optimization of a high-directivity microstrip patch antenna having a rectangular profile. Radioengineering, 2013. Vol. 22, 3 Setembre.
[30] Wu, Q., Chen, W., Yu, C., Wang, H., & Hong, W. (2024). Machine learning-assisted optimization for antenna geometry design. IEEE Transactions on Antennas and Propagation.
[31] Bhattacharyya, A. K. (1990). Effects of finite ground plane on the radiation characteristics of a circular patch antenna. IEEE transactions on antennas and propagation, 38(2), 152-159.
[32] Huang, J. (1983). The finite ground plane effect on the microstrip antenna radiation patterns. IEEE Transactions on Antennas and Propagation, 31(4), 649-653.
[33] Lier, E., & Jakobsen, K. (1983). Rectangular microstrip patch antennas with infinite and finite ground plane dimensions. IEEE Transactions on Antennas and Propagation, 31(6), 978-984.
[34] Prabhakar, H. V., Kummuri, U. K., Yadahalli, R. M., & Munnappa, V. (2007). Effect of various meandering slots in rectangular microstrip antenna ground plane for compact broadband operation. Electronics Letters, 43(16), 848-850.
[35] Debatosh, G., & Antar, Y. M. (2011). Microstrip and printed antennas: new trends, techniques and applications. First edit. John Wiley & sons ltd.
[36] Mahmoud. (2004). A new miniaturized annular ring patch resonator partially loaded by a metamaterial ring with negative permeability and permittivity. IEEE Antennas and Wireless Propagation Letters, 3, 19-22.

[37] Alù, A., Bilotti, F., Engheta, N., & Vegni, L. (2007). Subwavelength, compact, resonant patch antennas loaded with metamaterials. IEEE Transactions on Antennas and Propagation, 55(1), 13-25.
[38] Werfelli, H., Tayari, K., Chaoui, M., Lahiani, M., & Ghariani, H. (2016, March). Design of rectangular microstrip patch antenna. In 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP) (pp. 798-803). IEEE.

Chapter 7

Machine Learning Classification Techniques Using KNIME Platform

Bashar ALHAJAHMAD[1]
Musa ATAŞ[2]

[1] Assistant Professor, Siirt University, Faculty of Engineering, Department of Computer Engineering. Email: bashar.ahmad@siirt.edu.tr ORCID: 0009-0009-3455-7206
[2] Professor Doctor, Siirt University, Faculty of Engineering, Department of Computer Engineering. Email: musa.atas@siirt.edu.tr ORCID: 0000-0002-1214-3127

ABSTRACT

This study examines the application of machine learning (ML) classification techniques using KNIME, a powerful open-source data analytics platform. As artificial intelligence and machine learning continue to transform data analysis across various industries, understanding the effectiveness of different classification algorithms becomes critical. This research focuses on six widely-used algorithms applied to a cancer dataset from Kaggle: Naive Bayes, Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, and Random Forest. Using KNIME's intuitive workflow interface, we explore the steps of data preparation, model training, and evaluation. The results indicate that SVM achieved the highest accuracy at 99.12%, followed closely by Random Forest and KNN with 98.25%. In contrast, Naive Bayes demonstrated the lowest performance with an accuracy of 92.98%. These findings provide valuable insights for practitioners seeking to utilize ML for predictive analytics in real-world applications, highlighting the strengths and weaknesses of each algorithm.

Keywords: Artificial Intelligence, Machine Learning, KNN, Random Forest, SVM, Decision Tree, Logistic Regression.

1. INTRODUCTION

Artificial Intelligence (AI), particularly Machine Learning (ML), has rapidly advanced in recent years, especially in the context of data analysis and computing . ML enables systems to learn and improve from experience without being explicitly programmed, making it a prominent technology of the Fourth Industrial Revolution (Industry 4.0) (Sarker et al., 2020, Ataş, 2023a).

The effectiveness of an ML solution largely depends on the nature of the data and the performance of the learning algorithms (Michael et al.,2009). In ML, various tasks—such as classification, regression, clustering, feature engineering, dimensionality reduction, and reinforcement learning—are employed to build data-driven systems (Han et al.,2011).

KNIME, a versatile platform for data analysis, reporting, and integration, integrates numerous components for ML and data mining (KNIME, web page). Developed initially by a team of software engineers at the University of Konstanz, led by Michael Berthold, KNIME has evolved into a modular, scalable, open data processing platform (Lalit K., 2013). Its intuitive drag-and-drop workflow interface, alongside its strong visualization capabilities, has made it a popular choice among professional data analysts (Kalpana et al., 2014, Ataş, 2024).

This study aims to provide a comprehensive analysis of various ML classification techniques and their potential applications in real-world scenarios, using KNIME as the primary tool for implementation. We focus on six classification algorithms—Naive Bayes, SVM, Logistic Regression, KNN, Decision Tree, and Random Forest—applied to a cancer dataset obtained from Kaggle.

2. FEATURES OF KNIME

KNIME is a comprehensive analytics and data mining tool that utilizes an intuitive drag-and-drop workflow interface. It is a "low-code or no-code" platform that is excellent for predictive analytics and data mining, used by both professional data analysts and non-programmers. KNIME's graphical workflows represent each step of the analysis, making the process self-documenting (Acito, 2023). Key features that distinguish KNIME from its competitors include:

- Free to use on personal devices.
- Compatible with Windows, macOS, and Linux.
- Over 4000 nodes for data connections, transformations, ML, and visualization.
- Extensible via R or Python integration.
- Support for various file formats such as CSV, Excel, JSON, XML, and

even unstructured data like images and audio.

- Seamless integration with platforms like H2O and WEKA for additional analytics capabilities.
- Compatibility with reporting platforms such as Excel, Tableau, Spotfire, and Power BI.
- Large, active user community offering assistance and ready-to-use workflows.

While KNIME provides an open-source desktop application, its commercial version offers web deployment features. However, the open-source version includes all essential functionalities required for desktop analytics.

3. MACHINE LEARNING TASKS AND ALGORITHMS

This section explores various ML algorithms commonly used in classification, regression, clustering, association rule learning, and dimensionality reduction (Sarker, 2021, Ataş, 2023b). Predictive models, like the one shown in Figure 1, are trained using historical data (Step 1) and generate predictions for new test data (Step 2).

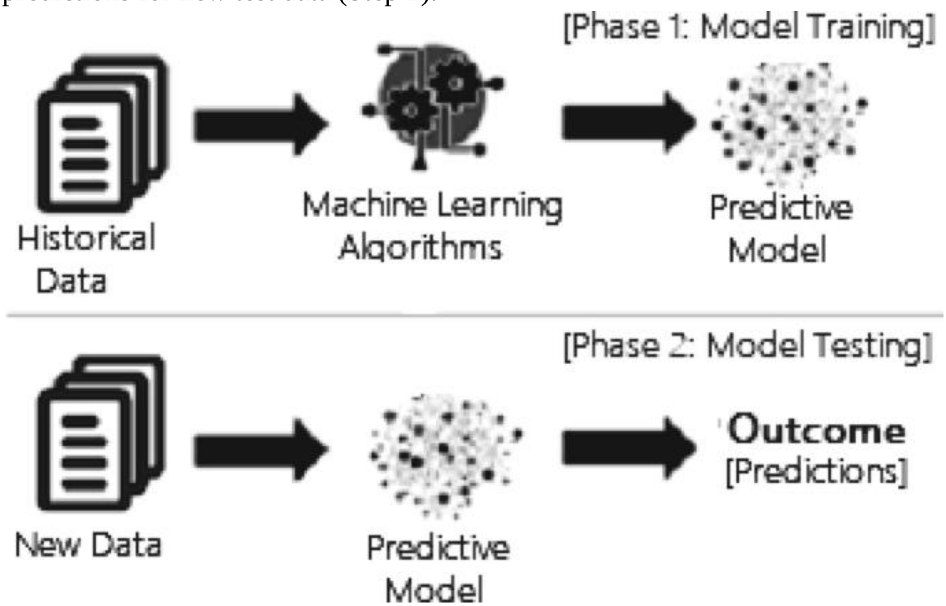


Figure 1: A general structure of a machine learning based predictive model considering both the training and testing phase

Each algorithm used in this analysis was chosen for its unique approach to classification, offering a wide spectrum of techniques from probabilistic methods (Naive Bayes) to ensemble learning (Random Forest) (Witten et al.,2005). This selection provided a comprehensive comparative analysis of model behavior under varying assumptions and methodologies:

- **Naive Bayes**: Based on Bayes' Theorem, this algorithm assumes independence between each pair of features, making it particularly effective for binary or multi-class classification problems such as spam filtering (John et al.,1995). Naive Bayes employs the assumption of feature independence, making it computationally efficient but potentially limited in capturing complex feature interactions.
- **Logistic Regression**: Logistic Regression is a parametric model that estimates the probability of class membership using the logistic function (also known as the sigmoid function). It is ideal for binary classification tasks where the relationships between features and the target variable are linear (LeCessie et al.,1992).
- **K-Nearest Neighbors (KNN)**: A lazy learning algorithm that classifies new data points based on similarity measures, often used in pattern recognition problems (Aha et al., 1991). KNN is a non-parametric method that relies on the proximity of data points in feature space. Its performance is heavily influenced by feature scaling and the choice of the "k" parameter.
- **Support Vector Machines (SVM):** SVM constructs a hyperplane in a high-dimensional space to separate data points from different classes, offering strong performance, especially when the margin between classes is maximized (Keerthi et al.,2001). SVM is particularly effective in cases with high-dimensional data, offering strong generalization by maximizing the margin between classes.
- **Decision Trees (DT):** Non-parametric supervised learning algorithms used for classification and regression tasks, known for their interpretability and ability to handle complex datasets (Pedregosa et al., 2011). Decision Trees (DT), which recursively split the dataset based on feature values, provide a highly interpretable model but are prone to overfitting, especially on smaller datasets.
- **Random Forest (RF):** An ensemble method that trains multiple decision trees on various sub-samples of the data and aggregates their predictions, resulting in improved accuracy and reduced overfitting (see Figure 2) (Breiman, 2001).

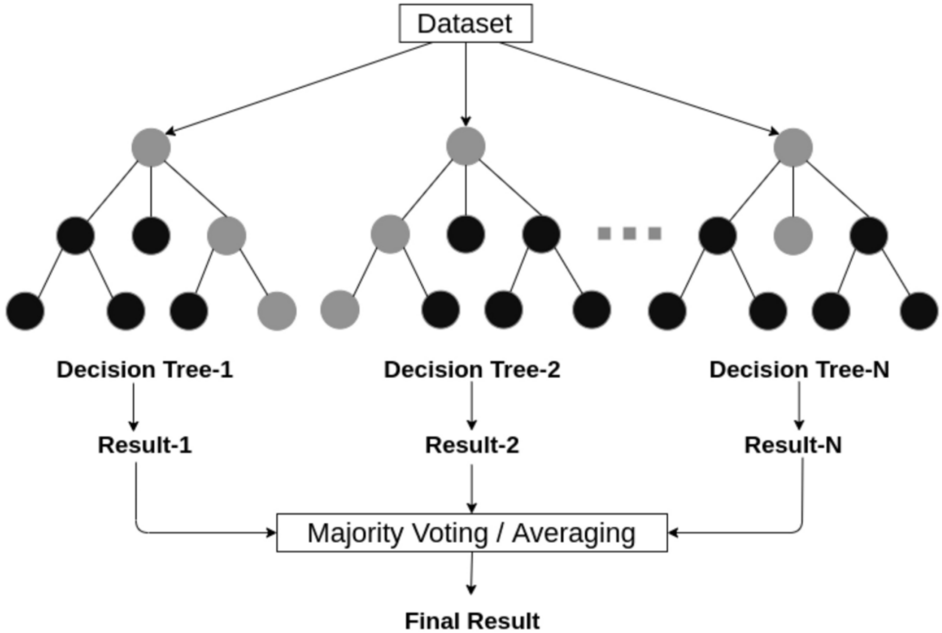


Figure 2: An example of a random forest structure considering multiple decision trees

4. DATA MODELING AND EVALUATION WITH KNIME

4.1 Dataset Description

The dataset used in this study, sourced from Kaggle, includes features of patients diagnosed with cancer (Cancer Data, web page). The dataset, containing patient data with diagnostic features of malignant and benign tumors, was first preprocessed to ensure quality input for machine learning algorithms. Key attributes of the dataset include:

- **id:** Unique identifier for each patient.
- **diagnosis**: Target variable indicating cancer type, either "M" (Malignant) or "B" (Benign).
- **radius_mean, texture_mean, perimeter_mean,** etc.: Visual characteristics of the cancer, represented as average values.

Essential preprocessing steps, such as normalization, were applied to standardize the data, making it suitable for algorithms sensitive to feature scaling, like KNN and SVM.

4.2 Implementation Process

The following steps outline the process of applying different classification algorithms in KNIME (Bashar et al., 2023):

1. **Data Preparation:** The dataset is imported using the "CSV Reader" node.

If necessary, feature scaling is applied via the "Normalizer" node.

2. **Data Splitting:** The data is divided into training (70%) and test (30%) sets using the "Partitioning" node, preparing it for further analysis.

3. **Model Training:** Different machine learning models are trained using KNIME's "Learner" nodes. In this study, we trained six classification models: Naive Bayes, SVM, Logistic Regression, KNN, Decision Tree, and Random Forest.

4. **Prediction:** After training, the models are tested on the test data to predict the outcome using the appropriate "Predictor" nodes for each classification algorithm.

5. **Model Evaluation:** The models' performance is evaluated using the "Scorer" node, which provides a confusion matrix and statistical metrics such as accuracy and error rates.

Figures 3 to 8 illustrate the various workflows used for classification in our study, showcasing each of the applied models. Each workflow includes the necessary learner and predictor components, as well as the scorer element, which is essential for displaying the model's accuracy and confusion matrix.

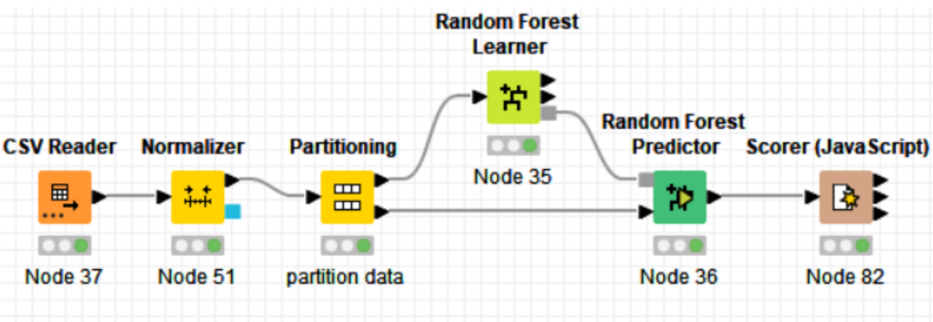


Figure 3: Random Forest classification workflow using Knime

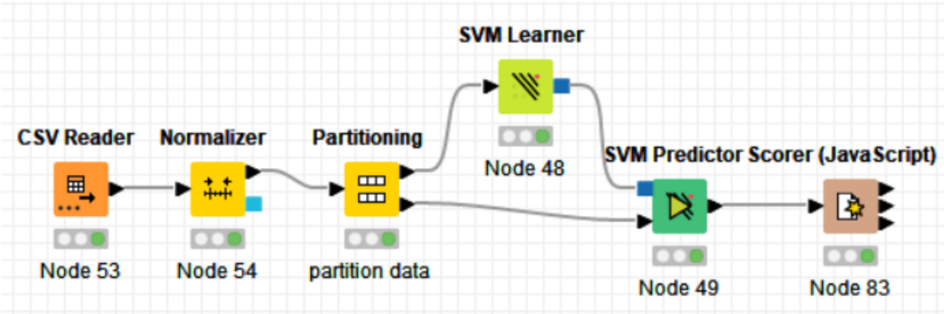


Figure 4: SVM classification workflow using Knime

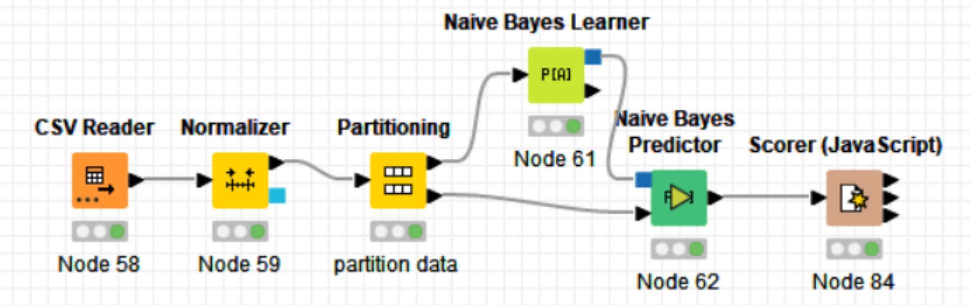


Figure 5: Naïve Bayes classification workflow using Knime

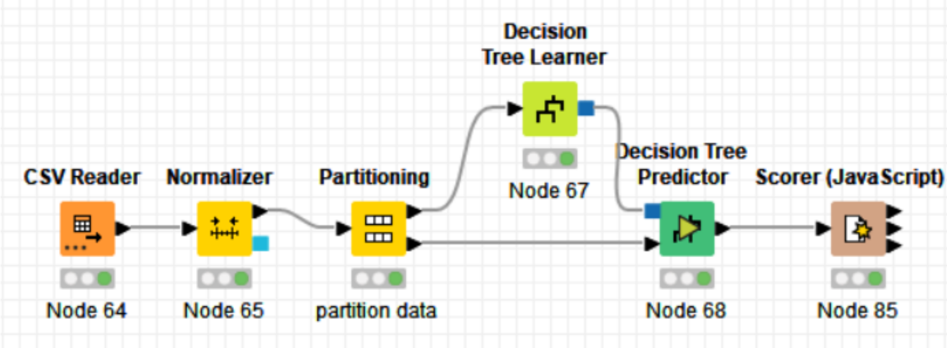


Figure 6: Decision Tree classification workflow using Knime

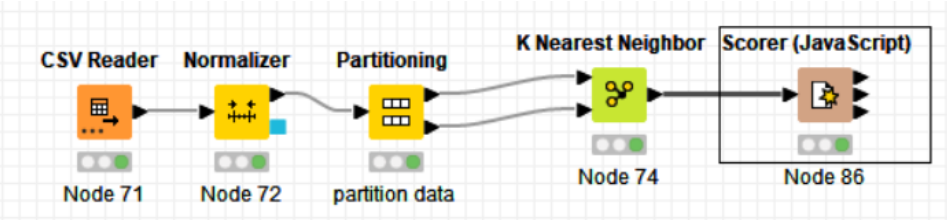


Figure 7: KNN classification workflow using Knime

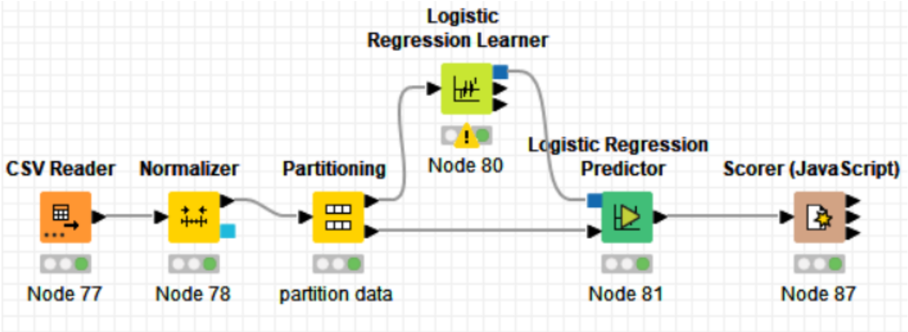


Figure 8: Logistic Regression classification workflow using Knime

5. MODEL PERFORMANCE EVALUATION

Figures 9 through 14 illustrate the confusion matrices and accuracy rates of the machine learning models evaluated in KNIME. The results show that the SVM algorithm achieved the highest accuracy at 99.12%, followed by Random Forest and KNN with 98.25% accuracy. Decision Tree and Logistic Regression performed moderately well, with 95.61% and 96.49% accuracy, respectively. Naive Bayes, although simple and fast, recorded the lowest accuracy at 92.98%.

Scorer View

Confusion Matrix

	B (Predicted)	M (Predicted)	
B (Actual)	73	1	98.65%
M (Actual)	1	39	97.50%
	98.65%	97.50%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
98.25%	1.75%	0.961	112	2

Figure 9: Confusion Matrix, Accuracy and Error rate for Random Forest Model

Scorer View

Confusion Matrix

	B (Predicted)	M (Predicted)	
B (Actual)	76	1	98.70%
M (Actual)	0	37	100.00%
	100.00%	97.37%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
99.12%	0.88%	0.980	113	1

Figure 10: Confusion Matrix, Accuracy and Error rate for SVM Model

Scorer View

Confusion Matrix

	B (Predicted)	M (Predicted)	
B (Actual)	70	4	94.59%
M (Actual)	4	36	90.00%
	94.59%	90.00%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
92.98%	7.02%	0.846	106	8

Figure 11: Confusion Matrix, Accuracy and Error rate for Naïve Bayes Model

Scorer View

Confusion Matrix

	B (Predicted)	M (Predicted)	
B (Actual)	69	3	95.83%
M (Actual)	2	40	95.24%
	97.18%	93.02%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
95.61%	4.39%	0.906	109	5

Figure 12: Confusion Matrix, Accuracy and Error rate for Decision Tree Model

Scorer View

Confusion Matrix

	B (Predicted)	M (Predicted)	
B (Actual)	75	1	98.68%
M (Actual)	1	37	97.37%
	98.68%	97.37%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
98.25%	1.75%	0.961	112	2

Figure 13: Confusion Matrix, Accuracy and Error rate for KNN Model

Scorer View

Confusion Matrix

	B (Predicted)	M (Predicted)	
B (Actual)	74	2	97.37%
M (Actual)	2	36	94.74%
	97.37%	94.74%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
96.49%	3.51%	0.921	110	4

Figure 14: Confusion Matrix, Accuracy and Error rate for Logistic Regression Model

6. MODEL TRAINING AND TESTING PROCESS

After preparing the data, the models were trained using KNIME's intuitive "Learner" nodes, which implement the learning algorithms. The training phase involved fitting the models to the training data and tuning hyperparameters where applicable. For instance, in SVM, the choice of kernel (linear or radial basis function) and regularization parameter (C) were crucial in optimizing performance, while in Random Forest, the number of trees and the depth of each tree were pivotal in balancing bias and variance. The trained models were then applied to the testing set using KNIME's "Predictor" nodes to generate predictions.

6.1 Model Evaluation and Results

Evaluation metrics, such as accuracy, precision, recall, F1-score, and confusion matrices, were used to assess model performance. These metrics provide a nuanced understanding of each model's strengths and limitations:

- SVM emerged as the top-performing algorithm with an accuracy of 99.12%, reflecting its robustness in handling complex data distributions. Its ability to maximize the margin between classes likely contributed to its high classification performance, especially in distinguishing between malignant and benign tumors.
- Random Forest and KNN followed closely with accuracies of 98.25%. Random Forest benefited from its ensemble nature, where the aggregation of multiple decision trees minimized overfitting and enhanced predictive stability. KNN, being sensitive to feature space, performed well after data normalization, although its performance may degrade with high-dimensional or sparse data.
- Logistic Regression and Decision Trees performed moderately well, achieving 96.49% and 95.61% accuracy, respectively. Logistic Regression's linear assumptions, while effective in this case, may not capture more intricate relationships between features, which could limit its performance in non-linear datasets. Decision Trees, though interpretable and easy to visualize, suffered from potential overfitting due to their hierarchical splitting structure, which often results in high variance if not pruned or aggregated (as in Random Forest).
- Naive Bayes showed the lowest accuracy at 92.98%, which is consistent with its simplifying assumption of feature independence. While computationally efficient and suitable for certain types of problems (e.g., text classification), Naive Bayes is generally outperformed by models that can capture more complex relationships between features, as was evident in this dataset.

6.2 Interpretation of Results and Practical Implications

The superior performance of SVM in this experiment highlights its utility for classification tasks where data points are well-separated and the goal is to maximize the decision boundary's margin. SVM's performance, however, is highly dependent on the choice of the kernel function, indicating that parameter tuning plays a critical role in optimizing the model. Its computational complexity also becomes a consideration when scaling to large datasets or high-dimensional feature spaces.

Random Forest's strong performance underscores the effectiveness of ensemble learning techniques in improving classification accuracy through model averaging, reducing the risk of overfitting that typically affects single decision trees. KNN's relatively high accuracy, despite being a simple algorithm, suggests that proximity-based learning remains effective in low-dimensional, well-normalized datasets, though it can become computationally expensive as the dataset grows.

Naive Bayes, while efficient, is limited by its strong independence assumption, making it less suitable for datasets with feature interactions, such as the cancer dataset used in this study. Nevertheless, its simplicity and speed make it a useful baseline model, especially in scenarios where computational resources are constrained.

7. FUTURE DIRECTIONS AND RECOMMENDATIONS

This analysis emphasizes the importance of algorithm selection based on the specific characteristics of the dataset. While SVM and ensemble methods like Random Forest performed well on this dataset, the choice of model should always consider factors such as dataset size, feature complexity, and computational resources. In practice, ensemble methods are particularly useful when interpretability is less critical, and maximizing predictive performance is the priority. Conversely, for applications requiring model interpretability (e.g., medical diagnosis), decision trees may still be favored, provided overfitting is addressed through techniques like pruning or ensemble learning.

Future research could extend this study by applying these algorithms to other medical datasets or incorporating additional techniques such as feature engineering, deep learning, or hybrid models that combine the strengths of multiple algorithms. Moreover, exploring advanced hyperparameter tuning methods, such as grid search or Bayesian optimization, may further improve the performance of machine learning models in specific applications.

8. CONCLUSION

In this study, we have demonstrated the application of various machine learning classification techniques using KNIME, a powerful and user-friendly data analytics platform. By focusing on six widely-used classification algorithms—Naive Bayes, Support Vector Machines (SVM), Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, and Random Forest—we applied these methods to a cancer dataset sourced from Kaggle, evaluating their respective performance.

The results highlight SVM's superior performance, achieving a 99.12% accuracy rate, while Random Forest and KNN also performed strongly. Naive Bayes, while fast, was the least accurate, with an accuracy of 92.98%. These findings underscore the importance of selecting appropriate algorithms based on the characteristics of the dataset and the application context.

9. REFERENCES

1. Acito, F. (2023). *Predictive analytics with KNIME: Analytics for citizen data scientists*. Switzerland: Springer.
2. Aha, D. W., Kibler, D., & Albert, M. (1991). *Instance-based learning algorithms*. *Machine Learning, 6*(1), 37–66.
3. Alhajahmad, B., Ataş, M., & Ataş, İ. (2023). *Utilizing KNIME as a Machine Learning Classification Tool: A Practical Demonstration with SVM, Random Forest, and Naive Bayes*. *2nd International Conference on Recent Academic Studies (ICRAS 2023)*, October 19-20, 2023, Konya, Turkey.
4. Ataş, İ., "Comparison of deep convolution and least squares GANs for diabetic retinopathy image synthesis". Neural Comput & Applic 35, 14431–14448, 2023a. https://doi.org/10.1007/s00521-023-08482-4
5. Ataş, İ., "Performance Evaluation of Jaccard-Dice Coefficient on Building Segmentation from High Resolution Satellite Images". Balkan Journal of Electrical and Computer Engineering, 11(1), 100-106, 2023b. https://doi.org/10.17694/bajece.1212563.
6. Ataş, İ., "The Effect of Latent Space Vector on Generating Animal Faces in Deep Convolutional GAN: An Analysis", DUJE, vol. 15, no. 1, pp. 99–106, 2024, doi: 10.24012/dumf.1393797.
7. Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., et al. (2009). *KNIME - The Konstanz Information Miner*. University of Konstanz Nycomed Chair for Bioinformatics and Information Mining, 11.
8. Breiman, L. (2001). *Random forests*. *Machine Learning, 45*(1), 5–32.
9. Cancer Data. (n.d.). Available at: https://www.kaggle.com/datasets/erdemtaha/cancer-data
10. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: Concepts and techniques*. Amsterdam: Elsevier.
11. John, G. H., & Langley, P. (1995). *Estimating continuous distributions in Bayesian classifiers*. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 338–345). Morgan Kaufmann Publishers Inc.
12. Kataria, L. (2013). *Implementation of Knime-Data Mining Tool*. *International Journal of Advanced Research in Computer Science and Software Engineering, 3*(11).
13. Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Krishna, M. K. R. (2001). *Improvements to Platt's SMO algorithm for SVM classifier design*. *Neural Computation, 13*(3), 637–649.

14. KNIME.com AG. (n.d.). *KNIME web site*. Retrieved from https://www.knime.com/
15. LeCessie, S., & Van Houwelingen, J. C. (1992). *Ridge estimators in logistic regression*. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 41*(1), 191–201.
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research, 12*, 2825–2830.
17. Rangra, K., & Bansal, K. L. (2014). *Comparative Study of Data Mining Tools*. *International Journal of Advanced Research in Computer Science and Software Engineering, 4*(6).
18. Sarker, I. H. (2021). *Machine learning: Algorithms, real-world applications and research directions*. *SN Computer Science, 2*(160).
19. Sarker, I. H., Hoque, M. M., Uddin, M. K., & Tawfeeq, A. (2020). *Mobile data science and intelligent apps: Concepts, AI-based modeling and research directions*. *Mobile Networks and Applications*, 1–19.

Chapter 8

Investigation of Tribological Properties of AISI 4140 Steel in Different Environmental Conditions

M. Tayyip ÖZDEMİR[1]
M. Salih GÜL[2]

[1] Res. Asst., Karabuk University, Faculty of Engineering, Department of Mechanical Engineering, Karabuk, Türkiye, tayyipozdemir@karabuk.edu.tr
[2] Asst. Prof., Karabuk University, Faculty of Engineering, Department of Mechanical Engineering, Karabuk, Türkiye, msalihgul@karabuk.edu.tr

INTRODUCTION

Tempered steels; These are steels that show high toughness in shrinkage as a result of the tempering process and are suitable for hardening in terms of the amount of carbon in their chemical composition (Jiang et al., 2017). The most widely used steel type among the tempered steels used in the machinery and automotive industry is AISI 4140 steel (Murwamadala & Rao, 2024). AISI 4140 steel is also called forged quality steel, alloy steel, low alloy structural steel and medium carbon steel. The most important feature of this steel material is that it can form a hard martensitic structure after quenching due to the Mo and Cr alloy elements it contains, thus allowing the properties such as toughness and strength to be provided together (Yalçin et al., 2024). For this reason, AISI 4140 steel is always a material with a wide range of uses (Hajian et al., 2024). Areas of use include aircraft and automobile manufacturing; high ductility parts such as axle shafts, crankshafts, splined shafts and similar parts, bandages and gears. They are also used in many places such as turbine engines, springs, machine steels, starter gears, ship chains and irons, railway wheels and shafts, cold drawn shafts and rods, brake rings and arms of turbo generators. One of the main reasons why AISI 4140 steel has such a wide range of uses is its high resistance to wear. These parts, especially those operating under heavy loads and challenging conditions, require materials with high wear resistance, which makes AISI 4140 an ideal choice.

Wear is defined as the breaking away of micro particles from the surface as a result of mechanical factors, causing undesirable changes in the material (Moshkovich et al., 2024) and this phenomenon is a process as old as human history. Wear, with its different mechanisms, significantly reduces the life of machine elements and engineering materials in many industrial sectors such as mining, agriculture, and machinery, causing great economic losses. Today's industrially developed countries are also making significant research expenditures to eliminate this material defect. Wear is not a property of the material itself, but a property of the engineering system. Wear has become a technical problem as well as a matter of economic interest. In technical terms, wear is the event of an undesirable change in the material as a result of the breaking and separation of micro particles on the surfaces of objects due to mechanical causes or mechanical energy. In order for the wear event in the material to be considered as wear, there must be a mechanical effect, friction, it must be slow but continuous, it must cause a change on the surface of the material, and it must occur unintentionally. In general, wear occurs in bearings, brakes, pistons, gears, crushing and grinding mills, road, soil and agricultural machinery, turbine blades, ore production devices and similar places. Although wear, which has a significant effect on the life of various machine elements and engineering materials, is a material problem that cannot be completely eliminated, it has been seen as a result of recent studies that wear characteristics depend on the type of material. In today's industry, in addition to production economy, workpiece quality, processing times and produced quantities, operator health and environmental protection are also of great importance. Cutting fluids are used to reduce the high heat generated during wear. However, the chemical components contained in cutting fluids carry contact risks that can harm human health and errors in waste management can have negative effects on the environment. Reducing the use of cutting fluids due to their harmful effects and high costs can negatively affect product quality and tool life. The MQL method cools the cutting tool by spraying oil droplets into the cutting area with compressed air, improves surface quality, removes chips and helps prevent tool wear (Gürbüz et al., 2017). Unlike other cooling methods, the MQL system provides both cooling and lubrication by sending a minimum amount of cutting fluid between the cutting tool and the workpiece. Water-based cooling processes are 3 to 4 times more efficient than MQL and reduce costs because they do not require pumps, cooling or filtration during cooling (Furness et al., 2017).

The MQL system uses cutting oil for lubrication and compressed air for cooling. These systems are suitable for both water-based and oil-based cutting fluids. The small amount of cutting fluid provided allows for an almost dry-processing-like operation. Since large amounts of fluid are not used, environmental pollution is reduced and costs are also reduced. MQL systems can be used in conventional machining operations such as turning, milling and drilling. Typically, between 20 and 150 ml/h of fluid is consumed during the process, and almost all of this fluid is used. MQL operates by relying on the lubricant delivered to the cutting zone. While some machining processes require external spraying, other processes require internal spraying. Internal spray systems can be more costly than external spray. External spray is applied directly to the cutting tool and workpiece surface. Dual-channel systems are designed to provide separate air and lubricant, so that the two components are mixed at or near the cutting edge. Internal spray systems, like external spray, can be used as single-channel or dual-channel.

In a single-channel system, the aerosol is generated at the applicator or at a point close to the cutting process. In a dual-channel system, the air and oil are delivered in separate ways and the aerosol is generated directly in front of the tool. The main advantages of external spray systems are their low cost and simple installation. However, the manual adjustment and positioning of these systems is a disadvantage, as they require constant adjustment to accommodate different length tools. In addition, losses can occur due to the dispersion of the lubricant as it is delivered to the cutting area. Despite these disadvantages, external spray systems have been used successfully in many machining processes.

The internal spray method provides direct lubricant delivery to the cutting zone via the machine spindle, tool holder and tool, and is generally preferred in high-speed operations. This system ensures that the aerosol reaches the cutting zone consistently and precisely. Better performance can be achieved with both internal and external sprays in dual-channel systems, but it is important that the channels are well protected. In machines designed for MQL systems, separate channels are carried to the spindle and run along the spindle. The lubricant is delivered to the tool holder via a distribution tube from the spindle. Air is transferred through a chamber surrounding the tube through which the fluid passes. In machines not specifically designed for MQL, the mixture is made at a point near the end of the spindle and the aerosol is delivered to the cutting zone via the tool. Previous studies on the MQL method have demonstrated the effectiveness and advantages of this technology, providing significant results such as reducing wear and improving surface quality in cutting processes; therefore, based on the findings in the literature, it would be useful to review previous research to better understand the potential of MQL.

Varadarajan et al. (Varadarajan et al., 2002) concluded that in minimum quantity lubrication, the cutting fluid penetrates into the workpiece surface and adheres tightly to the chip back face, contributing to the increase in plastic deformation due to the "Rebinder effect". This contributes to the curling of the chip by reducing the compressive stresses to some extent, thus reducing the tool-chip contact length. According to recent studies, the tool-chip contact length in cutting is shortest in minimum quantity lubrication; longer in wet cutting and longest in dry cutting. The cutting tool temperature is lower in minimum quantity lubrication due to the cooling occurring in both convection and evaporation forms. When compared to dry and wet cutting, the cutting forces are also lower. Liao et al. (Liao et al., 2007), investigated the milling of NAK80 steel with coated carbide tool using MQL. They also conducted pool cooling and dry cutting experiments for comparison. It was found that using MQL provided the best performance.

2. MATERIAL AND METHODS

In this study, wear test was applied to investigate the wear behavior of AISI 4140 stainless steel. Test conditions and sample preparation procedures are detailed below.

2.1. Sample Preparation

The chemical composition table of the AISI 4140 sample is shown in Table 1.

Table 1. Chemical composition of AISI 4140 steel.

Material	C	Si	Mn	P	S	Cr	Mo
AISI 4140	0.38-0.45	0.40	0.5-0.8	0.035	0.035	0.9-1.2	0.15-0.30

The test sample was prepared with dimensions of Ø35 × 10 mm. The surface of the sample was first sanded with 600 grit and then with 1200 grit sandpaper.

2.2. Experimental Environment Condition

Wear tests were carried out in two different environments: dry and MQL. Mackerel Ms brand oil was used as coolant and lubricant in the MQL environment. There was no coolant in the tests carried out in the dry environment. The tests were carried out at room temperature (approximately 23 ± 2°C) and relative humidity ranging from 40-60%. The MQL system was applied in spray form with a flow rate of 40 mL/h and an air pressure of 5 bar.

2.3. Wear Test

Experiments were carried out in a ball-on-flat wear tester conforming to ASTM G133 standards to investigate the tribological performance of AISI 4140 steel. The stroke length of the wear tester was determined as 10 mm and the total wear distance as 100 m. The experiments were carried out at a constant speed of 50 mm/s and a load of 40 N. Tungsten carbide (WC) was used as the abrasive ball.

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

3.1. Evaluation of Friction Force and Coefficient

As a result of the wear tests, friction forces were obtained. The graph of the friction forces obtained is given in Figure 1. The average friction force value obtained in the experiments carried out in a dry environment was calculated as 33.61 N. The average friction force value obtained in the experiments carried out in MQL was 16.78.

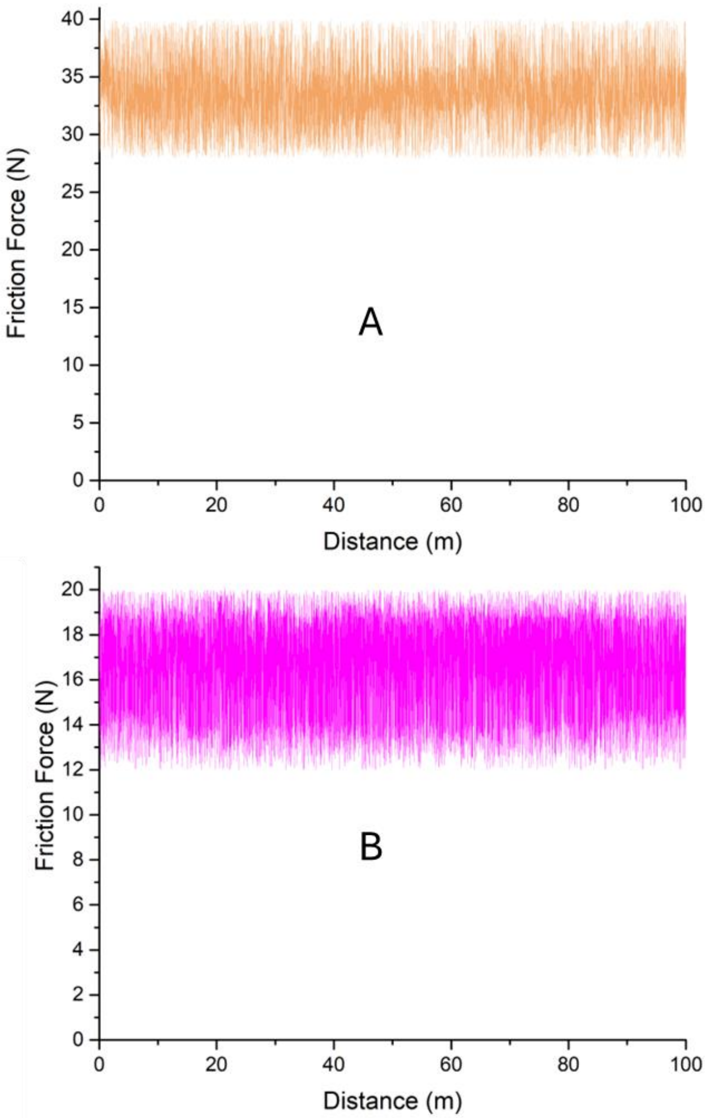


Figure 1. Friction force under different conditions a) dry, b) MQL

The friction coefficient values are given in Figure 2. While the average friction coefficient value obtained in the experiments conducted in a dry environment was 0.84, this value was 0.42 in MQL. As a result, there was a 50.07% decrease in the friction coefficient value in MQL compared to the dry environment.

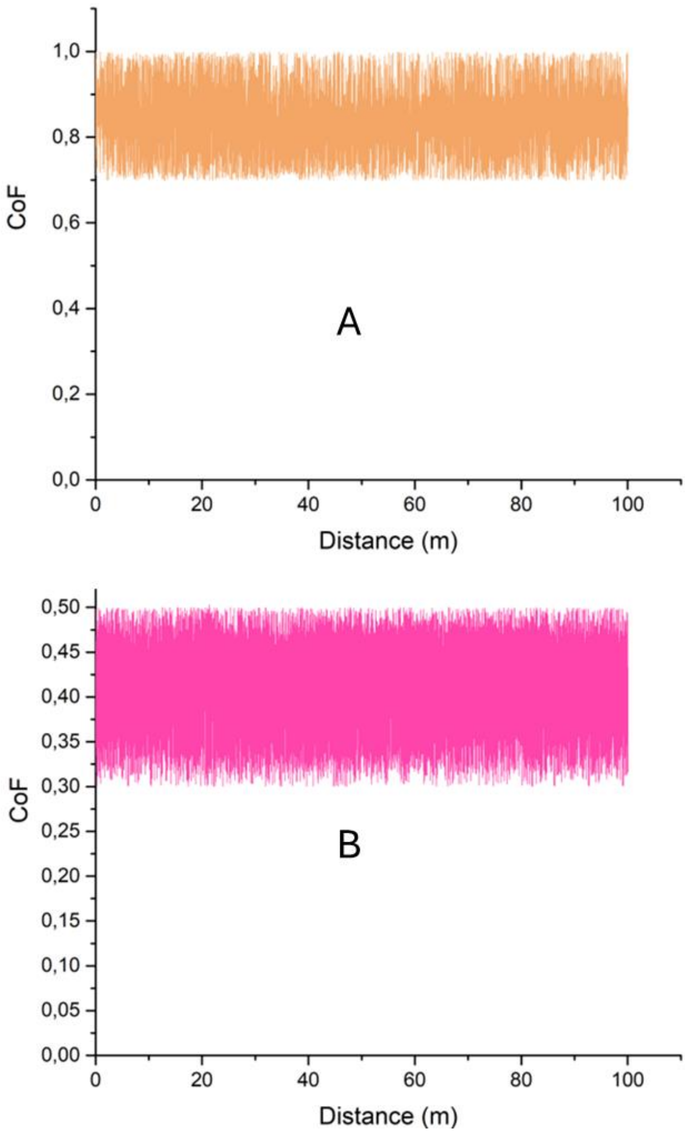


Figure 2. Friction coefficient under different conditions a) dry, b) MQL

When the results are examined in general, the wear tests performed in MQL showed better performance. This result can be supported by the study of Chuangwen et al. (Chuangwen et al., 2017). MQL enters the wear point and forms a thin oil film on the sample surface. This film reduces friction (Dhar et al., 2006; Yin et al., 2021).

3.2. Evaluation of Wear Depths

Many parameters are examined when determining tribological properties. One of these is the wear depth. The wear depth graph of the test samples is given in Figure 3. When Figure 3 is examined, the wear depth value obtained in the dry environment was obtained as 13.03 μm. This value decreased by 50.29% in MQL. This shows that MQL reduces the wear depth. In dry environments, the heat generated during wear is limited to dissipate, which can lead to faster deformation of the material and therefore more wear. MQL, on the other hand, prevents overheating by controlling the temperature in the wear area thanks to the cooling effect of the oil (Eltaggaz et al., 2023). This reduces material deformation, surface deterioration and therefore wear depth.

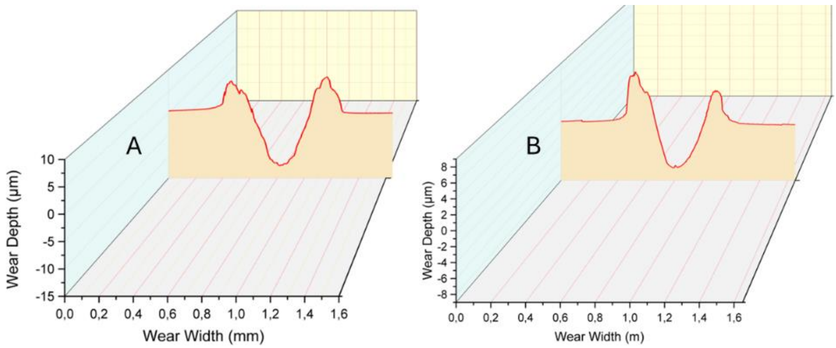


Figure 3. Wear depths in different environments a) dry, b) MQL

4. RESULTS

In this study, the tribological behavior of AISI 4140 stainless steel under different environmental conditions was investigated. The wear test was carried out in a ball-on-flat tester. The following results were obtained in the experiments carried out under dry and MQL environmental conditions:

- With the use of MQL, there was a significant decrease in the friction force and coefficient values. There was a 50.07% decrease in friction force in MQL compared to the dry environment.
- There was a 50.29% decrease in wear depth when passing from dry environment to MQL.
- MQL gave very good result on friction coefficient and wear depth.
- The use of the MQL method is important as one of the most ideal methods for lubricant costs, worker health and sustainability.

REFERENCES

Chuangwen, X., Ting, X., Huaiyuan, L., Zhicheng, S., Hongbing, J., & Mandong, L. (2017). Friction, wear, and cutting tests on 022Cr17Ni12Mo2 stainless steel under minimum quantity lubrication conditions. *The International Journal of Advanced Manufacturing Technology*, *90*(1–4), 677–689. https://doi.org/10.1007/s00170-016-9406-6

Dhar, N. R., Kamruzzaman, M., & Ahmed, M. (2006). Effect of minimum quantity lubrication (MQL) on tool wear and surface roughness in turning AISI-4340 steel. *Journal of Materials Processing Technology*, *172*(2), 299–304. https://doi.org/10.1016/j.jmatprotec.2005.09.022

Eltaggaz, A., Ali, S., Badwal, K., & Deiab, I. (2023). Influence of nanoparticle concentration in nanofluid MQL on cutting forces, tool wear, chip morphology when milling of Inconel 718. *The International Journal of Advanced Manufacturing Technology*, *129*(3–4), 1787–1800. https://doi.org/10.1007/s00170-023-12393-y

Furness, R., Shih, A., Stephenson, D., & Tai, B. (2017). Minimum quantity lubrication for sustainable machining. *Encyclopedia of Sustainable Technologies*, 477–485.

Gürbüz, H., Baday, Ş., & Gönülaçar, Y. E. (2017). Minimum Miktarda Yağlamanın Frezeleme İşlemleri Üzerine Etkisinin Araştırılması:Derleme. *Batman Üniversitesi Yaşam Bilimleri Dergisi*, *7*(2/2), 59–78.

Hajian, M., Ghanbari Adivi, B., Jamali, M., & Mofid, M. A. (2024). Investigating the Microstructure and Wear Properties of AISI 4140 Steel Clad with WC-Ti Composite Particles Via Gas Tungsten Arc Welding. *JOM*, *76*(6), 2824–2834. https://doi.org/10.1007/s11837-024-06471-z

Jiang, B., Wu, M., Zhang, M., Zhao, F., Zhao, Z., & Liu, Y. (2017). Microstructural characterization, strengthening and toughening mechanisms of a quenched and tempered steel: Effect of heat treatment parameters. *Materials Science and Engineering: A*, *707*, 306–314. https://doi.org/10.1016/j.msea.2017.09.062

Liao, Y. S., Lin, H. M., & Chen, Y. C. (2007). Feasibility study of the minimum quantity lubrication in high-speed end milling of NAK80 hardened steel by coated carbide tool. *International Journal of Machine Tools and Manufacture*, *47*(11), 1667–1676. https://doi.org/10.1016/j.ijmachtools.2007.01.005

Moshkovich, A., Lapsker, I., & Rapoport, L. S. (2024). The Effect of Plastic Deformation on the Flattening of Friction Surfaces. *Lubricants*, *12*(8), 276. https://doi.org/10.3390/lubricants12080276

Murwamadala, R. D., & Rao, V. V. (2024). Wear performance of AISI 4140 low-alloy steel PVD coated with TiN. *Advances in Materials and Processing Technologies*, *10*(2), 971–987. https://doi.org/10.1080/2374068X.2023.2185439

Varadarajan, A. S., Philip, P. K., & Ramamoorthy, B. (2002). Investigations on hard turning with minimal cutting fluid application (HTMF) and its comparison with dry and wet turning. *International Journal of Machine Tools and Manufacture, 42*(2), 193–200. https://doi.org/10.1016/S0890-6955(01)00119-5

Yalçin, E. D., Güler, O., Karabacak, A. H., Çanakçi, A., Varol, T., Kocaman, M., Çelebi, M., & Akçay, S. B. (2024). Pre-hardening water quenching effect on the wear resistance of TiCN coated 4140 steels. *Surface and Coatings Technology*, *480,* 130591. https://doi.org/10.1016/j.surfcoat.2024.130591

Yin, Q., Li, C., Dong, L., Bai, X., Zhang, Y., Yang, M., Jia, D., Li, R., & Liu, Z. (2021). Effects of Physicochemical Properties of Different Base Oils on Friction Coefficient and Surface Roughness in MQL Milling AISI 1045. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *8*(6), 1629–1647. https://doi.org/10.1007/s40684-021-00318-7

Chapter 9

The Use of Artificial Intelligence in the Analysis Process of Software Project Management

Fatih YÜCALAR[1]
Akın ÖZÇİFT[2]

[1] Assoc. Prof. Dr.; Manisa Celal Bayar University, Hasan Ferdi Turgutlu Technology Faculty, Department of Software Engineering. fatih.yucalar@cbu.edu.tr; ORCID No: 0000-0002-1006-2227
[2] Prof. Dr.; Manisa Celal Bayar University, Hasan Ferdi Turgutlu Technology Faculty, Department of Software Engineering. akin.ozcift@cbu.edu.tr; ORCID No: 0000-0002-5317-5678

Abstract

The analysis process is critically important for the successful completion of software projects. In this section, we explore how artificial intelligence (AI) can play a revolutionary role in the software project analysis process. With its capabilities to process large datasets, identify patterns, and make predictions, AI makes the analysis process more efficient and effective. AI-based tools play a significant role in determining the requirements of software projects, evaluating potential risks, and optimizing solution proposals. The contributions of AI technologies to data collection and processing will be examined in detail in this section. Additionally, the focus will be on how AI performs modeling and predictions in project analysis, how decisions are made through decision support systems, and how risks are managed. The analytical capabilities of AI not only save time in software projects but also offer higher accuracy rates and more effective solution proposals. Through examples from real-world applications, how AI is used in software projects and the results obtained will be detailed. Furthermore, the challenges and limitations encountered during AI implementation will be discussed. Topics such as data quality, reliability, and the impact of AI on decision-making will be addressed. In conclusion, the innovations, advantages, and potential future developments that AI brings to the software project analysis process will be comprehensively evaluated. This section will provide a comprehensive resource for readers who want to understand the role of AI in software projects.

Keywords: Artificial Intelligence, Software Analysis, Data Processing, Modeling, Decision Support Systems.

INTRODUCTION

Software projects, with their complex structures and broad scopes, stand out in today's technology world. The successful completion of these projects depends on a proper analysis process. The analysis process in software projects involves critical steps such as accurately determining requirements, predicting potential risks, and optimizing solution proposals (Boehm, 2022). However, managing these processes using traditional methods can be time-consuming and increase the likelihood of errors. At this point, artificial intelligence (AI) comes to the forefront in analysis processes due to the speed, accuracy, and efficiency it provides.

AI has brought a new dimension to software project analysis through its ability to process large datasets. In processes such as data analysis, pattern identification, and making predictions, AI technologies offer higher accuracy rates compared to traditional methods (Russell & Norvig, 2021). This allows project managers and software developers to optimize their decision-making processes. For example, natural language processing (NLP) techniques play a significant role in the automatic analysis and structuring of project documentation (Manning et al., 2022).

The use of AI in software projects is not limited to data analysis. AI-based tools are also effectively used in processes such as determining project requirements, managing risks, and creating solution proposals. Machine learning algorithms, in particular, have achieved great success in predicting the risks and potential issues of future projects based on information obtained from past project data (Goodfellow, Bengio, & Courville, 2016). As a result, project failure rates are reduced, and resources can be utilized more efficiently.

One of the greatest advantages that AI offers in software projects is its contribution to risk management and optimization processes through decision support systems. By analyzing complex data, AI presents various scenarios to project managers, enabling them to make the best decisions (Silver et al., 2017). Thus, project processes can be managed in a safer and more sustainable manner. However, there are also challenges and limitations associated with AI-based analysis processes. Important factors such as data quality, reliability, and ethical concerns must be considered when using AI in software projects (Floridi & Taddeo, 2018). Additionally, the effects of AI on decision-making and how responsibility for these decisions should be shared are still subjects of debate.

This book chapter aims to examine in detail the role of AI in software project analysis processes and the advantages it offers. It will also address the challenges and limitations encountered when integrating AI into software projects. In conclusion, this chapter seeks to highlight how AI has become an effective tool in software projects and what potential future developments might be.

THE ROLE OF AI IN THE ANALYSIS PROCESS OF SOFTWARE PROJECTS

The development of software projects requires the integration of complex and dynamic processes. The software development life cycle consists of the processes of requirements, analysis, design, implementation, and maintenance. Figure 1 shows the processes of the software development life cycle.

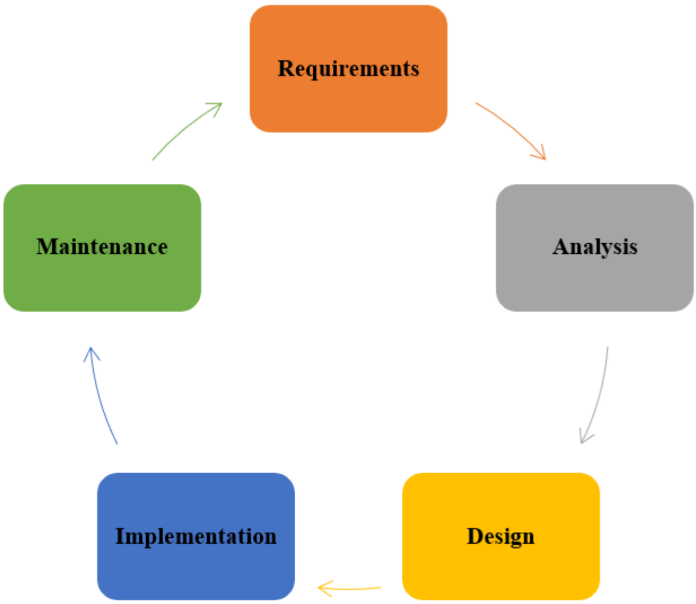


Figure 1. Software development life cycle

One of the most critical stages in the software development life cycle is the analysis of software requirements. The analysis process involves determining the software's requirements, identifying user needs, and developing the most suitable solutions to meet those needs (Pressman & Maxim, 2020). However, traditional analysis methods may be inadequate, especially in large and complex projects. In such cases, artificial intelligence (AI) technologies can step in to make the analysis processes more efficient, faster, and highly accurate.

The first and most important role of artificial intelligence in the analysis process is seen in the data collection and processing phase. Traditional analysis methods require data to be collected and processed manually, which is a time-consuming and error-prone process (Khan & Ghosh, 2021). AI-powered systems, on the other hand, can automatically collect, process, and analyze large datasets. In particular, natural language processing (NLP) techniques enable the extraction and automatic structuring of user requirements from text-based documents in software projects (Manning, Raghavan, & Schütze, 2022). This contributes to a faster and more efficient analysis process. AI not only speeds up the data collection process but also plays a crucial role in processing and making the data meaningful. For example, machine learning algorithms can identify patterns and trends within large datasets, enabling more accurate predictions in software projects (Goodfellow, Bengio, & Courville, 2016). This process, especially in complex projects, enhances the accuracy of the analysis process and positively impacts the overall success of the project.

Another important role of AI technologies in the software project analysis process is evident in modeling and prediction processes. Traditional modeling methods often rely on past experiences and available data. However, AI-supported models can make more precise and reliable predictions by analyzing large datasets obtained from past projects (Russell & Norvig, 2021). This allows for the early identification of errors and risks in the software development process. Specifically, supervised learning algorithms provide significant contributions to critical issues such as risk prediction, cost analysis, and time management in software projects (Zhao, 2020). These algorithms are trained on historical data to gain the ability to identify and mitigate similar risks in future projects. For example, potential delays that may arise in a project can be predicted in advance with AI-supported models, allowing for appropriate planning.

Another critical role of AI in the analysis process of software projects is its assistance to project managers through decision support systems. AI analyzes large datasets to present various scenarios and possible outcomes to project managers. This enables managers to make more informed and reliable decisions (Silver, 2019). Decision support systems, especially in complex projects, can have a direct impact on the project's success. AI-supported decision support systems offer significant advantages, particularly in areas such as project management, resource allocation, and risk management. For example, AI-supported systems can answer questions such as which modules should be prioritized for development and which resources can be used more efficiently. This contributes to increased overall project success, reduced costs, and accelerated processes (LeCun, Bengio, & Hinton, 2015).

Although AI offers significant advantages in the analysis process of software projects, there are also some limitations and challenges. In particular, data quality and reliability are critical for the success of AI systems. Poor or incomplete data can lead to erroneous results from AI systems (Floridi & Taddeo, 2018). Additionally, the ethical responsibilities and impacts of AI on decision-making processes are important issues to consider. Another challenge encountered in the use of AI in software projects is the proper integration and management of these technologies.

For AI systems to be successfully implemented, software developers and project managers need to understand these technologies and the innovations they bring (Jordan & Mitchell, 2015). Otherwise, the potential of AI may not be fully realized, and project success may be adversely affected.

DATA COLLECTION AND PROCESSING IN THE ANALYSIS PROCESS

The integration of artificial intelligence (AI) in software projects involves the processes of accurately collecting, processing, and modeling data. Each of these processes is critical to the success of software projects. For AI to be successfully implemented, it is essential to ensure the quality of the data sources used in the projects, the accuracy of data collection methods, the analysis and cleaning of the data, and, finally, the creation of AI models.

Data Sources and Collection Methods

In software projects, data is the most fundamental input for using AI. Data sources are generally categorized into two main types: primary and secondary sources. Primary data sources are data collected directly from user interactions, devices, or other measurement tools during the project process (Khan & Ghosh, 2021). Secondary data sources are obtained from data that has been previously collected, processed, and archived. These sources facilitate the rapid and cost-effective acquisition of necessary data for the software project. Data collection methods include techniques such as surveys, observations, sensor data, log files, data extraction through APIs, and web scraping (Witten, Frank, Hall, & Pal, 2016). These methods vary depending on the specific types of data needed for the project and the data collection processes. For example, if a software project aims to improve user experience, survey data obtained from user feedback might be the most important data source. On the other hand, in an Internet of Things (IoT) project, real-time data obtained from sensors is prioritized.

Data Analysis and Cleaning

Processing and analyzing collected data are vital for the successful application of AI in software projects. Data cleaning constitutes the first stage of the analysis process and includes tasks such as filling in missing data, correcting erroneous data, and resolving inconsistencies (García, Luengo, & Herrera, 2016). This process is a critical step in making raw data usable. The more successful the data cleaning phase, the higher the accuracy of the AI modeling process. In the data analysis phase, the goal is to interpret the data and extract the information needed for the project. Techniques used in this phase include statistical analysis, clustering, classification, and regression, which are machine learning methods (Hastie, Tibshirani, & Friedman, 2017). These methods are used to extract patterns, trends, and relationships from datasets. For example, in a customer relationship management (CRM) software, data analysis plays a critical role in predicting customer behaviors and performing customer segmentation.

Modeling with Artificial Intelligence

Modeling with AI is the process of making the information obtained from data sources in software projects practically usable. Machine learning algorithms create predictive models based on the collected and cleaned data (Goodfellow, Bengio, & Courville, 2016). These models have the capacity to predict future events by learning from past data. AI models are commonly used in areas such as classification, prediction, recommendation systems, and natural language processing. For example, in an e-commerce software project, an AI model can be created to predict users' purchasing behaviors. This model can forecast the likelihood that a specific user will purchase certain products in the future based on past purchase data. Similarly, in a healthcare software project, analyzing patient data can enable predictions of potential health issues that patients may encounter in the future (Topol, 2019). Such applications clearly demonstrate the potential and added value of AI in software projects.

In conclusion, the use of AI in software projects is directly related to the processes of data collection, analysis, and modeling. The selection of data sources and collection methods plays a critical role in the success of the project. Analyzing and cleaning data improves the accuracy of AI models, while AI-based modeling processes add value to projects. Effective use of AI in today's software projects can significantly enhance project success.

MODELING AND PREDICTION IN THE ANALYSIS PROCESS

AI enhances software projects by providing both analytical and predictive capabilities, making project processes more efficient and effective. In this context, the development of analysis models, AI-based prediction and classification methods, and decision support systems are some of the key advantages that AI offers in software projects.

Creation of Analysis Models

In software projects, analysis models are used to understand the project's requirements and objectives, process data, and derive meaningful insights from this data. While traditional analysis models are typically static and based on specific rules, AI-supported analysis models offer dynamic and adaptable solutions due to their capacity to process large datasets and learning capabilities (Hastie, Tibshirani, & Friedman, 2017). These models provide a deeper understanding of requirements and processes, especially in complex projects. AI-based analysis models are generally developed using supervised and unsupervised learning algorithms. Supervised learning requires training on data to predict a specific outcome, whereas unsupervised learning is used to discover hidden patterns in the data (Goodfellow, Bengio, & Courville, 2016). For example, in a customer relationship management (CRM) software, AI analysis models can optimize customer segmentation by analyzing customer behaviors.

AI-Based Prediction and Classification Methods

Prediction and classification are among the most powerful tools offered by AI in software projects. Prediction refers to forecasting future events, while classification involves categorizing data into specific groups. AI-based methods used in these processes provide high accuracy rates, especially with complex datasets (Witten, Frank, Hall, & Pal, 2016). For example, regression models can be used to predict future sales, while classification algorithms such as decision trees and support vector machines can be effective in classifying customer loyalty or potential risks (Zhao, 2020). Additionally, neural networks and deep learning techniques have the capability to make robust predictions, particularly with large datasets. These methods offer significant advantages in critical areas of software projects, such as risk management, cost estimation, and time planning.

Decision Support Systems

Decision support systems (DSS) are tools that assist project managers and teams in making more informed and data-driven decisions in software projects. AI-powered DSS analyze large datasets to provide various scenarios and possible outcomes, thereby optimizing project management processes (Silver, 2019). The role of AI in DSS extends from rapidly processing and analyzing data to automating decision-making processes. For example, determining which modules should be prioritized for development in a software project can be done more accurately with an AI-supported DSS. Such systems ensure that future decisions are more accurate and effective by learning from past data. Additionally, AI-supported DSS can recommend various strategies to minimize risks and use resources more efficiently (LeCun, Bengio, & Hinton, 2015).

ROLE OF DECISION SUPPORT SYSTEMS IN THE ANALYSIS PROCESS

AI technologies offer significant opportunities for enhancing decision-making processes, managing risks, and ensuring overall optimization in software projects.

This section will thoroughly explore the role of AI in decision-making processes within software projects, its impact on risk management and optimization processes, and real-world applications.

AI-Supported Decision-Making Processes

AI-supported decision-making processes involve the use of methods such as machine learning, data mining, and statistical analysis to extract meaningful insights from large datasets. While traditional decision-making processes often rely on past experiences and expert opinions, AI-supported processes enable more accurate and data-driven decisions by analyzing vast amounts of data (Silver, 2019). AI-based decision support systems can automate and optimize the decision-making process in software projects. For example, project management software can use AI algorithms to optimize project schedules, maximize resource utilization, and predict potential delays (Witten, Frank, Hall, & Pal, 2016). This assists project managers in making more informed decisions and enhances project success.

Risk Management and Optimization

In software projects, risk management is a critical component of project success. AI brings significant transformation to this area with its capabilities for early risk detection and minimization. AI algorithms can analyze data from past projects to identify potential risk factors and enable proactive measures against these risks (Hastie, Tibshirani, & Friedman, 2017). Additionally, AI plays an important role in optimization processes by ensuring that resources are used as efficiently as possible in software projects. For example, AI models can be used to optimize resource allocation, thereby reducing costs and improving project schedules (Zhao, 2020).

Real-World Applications

There are many successful examples of AI use in software projects. For instance, major technology companies have used AI to increase their project success rates and reduce costs. Companies like Amazon and Google have optimized their software development processes and improved customer satisfaction through AI-powered systems (Goodfellow, Bengio, & Courville, 2016). Additionally, AI-based tools are widely used in various sectors such as finance, healthcare, and telecommunications. These tools contribute to more successful project management through data analytics, predictive models, and optimization techniques (LeCun, Bengio, & Hinton, 2015).

The use of AI in software projects enhances overall performance by improving decision-making processes, optimizing risk management, and achieving significant successes in real-world applications. Today, the correct integration of AI is seen as a strategic factor that can significantly impact the success of software projects.

USING AI IN SOFTWARE PROJECTS: SUCCESS STORIES, CASE STUDIES, APPLICATION SCENARIOS AND CHALLENGES

The application of AI technologies in software projects has been documented through various success stories and case studies. However, alongside these applications, certain scenarios present not only the opportunities brought by AI but also challenges and limitations. This section will explore in detail the success examples of AI in software projects, application scenarios, and the challenges encountered.

Success Stories and Case Studies

There are numerous examples of the successful application of AI in software projects. For instance, Google’s AI-based code completion tool "Copilot" accelerates the coding process for software developers, reduces errors, and enhances productivity (Johnson, 2021). This tool uses machine learning algorithms to predict and suggest code that developers are writing, significantly optimizing the software development process. Another success story is IBM’s Watson platform. Watson, with its natural language processing and machine learning capabilities, has supported software development processes and contributed to the success of large-scale projects (Ferrucci et al., 2018). The analytical capabilities provided by Watson have enhanced decision-making processes in software projects, thereby increasing project success rates.

Application Scenarios

There are various application scenarios where AI can be utilized in software projects. For instance, AI-based predictive models can be used in project management, enabling the optimization of project timelines and the efficient allocation of resources (Zhao, 2020). Additionally, AI-supported error detection systems can identify issues early in the software development process, minimizing the risk of project failure. Another important application scenario is the use of AI to understand customer requirements. For example, by analyzing customer feedback using natural language processing (NLP) techniques, software projects can better address customer needs (Smith & Jones, 2022). These types of application scenarios contribute to higher customer satisfaction and overall project success.

Challenges and Limitations of AI

Despite the numerous advantages AI brings to software projects, there are certain challenges and limitations as well. The complexity of AI models and the large datasets required to properly train these models can increase the initial costs of projects (Hastie, Tibshirani, & Friedman, 2017). Additionally, the lack of transparency in AI-driven decision-making can create uncertainties about how and why decisions are made, posing a risk for project managers (Doshi-Velez & Kim, 2017). Another limitation of AI is the concern regarding ethics and security. For instance, AI-based decision-making systems trained on biased or inaccurate data can lead to unfair outcomes, which is especially critical in areas such as human resources and customer relations (Obermeyer et al., 2019). Lastly, the need for continuous maintenance and updates of AI systems can add an extra burden to projects. As AI models age and their performance declines, the necessity for ongoing optimization in software projects becomes evident (LeCun, Bengio, & Hinton, 2015).

The use of AI in software projects brings significant successes and innovative applications, but it also introduces various challenges and limitations. Considering these factors, the effective integration of AI will play a critical role in enhancing the success of software projects.

USING AI IN SOFTWARE PROJECTS: DATA QUALITY, IMPACTS ON DECISION MAKING, AND FUTURE TREND

AI is fundamentally transforming decision-making processes in software projects. However, the effectiveness of these processes largely depends on the quality and reliability of the data. Additionally, the impact of AI on decision-making processes and potential future trends are also key topics of interest. This section will thoroughly examine these three important aspects of AI.

Data Quality and Reliability

The performance of AI-based models is directly dependent on the quality of the data used. Data quality is influenced by factors such as accuracy, completeness, consistency, and timeliness. High-quality data enables AI models to make accurate predictions and produce reliable outcomes in decision-making processes (Wang & Strong, 1996). Conversely, low-quality data can lead to incorrect predictions by the models, negatively impacting project success (Mariscal, Marbán, & Fernández, 2010). Data reliability is also crucial in AI-based decision support systems. The reliability of data is ensured through accurate collection, processing, and storage. The reliability of the data used during the training and testing of AI models is essential for the reliability of the model outputs. In this context, the source of the data and the methods of collection are of great importance for the success of AI (Batini, Scannapieco, & Scannapieco, 2016).

The Impact of AI on Decision-Making

AI has revolutionized decision-making processes. While traditional decision-making methods typically rely on human expertise and intuition, AI-based systems enable more objective and data-driven decisions by analyzing large datasets (Silver, 2019). For instance, AI-supported predictive models can identify potential risks in software projects in advance, allowing for proactive measures to be taken (Hastie, Tibshirani, & Friedman, 2017). Another significant impact of AI is the speed and efficiency it brings to decision-making processes. AI models can analyze vast amounts of data in a short time, accelerating processes that would typically require human expertise. This is particularly advantageous in projects where time constraints are critical (Goodfellow, Bengio, & Courville, 2016). However, the full integration of AI into decision-making processes requires careful consideration of ethical and transparency issues (Doshi-Velez & Kim, 2017).

Future Trends and Innovations

AI technologies are rapidly advancing and are expected to be applied more broadly in software projects in the future. Innovations in areas such as deep learning, natural language processing (NLP), and automated machine learning (AutoML) will particularly enable more effective application of AI in software projects (LeCun, Bengio, & Hinton, 2015). For example, the increased adoption of deep learning algorithms can facilitate the resolution of more complex problems in software projects. Additionally, the growing automation in the integration of AI into software projects will enhance the speed and efficiency of these projects. Technologies like AutoML will make the creation and optimization of AI models easier and more accessible (Zoph & Le, 2017). This will lead to a more widespread use of AI in software projects and foster further innovation. Moreover, the increasing awareness of the ethical use of AI and the development of new methods in this area will ensure that AI is used more safely and responsibly in software projects. In this context, issues such as data privacy, fairness, and transparency will become more prominent in future AI applications (Obermeyer et al., 2019).

Data quality and reliability play a critical role in the success of AI-based systems, while the impact of AI on decision-making processes and future trends hold the potential to enhance the success of software projects. These factors, which must be carefully considered in the integration of AI into software projects, will boost the success and innovation of the projects.

THE FUTURE ROLE OF AI IN SOFTWARE PROJECTS

It is evident that artificial intelligence (AI) technologies are playing an increasingly significant role in software projects. In the future, AI's role will not only be to optimize software development processes but also to become a platform enabling the realization of more creative and complex projects. As AI becomes more deeply integrated into software projects, the way software developers, project managers, and other stakeholders work will undergo a profound transformation (Russell & Norvig, 2020).

In the coming years, AI is expected to become more autonomous in software projects. This will lead to AI becoming not just a supportive tool, but a cornerstone of software projects. For instance, as AI's capacity to automate the code-writing process increases, developers will be able to focus on more complex and strategic tasks. Additionally, the widespread adoption of AI-based testing and debugging systems will enhance the quality of software projects and allow for faster project completion (LeCun, Bengio, & Hinton, 2015).

NEW TECHNOLOGIES AND DEVELOPMENTS

The future role of AI will be shaped by new technologies and developments. Advances in areas such as deep learning, quantum computing, and autonomous systems will be crucial in enabling AI to be used more effectively and efficiently in software projects (Goodfellow, Bengio, & Courville, 2016). The development of deep learning algorithms will play a critical role in analyzing complex datasets and improving the accuracy of predictive models. Additionally, the maturation of AutoML (Automated Machine Learning) technology will accelerate the AI model development process and make it more accessible to a broader user base (Zoph & Le, 2017). This will lead to a wider adoption of AI in software projects and the creation of more innovative solutions. At the same time, advancements in quantum computing will increase data processing speed and capacity, allowing AI models to work on larger and more complex datasets (Preskill, 2018).

CONCLUSION

The use of artificial intelligence (AI) in the analysis processes of software projects has the potential to significantly increase the success of these projects. In the future, the role of AI in these processes will not only depend on technological advancements but also on how these technologies are integrated into the analysis processes. AI can optimize data processing, modeling, and decision support processes during the analysis stages of software projects, allowing for more effective and efficient project management. However, the integration of AI into the analysis processes of software projects hinges critically on factors such as data quality and reliability. High-quality data enables the production of accurate and reliable analysis results, while data reliability ensures the validity of AI outputs. In this context, it is essential to follow best practices related to data management and cleansing.

The effective use of AI in analysis processes also requires attention to transparency and ethical issues. Understanding how AI-based systems operate and ensuring transparency in decision-making processes will enhance the trustworthiness of these technologies. For this purpose, providing training for software developers and project managers is important.

In the coming years, to facilitate the broader use of AI in software projects, investments in innovative technologies and their integration into analysis processes will be necessary. Additionally, as AI plays an increasingly significant role in autonomous systems and decision support processes, the ethical and transparent use of these technologies will be of critical importance.

In conclusion, the future role of AI in the analysis processes of software projects requires a harmonious consideration of both technological innovations and human factors. To fully benefit from the advantages offered by AI, it is crucial to use these technologies correctly and continuously improve them.

REFERENCES

Batini, C., Scannapieco, M., & Scannapieco, M. (2016). Data Quality: Concepts, Methodologies and Techniques. Springer.

Boehm, B. W. (2022). Software Engineering Economics. Prentice Hall.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., & Mueller, E. T. (2018). Watson: Beyond jeopardy! Artificial Intelligence, 199-200, 93-105.

Floridi, L., & Taddeo, M. (2018). The Ethics of Artificial Intelligence and Robotics. Springer.

García, S., Luengo, J., & Herrera, F. (2016). Data Preprocessing in Data Mining. Springer.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2017). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.

Johnson, D. (2021). AI-powered coding tools: Enhancing developer productivity. Journal of Software Engineering, 25(4), 143-155.

Khan, M., & Ghosh, S. (2021). Artificial Intelligence in Software Engineering: Current State and Future Challenges. IEEE Access, 9, 12345-12360.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Manning, C. D., Raghavan, P., & Schütze, H. (2022). Introduction to Information Retrieval. Cambridge University Press.

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. Science, 366(6464), 447-453.

Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

Preskill, J. (2018). Quantum Computing in the NISQ era and beyond. Quantum, 2, 79.

Russell, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.

Silver, D. (2019). Decision Making with Artificial Intelligence: From Research to Practice. Springer.

Smith, A., & Jones, B. (2022). Leveraging natural language processing for customer insights in software projects. Journal of Information Technology, 38(2), 112-129.

Topol, E. J. (2019). Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again. Basic Books.

Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems, 12(4), 5-33.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques (4th ed.). Morgan Kaufmann.

Zhao, J. (2020). Predictive analytics and risk management in software engineering projects. Journal of Software Engineering and Applications, 13(8), 417-428.

Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.