Emerging Trends in Mechatronics

Ehsan Momeni
Danial Jahed Armaghani
Aydin Azizi *Editors*

Artificial Intelligence in Mechatronics and Civil Engineering

Bridging the Gap

Springer

Emerging Trends in Mechatronics

Series Editor

Aydin Azizi, Oxford, UK

Ehsan Momeni · Danial Jahed Armaghani ·
Aydin Azizi
Editors

Artificial Intelligence in Mechatronics and Civil Engineering

Bridging the Gap

Springer

Editors
Ehsan Momeni
Department of Civil Engineering
Faculty of Engineering
Lorestan University
Khorramabad, Iran

Danial Jahed Armaghani
Civil and Environmental Engineering
University of Technology Sydney
Sydney, NSW, Australia

Aydin Azizi
School of Engineering, Computing
and Mathematics
Oxford Brookes University
Oxford, UK

ISSN 2731-4855 ISSN 2731-4863 (electronic)
Emerging Trends in Mechatronics
ISBN 978-981-19-8789-2 ISBN 978-981-19-8790-8 (eBook)
https://doi.org/10.1007/978-981-19-8790-8

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

About This Book

This book aimed to cover the application of artificial intelligence and simulation techniques in Engineering. The book highlights the successful implementation of different soft computing techniques in various areas of engineering more especially in Civil, Electronic, Mechatronic, and Mining Engineering. This book comprises nine chapters which, in overall, shed lights on the importance of simulation techniques in solving complex engineering problems.

Chapter "Optical Resistance Switch for Optical Sensing" deals with optical resistance switch for optical sensing. As regards optical switches can be suitable candidates for optical sensing applications. This chapter focuses on optical switch devices for optical sensing applications. The optical structures studied are photonic crystal, plasmonic, and graphene structures which are good platforms for the realization of optical devices such as optical switches and sensors. Photonic crystals can be considered as an efficient basis for the realization of optical switches and sensors. It is because photonic band-gaps with sharp transient edges which are suitable for switching and sensing mechanisms can be generated at their transmission spectra. Surface plasmon polaritons have drawn extensive attention in recent decades due to their ability to break the traditional diffraction limit of light. Graphene structures have remarkable advantages such as ultra-low ohmic losses, thin thickness, tunability of complex conductivity, and mechanical strength. In this chapter, relying on theoretical models and numerical simulations, it is shown that optical sensing mechanism can be achieved by using optical devices such as graphene switches, nanomaterial heterostructures (plasmonic and PC structures)-based switch, and plasmonic amplitude modulators.

Chapter "Empirical, Statistical, and Machine Learning Techniques for Predicting Surface Settlement Induced by Tunnelling" deals with the application of simulation techniques in assessing the tunnelling-induced settlement in urban areas. Tunnels have been constructed in many countries around the world for different purposes, such as the metro system to mitigate traffic congestion. Since the construction of urban tunnels is typically conducted at shallow depths, specific concerns such as structural damage inevitably arise. Surface settlement induced by tunnelling is one of the common problems encountered during and after tunnelling construction. Therefore, accuracy in the prediction of surface settlement induced by tunnelling is important to prevent structural damage. Several methods have been previously proposed to compute tunnelling-induced surface settlement, such as empirical, numerical, laboratory, statistical, and machine learning. Each of these models has advantages and disadvantages. This study deeply investigates the available techniques to estimate settlement induced by tunnelling and reviews the most important findings and solutions. Among the existing techniques, machine learning seems to be the most suitable technique in estimating settlement induced by tunnelling. These techniques, with their behind calculations and assumptions, are able to identify the best relations between independent and dependent parameters and, therefore, to solve complex and non-linear problems. The discussion provided in this chapter can be advantageous to those who are interested to conduct research or design in the same field.

Chapter "A Review on the Feasibility of Artificial Intelligence in Mechatronics" deals with a review on the application of artificial intelligence in mechatronic. Artificial intelligence has become a valuable tool in various fields with the increasing progress of science and information production. With the processing of big data and increased productivity, new challenges have appeared in the design and application of control systems. In some systems, the interaction between humans and robots is essential, while real-time decision-making plays a vital role in other types of systems. This chapter presents a review of artificial intelligence methods in mechatronics. For this purpose, the leading intelligent control methods in technical systems are reviewed and discussed in the initial part of this chapter, including reinforcement learning, fuzzy logic method, artificial neural networks, optimization techniques, and adaptive control methods. The rest of the chapter reviews the applications of intelligent approaches in engineering control problems. This part is categorized into five subsections: iterative learning, parametric optimization, identification, controller tuning, and control problems as stabilization. Finally, in the conclusion of the chapter, the main challenges in improvements of intelligent control methods are listed.

Chapter "Feasibility of Artificial Intelligence Techniques in Rock Characterization" deals with the feasibility of soft computing and simulation techniques in assessing the rock engineering properties. This chapter reviews the recent works which highlight the workability of the aforementioned techniques in predicting unconfined compressive strength of rock samples. Based on the provided discussion in this chapter, the artificial intelligence-based predictive models are quick, economic, and feasible tools in rock characterization.

Chapter "A Review on the Application of Soft Computing Techniques in Foundation Engineering" reviews the application of artificial intelligence methods in foundation engineering. This chapter shed light on many studies which underline the feasibility of simulation-based techniques in assessing the bearing capacity and settlement of various types of foundation including shallow, deep, and skirted foundations.

Chapter "Application of a Data Augmentation Technique on Blast-Induced Fly-Rock Distance Prediction" deals with the application of a data augmentation technique on blast-induced fly-rock distance prediction. Fly-rock induced by blasting is an inevitable phenomenon in quarry mining, which can give rise to severe hazards, for example, causing damage to buildings and human life. Thus, successful estimation of fly-rock distance is crucial. In this regard, many researchers attempt to develop empirical, statistical, or machine learning models to accurately predict fly-rock distance. However, for most previous research, a worrying drawback is that the amount of data related to fly-rock distance prediction is insufficient because the measurement work of fly-rock distance is costly for manpower and material resources. In Chapter six, to deal with the problem of data shortage, authors first separated the original dataset which was collected from four granite quarry sites in Malaysia into two parts, i.e., the training and testing sets, and then adopted a data augmentation technique termed tabular variational autoencoder (TVAE) to augment the amount of the training (true) data, so as to generate a fresh synthetic data set. Subsequently, we utilized several statistical methods or plots, such as the boxplot method, kernel density estimation, cumulative distribution function, and heatmap method, to testify to the effectiveness of the synthetic data generated by the TVAE model. Lastly, several commonly used machine learning models were developed to verify whether the mixed data set which is obtained by merging the training and synthetic data sets can benefit from the addition of the synthetic data. The work of verification is implemented on the testing data set. The results demonstrate that the size of the training data set has increased from the initial 131 to 1000 to obtain a synthetic data set, and the results of statistical methods proved that the synthetic data set not only preserves the inner characteristics of the training data set but also generalizes more diversities compared to the training data set. Further, by comparing the performance of five machine learning model on three data sets (i.e., the training, synthetic, and mixed data sets), it can be concluded that the overall performance of all machine learning models on the mixed data set outperforms that on the training and synthetic data sets. Consequently, it can be asserted that the application of the data augmentation technique on the fly-rock distance issue is fruitful in the present study and has profound engineering application value.

Chapter "Forecast of Modern Concrete Properties Using Machine Learning Methods" reviews machine learning methods that have been used in the previous researches to forecast concrete properties. Major classifications and main steps of machine learning techniques are introduced then their application in concrete science is discussed. Artificial neural network, fuzzy logic, decision tree, support vector machine, gene expression programing, bagging, and boosting are the most commonly machine learning methods. A typical machine learning-based studies comprise problem description, data collection, data pre-processing, model development, and model assessment. Concretes noticed in this review include ordinary, self-consolidation, ultra-high-performance, alkali-activated, and recycled aggregate types. Inputs, prediction methods, and outputs for each concrete type is reviewed in different researches and summarized as review result. The chapter illustrates that machine learning techniques are capable of predicting a wide range of properties, including mechanical properties, freshness properties, and durability properties. According to this chapter, concrete compressive strength is the most frequent property and artificial neural network is the most frequently machine learning method that different researchers dealt with. Activation functions, network architecture, learning rules, and hybrid models of artificial neural network are discussed in details. In general, this chapter suggests that machine learning methods perform better than classical regression models.

Chapter "Reliability-Based Design Optimization of Detention Rockfill Dams and Investigation of the Effect of Uncertainty on Their Performance Using Meta-Heuristic Algorithm" deals with Reliability-based design optimization of detention rockfill dams. Detention rockfill dams are one of the most popular structures which are used for reducing the flood peak and also increasing the downstream response time against the occurrence of floods. Designing such structures is a critical issue because of the complex interaction between the coarse porous rockfill material and the flow, the non-Darcian flow in rockfills, their stable encounter with floods, reliable flood peak reduction, and safe flood discharge release downstream. In this chapter, the preliminary design is performed using a simulation model in order to obtain the initial height and length of the dam and then using a simulation-optimization model, the optimization of the dams is done to provide structural safety factors. Also, both preliminary and optimal designs are investigated for their reliabilities under uncertain conditions. Using the Monte Carlo simulation method as well as the simulation and optimization models, the effects of uncertainty in design parameters are investigated. In this step, LHS is used for generating samples, and the rejection rule is implied for deleting some samples which are greater than 15% uncertainty. The uncertainty of input parameters in model design consideration is investigated including the uncertainties of inlet hydrograph, volume-elevation relation of the reservoir, non-Darcian stage-discharge equation in coarse rockfill material, and non-linear flood routing in detention rockfill dams. At the end of this chapter, a reliability-based design optimization (RBDO) of the detention rockfill dam was carried out using self-adaptive NSGA-II. The first objective function in this multi-objective algorithm is the minimization of the dam cost and the second one is the maximization of the reliability function.

Chapter "Machine Learning in Mechatronics and Robotics and Its Application in Face-Related Projects" deals with various aspects of face-related projects. Implementing non-verbal information is one of the ways to create communication between people. Using this type of information can improve the interaction process in human–robot interaction. One of the aspects of people's non-verbal information is facial images, which can play an essential role in the development of mechatronic and robotic systems. This chapter discusses some uses of facial images, such as facial recognition, and facial expression recognition. Using these images and applications, authors explored and developed mechatronic and robotic systems which were based on special access for different persons and changes in facial expressions. As mentioned earlier, in chapter nine, different aspects of face-related projects are explored, apart from that, some ideas that can be applied to create new approaches are discussed. Since these projects perform based on a procedure, the flow of the facial projects from face detection to facial expression recognition and other applications are discussed. This chapter helps the readers get acquainted with using facial images in human–robot interaction. Furthermore, enough information is provided to understand the process of the methods used before, from traditional to modern approaches, and to know the process of developing new methods. In the different parts, the datasets related to each application are introduced to have a complete view of facial image-based projects. By understanding these applications and how to create innovative methods, the way to use facial images is paved to provide new face image-based systems.

Contents

Optical Resistance Switch for Optical Sensing

Shiva Khani, Ali Farmani, and Pejman Rezaei

1 Introduction

Optical components [1–3] are superior to semiconductor electronic ones [4, 5] in terms of higher computational speeds and lower noise values. Consequently, the development of optical devices is growing in prosperity. Photonic crystal (PC) structures [6–8] can be a suitable candidate for the realization of different passive and active optical devices such as PC filters [9, 10], sensors [11–13], splitters [14, 15], slow lights [16], demultiplexers [17, 18], switches [19–21], modulators [22, 23], logic gates [24, 25], and so on. PCs are periodic arrangements of dielectric layers with alternating refractive indexes. The distribution of the dielectric layers' refractive index is periodic in one, two, or three dimensions [26]. Due to the remarkable capability to appear photonic band-gaps (PBGs) with sharp transient edges, PCs are considered as an efficient basis for the realization of optical switches and sensors [27]. In the frequency range of the PBG, no optical wave is allowed to propagate through the PC structure. The refractive index and structural parameters of the PC structure determine the PBG frequency range [28]. The relatively large size (micrometer size) of the PC structures is their main drawback compared to other optical structures such as plasmonic [29–31] and graphene [32, 33] structures.

Surface plasmon polaritons (SPPs) due to their remarkable capabilities to overcome the diffraction limit and manipulate light in a nano-scale domain are considered as an efficient basis for the realization of highly integrated optical circuits [34–36].

S. Khani · P. Rezaei
Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran
e-mail: shiva.khani@semnan.ac.ir

P. Rezaei
e-mail: prezaei@semnan.ac.ir

A. Farmani (✉)
School of Electrical and Computer Engineering, Lorestan University, Khoramabad, Iran
e-mail: farmani.a@lu.ac.ir

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*,
Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_1

SPPs are created at the interface between an insulator and a metal layer with different signs of dielectric constants through the interaction of surface electrons and incident photons [37]. Accordingly, all of the conventional PC components can be redesigned based on SPPs with more miniaturized footprints. Over the last few years, a variety of devices based on SPPs have been proposed, i.e., plasmonic filters [38, 39], Bragg reflectors [40, 41], sensors [42–44], demultiplexers [45, 46], Mach–Zehnder interferometers [47], and Switches [48–51]. Another advantage of metal–insulator-metal (MIM) plasmonic structures is the possibility of integrating such configurations with microwave circuits [52, 53]. It is because MIM structures have also been used for the realization of previous microwave microstrip components [54].

Due to the large negative real part and small imaginary part of permittivity of the noble metals such as silver and gold in optical frequencies, such metal materials have been widely used in plasmonic devices [55]. However, plasmonic components designed using these metals have some drawbacks. First of all, such structures are inherently lossy [56]. Secondly, these structures cannot be tuned freely when the geometry parameters of them are fixed [57–59]. This motivates researchers to use other materials with low ohmic losses and tunability property such as graphene structures [33, 60, 61].

Graphene is a two-dimensional (2D) monolayer sheet of carbon atoms packed into a honeycomb lattice [62, 63]. Graphene can behave like a thin metal film with negative permittivity by means of chemical doping, electric and magnetic fields [64, 65]. Meanwhile, graphene structures have remarkable advantages compared to noble metals. Such features include ultra-low ohmic losses of graphene, much thinner thickness of graphene than metal layers, tunability of graphene's complex conductivity, and mechanical strength of graphene structures. The mentioned properties of graphene make it an appropriate choice to design highly integrated graphene-plasmonic (GP) devices [66–69].

Among the various optical devices, optical switches [70, 71] and sensors [72–74] are popular topics and have a wide range of applications. Up to now, various approaches based on different structures such as PC [75, 76], plasmonic [77, 78], graphene [79, 80], etc., have been adopted to design optical switches and sensors. An all-optical switch based on a directional coupler structure in a 2D PC lattice with Kerr optical nonlinearity has been proposed in [21]. In the proposed structure reported in [21], the high switching speed and low reflection coefficient for the control signal makes the presented switch suitable for all-optical integration purposes. In other approaches [2, 29, 48, 50], novel plasmonic topologies using two isolated MIM waveguides to pass the data and control signals have been proposed for all-optical switching applications. Using such a method (isolated data and control paths) prevents the intermodulation between the data and control signals and reduces harmonic distortion. Other methods including optical switches using a deposited graphene nanoribbon on a silica (SiO2)/ silicon (Si) substrate [81], a terahertz switch based on a graphene monolayer, poly-methyl methacrylate (PMMA) [82], all-optical multi-channel switch using square ring resonators [83], and so on have also been presented.

Over the last few years, there are also various suggestions for realizing optical sensors using different structures. Due to the sharp edge of the Fano resonance shape, such a resonance mode can improve the sensitivity and figure of merit (FOM) of optical sensors. As a result, many resonators and optical configurations have been presented to obtain the Fano resonance shape, including an asymmetric MIM waveguide structure [84], splitting ring resonator and tooth-shaped resonator [85], and dual ring resonators [86]. Another method that has been used in [87, 88] is using optical structures with Plasmon-induced transparency (PIT) resonance modes.

In addition to the mentioned published optical switch and sensor structures, there are also reports on structures to realize the function of both optical switches and sensors simultaneously. In such structures, multifunctional devices are used as optical switches and sensors using two different insulator materials [80, 89–92]. The extensive design of dual-functional devices (optical switch and sensor) includes plasmonic nano-disk resonators [89], dumbbell-shaped cavity slots [90], rectangular resonators [91], two pairs of graphene nano-rings and a graphene nanoribbon [92], and a plasmonic graphene-based structure [80].

In this chapter, first as a case study, graphene and nanomaterial heterostructures (combination of plasmonic and PC structures) are used to design optical switches. Thereafter, to study the modulation characteristics, an optical modulator has been designed using plasmonic structures. Finally, the summary and conclusions are discussed in the last section.

2 Graphene Optical Switch

Optical switches are used for light routing and switching. Accordingly, they have found wide applications in many optical systems. On the other hand, optical switches can also be used for optical sensing applications. It is mainly due to the high wavelength shifts of optical switches. Such devices have been presented in [93–95]. Among the most important optical structures that have found wide application for switching mechanisms are graphene structures.

The optical features of graphene can be tuned by varying its Fermi level. The Fermi level and therefore the chemical potential (μ_c) and the surface conductivity of graphene can be changed by applying an external voltage. When the chemical potential is higher than the threshold value, graphene acts as a metal layer, while graphene can be considered as a dielectric layer for the chemical potential lower than the threshold value. The surface conductivity of graphene (σ) which is composed of the intraband and interband parts can be calculated by the Kubo formula [96, 97].

$$\sigma(\omega) = \sigma_{\text{intra}}(\omega) + \sigma_{\text{inter}}(\omega) \tag{1}$$

where σ_{intra} and σ_{inter} are the intraband and interband transitions, respectively, and they can be expressed as

$$\sigma_{\text{intra}} = \frac{2ie^2k_BT}{\pi\hbar^2\left(\omega + i\tau^{-1}\right)}\ln\left(2\cosh\left(\frac{E_f}{2k_BT}\right)\right) \tag{2}$$

$$\sigma_{\text{inter}} = \frac{e^2}{4\hbar}\left\{\frac{1}{2} + \frac{1}{\pi}arc\tan\left(\frac{\hbar\omega - 2E_f}{2k_BT}\right) - \frac{i}{2\pi}\ln\left(\frac{\left(\hbar\omega + 2E_f\right)^2}{\left(\hbar\omega - 2E_f\right)^2 + (2k_BT)^2}\right)\right\} \tag{3}$$

where k_B is the Boltzmann constant, E_f demonstrates the Fermi level of graphene, $\hbar$ shows the reduced Planck constant, and τ is the carrier relaxation time. In [98], all-optical switches based on graphene plasmon structures were proposed. The general structure of the designed graphene plasmon switch is shown in xxx. This structure is composed of a novel combination of insulator–metal-insulator (IMI) and graphene waveguides designed in the form of a Mach–Zehnder interferometer (MZI) topology. In this structure, the metal material in the MZI structure is silver. The complex relative permittivity of silver is described by the accurate Drude–Lorentz model [99]:

$$\varepsilon_m(\omega) = 1 - \frac{\omega_{\text{p}}^2}{\omega(\omega + \text{j}\gamma)} + \sum\nolimits_{n=1}^{5}\frac{f_n\omega_{\text{n}}^2}{\omega_{\text{n}}^2 - \omega^2 - j\omega\gamma_n} \tag{4}$$

where $\omega_p = 2002.6 \times 10^{12}$ Hz is the bulk plasma frequency of metal, $\gamma = 11.61 \times 10^{12}$ Hz is a damping constant, ω is the angular frequency of the incident light. Also, values of resonant frequencies (ω_n), damping constants (γ_n), and weights (f_n) are Fig. 1 given in Table 1. The real and imaginary parts of the silver permittivity are shown in Fig. 2a.

The insulator material around the MZI structure is a Kerr non-linear material which is chosen to be GaAs. The bottom layer (black area) is a graphene layer that is placed on a silica substrate. The graphene permittivity is characterized by [65]:

$$\varepsilon_g(\omega) = 1 + i\left(\frac{\sigma(\omega)}{\varepsilon_0\omega t_g}\right) \tag{5}$$

where t_g is the graphene thickness, and ε_0 is the permittivity of vacuum. Figure 2b shows the real and imaginary parts of the graphene permittivity. The structural parameters of the proposed optical switch (Fig. 1) are as follows: $d_1 = 300$, $d_2 = 3000$, $d_3 = 3000$, $d_4 = 350$, $d_5 = 50$, $d_6 = 500$, $d_7 = 600$, $d_8 = 1000$, and $d_9 = 100$ (all in nm).

Figure 1 shows that there is a metal layer in the shape of an MZI above the graphene sheet so that the graphene region under the metal plate turns into waveguides by applying the DC voltage. It means that when the DC voltage is applied between two layers (metal and graphene), a capacitive structure in the form of the MZI is generated. The created capacitor is studied in two modes of DC and AC in the following. After applying the DC voltage, the chemical potential of the created MZI structure on the graphene layer modulated from 0 to 0.31 eV. Accordingly, surface plasmon waves

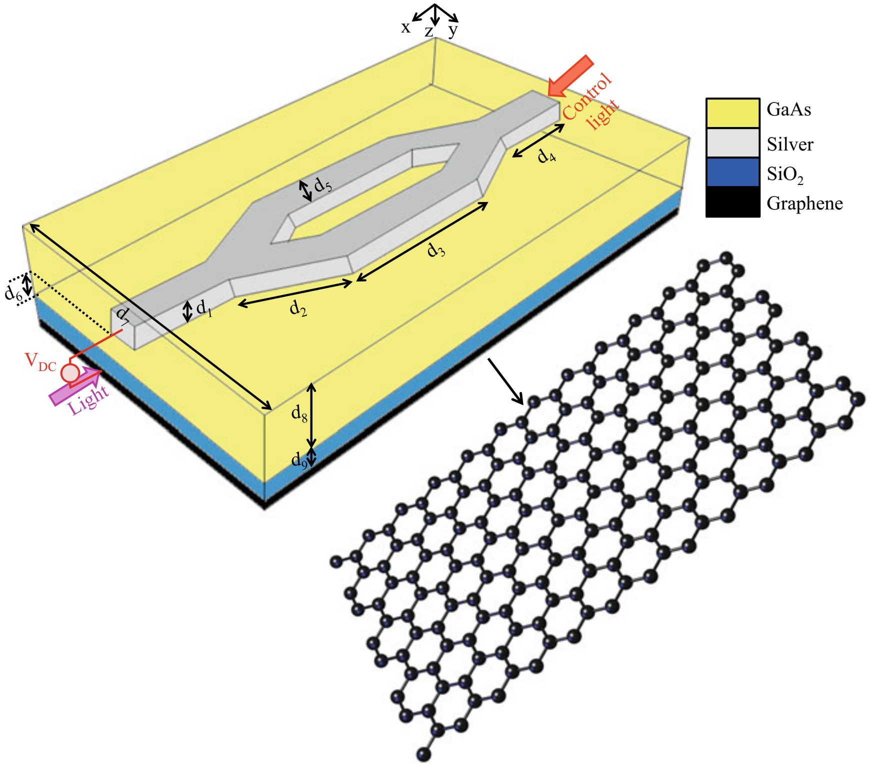


Fig. 1 General structure of the proposed all-optical graphene plasmon switch

Table 1 Parameters of the Drude–Lorentz model for silver

n	ω_n (THz)	γ_n (THz)	f_n
1	197.3	939.62	7.9247
2	1083.5	109.29	0.5013
3	1979.1	15.71	0.0133
4	4392.5	221.49	0.8266
5	9812.1	584.91	1.1133

can propagate in the graphene MZI structure, while these waves cannot excite in other sections of the graphene layer ($\mu_c = 0$ ev).

A. DC mode of the gate capacitor

Figure 3a shows the order of the created capacitor layers in Fig. 1. As seen in this figure, two insulator materials of SiO$_2$ and GaAs exist between the two capacitor plates (metal and graphene plates). It is because graphene is created by the plasma-enhanced chemical vapor deposition (PECVD) method on a silica layer and the

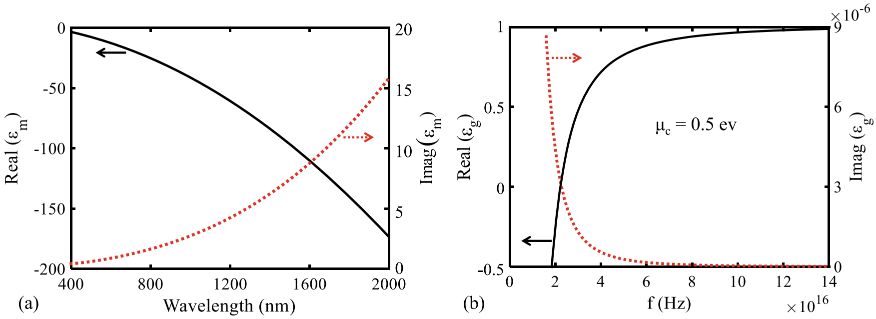


Fig. 2 Real and imaginary parts of the **a** silver permittivity, **b** graphene permittivity

non-linear material of GaAs is used for light modulation. As a result, the equivalent capacitance includes two series capacitors. Such an equivalent capacitance can be obtained as

$$C_{ox} = \frac{1}{\frac{1}{C_{\mathrm{SiO_2}}} + \frac{1}{C_{\mathrm{GaAs}}}} = \frac{\varepsilon_0}{\frac{d^{nm}_{\mathrm{SiO_2}}}{\varepsilon_r^{SiO_2}} + \frac{d^{nm}_{\mathrm{GaAs}}}{\varepsilon_r^{\mathrm{GaAs}}}} \tag{6}$$

As seen in Eq. 6, the capacitance value depends on four parameters of the heights of the insulator layers ($d^{nm}_{\mathrm{SiO_2}}$ and d^{nm}_{GaAs}) and the permittivity values of the insulator materials ($\varepsilon_r^{\mathrm{SiO_2}}$ and $\varepsilon_r^{\mathrm{GaAs}}$). To achieve the appropriate dimensions of the two insulator materials ($d^{nm}_{\mathrm{SiO_2}}$ and d^{nm}_{GaAs}), the vertical propagation length of the generated surface plasmon waves in the main and control waveguides (graphene and metal waveguides)

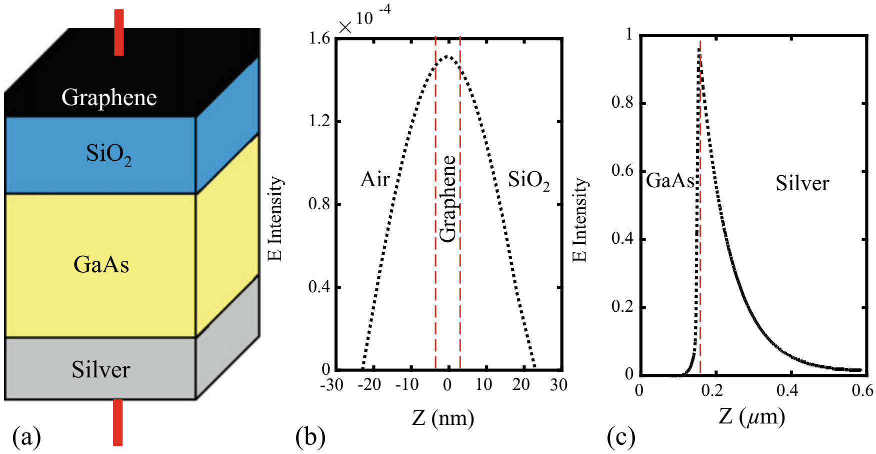


Fig. 3 **a** Order of the created capacitor layers in Fig. 1, **b** light penetration depth of the graphene (main) waveguide, **c** light penetration depth of the metal (control) waveguide

should be considered. The penetration depth in z-direction for two waveguides is shown in Figs. 3b and c. According to Figs. 3b and c, the values more than 500 and 20 nm should be considered for the heights of GaAs and SiO_2 layers, respectively. It is because of two reasons. First, existing the possibility of light modulation in the metal waveguide and secondly avoiding interferences between the two main and control waveguides.

As mentioned, two parameters of the permittivity values of SiO_2 and GaAs also affect the capacitance value. As a result, the $\varepsilon_r^{SiO_2}$ and ε_r^{GaAs} values should be determined. In all conditions, the permittivity of SiO_2 is constant and equal to 3.9. Therefore, the capacitance value can be expressed as

$$C_{ox} = \frac{\varepsilon_r^{\mathrm{GaAs}}\varepsilon_0}{\frac{20\,\mathrm{nm}+d_1}{3.9} \times \varepsilon_r^{\mathrm{GaAs}} + 500\,\mathrm{nm} + d_2} \tag{7}$$

when light is not modulated inside the GaAs material, the permittivity of GaAs is equal to 12.9. By considering $d_{\mathrm{SiO_2}}^{\mathrm{nm}} = 100$ and $d_{\mathrm{GaAs}}^{\mathrm{nm}} = 500$ nm, the DC gate capacitor can be calculated from Eq. 7. The obtained C_{ox}^{DC} is equal to 137.4 μf, which refers to chemical potential. The calculation procedure shows that the material type of the non-graphene gate is not important in this section. To achieve a chemical potential of 0.31 eV in graphene MZI structure, the applied voltage value can be calculated from Eq. 8 which is equal to 82 v.

$$|V_{\mathrm{DC}}| = \left(\frac{e}{\pi C_{ox}^{\mathrm{DC}}}\right)\left(\frac{\mu_c}{\hbar v_f}\right)^2 \tag{8}$$

In addition, by applying the obtained DC voltage, the gate capacitance should be equal to 128.8 μf for the chemical potential of graphene to be equal to 0.3 eV. This change in capacitance value is created by the light modulation in the AC mode of the gate capacitor.

B. AC mode of the gate capacitor

The use of surface plasmons between the non-graphene capacitor plate and insulator material is one of the techniques to change the capacitance value. The important point is that the calculation of the capacitance changes the chemical potential level in switching structure. In the micrometer frequency range, the refractive index of GaAs changes as.

$$n^2 - 1 = 4.372514 + \frac{5.466742\,\lambda^2}{\lambda^2 - (0.4431307)^2} + \frac{0.0242996\,\lambda^2}{\lambda^2 - (0.8746453)^2} + \frac{1.957522\,\lambda^2}{\lambda^2 - (36.9166)^2} \tag{9}$$

The permittivity variations of GaAs versus wavelength are shown in Fig. 4a. This figure is obtained from Eq. 9. The wavelength range of 1150–1550 nm is among the most important frequency ranges in telecommunication applications and optical integrated circuits. Consequently, this range is considered for GaAs permittivity variations. The capacitance value in the range of 1150–1550 nm is calculated using Eq. 8 and its variations are shown in Fig. 4b. Also, Fig. 4c shows the relationship between the wavelength of the control waves and the variation of the chemical potential. Based on the obtained results, the best wavelength for the control light is equal to 1300 nm.

After studying the created capacitor in two DC and AC modes, the operation of the proposed optical switch based on the main and control waveguides is investigated in the following. As mentioned above, the main waveguide includes air, graphene, and silica layers. Figure 5 shows this structure. In this figure, control light is not applied to the structure. Accordingly, the chemical potential of the total MZI structure is 0.31 eV. It is worth mentioning that the chemical potential of other places (black places) of the graphene layer is equal to 0 eV.

Figure 6 shows the field profile of the main waveguide by applying the main light source. As seen in Fig. 6, when the chemical potential level of two arms of the

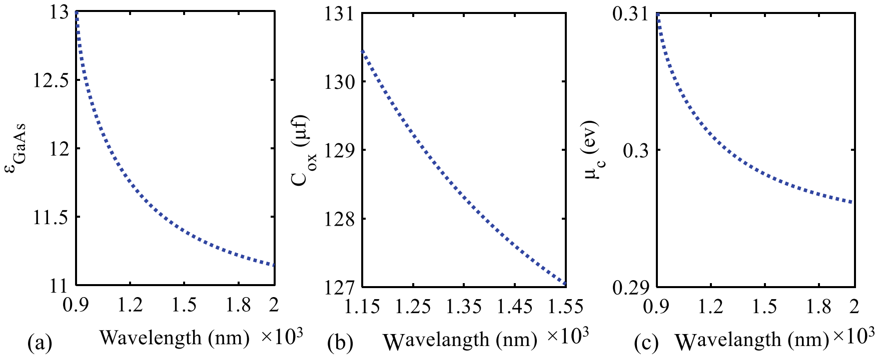
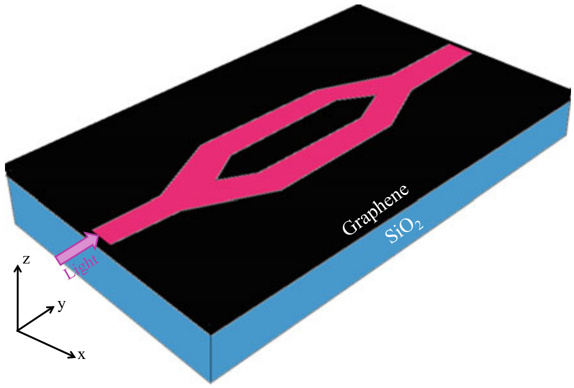


Fig. 4 **a** GaAs permittivity, **b** variations of the waveguide capacitance, and **c** variations of the chemical potential for the proposed switch

Fig. 5 Main waveguide of the optical switch (Fig. 1)



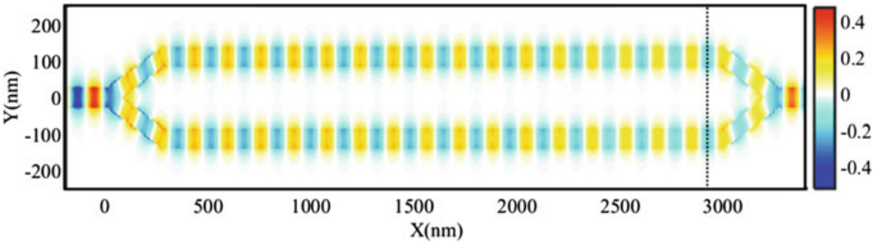


Fig. 6 Field profile of E_x at 1.42 µm for the main waveguide (with the chemical potential of 0.31 eV for both arms)

MZI structure is the same, the phase difference of waves between these two arms is 0 degrees. Consequently, the waves can transmit to the output port of the MZI structure.

The next is that the main and control light sources be applied, simultaneously. In this case, the control signal induces the Kerr effect in the non-linear medium (GaAs material) and one of the arms has a different chemical potential compared to the whole main waveguide. Figure 7 shows the mechanism of wave propagation for the different chemical potential levels for the two arms of the main waveguide (0.31 eV for the upper arm and 0.3 eV for the lower arm). As seen, after spending almost 2 μm from the arms, the phase difference of waves between two arms is equal to 180 degrees. Accordingly, the waves cannot be transmitted to the output port.

To provide a better view of the switching operation of the proposed optical switch, the transmission spectra of the main waveguide for the same and different chemical potentials of two arms have also been presented in Fig. 8. By comparison of two cases (Figs. 8a and b), the wavelength range of 1300–1700 nm is the best range for the main waves. It is because, in this wavelength range, the transmittance values are more than 40% and less than 10% for the identical and unequal waveguides, respectively.

Based on the obtained results, to achieve a phase difference of 180° between two arms of the graphene MZI structure, it is needed that a suitable control waveguide be designed. In other words, the control light should affect only one of the arms. For

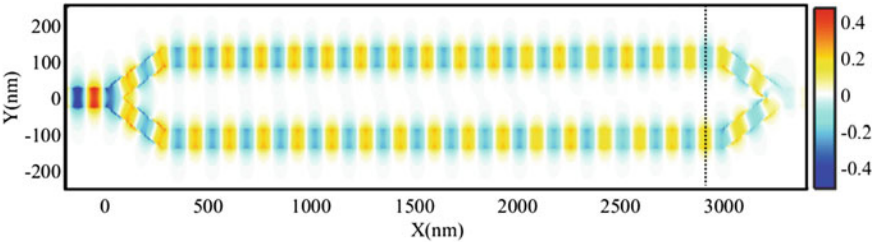


Fig. 7 Field profile of E_x at 1.42 µm for the main waveguide (with the chemical potential of 0.31 eV for the upper arm and 0.3 eV for the lower arm)

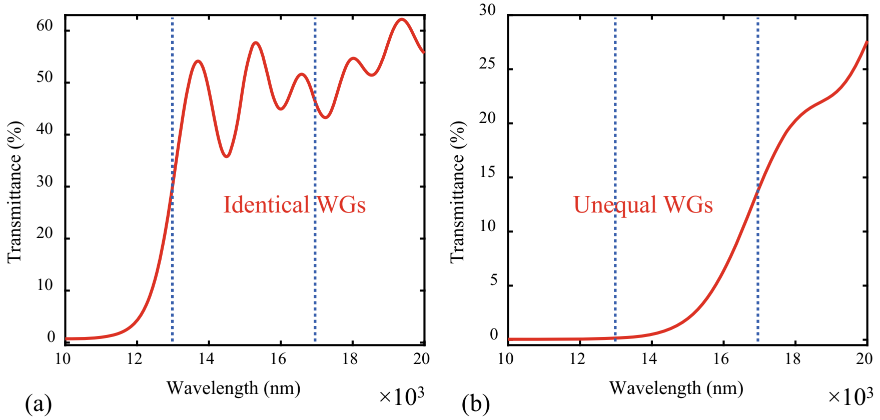


Fig. 8 Transmission spectrum of main waveguide for **a** the same chemical potentials of two arms, **b** the different chemical potentials of two arms

this purpose, two different techniques are proposed. The first one is that by grating one of the arms, the propagation of the surface plasmons is prevented to this arm. Figure 9a shows this structure (switch I). The presented structural parameters in this figure are as follows: $d_{10} = 500$, $d_{11} = 750$, $d_{12} = 20$, and $d_{13} = 500$ (all in nm). In the second technique, different plasmon frequencies are used for two arms of the metal MZI structure. Accordingly, one of the MZI arms is filled with vanadium material. The proposed structure (switch II) is shown in Fig. 9b. In the following, each of the mentioned methods has separately been studied.

Figure 10 shows the field profiles of the proposed switch I for both arms (simple and grating arms) with two main and control light sources. As seen in Fig. 10a, the waves can propagate through the simple arm. For the grating arm, the incident light cannot reach the end of the arm. Also, Fig. 11 shows the transmission spectra of the

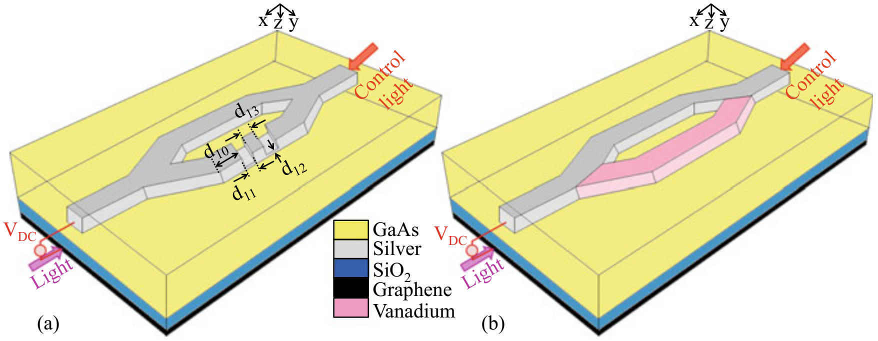


Fig. 9 The proposed all-optical graphene plasmon switch using a control waveguide with **a** a grating structure in one of the arms (switch I), **b** different metal material in one of the arms (switch II)

main waveguide for the simple and grating arms. As seen, the transmission value for the simple and grating arms are almost equal to 55% and 2%, respectively. Although the first proposed structure (switch I) has a suitable switching operation, it should be noted that the designed structure has some drawbacks. First of all, light propagation cannot be stopped at the beginning of the grating arm and it passes about to 1 μm of the arm. Secondly, due to the small size of the d_{12}, the fabrication process of this structure is difficult. Therefore, the second method has also been proposed (Fig. 9b).

As mentioned above, in the second method (switch II), two different metal materials of silver and vanadium which have different plasmon frequencies are used for two arms of the control waveguide. Due to the good electrical conductivity of vanadium, this metal is replaced in one of the silver arms. This causes that the incident light cannot propagate at the boundary of GaAs-vanadium. Figure 12 shows the field profile of this case. As seen, light propagation can be stopped at the beginning of the vanadium MZI arm.

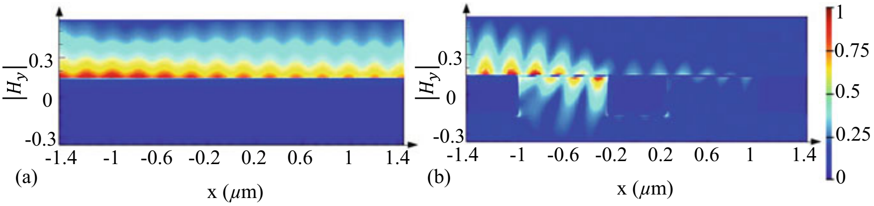


Fig. 10 Field profile of $|H_y|$ for the **a** simple arm, **b** grating arm

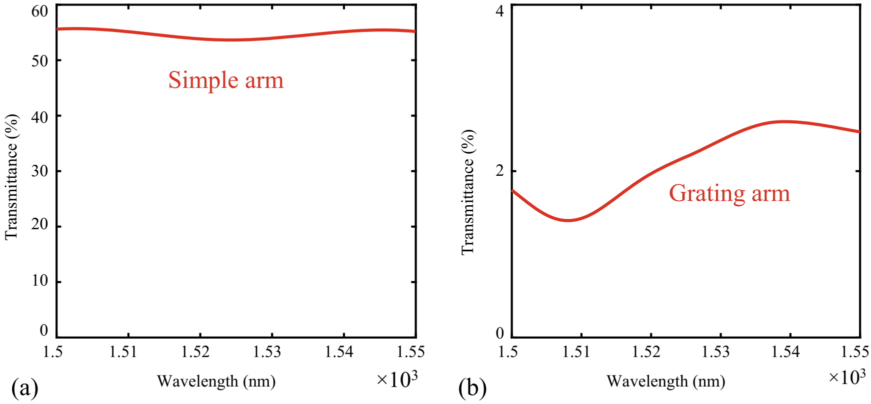


Fig. 11 Transmission spectrum of the main waveguide for the **a** simple arm, **b** grating arm

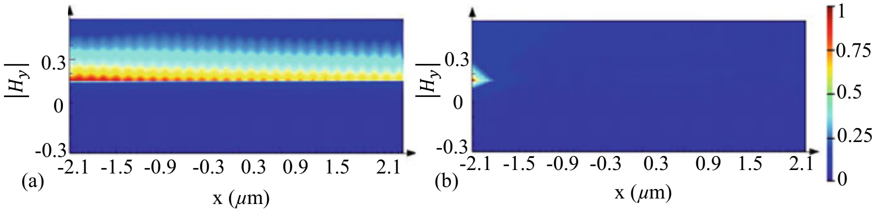


Fig. 12 Field profile of $|H_y|$ for the **a** silver arm, **b** vanadium arm

3 Nanomaterial Heterostructures-Based Switch

To benefit from the advantages of different optical structures, the combination of such structures can be used to design optical switches. The idea of using two different optical structures (plasmonic and PC configurations) has been used to design an all-optical switch in [100]. The design process of the proposed switch in [100] is discussed in the following.

The proposed heterostructure all-optical switch is shown in Fig. 13. As seen in this figure, the 1D PC structure used in this configuration is located at the intersection of MIM plasmonic and insulator waveguides. The insulator and metal materials in the horizontal MIM plasmonic waveguides are air and silver, respectively. The complex relative permittivity of silver is described by the Drude–Lorentz model (Eq. 4) [101]. In addition, the material used in the insulator waveguides is assumed SiO_2.

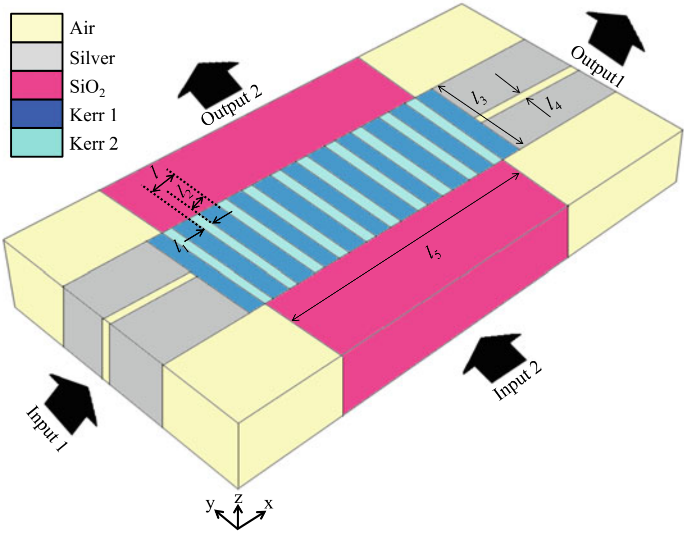


Fig. 13 Proposed heterostructure all-optical switch

The zoomed view of the central section of the proposed switch (PC section) is shown in Fig. 14a. This 1D PC structure is composed of the alternating non-linear insulator layers with the high refractive index of $n_H = 3.4$ (layers with the thickness of l_1) and the low refractive index of $n_L = 1.77$ (layers with the thickness of l_2). The structural parameters of the proposed switch are as follows: $l_1 = 120$, $l_2 = 290$, $l_3 = 700$, $l_4 = 100$, and $l_4 = 3836$ (all in nm). Also, the lattice constant of the 1D PC structure is l which is equal to $l = l_1 + l_2$.

The obtained band diagram of the used 1D PC structure using the plane wave expansion (PWE) method is shown in Fig. 14b. As seen, there is a PBG at the wavelength range of extended from 1534 to 2320 nm. To verify the performance of the used 1D PC structure, its transmission spectrum is also obtained using the finite-difference time-domain (FDTD) method (Fig. 14c). It can be seen that the transmission value is equal to zero at the mentioned PBG wavelength range. Accordingly, the sharp edges of the created PBG can be used for the switching mechanism. Based on the obtained

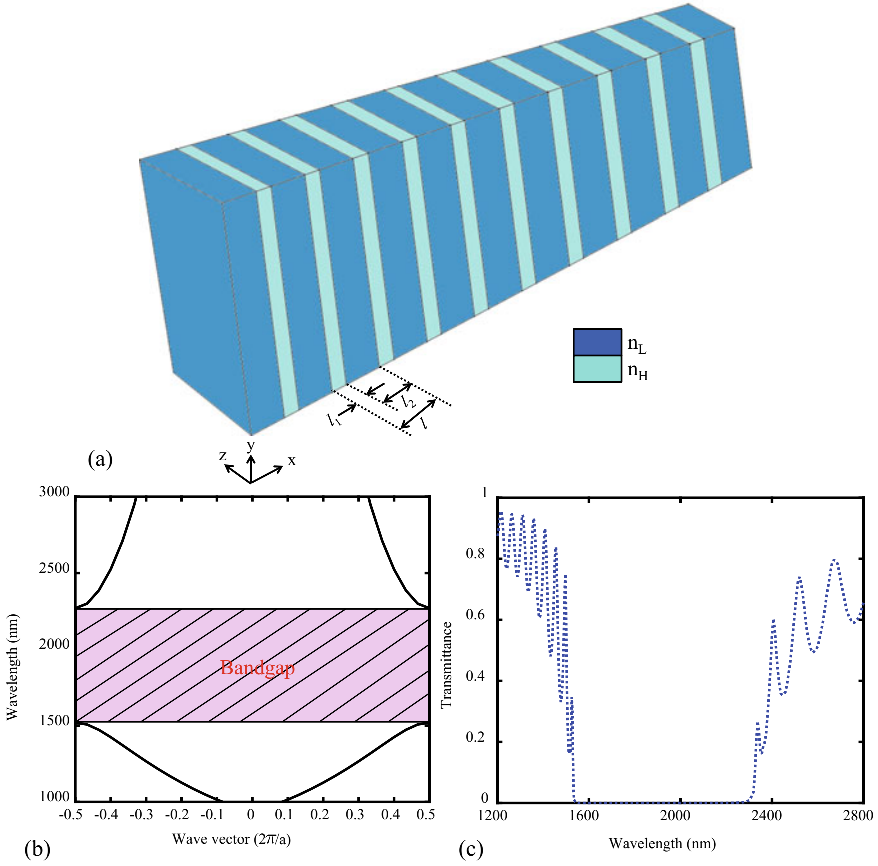


Fig. 14 **a** Proposed structure of 1D PC, **b** its band diagram, **c** its transmission spectrum

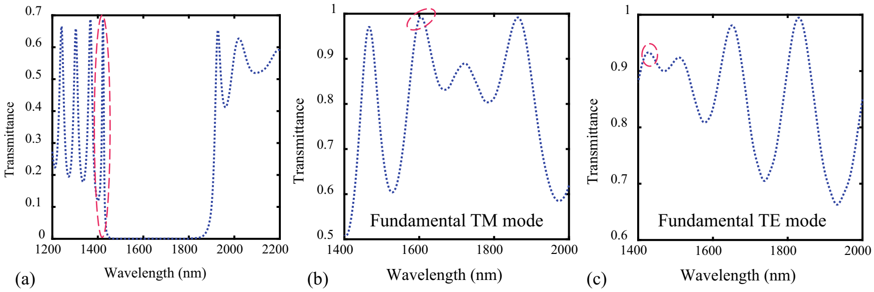


Fig. 15 Transmission spectrum of the proposed switch (Fig. 13) **a** from Input 1 to Output 1, **b** from Input 2 to Output 2 for TM mode, and **c** from Input 2 to Output 2 for TE mode

results, the horizontal and vertical waveguides (MIM plasmonic and insulator waveguides) are used for data and control signals, respectively. Suitable isolation between two signals can be achieved using this method.

After investigating the performance of the used 1D PC structure located at the center of the proposed topology (Fig. 13), the switching mechanism of the designed all-optical switch is studied. For this purpose, the transmission spectra of the designed switch for data (from Input 1 to Output 1) and control signals (from Input 2 to Output 2) are obtained. Figure 15a shows the transmission spectrum of the data signal when only the data signal source (located at Input 1) is "on". As seen, the created sharp PBG edge (low edge) can be selected for the data wavelength. Also, the transmission spectra of the control signals for TM and TE modes are shown in Figs. 15b and c, respectively. In this case, only the control signal source (located at Input 2) is applied. As seen in Fig. 15b and c, the wavelengths of 1606 and 1427 nm for fundamental TM and TE modes can be suitable choices for the control wavelengths. It is because these two wavelengths can propagate from Input 2 to Output 2 almost completely and they are located at the PBG of the data signal.

At the next step, both data and control signal sources are simultaneously applied to Input 1 and Input 2, respectively. In this case, the refractive indexes of Kerr materials (Kerr 1 and Kerr 2) change, and the transmission spectrum of the data signal shifts to higher wavelengths. According to the used Kerr material type, three situations can occur. These three situations are as follows:

A. Situation 1: $n_{2L} \gg n_{2H}$

When the non-linear refractive index coefficient (n_2) of the wide layers (layers with the thickness of l_2 and the refractive index of n_L) is much more than the narrow layers (layers with the thickness of l_1 and the refractive index of n_H), only the refractive indexes of the wide layers change. As a result, in this situation, the data transmittance shift is only due to the refractive index changes of n_L. This case is shown in Fig. 16a. As seen by changing the refractive index of n_L ($\Delta n_L = 0.01$ and $\Delta n_H = 0$), the data signal can propagate to the output port. In other words, the transmittance value at the

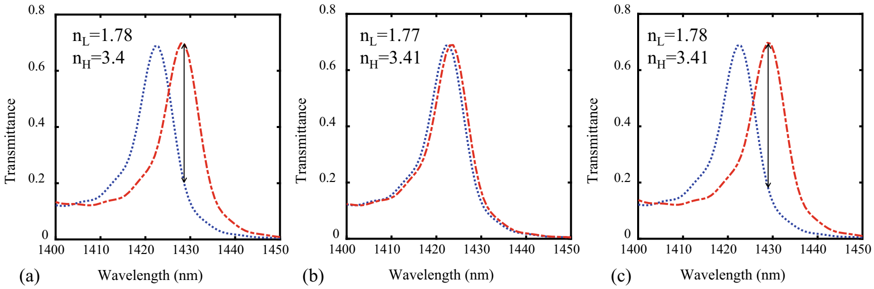


Fig. 16 Transmission spectra of the proposed switch (Fig. 13) without and with control signal source for **a** situation 1, **b** situation 2, and **c** situation 3

data wavelength (1429 nm) increases from 0.15 (blue curve) to 0.698 (red curve). As a result, the switching operation can occur in this case.

B. Situation 2: $n_{2H} \gg n_{2L}$

Based on the mentioned explanations in the previous case, when $n_{2H} \gg n_{2L}$, the data transmittance shift is only caused by the refractive index changes of n_H. Figure 16b shows this case ($\Delta n_H = 0.01$ and $\Delta n_L = 0$). As seen in this figure, by increasing the refractive index of n_H, the data signal shift is very imperceptible. Accordingly, the switching operation cannot occur in this case.

C. Situation 3: $n_{2H} \simeq n_{2L}$

In the last situation, the non-linear effect in the narrow layers is equal to the non-linear effect in the wide layers of the PC structure. Consequently, by applying the control signal source, both refractive indexes of n_H and n_L increase, simultaneously. This case is shown in Fig. 16c ($\Delta n_H = 0.01$ and $\Delta n_L = 0.01$). This figure shows that the transmittance value of the data signal increases from 0.15 to 0.7, and the switching operation occurs.

To give a better view of the refractive index changes on the switching mechanism of the proposed structure, the field profiles of E_y magnitude for two moods of "off" and "on" states of the designed switch are presented in Fig. 17. As seen in Fig. 17a, when only the data signal source is applied, the data wavelength cannot propagate from Input 1 to Output 1. In other words, the designed switch is at the "off" state. As reviewed, by applying two data and control signal sources for two situations 1 and 3, the switching operation can occur. The field profiles of the two situations are shown in Fig. 17b and c, respectively. As seen in these figures, the incident light can pass through the structure in these cases.

After investigating the general switching operation of the proposed structure, its switching mechanism is studied using practical Kerr materials with the mentioned properties. Transmission spectra of the designed switch using different practical Kerr materials are shown in Fig. 18. In the first, second, and third switches (S_1, S_2, S_3), the Kerr materials with the non-linear properties of $n_{2(CS2)} \simeq n_{2(Si)}$, $n_{2(AuSiO2)} \gg n_{2(GaAs)}$,

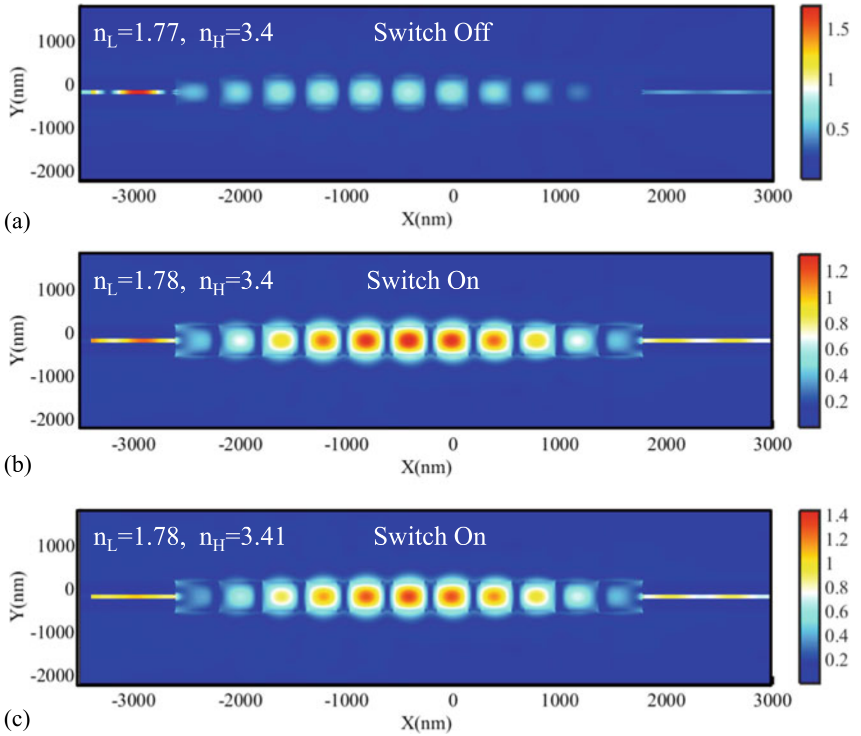


Fig. 17 Field profile of $|E_y|$ at the data wavelength (1429 nm) **a** with only data signal source ("off state"), **b** with both data and control signal sources for situation 1 ("on state"), and **c** with both data and control signal sources for situation 3 ("on state")

and $n_{2(\mathrm{GGS\text{-}Au10})} \simeq n_{2(\mathrm{Si})}$ are used, respectively. Also, Table 2 shows the features of such switches. As seen in this table, based on the transmittance values of the "on" and "off" states of such switches (T_{on} and T_{off}), the switching operation occurs in all three cases.

To provide a better view of the proposed switch operation, its time-domain behavior is also presented. In this section, two non-linear Kerr materials with the refractive indexes of 1.77, 3.4 and the $\chi^{(3)}$ values of 2×10^{-18}, 4.8×10^{-18} m^2/v^2 are used to fill the PC structure to achieve an all-optical switch. Figures 19a shows the time-domain data and control signal sources inserted into the input ports (Input 1 and Input 2) of the proposed switch. Here, the wavelengths of the data signal and the control signal (fundamental TE mode) are chosen 1926 and 1427 nm, respectively.

As seen in Fig. 19a, the data signal source is a continuous-wave signal. Also, the time-domain control signal source is a continuous-wave signal with an input power of 29.3 MW/cm. This signal is at the "on" state from 2.75 to 4.25 ps. The transition time of 500 fs is considered for the rising and falling edges. Figure 19b shows the time-domain output signal derived from data and control input sources. As seen, when only the data signal source is on, the output power is more than 0.6. In 2.75 to

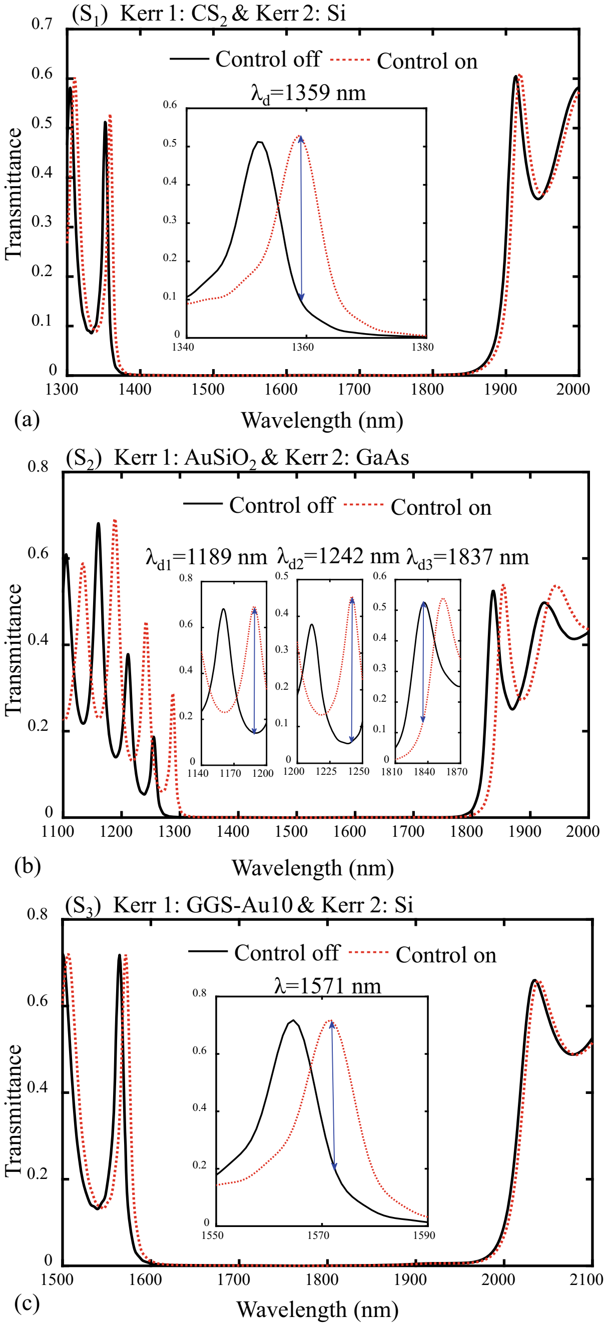


Fig. 18 Transmission spectra of the proposed switch (Fig. 13) without and with control signal source for two Kerr materials of **a** CS_2, Si, **b** $AuSiO_2$, GaAs, and **c** GGS-Au10, Si

Table 2 Mechanism of the heterostructure all-optical switch (Fig. 13) with different Kerr materials

Switch	Kerr material	n_0	n_2 (cm^2/w)	$\chi^{(3)}$ (m^2/v^2)	λ_d (nm)	T_{on}	T_{off}	I (MW/cm^2)
S_1	CS_2 Si	1.63 3.5	0.3×10^{-13} 0.43×10^{-13}	4×10^{-20} 2.8×10^{-19}	1359	0.525	0.09	232,000
S_2	$AuSiO_2$ GaAs	1.47 3.4	2.7×10^{-9} 1.59×10^{-13}	2.37×10^{-15} 9.7×10^{-19}	1189 1242 1837	0.692 0.454 0.53	0.14 0.06 0.16	22.2
S_3	GGS-Au10 Si	2 3.5	1.8×10^{-13} 0.43×10^{-13}	3.8×10^{-19} 2.8×10^{-19}	1571	0.72	0.19	66,600

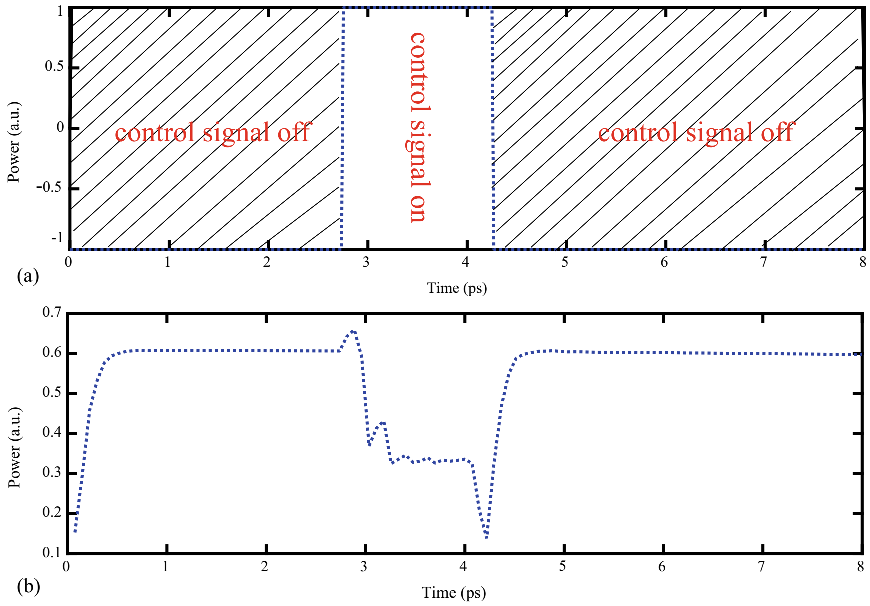


Fig. 19 Time-domain representations of **a** data and control signals, **b** output signal for the proposed heterostructure all-optical switch

4.25 ps (when both data and control signal sources are on), the transmitted power to the output port decreases. The main drawback of the time-domain output signal is its low contrast ratio value. To improve the contrast ratio value, a suppression filter should be added to the output port which is discussed in the following.

Figure 20 shows the proposed heterostructure all-optical switch with its suppression filter. The designed filter is composed of seven unequal stubs coupled to a MIM plasmonic waveguide. The structural parameters of the designed suppression filter

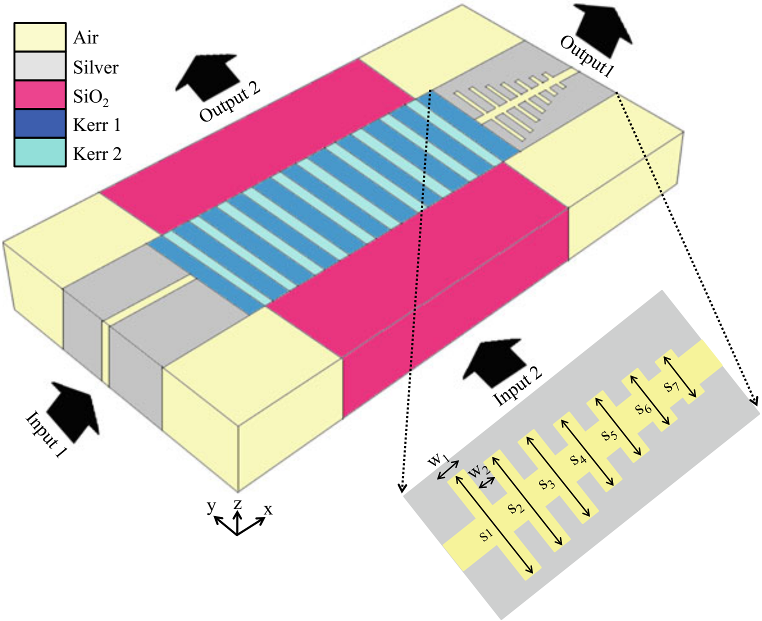


Fig. 20 Proposed heterostructure all-optical switch with the suppression filter at the output port

are as follows: $s_1 = 400$, $s_2 = 360$, $s_3 = 320$, $s_4 = 280$, $s_5 = 240$, $s_6 = 200$, $s_7 = 160$, $w_1 = w_2 = 50$ (all in nm).

Figure 21 shows the time-domain input and output signals of the proposed heterostructure all-optical switch with the suppression filter at the output port (Fig. 20). It is worth mentioning that the data and control input signals are similar to Figs. 19a. As seen, here the switching operation also occurs, but the output power for the “off” state reduces due to the existence of the suppression filter. As a result, the contras value increases. Figure 21b shows that the values of the rise and fall times are equal to 250 fs.

To achieve higher transmittance values in edges and out of the PBG, tapered resonators are added to the first proposed all-optical switch. The schematic of the improved proposed all-optical switch is shown in Fig. 22. The values of the structural parameters of the tapered resonators are equal to $d_6 = 117$ and $d_7 = 250$ nm. Other parameters have been already explained.

The transmittance spectra of the data and control signals are shown in Fig. 23. According to Fig. 23a, the transmittance values in all of the wavelengths (obviously, except for the PBG) increase. Increasing the transmittance values in the PBG edges provides all-optical switches with more transmittance values for the “on” state which is because there is more coupling strength between MIM plasmonic waveguides and the PC structure in this all-optical switch. As known, there is a trade-off between designing parameters of all-optical switches. Increasing the coupling effect between

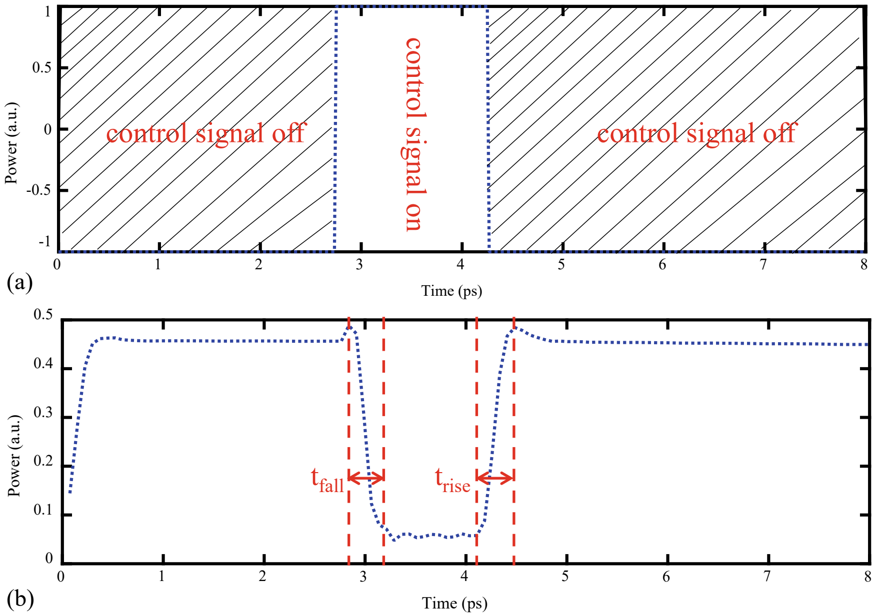


Fig. 21 Time-domain representations of **a** data and control signals, **b** output signal for the proposed heterostructure all-optical switch with the suppression filter at the output port

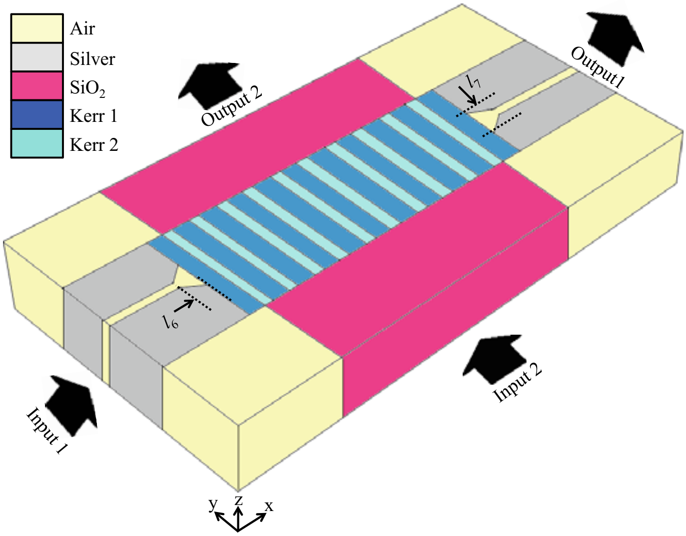


Fig. 22 Improved proposed heterostructure all-optical switch

MIM plasmonic waveguides and PC leads to slower sharpness in transition from the maximum transmittance to the minimum transmittance.

Since the existing coupling between the insulator waveguides and PC structure does not change, the transmittance spectrum of the pump signal (from Input 2 to Output 2) is similar to the previous all-optical switch (Figs. 15b and c). Here the switching mechanism of the improved proposed all-optical switch for different three situations is also investigated.

Figure 24 shows the transmittance spectra for three different situations including $n_{2L} \gg n_{2H}$, $n_{2H} \gg n_{2L}$, and $n_{2L} \approx n_{2H}$ for the improved proposed all-optical switch. As mentioned, for two modes of $n_{2L} \gg n_{2H}$ (Fig. 24a) and $n_{2L} \approx n_{2H}$ (Fig. 24c) the switching operation occurs. In contrary, the switching operation for $n_{2H} \gg n_{2L}$ cannot be obtained (Fig. 24b). Because the higher refractive index layers have a lower effect on the data signal transferring from Input 1 to Output 1. It should be noted that due to the lower sharpness of the PBG edges for the improved proposed all-optical switch, more control light intensity is needed to increase the refractive index of Kerr layers by a value of 0.02.

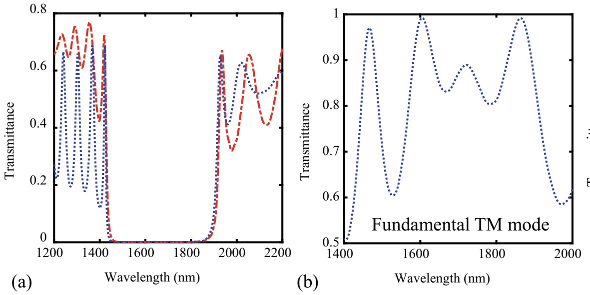


Fig. 23 Transmission spectrum of the improved proposed switch (Fig. 22) **a** from Input 1 to Output 1, **b** from Input 2 to Output 2 for TM mode, and **c** from Input 2 to Output 2 for TE mode

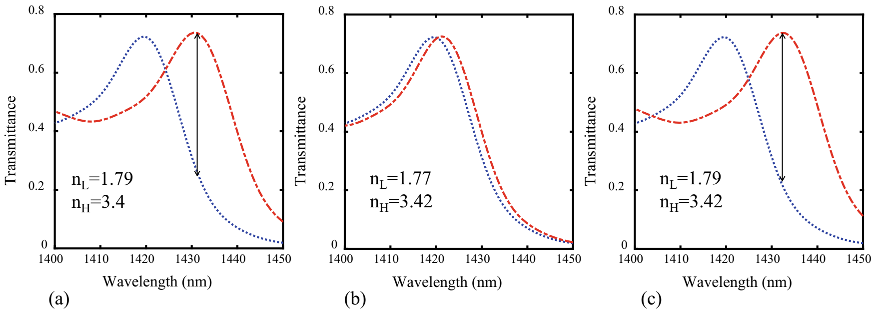


Fig. 24 Transmission spectra of the improved proposed switch (Fig. 13) without and with control signal source for **a** situation 1, **b** situation 2, and **c** situation3

The switching wavelength for the improved proposed all-optical switch in Figs. 24a and c is equal to 1432 nm. The maximum transmittance values at the "on" state of the improved all-optical switch are equal to 0.74 and 0.735 for situation 1 (Fig. 24a) and situation 2 (Fig. 24c), respectively. Also, the transmittance value at the "off" state of the switch is equal to 0.185 which is close to zero. Finally, providing suitable isolation between data and control signals, and requiring lower control intensities cause that the proposed structure can be a good candidate to use in other integrated optical circuits.

4 Modulation Characteristics

To bring electrical signals to the optical domain, optical systems usually require an optical modulator. As a result, optical modulators have found wide application in many optical systems. In optical modulators, by applying the control signal, the phase, frequency, or amplitude of the data signal can be modulated. This mechanism can produce phase modulators, frequency modulators, and amplitude modulators, respectively. In addition, the classification of optical modulators is performed based on the different methods which are used to design them. These methods are the electro-optic method [102, 103], thermo-optic method [104], and exploration of non-linear effects [105]. Since the optical nonlinearity is used for the active control of optical signals in all-optical modulators, such modulators have high modulation speeds. Consequently, numerous optical topologies have been used to design all-optical modulators so far [106–108]. The main drawbacks of the most designed optical modulators in the literature can be summarized as follows. First, in most of them, the time-domain behaviors of modulators have not been investigated. However, to prove the correct operation of modulators, it is necessary to present time-domain simulations when the non-linear Kerr effect is used. Also, most of them use a common path for both data and control signals, while an ideal modulator should be able to provide suitable isolation between the data and control signals.

One of the most applied optical structures to design optical modulators is plasmonic structures. In [109], a novel and simple topology has been proposed to design a plasmonic all-optical amplitude modulator. In this structure, it is tried to reform the mentioned disadvantages. The step-by-step designing procedure of the designed all-optical amplitude modulator is investigated in the following.

Figure 25 shows the proposed structure of the designed basic filter which is used to the realization of the proposed all-optical modulator. As seen in this figure, the filter structure consists of a non-linear nano-disk resonator, surrounded by a silicon ring-shaped resonator which is located inside a circular air resonator, and a horizontal MIM plasmonic waveguide. The metal and Kerr materials used in this structure are silver and InGaAsP, respectively. Silver is characterized by a comprehensive and accurate Drude–Lorentz model (Eq. 4). Also, the third-order non-linear susceptibility ($\chi^{(3)}$) and the linear dielectric constant (ε_0) of the Kerr material (InGaAsP) are equal to

$1 \times 10^{-18} \frac{\mathrm{m}^2}{\mathrm{V}^2}$ and 2.25, respectively. The structural parameters values of the proposed filter are as follows: $r_1 = 517$, $r_2 = 570$, $r_3 = 650$, and $w = 50$ (all in nm).

Figure 26a shows the transmission spectrum of the designed basic filter using FDTD simulations in the wavelength range of 600–1000 nm. As seen in this figure, the transmission spectrum has several resonance modes due to the presence of different resonance wavelengths at the designed resonator. Based on the obtained results, two of the generated resonance modes should be chosen for the data and control signals. Since it is needed that the control signal should pass almost completely through the structure and have gradual transitions, the resonance mode located at the wavelength of 702 nm can be a suitable choice for the control signal. Figure 26b shows the zoomed view of the selected control mode. Furthermore, the data signal should have a sharp edge and a high difference between its notch and peak transmittances values. Therefore, the Fano resonance mode located at the wavelength of 702 nm is selected for the data signal. The zoomed view of this signal is also shown in Fig. 26c.

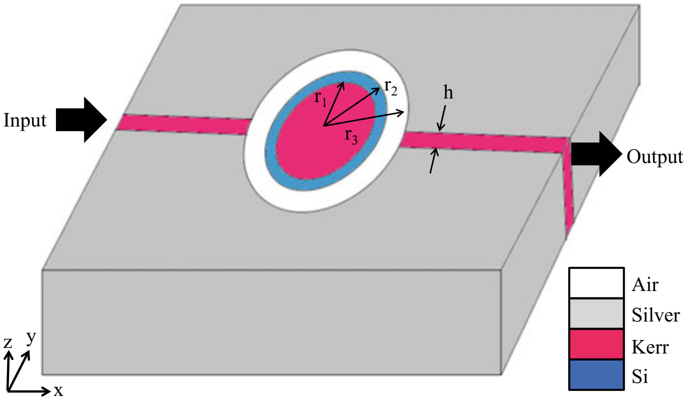


Fig. 25 Proposed structure of the designed basic filter

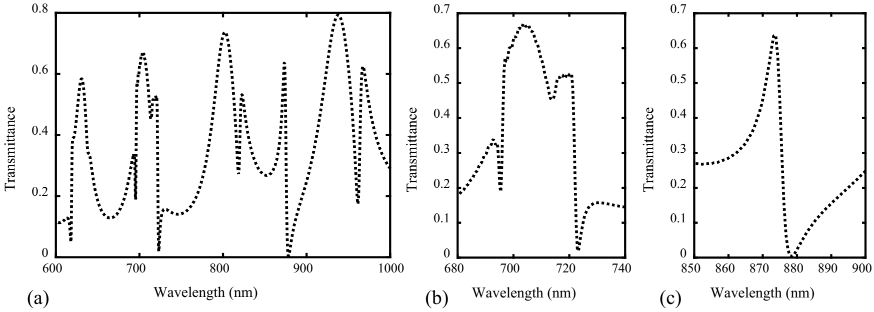


Fig. 26 **a** Transmission spectrum of the designed basic filter, **b** zoomed view of the selected mode for the control wavelength, and **c** zoomed view of the selected mode for the data wavelength

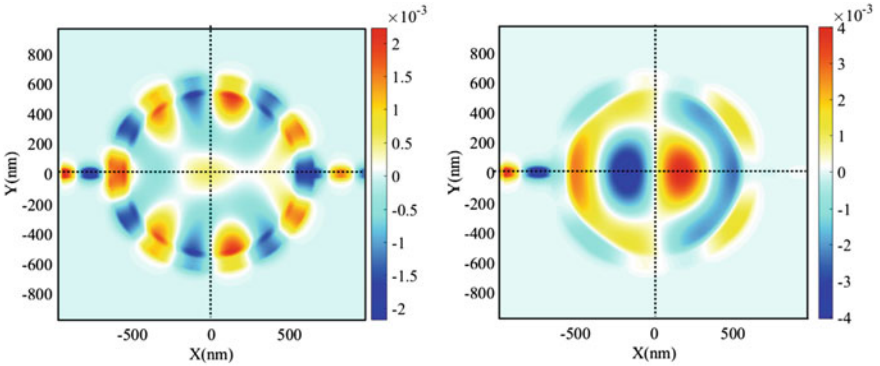


Fig. 27 Field profile of Re (H_z) for the basic filter at the resonance mode of **a** 702 nm and **b** 878 nm

As mentioned above, the goal of this design is that two isolated paths be used for data and control signals. As a result, it is needed that there be another waveguide to pass the control signal. Due to the symmetrical structure of the proposed resonator, another waveguide can be vertically coupled to the resonator. Accordingly, the selected data and control modes that pass through the horizontal and vertical waveguides, respectively should guarantee minimum cross-talk between two waveguides. For this purpose, the field profiles of the basic filter at the selected data and control wavelengths are studied.

Figure 27a and b shows the field profiles of the proposed basic filter at the wavelengths of 702 and 878 nm, respectively. As seen, both data and control wavelengths have even and odd symmetry along the horizontal and vertical axes, respectively. Therefore, each of the two selected wavelengths can pass through a waveguide (horizontal or vertical) without any leakage to another waveguide.

By adding the vertical waveguide to the basic filter, an all-optical amplitude modulator can be designed. Figure 28 shows the general structure of the proposed all-optical amplitude modulator. As seen in this figure, the horizontal and vertical waveguides are considered to pass the data and control signals, respectively. Using this method, an all-optical amplitude modulator with suitable isolation between the data and control signals can be achieved.

The transmission spectra of the data signal (from Data input to Data output port) with only the data signal source and control signal (from Control input to Control output port) with only the control signal source are shown in Fig. 29a and b, respectively. It is worth mentioning that the transmittances of the data and control signals be compared with the selected data and control signals of the basic filter in these figures. As seen, the transmission spectra of the designed modulator almost match the transmittances of the basic filter. In other words, the data and control wavelengths pass through the horizontal and vertical waveguides, respectively without any leakage. The next step to investigate the modulation operation of the proposed amplitude modulator is that the data and control signal sources be applied, simultaneously. This case is shown in Fig. 29c. As seen in this figure, the transmittance

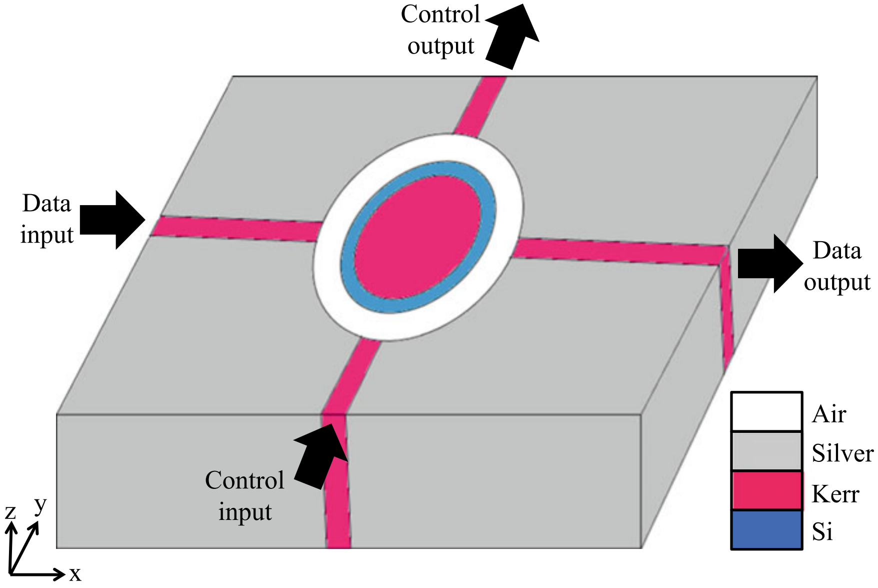


Fig. 28 Proposed structure of the all-optical amplitude modulator

value of the data wavelength (878 nm) modulates from 1.2 to 72.2% by applying the control signal source. It is because the control signal induces the Kerr effect in the non-linear medium.

The field profile of H_z magnitude is also shown in Fig. 30 to clarify the operating mechanism of the proposed amplitude modulator. Figure 30a shows that the data wavelength cannot transmit to the output port by applying only the data signal source.

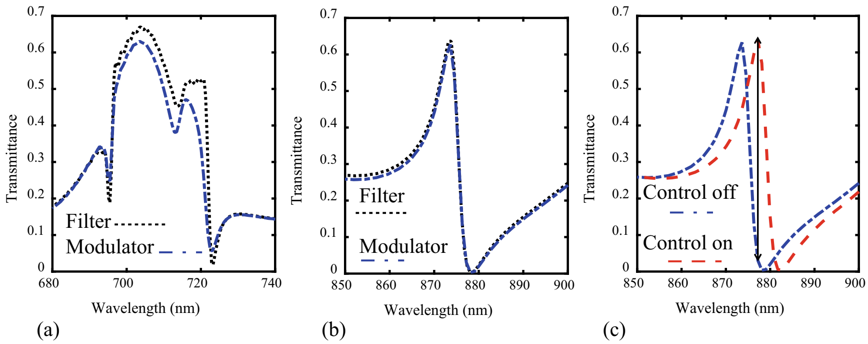


Fig. 29 Transmission spectra of **a** the proposed modulator (from Data input to Data output) with only data signal source and basic filter, **b** the proposed modulator (from Control input to Control output) with only control signal source and basic filter, and **c** the proposed modulator with both signal sources

In addition, as expected, by applying only the control signal source, the control wavelength can propagate through the structure. As discussed, the data wavelength can pass through the structure in the presence of both data and control signal sources. Figure 30c shows this case.

After designing the proposed all-optical amplitude modulator, to provide a better view of the designed modulator operation, its time-domain behavior is presented. For this purpose, the values of the input power of the data and control signal sources need to be specified. To determine the input power of the data signal source (P_{data}), a continuous wave at the wavelength of 878 nm is applied to Data input port. Figure 31a shows the normalized transmission of the data signal versus the variations of P_{data} (input power of the data signal source). As seen in this figure, when P_{data} increases from 0.195 to 0.78 W/μm, the maximum normalized transmission value does not change significantly, while by increasing P_{data} higher than 0.78 W/μm, the maximum normalized transmission value starts to increase. As discussed, the data wavelength cannot propagate through the Data output port by applying only the data signal source. As a result, the best value for the input power of the data signal source is equal to 0.78 W/μm.

To adjust the input power of the control signal source (P_{control}), the data and control signal sources are applied to the input ports (Data and Control input ports), simultaneously. The control signal source is a continuous-wave signal at the wavelength of 702 nm. The variations of the extinction ratio for different values of P_{control} are also shown in Fig. 31b. The extinction ratio of the proposed amplitude modulator can be defined as

$$ER = 10 \log \frac{T_{\text{on}}}{T_{\text{off}}} \tag{10}$$

where T_{on} and T_{off} are the transmittances in the presence and absence of the control signal source, respectively. As seen in Fig. 31b, the extinction ratio increases initially and then decreases by increasing P_{control}. Therefore, $P_{\text{control}} = 34.3$ W/μm for the highest extinction ratio value of 18.53 dB is an appropriate choice for designing the proposed amplitude modulator.

After determining the input power of the data and control signal sources (P_{data} and P_{control}), the time-domain behavior of the output signal is studied. The time-domain data and control signal sources inserted into the proposed modulator are shown in Fig. 32a and b, respectively. Figure 32a shows that the data signal source be a continuous-wave signal at the data wavelength of 878 nm with the selected input power of 0.78 W/μm. In addition, the time-domain control signal source is a continuous-wave signal at the control wavelength of 702 nm and the input power of 34.3 W/μm. As seen in Fig. 32b, this signal is “off” from 0 to 4 ps and 8 to 12 ps and it is “on” from 4.5 to 7.5 ps. The transition time of 500 fs is considered for the rising and falling edges.

The time-domain output signal derived from data and control signal sources is shown in Fig. 32c. As seen in this figure, in 0 to 4 ps and 8 to 12 ps (when only the data signal source is “on”), the output power is close to zero. When both data and control

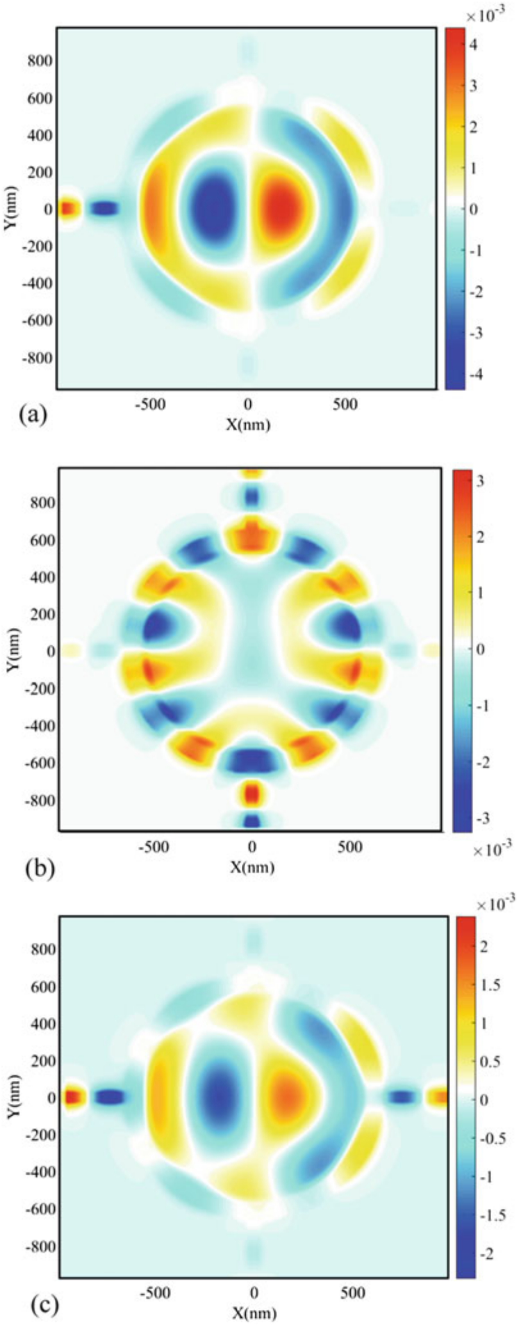


Fig. 30 Field profile of Re (H_Z) at **a** the data wavelength of 878 nm with only the data signal source, **b** the control wavelength of 702 nm with only the control signal source, and **c** the data wavelength of 878 nm with both data and control signal sources

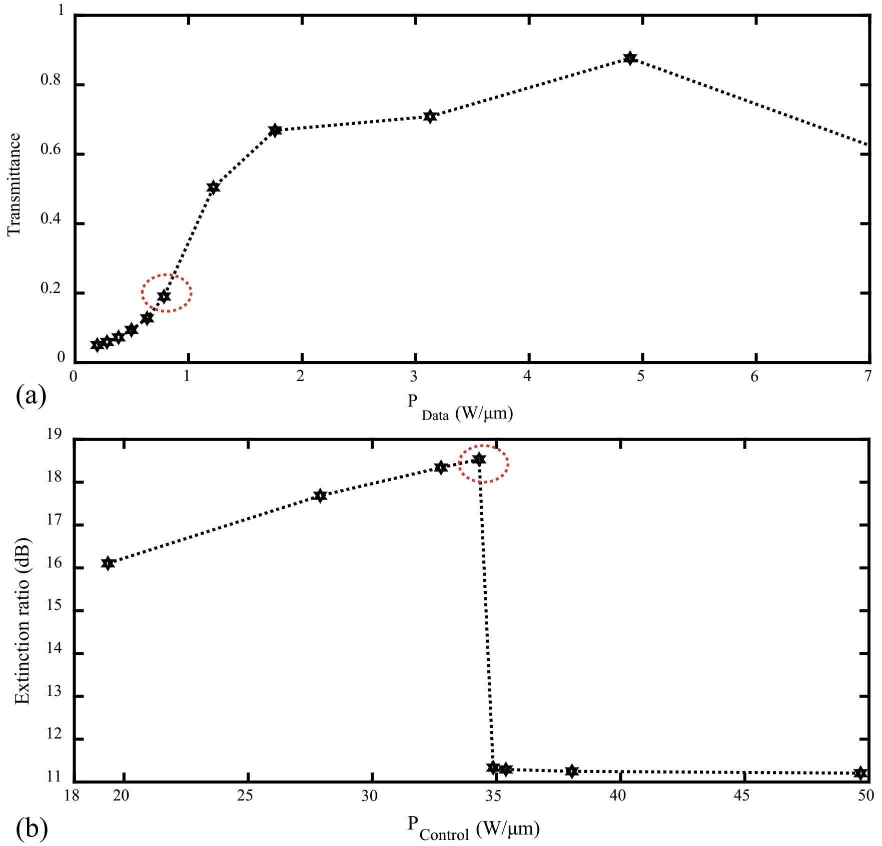


Fig. 31 The transmittance of the data wavelength for different values of P_{data}, **b** the extinction ratio of the output signal for different values of P_{control}

sources are “on” (in 4.5 to 7.5 ps), the amplitude of the data signal increases. In other words, the modulation operation occurs by applying the control signal source.

An important point for designing optical modulators is that the control signal should modulate the data signal and should not leak to the output port. For this purpose, the Fourier transform of the time-domain output signal is also obtained (Fig. 33). As seen in Fig. 33, the control wavelength of 702 nm leaks to the output port. Consequently, the control wavelength should be filtered at the output port.

Figure 34 shows the designed suppression filter and its transmission spectrum. This filter is composed of five unequal stubs coupled to a MIM plasmonic waveguide. The structural parameters of the proposed suppression filter are as follows: $a_1 = 100$, $a_2 = 97$, $a_3 = 94$, $a_4 = 91$, $a_5 = 88$, $w = 60$, and $g = 100$ (all in nm).

The transmission spectrum of the suppression filter (Fig. 34b) shows that the data wavelength can pass through the designed filter, while the control wavelength is blocked to the output port. As a result, by coupling the suppression filter at the Data

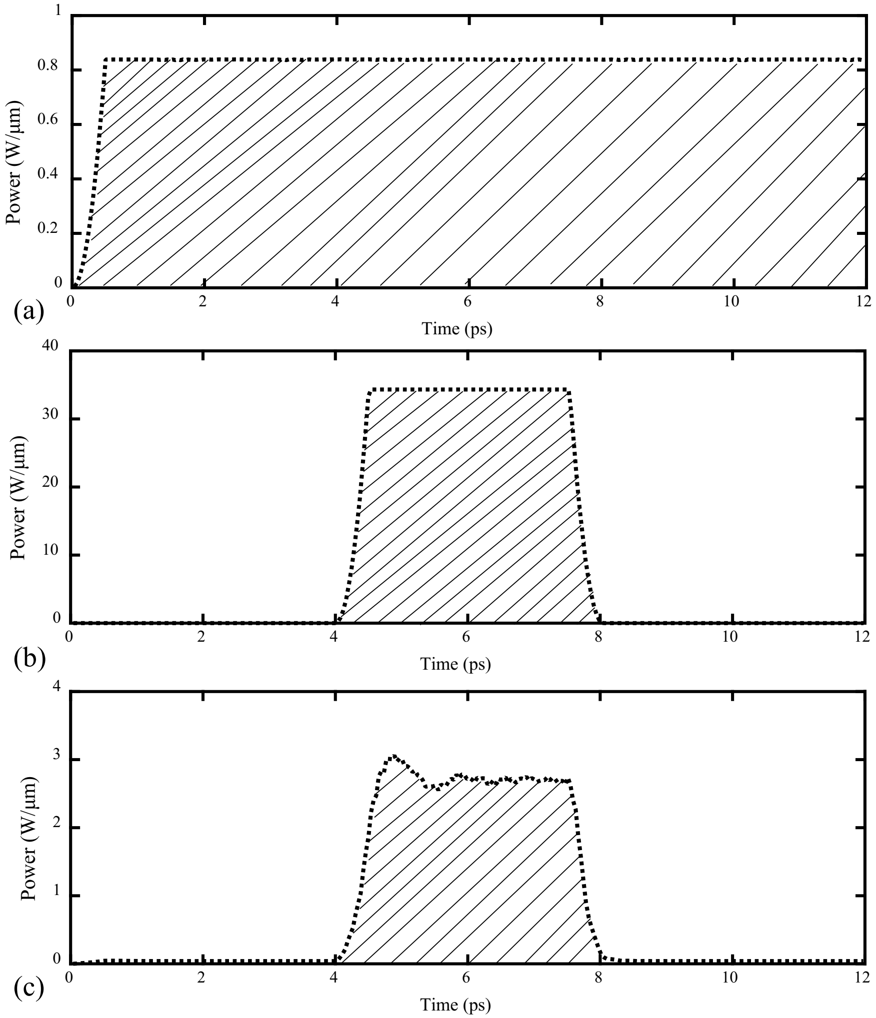


Fig. 32 Time-domain representations of **a** data signal, **b** control signal, and **c** output signal for the proposed all-optical amplitude modulator (Fig. 28)

output port of the proposed modulator, the control wavelength cannot propagate at the output port. The operating mechanism of the suppression filter can be also explained by presenting the field profile. Figure 35 shows the field profile of Re (H_z) at both wavelengths of 702 and 878 nm. As seen, there is no light transmission between the input and output ports at the wavelength of 702 nm, while the incident light at the wavelength of 878 nm can pass through the plasmonic suppression filter.

By the combination of the designed all-optical amplitude modulator (Fig. 28) and suppression filter (Fig. 34a), the improved all-optical amplitude modulator is formed. Figure 36 shows this structure.

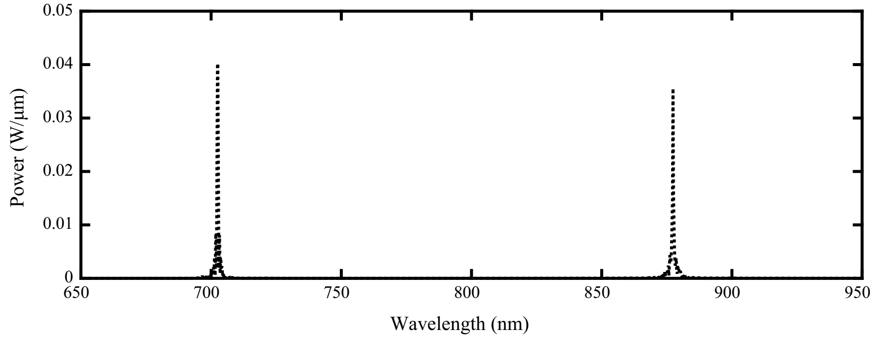


Fig. 33 Fourier transform of the time-domain output signal for the proposed all-optical amplitude modulator (Fig. 28)

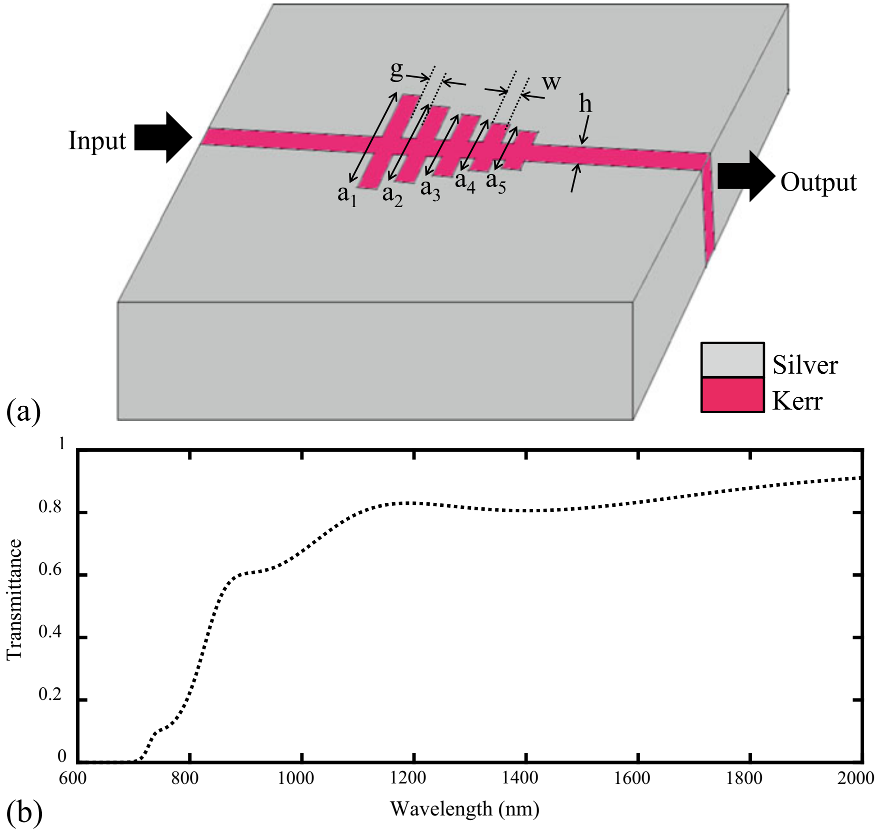


Fig. 34 Proposed structure of the suppression filter, **b** its transmission spectrum

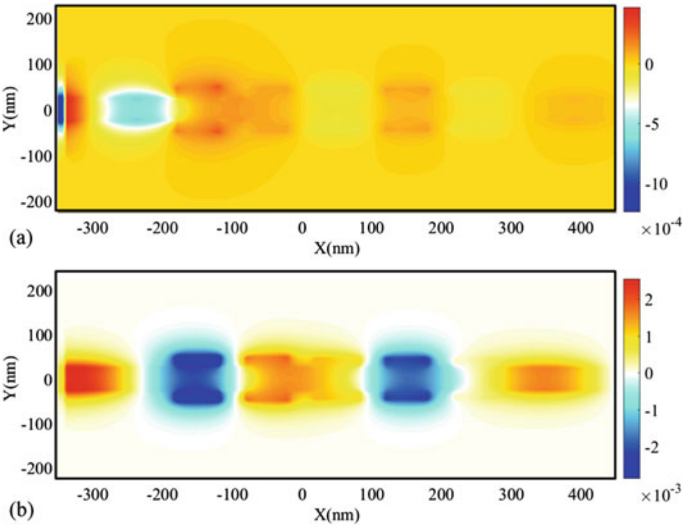


Fig. 35 Field profile of Re (H_z) at the wavelength of **a** 702 nm **b** 878 nm

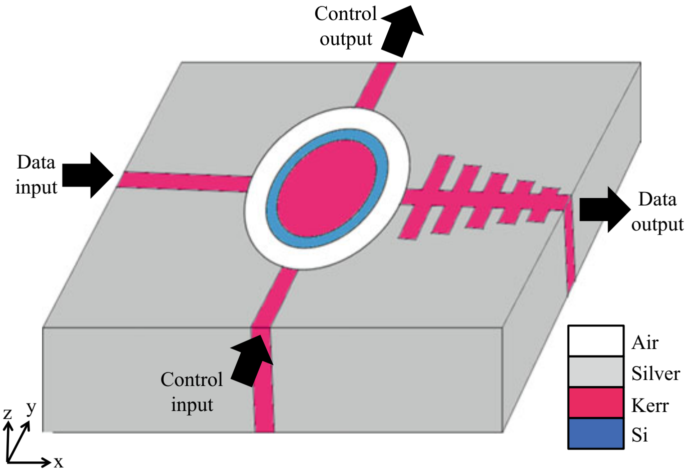


Fig. 36 Proposed structure of the improved all-optical amplitude modulator

Here, the time-domain behavior of the improved modulator is also investigated. The continuous-wave signals for the data and control signal sources are shown in Fig. 36a and b, respectively. As seen, such signals are similar to Fig. 32a and b.

Figure 36c shows the output signal derived from these two input signals. As seen in this figure, the amplitude modulation with a lower output power value also occurs in this case. It is because of the existence of the suppression filter at the output port.

To better analyze the results, the Fourier transform of the output time-domain signal of Fig. 37c is presented in Fig. 38. As seen, after adding the suppression filter which passes the data wavelength and prohibits the propagation of the control wavelength, only the data wavelength can transmit to the output port. As a result, the mentioned problem is solved using this method (coupling the suppression filter and all-optical modulator).

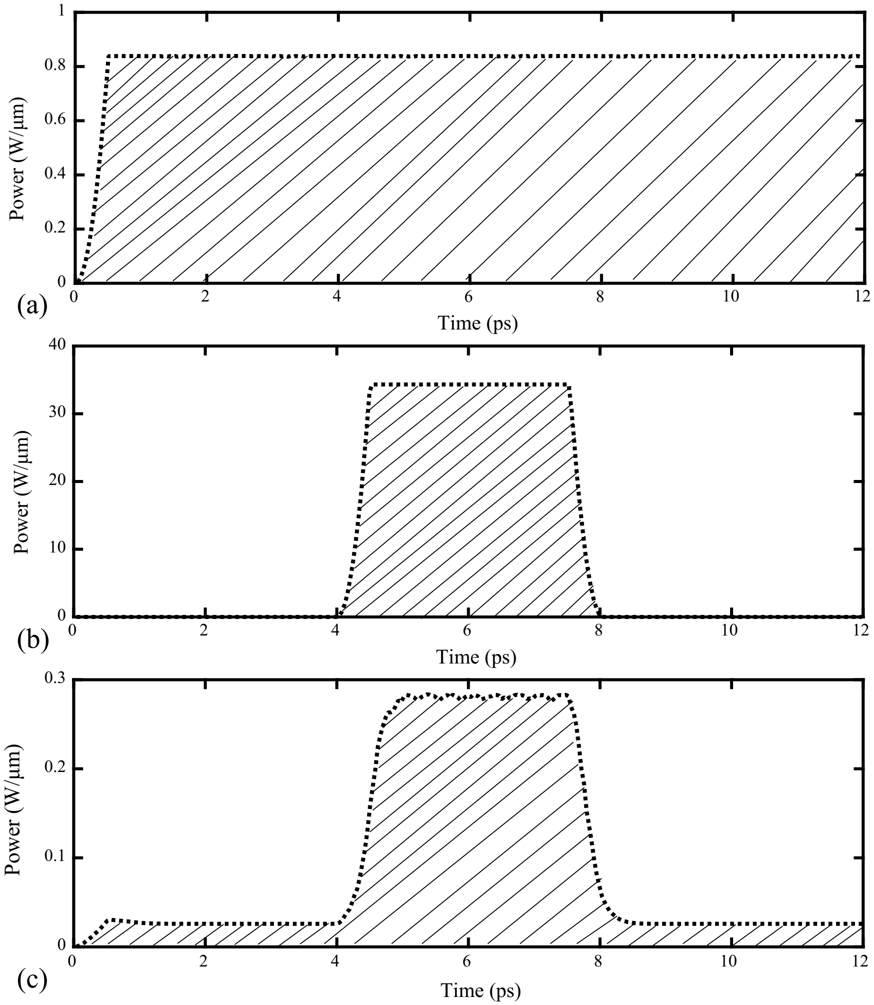


Fig. 37 Time-domain representations of **a** data signal, **b** control signal, and **c** output signal for the proposed improved all-optical amplitude modulator (Fig. 36)

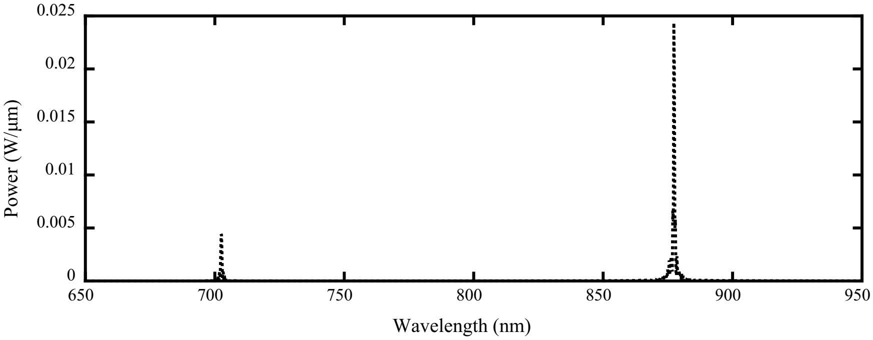


Fig. 38 Fourier transform of the time-domain output signal for the proposed improved all-optical amplitude modulator (Fig. 37)

5 Summary

Since the discovery of optical structures and initial studies on the optical switches for optical sensing, there have been a lot of publications to the literature on this topic, far more than those cited here. Optical switches can be good candidates for optical sensing applications. In this chapter, some of the publications have been studied. Finally, the main properties of the discussed structures in the previous sections are given in Table 3. Such features are the setup and topology of the optical devices, their mechanism operation, dimensions (2D/3D), and isolated waveguides. Also, it has been investigated whether the time-domain simulations are performed or not. In summary, based on the mentioned properties, these structures have the potential to be used as optical sensors.

Table 3 Summary of optical structures performance

Refs.	Setup	Topology	Mechanism operation	2D/3D	Isolated waveguide	Time simulation
Armaghani et al. [98]	Graphene and plasmonic	MZI	All-optical switch	3D	Yes	Yes
Khani et al. [100]	Plasmonic and PC	Cross waveguides	All-optical switch	2D	Yes	Yes
Khani et al. [109]	Plasmonic	Disk resonator and cross waveguides	All-optical amplitude modulator	2D	Yes	Yes

References

1. Miller KJ, Hallman KA, Haglund RF, Weiss SM (2017) Silicon waveguide optical switch with embedded phase change material. Opt Express 25(22):26527–26536
2. Khani S, Danaie M, Rezaei P (2021) Plasmonic all-optical metal–insulator–metal switches based on silver nano-rods, comprehensive theoretical analysis and design guidelines. J Comput Electron 20(1):442–457
3. Sani MH, Saghaei H, Mehranpour MA, Tabrizi AA (2020). A novel all-optical sensor design based on a tunable resonant nanocavity in photonic crystal microstructure applicable in MEMS accelerometers. Photonic Sens 1–15
4. Shenai K (2018) The figure of merit of a semiconductor power electronics switch. IEEE Trans Electron Devices 65(10):4216–4224
5. Picard JF, Schaub SC, Rosenzweig G, Stephens JC, Shapiro MA, Temkin RJ (2019) Laser-driven semiconductor switch for generating nanosecond pulses from a megawatt gyrotron. Appl Phys Lett 114(16):164102
6. Fasihi K, Bashiri S (2020) A new 2 × 1 photonic crystal multiplexer assisted by Fano resonances and Kerr nonlinear effect. Photonics Nanostruct Fundam Appl 42, 100837
7. Farmani A, Mir A, Irannejad M (2019) 2D-FDTD simulation of ultra-compact multifunctional logic gates with nonlinear photonic crystal. JOSA B 36(4):811–818
8. Tavousi A, Rakhshani MR, Mansouri-Birjandi MA (2018) High sensitivity label-free refractometer based biosensor applicable to glycated hemoglobin detection in human blood using all-circular photonic crystal ring resonators. Optics Commun 429:166–174
9. Foroughifar A, Saghaei H, Veisi E (2021) Design and analysis of a novel four-channel optical filter using ring resonators and line defects in photonic crystal microstructure. Opt Quant Electron 53(2):1–12
10. Rakhshani MR, Mansouri-Birjandi MA (2013) Realization of tunable optical filter by photonic crystal ring resonators. Optik 124(22):5377–5380
11. Sani MH, Ghanbari A, Saghaei H (2022) High-sensitivity biosensor for simultaneous detection of cancer and diabetes using photonic crystal microstructure. Opt Quant Electron 54(1):1–14
12. Aly AH, Zaky ZA, Shalaby AS, Ahmed AM, Vigneswaran D (2020) Theoretical study of hybrid multifunctional one-dimensional photonic crystal as a flexible blood sugar sensor. Phys Scr 95(3):035510
13. Okamoto K, Tsuruda K, Diebold S, Hisatake S, Fujita M, Nagatsuma T (2017) Terahertz sensor using photonic crystal cavity and resonant tunneling diodes. J Infrared Millimeter Terahertz Waves 38(9):1085–1097
14. Gao YF, Sun JP, Xu N, Jiang Z, Hou QC, Song H, Zhang C et al (2021) Manipulation of topological beam splitter based on honeycomb photonic crystals. Optics Commun 483:126646
15. Xu L, Wang Y, El-Fiky E, Mao D, Kumar A, Xing Z, Plant DV (2019) Compact broadband polarization beam splitter based on multimode interference coupler with internal photonic crystal for the SOI platform. J Lightwave Technol 37(4):1231–1240
16. Kondo K, Tatebe T, Hachuda S, Abe H, Koyama F, Baba T (2017) Fan-beam steering device using a photonic crystal slow-light waveguide with surface diffraction grating. Opt Lett 42(23):4990–4993
17. Mohammadi M, Seifouri M, Olyaee S, Karamirad M (2021) Optimization and realization all-optical compact five-channel demultiplexer using 2D photonic crystal based hexagonal cavities. J Comput Electron 20(2):984–992
18. Tavousi A (2019) Wavelength-division demultiplexer based on hetero-structure octagonal-shape photonic crystal ring resonators. Optik 179:1169–1179
19. Rajasekar R, Raja GT, Robinson S (2020) Numerical investigation of reconfigurable photonic crystal switch based on phase change nanomaterial. IEEE Trans Nanotechnol 19:545–552
20. Shiramin LA, Xie W, Snyder B, De Heyn P, Verheyen P, Roelkens G, Van Thourhout D (2017) High extinction ratio hybrid graphene-silicon photonic crystal switch. IEEE Photonics Technol Lett 30(2):157–160

21. Dou Y, Qiucui L, Xunya J (2019) New dynamic mechanism of interplay between nonlinearity and Bragg scattering in a femtosecond all-optical photonic crystal switch. J Opt 21(5):055501
22. Moradi M, Mohammadi M, Olyaee S, Seifouri M (2021) Design and simulation of a fast all-optical modulator based on photonic crystal using ring resonators. Silicon 1–7
23. Furukado Y, Abe H, Hinakura Y, Baba T (2018) Experimental simulation of ranging action using Si photonic crystal modulator and optical antenna. Opt Express 26(14):18222–18229
24. Salimzadeh A, Alipour-Banaei H (2018) An all optical 8 to 3 encoder based on photonic crystal OR-gate ring resonators. Opt Commun 410:793–798
25. Rezaei MH, Zarifkar A, Miri M (2018) Ultra-compact electro-optical graphene-based plasmonic multi-logic gate with high extinction ratio. Opt Mater 84:572–578
26. Rezaee S, Zavvari M, Alipour-Banaei H (2015) A novel optical filter based on H-shape photonic crystal ring resonators. Optik 126(20):2535–2538
27. Scholz S, Hess O, Rühle R (1998) Dynamic cross-waveguide optical switching with a nonlinear photonic band-gap structure. Opt Express 3(1):28–34
28. Liu Z, Zhou J, Li B (2013) Photonic band gap effect on energy transfer in Tb^{3+}, Eu^{3+} CO-Doped TiO_2 inverse opal films. Mod Phys Lett B 27(14):1350099
29. Khani S, Danaie M, Rezaei P (2019). All-optical plasmonic switches based on asymmetric directional couplers incorporating Bragg gratings. Plasmonics 1–11
30. Mansuri M, Mir A, Farmani A (2019) Numerical modeling of a nanostructure gas sensor based on plasmonic effect. J Optoelectronical Nanostruct 4(2):29–44
31. Khani S, Danaie M, Rezaei P (2018) Realization of single-mode plasmonic bandpass filters using improved nanodisk resonators. Opt Commun 420:147–156
32. Ghahari F, Walkup D, Gutiérrez C, Rodriguez-Nieva JF, Zhao Y, Wyrick J, Stroscio JA et al (2017) An on/off Berry phase switch in circular graphene resonators. Science 356(6340):845–849
33. Farmani H, Farmani A (2020) Graphene sensing nanostructure for exact graphene layers identification at terahertz frequency. Phys E 124:114375
34. Barnes WL, Dereux A, Ebbesen TW (2003) Surface plasmon subwavelength optics. Nature 424(6950):824–830
35. Bozhevolnyi SI, Volkov VS, Devaux E, Laluet JY, Ebbesen TW (2006) Channel plasmon subwavelength waveguide components including interferometers and ring resonators. Nature 440(7083):508–511
36. Gramotnev DK, Bozhevolnyi SI (2010) Plasmonics beyond the diffraction limit. Nat Photonics 4(2):83–91
37. Chen Z, Yu L (2014) Multiple Fano resonances based on different waveguide modes in a symmetry breaking plasmonic system. IEEE Photonics J 6(6):1–8
38. Khani S, Danaie M, Rezaei P (2019) Tunable single-mode bandpass filter based on metal–insulator–metal plasmonic coupled U-shaped cavities. IET Optoelectron 13(4):161–171
39. Kokabi M, Ghorbani S, Moayed SH (2021) Tunable bandpass plasmonic filter based on graphene as the nonlinear Kerr material. Laser Phys 31(2):026201
40. Pathiranage SPY, Gunapala SD, Premaratne M (2021). Tunable plasmonic resonator using conductivity modulated Bragg reflectors. J Phys Condens Matter
41. Zhuang H, Zhuang J, Kong F, Li K (2020) Tunable Bragg reflector with parallel bulk Dirac semimetals at terahertz frequencies. J Mod Opt 67(11):1010–1016
42. Mansouri M, Mir A, Farmani A, Izadi M (2021) Numerical modeling of an integrable and tunable plasmonic pressure sensor with nanostructure grating. Plasmonics 16(1):27–36
43. Khani S, Hayati M (2021). An ultra-high sensitive plasmonic refractive index sensor using an elliptical resonator and MIM waveguide. Superlattices Microstruct 106970
44. Khani S, Hayati M (2022) Optical biosensors using plasmonic and photonic crystal band-gap structures for the detection of basal cell cancer. Sci Rep 12(1):1–19
45. Khani S, Danaie M, Rezaei P (2018) Double and triple-wavelength plasmonic demultiplexers based on improved circular nanodisk resonators. Opt Eng 57(10):107102
46. Khani S, Farmani A, Mir A (2021) Reconfigurable and scalable 2, 4-and 6-channel plasmonics demultiplexer utilizing symmetrical rectangular resonators containing silver nano-rod defects with FDTD method. Sci Rep 11(1):1–13

47. Ghosh S, Rahman BMA (2019) Design of on-chip hybrid plasmonic Mach-Zehnder interferometer for temperature and concentration detection of chemical solution. Sens Actuators, B Chem 279:490–502
48. Khani S, Danaie M, Rezaei P (2020) Compact and low-power all-optical surface plasmon switches with isolated pump and data waveguides and a rectangular cavity containing nano-silver strips. Superlattices Microstruct 141:106481
49. Farmani A, Zarifkar A, Sheikhi MH, Miri M (2017) Design of a tunable graphene plasmonic-on-white graphene switch at infrared range. Superlattices Microstruct 112:404–414
50. Khani S, Danaie M, Rezaei P (2020) Realization of a plasmonic optical switch using improved nano-disk resonators with Kerr-type nonlinearity: a theoretical and numerical study on challenges and solutions. Opt Commun 477:126359
51. Negahdari R, Rafiee E, Emami F (2019) Realization of all-optical plasmonic MIM split square ring resonator switch. Opt Quant Electron 51(7):1–14
52. Khani S, Hayati M (2017) Compact microstrip lowpass filter with wide stopband and sharp roll-off. Microw J 60(11):86–92
53. Valizade A, Rezaei P, Orouji AA (2015) Design of reconfigurable active integrated microstrip antenna with switchable low-noise amplifier/power amplifier performances for wireless local area network and WiMAX applications. IET Microwaves Antennas Propag 9(9):872–881
54. Khani S, Danaie M, Rezaei P, Shahzadi A (2020) Compact ultra-wide upper stopband microstrip dual-band BPF using tapered and octagonal loop resonators. Frequenz 74(1–2):61–71
55. Pitarke JM, Silkin VM, Chulkov EV, Echenique PM (2006) Theory of surface plasmons and surface-plasmon polaritons. Rep Prog Phys 70(1):1
56. Jablan M, Buljan H, Soljačić M (2009) Plasmonics in graphene at infrared frequencies. Phys Rev B 80(24):245435
57. Li S, Liu H, Sun Q, Huang N (2016) Multi-channel terahertz wavelength division demultiplexer with defects-coupled photonic crystal waveguide. J Mod Opt 63(10):955–960
58. Zhang L, Tang L, Wei W, Cheng X, Wang W, Zhang H (2016) Enhanced near-infrared absorption in graphene with multilayer metal-dielectric-metal nanostructure. Opt Express 24(18):20002–20009
59. Dai D, Wang J, Chen S, Wang S, He S (2015) Monolithically integrated 64-channel silicon hybrid demultiplexer enabling simultaneous wavelength-and mode-division-multiplexing. Laser Photonics Rev 9(3):339–344
60. Hamouleh-Alipour A, Mir A, Farmani A (2020). Analytical modeling and design of a Graphene metasurface sensor for thermo-optical detection of terahertz plasmons. IEEE Sens J
61. Milaninia KM, Baldo MA, Reina A, Kong J (2009) All graphene electromechanical switch fabricated by chemical vapor deposition. Appl Phys Lett 95(18):183105
62. Novoselov KS, Geim AK, Morozov SV, Jiang D, Katsnelson MI, Grigorieva I, Firsov AA (2005) Two-dimensional gas of massless Dirac fermions in graphene. Nature 438(7065):197–200
63. Zhang Y, Tan YW, Stormer HL, Kim P (2005). Experimental observation of the quantum Hall effect and Berry's phase in graphene. Nature 438(7065):201–204
64. Koppens FH, Chang DE, Garcia de Abajo FJ (2011) Graphene plasmonics: a platform for strong light–matter interactions. Nano Lett 11(8):3370–3377
65. Vakil A, Engheta N (2011) Transformation optics using graphene. Science 332(6035):1291–1294
66. Aghaee T, Orouji AA (2020) Reconfigurable multi-band, graphene-based THz absorber: circuit model approach. Results Phys 16:102855
67. Hamzavi-Zarghani Z, Yahaghi A, Matekovits L, Farmani A (2019) Tunable mantle cloaking utilizing graphene metasurface for terahertz sensing applications. Opt Express 27(24):34824–34837
68. Aghaee T, Orouji AA (2021) Dual-band terahertz absorber based on graphene periodic arrays of disks and ribbons: circuit model approach. J Comput Electron 20(1):611–625

69. Khosravian E, Mashayekhi HR, Farmani A (2020) Tunable plasmonics photodetector in near-infrared wavelengths using graphene chemical doping method. AEU-Int J Electron Commun 127:153472
70. Tian X, Luo H, Wei R, Zhu C, Guo Q, Yang D, Qiu J (2018) An ultrabroadband Mid-infrared pulsed optical switch employing solution-processed bismuth oxyselenide. Adv Mater 30(31):1801021
71. Monfared SA, Seifouri M, Hamidi SM, Mohseni SM (2021) Electro-optical switch based on one-dimensional graphene-plasmonic crystals. Opt Mater 115:111051
72. Khani S, Hayati M (2022) Optical sensing in single-mode filters base on surface plasmon H-shaped cavities. Optics Commun 505:127534
73. Ismail M, Khan MI, Akhtar K, Khan MA, Asiri AM, Khan SB (2018) Biosynthesis of silver nanoparticles: a colorimetric optical sensor for detection of hexavalent chromium and ammonia in aqueous solution. Phys E 103:367–376
74. Jiang WS, Xin W, Xun S, Chen SN, Gao XG, Liu ZB, Tian JG (2017) Reduced graphene oxide-based optical sensor for detecting specific protein. Sens Actuators, B Chem 249:142–148
75. Rajasekar R, Parameshwari K, Robinson S (2019) Nano-optical switch based on photonic crystal ring resonator. Plasmonics 14(6):1687–1697
76. Ahmed AM, Mehaney A (2019) Ultra-high sensitive 1D porous silicon photonic crystal sensor based on the coupling of Tamm/Fano resonances in the mid-infrared region. Sci Rep 9(1):1–9
77. Nozhat N, Granpayeh N (2014) All-optical nonlinear plasmonic ring resonator switches. J Mod Opt 61(20):1690–1695
78. Khodadadi M, Moshiri SMM, Nozhat N (2020) Theoretical analysis of a simultaneous graphene-based circular plasmonic refractive index and thickness bio-sensor. IEEE Sens J 20(16):9114–9123
79. Liang C, Niu G, Chen X, Zhou Z, Yi Z, Ye X, Xiao S et al (2019) Tunable triple-band graphene refractive index sensor with good angle-polarization tolerance. Optics Commun 436:57–62
80. Nejat M, Nozhat N (2020) Sensing and switching capabilities of a graphene-based perfect dual-band metamaterial absorber with analytical methods. JOSA B 37(5):1359–1366
81. Li J, Tao J, Chen ZH, Huang XG (2016) All-optical controlling based on nonlinear graphene plasmonic waveguides. Opt Express 24(19):22169–22176
82. Luo L, Wang K, Ge C, Guo K, Shen F, Yin Z, Guo Z (2017) Actively controllable terahertz switches with graphene-based nongroove gratings. Photonics Res 5(6):604–611
83. Zhang Z, Yang J, He X, Han Y, Zhang J, Huang J, Xu S (2018) All-optical multi-channel switching at telecommunication wavelengths based on tunable plasmon-induced transparency. Optics Commun 425:196–203
84. Wang M, Zhang M, Wang Y, Zhao R, Yan S (2019) Fano resonance in an asymmetric MIM waveguide structure and its application in a refractive index nanosensor. Sensors 19(4):791
85. Zhang Y, Kuang Y, Zhang Z, Tang Y, Han J, Wang R, Liu W et al (2019) High-sensitivity refractive index sensors based on Fano resonance in the plasmonic system of splitting ring cavity-coupled MIM waveguide with tooth cavity. Appl Phys A 125(1):1–5
86. Shi X, Ma L, Zhang Z, Tang Y, Zhang Y, Han J, Sun Y (2018) Dual Fano resonance control and refractive index sensors based on a plasmonic waveguide-coupled resonator system. Optics Commun 427:326–330
87. Shahamat Y, Ghaffarinejad A, Vahedi M (2020) Plasmon induced transparency and refractive index sensing in two nanocavities and double nanodisk resonators. Optik 202:163618
88. Farmani A (2019) Three-dimensional FDTD analysis of a nanostructured plasmonic sensor in the near-infrared range. JOSA B 36(2):401–407
89. Bian ZY, Liang RS, Zhang YJ, Yi LX, Lai G, Zhao RT (2015) Multifunctional disk device for optical switch and temperature sensor. Chin Phys B 24(10):107801
90. Bazgir M, Jalalpour M, Zarrabi FB, Arezoomand AS (2020) Design of an optical switch and sensor based on a MIM coupled waveguide using a DNA composite. J Electron Mater 49(3):2173–2178

91. Shahamat Y, Vahedi M (2019) Mid-infrared plasmonically induced absorption and transparency in a Si-based structure for temperature sensing and switching applications. Optics Commun 430:227–233
92. Jiang X, Chen D, Zhang Z, Huang J, Wen K, He J, Yang J (2020) Dual-channel optical switch, refractive index sensor and slow light device based on a graphene metasurface. Opt Express 28(23):34079–34092
93. Ghafari S, Forouzeshfard MR, Vafapour Z (2019) Thermo optical switching and sensing applications of an infrared metamaterial. IEEE Sens J 20(6):3235–3241
94. Shahamat Y, Vahedi M (2017) Pump-tuned plasmon-induced transparency for sensing and switching applications. Optics Commun 401:40–45
95. Poorgholam-Khanjari S, Razavi Z, Zarrabi FB (2021) Reconfigurable optical rectangular particle array absorber based on metal–DNA–metal structure as a refractive index sensor and optical switch. Optics Commun 489:126866
96. Drachev VP, Chettiar UK, Kildishev AV, Yuan HK, Cai W, Shalaev VM (2008) The Ag dielectric function in plasmonic metamaterials. Opt Express 16(2):1186–1195
97. Falkovsky LA (2008). Optical properties of graphene. In: Journal of physics: conference series, vol 129, no 1. IOP Publishing, p 012004
98. Armaghani S, Khani S, Danaie M (2019) Design of all-optical graphene switches based on a Mach-Zehnder interferometer employing optical Kerr effect. Superlattices Microstruct 135:106244
99. Khani S, Danaie M, Rezaei P (2019) Design of a single-mode plasmonic bandpass filter using a hexagonal resonator coupled to graded-stub waveguides. Plasmonics 14(1):53–62
100. Khani S, Danaie M, Rezaei P (2020) Hybrid all-optical infrared metal-insulator-metal plasmonic switch incorporating photonic crystal bandgap structures. Photonics Nanostruct Fundam Appl 40, 100802
101. Ahmed I, Khoo EH, Kurniawan O, Li EP (2011) Modeling and simulation of active plasmonics with the FDTD method by using solid state and Lorentz-Drude dispersive model. JOSA B 28(3):352–359
102. Haffner C, Chelladurai D, Fedoryshyn Y, Josten A, Baeuerle B, Heni W, Leuthold J (2018) Low-loss plasmon-assisted electro-optic modulator. Nature 556(7702):483–486
103. Bhasker P, Norman J, Bowers J, Dagli N (2020) Low voltage, high optical power handling capable, bulk compound semiconductor electro-optic modulators at 1550 nm. J Lightwave Technol 38(8):2308–2314
104. Li F, Tang T, Li J, Luo L, Li C, Shen J, Yao J (2020) Chiral coding metasurfaces with integrated vanadium dioxide for thermo-optic modulation of terahertz waves. J Alloy Compd 826:154174
105. Liao C, Li C, Wang C, Wang Y, He J, Liu S, Wang Y (2019) High-speed all-optical modulator based on a polymer nanofiber Bragg grating printed by femtosecond laser. ACS Appl Mater Interfaces 12(1):1465–1473
106. Gardes FY, Brimont A, Sanchis P, Rasigade G, Marris-Morini D, O'Faolain L, Martí J (2009) High-speed modulation of a compact silicon ring resonator based on a reverse-biased pn diode. Opt Express 17(24):21986–21991
107. Moradiani F, Seifouri M, Abedi K, Gharakhili FG (2021) High extinction ratio all-optical modulator using a vanadium-dioxide integrated hybrid plasmonic waveguide. Plasmonics 16(1):189–198
108. Sun F, Xia L, Nie C, Qiu C, Tang L, Shen J, Du C (2019) An all-optical modulator based on a graphene–plasmonic slot waveguide at 1550 nm. Appl Phys Express 12(4):042009
109. Khani S, Danaie M, Rezaei P (2021) Fano Resonance using surface plasmon polaritons in a nano-disk resonator coupled to perpendicular waveguides for amplitude modulation applications. Plasmonics 1–18

Empirical, Statistical, and Machine Learning Techniques for Predicting Surface Settlement Induced by Tunnelling

Chia Yu Huat, Danial Jahed Armaghani, Ehsan Momeni, and Sai Hin Lai

Abstract Tunnels have been constructed in many countries around the world for different purposes, such as the metro system to mitigate traffic congestion. Since the construction of urban tunnels is typically conducted at shallow depths, specific concerns such as structural damage inevitably arise. Surface settlement induced by tunnelling is one of the common problems encountered during and after tunnelling construction. Therefore, accuracy in the prediction of surface settlement induced by tunnelling is important to prevent damage to the existing structures. Several methods have been previously proposed to compute tunnelling-induced surface settlement, such as empirical, numerical, laboratory, statistical, and machine learning. Each of these models has advantages and disadvantages. This study deeply investigates the available techniques to estimate settlement induced by tunnelling and reviews the most important findings and solutions. Among the existing techniques, machine learning seems to be the most suitable and accurate in estimating settlement induced by tunnelling. These techniques, with their behind calculations and assumptions, are able to identify the best relations between independent and dependent parameters and, therefore, to solve complex and non-linear problems. The discussion provided in this study can be useful to those who are interested to conduct research or design in the same field.

C. Y. Huat
Department of Civil Engineering, Faculty of Engineering, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
e-mail: 17107717@siswa.um.edu.my

D. J. Armaghani (✉)
Department of Urban Planning, Institute of Architecture and Construction, South Ural State University, Engineering Networks and Systems, 454080 Chelyabinsk, Russia
e-mail: danialarmaghani@susu.ru

E. Momeni
Department of Civil Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran
e-mail: Momeni.E@lu.ac.ir

S. H. Lai
Department of Civil Engineering, Faculty of Engineering, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
e-mail: laish@um.edu.my

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*, Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_2

Keywords Tunnel construction · Surface settlement · Machine learning · Empirical · Statistical

Notation

S	Surface settlement in the transverse section at a distance
$S_{\max}$	Maximum surface settlement
x	Distance from the centreline of the tunnel
i	Point of inflection (settlement through)
V_s	Volume loss of the soil in m^3/m
D	Tunnel diameter
C	Tunnel cover
P	Pillar width
Q	Diagonal distance between the tunnel
$\mathrm{S_v}$ (x)	Transverse surface settlement of twin tunnelling
x_1	Distance from the centreline of the first tunnel
i_x	Point of inflection (settlement through) for 1st or 2nd tunnel
z	Tunnel depth axis
k	Empirical constant
Z'	Depth of the calculated settlement trough from the surface settlement
R	Tunnel radius
$\mathrm{S_{mod}}$ (x)	Modified surface settlement
F	Modification factor
d	Distance between tunnel from center to center
A	Multiple of $\mathrm{i_x}$ for full trough width.
M	Maximum modification factor
α	Coefficient of the construction
$\mathrm{S_m}$	Mean settlement directly above the tunnel
β	Laboratory test constant
γ	Laboratory test constant
y	Output variable
a_n	Regression coefficient
x'_n	Independent variable
c'	Constant
c	Soil cohesion
Φ	Soil friction angle
E	Soil elastic modulus
V	Penetration rate
F'	Thrust force
P	Grouting pressure
n	Percentage of grout fill
HR	Horizontal to vertical stress ratio
EI_{lining}	Flexural stiffness of tunnel lining

E_{grout}	Elastic modulus of grout
ML	Machine Learning
VL	Volume loss of the soil in %
NATM	New Austrian Tunnelling Method
EPBM	Earth Pressure Balance Machine
CCM	Convergence-Confinement method
MLR	Multiple Linear Regression
SC	Soft Computing
GP	Genetic Programming
GEP	Gene Expression Programming
ANN	Artificial Neural Network
FL	Fuzzy Logic
GA	Genetic Algorithm
BP	Back Propagation
RMSE	Root Mean Square Error
R	Correlation coefficient
ANFIS	Adaptive Neuro-Fuzzy Inference System
PSO	Particle Swarm Optimisation
MR	Multiple Regression
FS	Fuzzy System
PCA	Principal Component Analysis
TBM	Tunnelling Boring Machine
POD	Proper Orthogonal Decomposition
ANFIS-PC	Adaptive Neuro-Fuzzy Inference System-Principal Component
BPNN	Back Propagation Neural Network
GRNN	General Regression Neural Network
ELM	Extreme Learning Machine
SVM	Support Vector Machine
RF	Random Forest
FCM	Fuzzy C Means Clustering
RSE	Relative Strength of Effects
TGML	Theory-Guided Machine Learning
IBM	International Business Machines

1 Introduction

A tunnel is a cross-sectional structure with the opening driven through the soil, rock mass, or mixture [1]. High population growth and limited space have caused a high demand for transportation infrastructure usage. Tunnels can be built by several approaches, such as cut and cover, immersed, jack box tunnel, drilling and blasting, and mechanized tunnelling [2]. For instance, cut and cover [3–5] is one of the oldest methods for tunnel construction where the concepts involve excavation of the trench or rectangular hole from the ground surface to the required excavation depth. Then, the actual tunnel is constructed, and subsequently, the excavated trench is backfilled again. Immersed tunnelling is carried out under the sea, and this method comprises a prefabricated tunnel before being placed under the water. Figure 1 illustrates the old construction methods of tunnelling. As time has passed, technologies have improved; now tunnelling can also be carried out using mechanized tunnelling.

Tunnels are built in a variety of geological conditions, including soft ground, rock and soil mixtures, hard rock or weathered rock, and so on. Different tunnelling methods and geological conditions possess several problems and difficulties. Several issues can be encountered during the process of tunnel construction in soil or weathered rock. In modern cities, most of the tunnelling works are carried out in urban areas and highly populated cities, hence, controlling the surface settlement is a crucial factor [6] for designing this type of structure as it can affect the on-surface structure integrity. Several methods can be used to predict surface settlement. In the current industry practice, the common methods for the determination of surface settlement can be categorized into empirical, semi-empirical and numerical, statistical, and Machine Learning (ML). Among them, ML is a new area that can be used for prediction with a high level of prediction capacity. However, currently, the implementation of ML techniques is not common for predicting surface settlement in the industry. ML is an evolving branch of computational algorithms that are designed to emulate human intelligence based on data given by learning from the surrounding environment [7]. In this study, available techniques for estimating the tunnel-induced surface settlement are discussed and compared. The advantages and disadvantages of the available techniques will be further explained. A future approach for assessing surface settlement is also recommended to practitioner engineers.

2 Settlements and Volume Losses Due to Tunnelling

Different tunnel configurations have different impacts on surface settlement. In the early era of tunnelling, mostly single tunnels were constructed. As time passed and with the improvement of technology, twin tunnels can now be seen across many cities in various countries. In this section, the impacts of single and twin tunnels on the settlement will be further described.

2.1 Surface Settlement Induced by Single Tunnels

Peck [8] is one of the early researchers who proposed the well-known empirical formula for surface transverse settlement caused by tunnelling, which is widely used in the industry. The proposed formula is based on the field observation and simplification of Litwiniszyn's [9]. formula. According to the formula, the loss of ground into the tunnel is the major source of surface settlements and that loss of

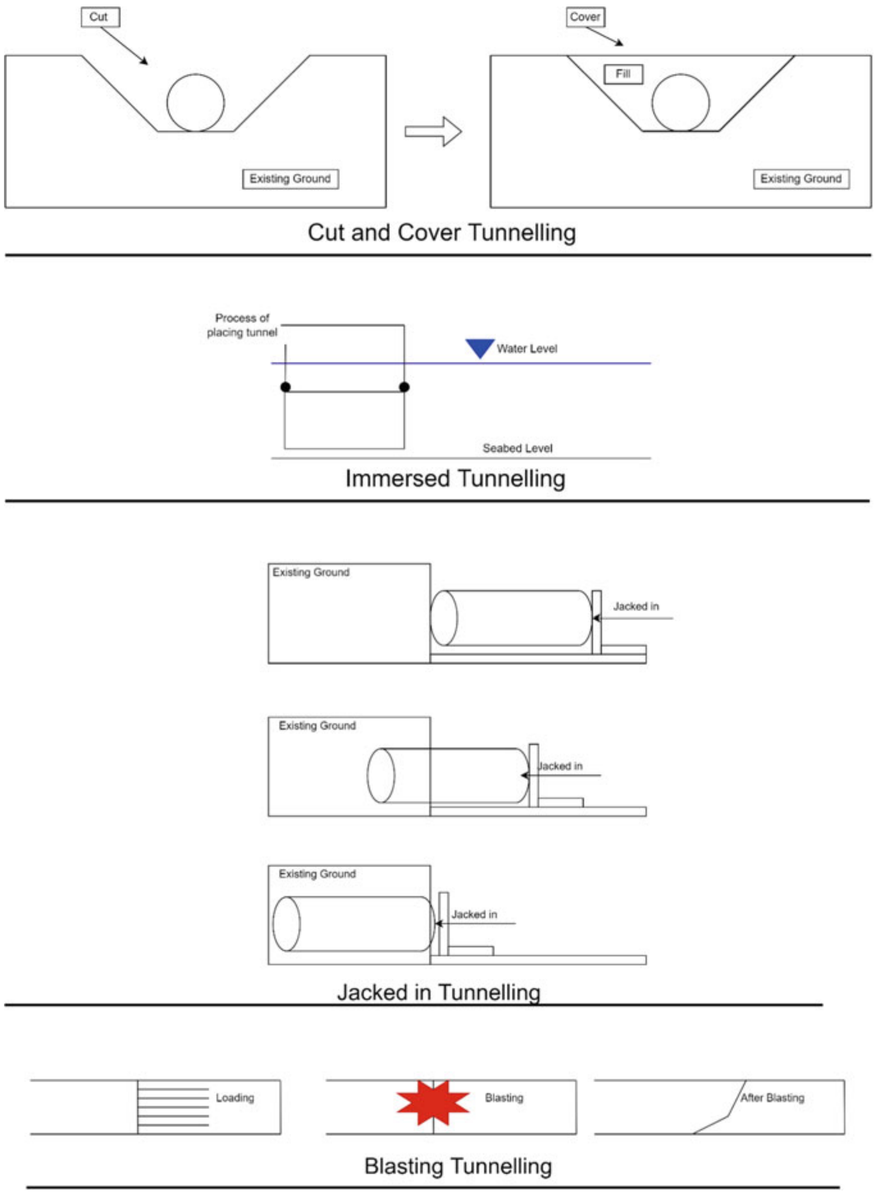


Fig. 1 Illustration of the tunnel construction methods

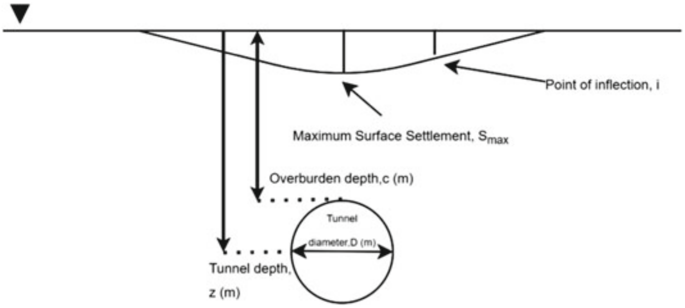


Fig. 2 Transverse settlement due to tunnelling

ground is related to the method of construction, type of soil, groundwater conditions, geometry, and depth of the tunnel. His proposed formula indicated that the pattern of surface settlements caused by ground loss due to tunnels can be approximated by a Gaussian probability curve as illustrated in Fig. 2. The empirical formula is as follows:

$$S = S_{\max} e^{\frac{-x^2}{2i^2}} \tag{1}$$

where S is the surface settlement in the transverse section at distance, x is the distance from the centreline of the tunnel, i is the point of inflection (settlement trough) and the maximum surface settlement can be expressed using this formula:

$$S_{\max} = \frac{V_s^2}{2 \times \pi \times i} \tag{2}$$

where V_s is the volume loss of the soil (m^3/m).

The inflection point, i can be defined as the distance from the tunnel center to the point where the concavity curve changes from positive to negative and it is one of the crucial parameters used for the calculation of the surface settlement.

According to Peck's [8] formula, the assumption of the settlement is based on the volume loss during tunnelling which is equivalent to the volume of the surface settlement trough for clayey soil as illustrated in Fig. 3. Thus, the volume loss, V_s of the excavated area can be expressed in the following formula:

$$V_s = \frac{Volume\ Loss(VL)\ (\%)}{100}\ \frac{\pi D^2}{4} \tag{3}$$

The key inputs in the calculation of the maximum surface settlement are subject to the volume loss and the settlement trough. As result, several studies have been conducted to investigate the input of Vs for various ground conditions and tunnelling methods, as shown in Table 1.

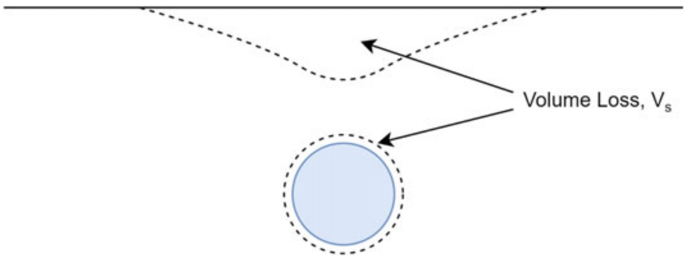


Fig. 3 Volume loss of the tunnelling

Table 1 Volume loss of the soil with various ground conditions

Reference	Ground condition	VL (%)	Method of tunnelling
Attewell and Farmer [10]	London clay	1.44	Hand excavation shield tunnelling
O'Reilly and New [11]	London clay	1.0–1.4	Open face shield-driven tunnels
Kavvadas et al. [12]	Weak rock	0.2	New Austrian tunnelling method (NATM)
Standing et al. [13]	London clay	2.9–3.3	Shield tunnelling
Mair and Taylor [14]	Stiff clay	1.0–2.0	Open face method
	Siff clay	0.5–1.5	NATM
	Sand	0.5	Closed face tunnelling boring machine
	Soft clay	1.0–2.0	Closed face tunnelling boring machine
Wan et al. [15]	London clay	0.8%	Earth pressure balance machine (EPBM)

For the ease of calculation of the surface settlement, Eqs. (2) and (3) can be merged to develop Eq. (4):

$$S_{\max} = 0.313\frac{(VL)D^2}{i} \tag{4}$$

It can be seen that different tunnelling approaches and ground conditions depict various recommendation range values and empirical equations. As shown in Table 1, tunnel geometry, geological formation, and tunnel excavation are some of the factors affecting the surface settlement induced by blasting.

2.2 Surface Settlement Induced by Twin Tunnels

Many metro tunnelling projects are designed and constructed with twin tunnels due to highly densely populated urban areas with numerous buildings on the surface [16, 17]. Several empirical formulas were developed for a single tunnel instead of a twin tunnel. In comparison with a single tunnel, surface settlements induced by twin tunnelling can cause wider susceptible areas and higher settlements. Unlike single tunnels, most twin tunnels in various countries are constructed using tunnelling machines. Similar to the single tunnels, surface settlement induced by the twin tunnels can be caused by the ground condition, tunnel geometry, and tunnelling excavation methods.

Besides, twin tunnels have more factors that can affect the surface settlement due to tunnelling in comparison with single tunnels, such as the distance between two tunnel axes, the period of excavation between the 1st and 2nd excavated tunnels, and variation of the shield operation for the 1st and 2nd excavated tunnels. There are four (4) types of typical twin tunnel configurations, which can be seen in Fig. 4 [18].

For the simplification of the computation of surface settlement induced by twin tunnels, engineers and researchers tend to use the superposition of the individual

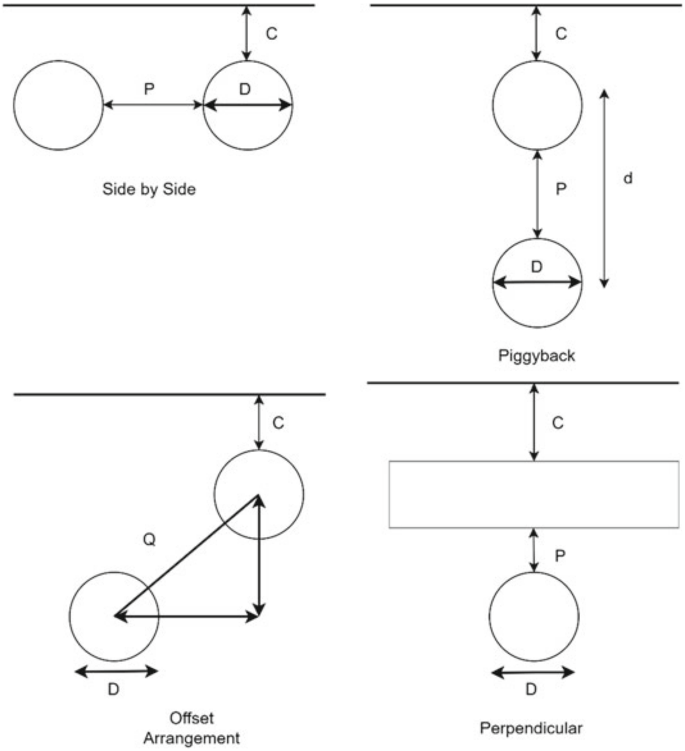


Fig. 4 Typical configuration of the twin tunnels

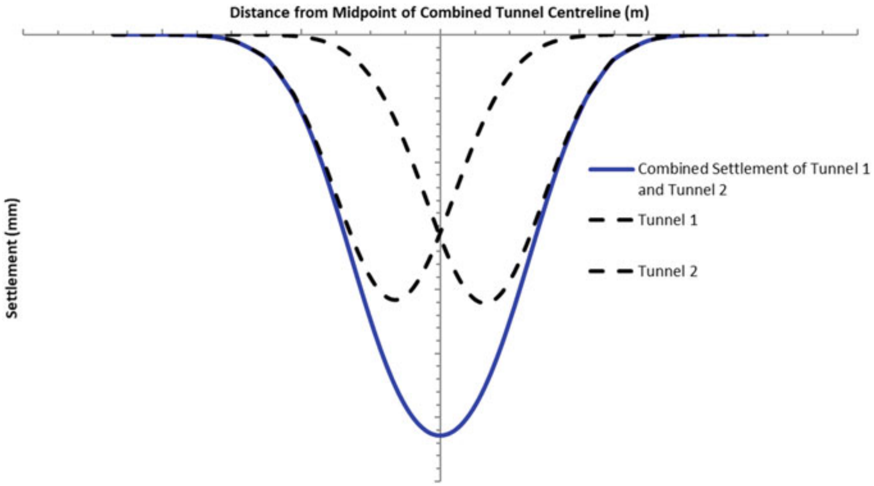


Fig. 5 Superposition of Gaussian settlement distribution

settlement troughs based as shown in Fig. 5. In this figure, the shape of the surface settlement is based on the empirical Gaussian distribution.

$$S_v(x) = S_{\max}\left[e^{-\frac{x_1^2}{2i_x^2}} + e^{-\frac{(x_1-d)^2}{2i_x^2}}\right] \tag{5}$$

where d is the distance between the tunnel center, x_1 is the distance from the centreline of the first tunnel, i_x is the point of inflection (settlement trough) for 1st or 2nd tunnel.

Under these circumstances, the approach does not consider the influence of the existing tunnel on the new tunnel construction movement. The concept of superposition of the surface settlement applies to parallel twin tunnels if the ratio of the distance between tunnel centers to the diameter of the tunnel is larger than 2.7 and it is irrelevant to the ratio of the overburden depth on the tunnel [16, 19].

3 Methods for Predicting Surface Settlement Induced by Tunnelling

In this section, available methodology categories in predicting surface settlement induced by tunnelling will be discussed in consequent sub-sections.

3.1 Empirical Formula

Peck's [8] has been widely used by researchers to estimate surface settlement induced by tunnelling. O'Reilly and New [11] have further modified the empirical formula from Peck [8] based on more field data from the actual tunnelling works. In their approach, several assumptions were made, i.e., the soil movements happened along radial paths that were directed toward the tunnel and the condition was in plain strain with constant volume deformation. This assumption implies that the width of the zone of surface settlement decreases linearly, and the formula is as follows:

$$i = kz \tag{6}$$

where i is the trough width parameter at the depth of the tunnel axis, z, and k are empirical constants of the ground (trough width parameter). It is important to understand that this equation does not apply to very shallow tunnels with a tunnel depth-to-diameter ratio of less than 1. Various ground conditions give different values of k, which can be seen in Table 2.

O'Reilly and New [11] found that most cases show $i = 0.5z$ irrespective of the tunnel in soft clay or stiff clay. This finding has shown similarities with Fujita [19], where the case studies were based on the tunnels constructed in clay and various types of construction methods were utilized, such as hand mine tunnels, blind shields, slurry shields, and tunnelling machines.

Mair and Taylor found that $k = 0.35$ for tunnelling in granular soils [14]. Many researchers have worked on it and proposed empirical formulas for the calculation of i, which are tabulated in Table 3.

Terzaghi [22] is the first study to claim that the second tunnelling-induced settlement of the twin tunnels is always bigger than the first constructed tunnel. Cording [23] showed that the second-driven tunnel has a greater surface settlement in comparison with the first excavated tunnel. The impact of the first constructed tunnel on the subsequent tunnel is due to the changes in the stress state in the soil [24]. To consider such a situation, Addenbrooke et al. [25] and Hunt [26] have further developed a new empirical approach. Addenbrooke et al. [25] proposed a design chart (Fig. 6) that comprises the eccentricity of the maximum surface settlement and the volume loss that is induced by the second constructed tunnel with respect to the pillar width.

To consider the overlapping zone (i.e., the zone that is disturbed by the previously excavated tunnel), another empirical formula was proposed by Hunt [26] for the modified superposition that takes this into consideration (Fig. 7). In the region of the

Table 2 Summary of the settlement trough for various types of soil [11]

Soil type	Tunnelling methods	Empirical constant, k
Siff fissured clays	Shield tunnelling or hand excavation	0.4–0.5
Glacial deposits	Shield tunnelling with free air and compressed air	0.5–0.6
Silty clay	Shield tunnelling with compressed air	0.6–0.7

Table 3 Summary of empirical formulas for estimation of settlement trough width

Reference	Empirical formula	Ground condition	Tunnelling excavation method
O'Reilly and New [11]	$i = kz$ Cohesive soil $i = 0.43z + 1.1$ Granular soil $i = 0.28z - 0.1$	Where trough width parameter, k varies according to the soil type	Shield tunnelling
Mair et al. [20]	$i = z\left[0.175 + 0.325\left(1 - \frac{Z'}{z}\right)\right]$	Clay, Z' is the depth of the calculated settlement trough from the surface settlement	Centrifuge model test
Loganathan and Poulos [21]	$i = R \times 1.15 \times \left(\frac{z}{2R}\right)^{0.9}$	Clay, where z is the tunnel depth and R is the radius of the tunnel	Tunnelling machine

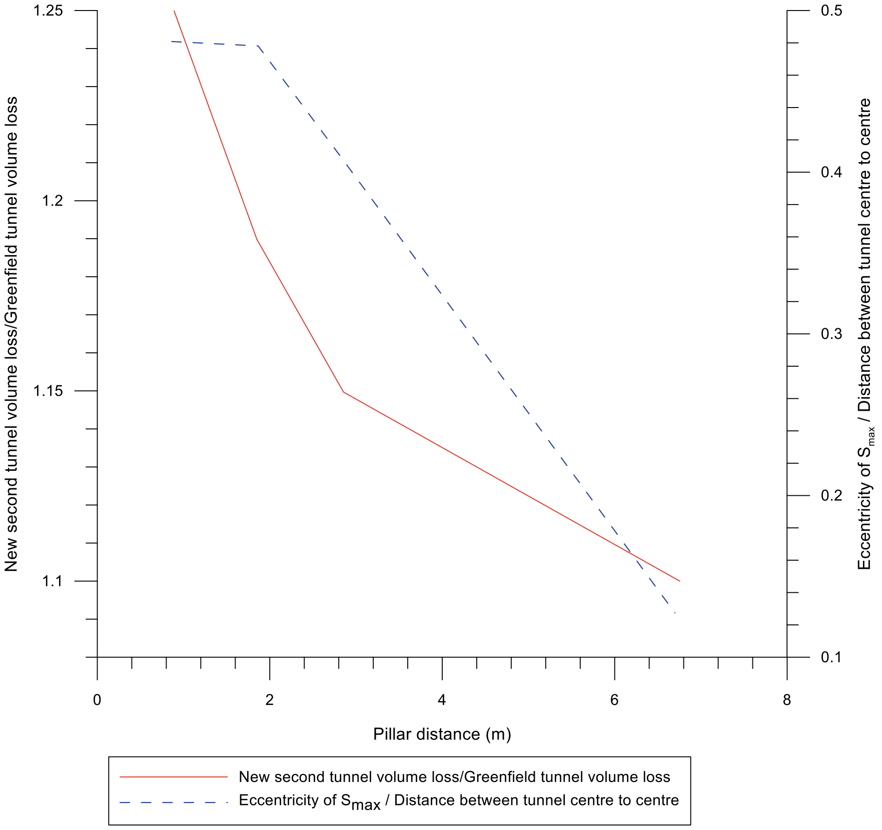
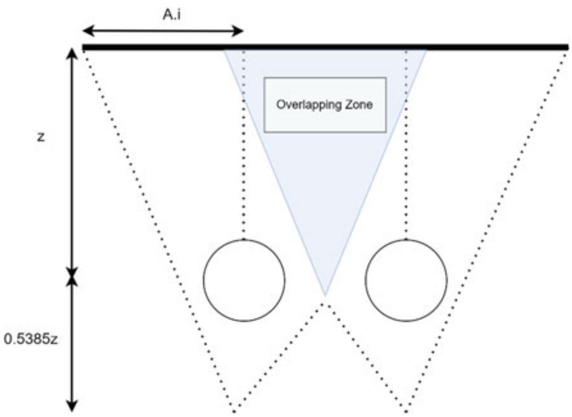


Fig. 6 Design chart to determine the settlement induced by tunnelling

Fig. 7 Definition of the overlapping zone



overlapping zone, the surface settlement above the second tunnel can be described as follows:

$$S_{\mathrm{mod}}(x) = F.S_v(x) \tag{7}$$

$$F = \left[M\left(1 - \frac{|d + x|}{A.i_x} \right) \right] \tag{8}$$

where

A multiple of i_x for full trough width
d distance between tunnel (from center to center)
x distance to the tunnel centerline
M maximum modification factor

The value of maximum modification recommended by Hunt [26] is between 60 and 150% where these values are subjected to change of volume loss for the second tunnel. This empirical formula shows that parameter selection plays a crucial role in ensuring accurate outputs.

Based on all of the proposed empirical methods for the computation of the surface settlement due to tunnelling, it can be observed that the factors affecting the surface settlement can be classified into three (3) categories, namely tunnel geometry, geological conditions, and tunnelling construction approach.

(i) **Tunnel Geometry**
Many studies have shown that the tunnel dimension, the distance between two tunnels, and the construction sequence are the most effective factors in reducing surface settlement caused by tunnelling [27]. The findings also show that the first constructed tunnel position has an impact on the second constructed tunnel.

Some researchers [28] demonstrated that the depth and pillar width of tunnels have a strong impact on the shape of the settlement from the twin tunnels.

(ii) **Geological Condition**

Geological conditions such as type of soil, profiling of the soil, and drained or undrained soil conditions can be considered influential factors in soil settlement.

(iii) **Tunnelling Construction Approach**

The sequence of tunnel construction is showing a significant impact on the surface settlement. Tunnelling operational parameters have an impact on the surface settlement [29]. However, these parameters are not considered in the empirical formulas. Tunnelling machine operational parameters should be considered in studies because many tunnelling projects currently use a mechanized excavation approach to excavate tunnels.

3.2 Numerical Analysis Modeling

Another method to determine the induced settlement due to tunnelling is the use of finite element methods. In the tunnel engineering field, numerical modelling of stresses and displacements is typically used in engineering practice. It is attributed to the fact that tunnel construction is related to the removal of the ground and the installation of a support system in the confined space. The process of ground excavation induces ground stress relief and redistribution around the tunnel circumferences and causes a decrease in radial stress and an increase in tangential stress. As such, these complex interactions between the excavation and construction steps have a strong impact because the distribution of stresses and strain around the tunnel periphery can be captured by numerical analysis. Besides, tunnels are normally excavated in non-homogenous ground conditions, which can be effectively considered in the finite element via initial stress distribution. Practising engineers always prefer to analyze the model using 2D finite elements because of its simplicity and economic approach in comparison to 3D methods, as more parameters are required to ensure the simulation of construction is correct, which is time-consuming. Nevertheless, the 2D numerical approach is a practical tool for carrying out parametric studies and for the analysis and design of a site-specific problem. A varied range of findings can be seen from different studies, as tabulated in Table 4. In addition, Moller [30] made a comparison between 2D and 3D numerical analysis. The finding shows a good comparable result between 2D and 3D analyses.

Although numerical analysis is one of the common methods used for the determination of the surface settlement induced by tunnelling, it is good to note that the finite element method is time-consuming and we are not able to get a reasonable error between the actual surface settlement and the one obtained by these methods.

Table 4 Summary of the findings on the determination of surface settlement induced by tunnelling using finite element methods

Reference	Software	Objectives of the studies	Modeling method	Brief finding
Do et al. [31]	FLAC 2D and 3D	To provide a 2D numerical investigation to highlight the influence of two equivalent approaches which are Convergence-the confinement method (CCM) and the VL method (VLM) on the behavior of a tunnel built in an urban area for surface settlement and structural lining forces with the effect of segment joints taking into consideration	CCM method VL method 3D model	2D settlement trough from CCM matches well with the 3D modelling
Rukdeechuai et al. [32]	Plaxis 2D	Using three (3) soil models namely the Mohr–Coulomb model, hardening soil model, and hardening soil model with small strain stiffness for comparison	Contraction method	Maximum ground surface settlement predicted by different soil models (Mohr–Coulomb model, hardening soil model and hardening soil model with small stiffness)

(continued)

Table 4 (continued)

Reference	Software	Objectives of the studies	Modeling method	Brief finding
Möller and Vermeer [33]	Plaxis 2D	2D finite element analysis for both open face and closed face tunnelling. (Comparison of numerical installation methods)	Grout pressure method	The grout pressure method predicts the shape of the settlement trough perfectly well in comparison with the stress reduction method. To predict the correct maximum surface settlement a very slight reduction of pressure in the grout pressure method is required
Ercelebi et al. [34]	Plaxis 2D	Surface settlement prediction using finite element method, semi-theoretical and analytical method	Contraction method	Plaxis finite element modeling gives the most realistic results in comparison with the semi-theoretical and analytical method provided that the volume loss assumption is correct
Likitlersuang et al. [35]	Plaxis 2D	To present simplified methods for ground settlement computation of tunnelling works using finite element program	Contraction method Stress reduction Modified grout pressure	All three methods show a sensible degree of matching predicted surface settlements. However, these approaches have several limitations which are: (1) Contraction method shows the unrealistic shape of structure forces in the tunnel lining (2) Calculated pore water pressure from the stress reduction method is misread, thus, not suitable for long-term analysis (3) Shield loss component is ignored in the modified grout pressure method 2D finite element modeling can be used to solve 3D problems of tunnelling indued ground surface settlement

(continued)

Table 4 (continued)

Reference	Software	Objectives of the studies	Modeling method	Brief finding
Hieu et al. [36]	Plaxis 2D	Transverse ground movement of nine sections was back analyzed using empirical and finite element analysis with a total of 19.7 km total distance of the underground structure	Contraction Stress reduction method	The width of the ground settlement trough obtained by FEM analysis using contraction method and stress reduction methods is wider than the field data and empirical equation Correlation among volume loss in empirical method, contraction ratio, and stress reduction factor was developed and can be used for estimation of the contraction ratio or stress reduction factor for a given volume loss

3.3 Laboratory Works

Laboratory tests are one of the methods for determining the surface settlement induced by tunnelling. A centrifuge laboratory model is required for the simulation of the actual field tunnelling. The laws of geotechnical centrifuge scaling were described by Taylor [37]. Nomoto et al. [38] performed centrifuge model tests for simulating the tunnelling process in dry sand using a newly developed 100 mm diameter shield tunnel. Three (3) types of tunnelling construction, namely the "buried-tube test". "Tail void test" and "shield test" were carried out to measure the lining stresses, simulate the process of tail void formation and build a full complete process of shield construction, respectively. Based on their laboratory tests, it was found that laboratory results were close to the previous relevant field data. They also stated that this could apply to the actual tunnel design problems. Besides, they have also proposed equations for both transverse and longitudinal settlement. For transverse settlement, their proposed equation for the surface settlement induced by tunnelling can be defined as follows:

$$S_v(x) = S_m.e^{-\alpha.x^2} \tag{9}$$

where S_m is the mean settlement directly above the tunnel, α is the coefficient of the construction in each test which can be expressed as follows:

$$\alpha = \beta.\left(\frac{C}{D}\right)^{-\gamma} \tag{10}$$

where β and γ are the laboratory test constants.

Other than sandy soil material, Hunt [26] carried out the centrifuge tunnelling test in clay with the condition of the heavily over-consolidated sample of clay. However, the tests conducted for this study did not reach any conclusive findings and several factors are the reasons for the difficulty of laboratory modeling in clay, such as sample preparation was very time-consuming and required a lot of labor work, as stated by Hunt [26].

In another study, Divall [39] carried out centrifuge tests for both single and twin tunnels. He used two approaches, which are air and water support for single tunnels. He found that the centrifuge model test of single tunnel results can fit the Gaussian curve with the variables of the point of inflection and the maximum surface settlement. As for twin tunnels (i.e., side by side types), the second constructed tunnel volume loss has shown a consistent increase in comparison with the VL of the first constructed tunnel. Their findings are similar to the findings obtained by Mair and Taylor [14]. Although laboratory tests can be carried out to determine the surface settlement due to tunnelling, this method is time-consuming and costly [40].

Table 5 Summary of the proposed statistical models to predict surface settlement due to tunnelling

Reference	Proposed equation
Moeinossadat et al. [42]	$S_{\max} = \dfrac{117Z}{D} + 0.031c + 0.643\Phi - 0.469E + 0.828V - 2.028F' + 84.699P + 0.085n$
Moghaddasi and Noorian-Bidgoli [43]	$S_{\max} = 1.0236 - 0.1814HR - 0.2338c - 0.8664E$
Anato et al. [41]	$S_{\max} = -1.1 \times 10^{-5} EI_{\text{lining}} - 3.63x10^{-4} E_{\text{grout}} + 3.11 \times 10^{-4} - 3.14 \times 10^{-3} V - 35.136$

c is soil cohesion; Φ is soil friction angle; E is soil elastic modulus; V is penetration rate; F' is thrust force; P is grouting pressure; n is the percentage of grout fill; HR is horizontal to vertical stress ratio; EI_{lining} is the flexural stiffness of tunnel lining; E_{grout} is elastic modulus of grout

3.4 Statistical Models

Several researchers carried out statistical models for the prediction of the surface settlement induced by tunnelling [41–43]. A common technique, i.e., the Multiple Linear Regression (MLR), was used to predict the surface settlement induced by tunnelling. MLR is an approach that can be used to determine the linear relationship of the model between a dependent variable (output) and several independent variables (inputs). This method is similar to the regression line, which can identify the best fit line of the independent variables. This MLR formula can be expressed as follows:

$$y = a_1x_1' + a_2x_2' + \cdots + a_nx_n' + c' \tag{11}$$

where a is the regression coefficient, x' is an independent variable, and c' is constant. Some researchers have proposed several MLR equations for the prediction of surface settlement due to tunnelling, and this formula can be seen in Table 5.

3.5 Machine Learning (ML)

Soft Computing (SC) is a group of techniques that are completely different from conventional computing methods. SC methods can provide an approximate solution to complicated actual life problems [44]. ML is one of the components in the SC group that provides the solution based on the approximation and this method is tolerant to imprecision and uncertainty. In this section, four (4) ML methods namely Genetic Programming (GP), Gene Expression Programming (GEP), Artificial Neural Network (ANN), and Fuzzy logic (FL) will be briefly explained. The advantages and disadvantages of these techniques can be seen in Table 6.

Table 6 Advantages and disadvantages of the main ML methods [45]

Type of MLs	Advantages	Disadvantages
ANN	(i) The ability to learn and adapt (ii) Capability of fault tolerance (iii) Trained model based on historical data	(i) Black box (ii) Applicable for quantitative information (iii) No reason capability
FL	(i) Using rules to represent the knowledge (ii) Capability of fault tolerance (iii) Reasoning capability	(i) No learning capability (ii) Able to handle quality information only
GEP and GP	(i) Systematic random search (ii) Able to supply multiple solutions	(i) The rate of convergence is slow and close to the optimal solution

3.5.1 Genetic Programming (GP)

Genetic Algorithm (GA) is an algorithm based on the concept of genetic and natural, which are genes or chromosomes, to solve search and optimization problems. GA was developed by Holland John [46]. Cramer [47] founded GP with the tree patterns to evolve the programs. Subsequently, GP was developed as part of the extension of GA by Koza [48]. Most of the operators from GA can be implemented in GP. The main difference between GA and GP is that GA is used as a parameter optimization tool to produce the best value with a given set of model parameters, whereas GP creates a structure that represents a dataset with input variables and corresponds to the output. The GP is also known as the tree-based GP, where the algorithm works in a tree approach. Figure 8 illustrates the components of GP for the analysis.

GP uses four steps to solve problems. First, it generates the initial population comprised of two important elements, which are functions (operator) and terminals (variables and constants). The functions are identified from the set of operators, whereas the terminal is confirmed based on the variables and constant sets. The function set includes mathematical operators such as “log” and “-”. After the random population is created, each individual in the population is evaluated for fitness based on how well the problems can be solved. At this stage, GP is carrying out fitness

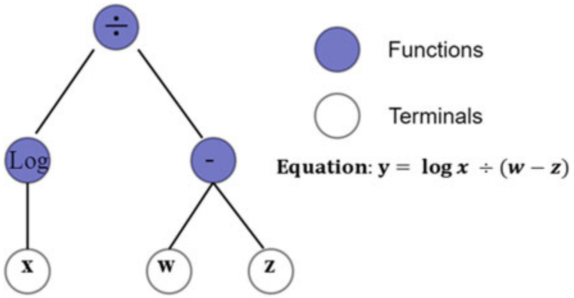


Fig. 8 Illustration of GP as tree representation

proportionate selection, also known as roulette wheel selection, to select the useful recombination solution.

At the last stage, new populations are created based on two methods: cross-over and mutation. In this cross-over stage, two (2) trees are selected randomly from the population and each node from each tree is selected randomly where the sub-trees exchange for the generation of two offspring. As for the mutation stage, a random change to the structure in the population is introduced, and it works by randomly removing the subtree and replacing it with the tree structure. Finally, the process is ceased when no improvement in the solution can be seen or an acceptable solution is achieved.

3.5.2 Gene Expression Programming (GEP)

GEP was founded by Ferreira [49] based on the concept of GA and GP. The core difference between GA, GP, and GEP is as follows: GA is the individuals with linear strings of fixed length, also known as chromosomes; GP is a non-linear structure with different sizes and shapes of parse trees; and finally, GEP is the individuals with linear stings of fixed length and can subsequently be formed as non-linear entities of different sizes and shapes. Several researchers [50, 51] found that GEP can overcome the shortcomings of the GA and GP models. Similar to the GA and GP, the GEP model commences with the random generation of chromosomes in the population. The symbolism of these chromosomes is expressed as trees with various shapes and sizes (expression trees) [52]. Next, the fitness of each is evaluated by a fitness function. Then, the best chromosomes of the first generation are selected via the Roulette method and will be copied for the next generation. These new copied generations are subjected to similar developmental processes, which are genome expression, selection of the environment, and modified reproduction [49]. The process is repeated (for a certain number of generations) until the desired results are achieved.

GEP has two main advantages where the genetic diversity is simplified as genetic operators at the chromosome level and the evolution is more complex programs that comprise several subprograms [53]. Figure 9 illustrates the example of the GEP expression tree and chromosome of a gene.

3.5.3 Artificial Neural Network (ANN)

ANN is one of the ML methods that follow the mechanism of the human brain's biological nervous system for the information-transfer process. The concept of this algorithm was founded by Rosenblatt [54]. ANN can solve the complex and non-linear inputs (variables) and outputs (predictor) [55, 56]. There are many types of ANNs, however, multilayer feed-forward ANN is the most widely used [57, 58]. This ANN type consists of several layers which are input, hidden, and output layers where they are connected through some hidden nodes via different connection weights [59]. The activation function of every node act to transmit the signals to other nodes or

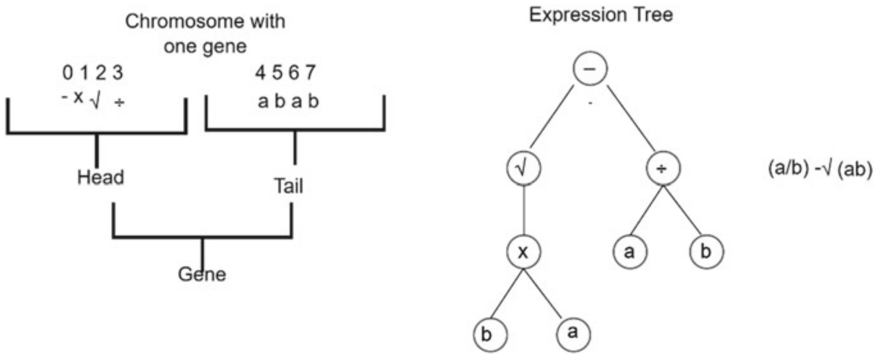


Fig. 9 GEP expression tree and chromosome with one gene

the output of the network. This activation function for every node is applied to the net input of the node. ANN is required to be trained using other learning algorithms to achieve the desired outcome. The Back Propagation (BP) algorithm is the most common method for the training of an ANN [60]. BP comprises two passes, which are the forward pass and the back pass. It uses the transfer function where the outputs are computed and the errors of the outputs are identified. If the error is high (the error can be evaluated based on the Root Mean Square Error, RMSE), then the error is returned to the network and updates the individual weights. This is known as a backward pass. The process of moving forward and backward is repeated many times until reaching a negligible error (often assessed by RMSE). Figure 10 shows an ANN model with the forward and backward propagation process.

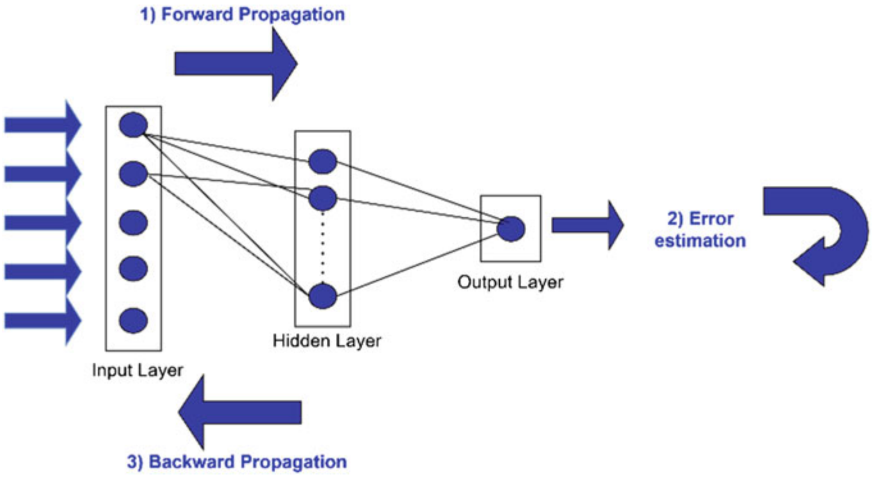


Fig. 10 Illustration of the ANN with the forward and backward propagation

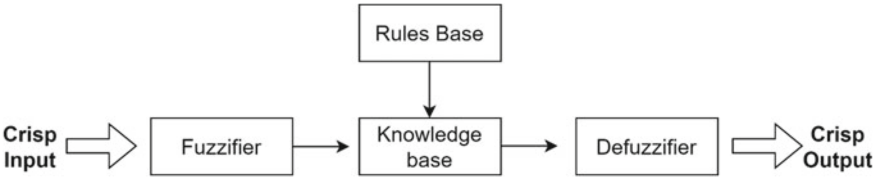


Fig. 11 The structure of the fuzzy logic system

3.5.4 Fuzzy Logic (FL)

Zadeh [61] initiated FL to identify the vagueness of the information by using a computation approach based on the degree of truth. In general, it is formalized as a mathematical approach to describe the ambiguity of information. The FL mechanism follows the way of making decisions when uncertainty is encountered. It also provides an effective way to illustrate complicated data. An outline diagram illustrating the FL is shown in Fig. 11. In general, FL comprises four elements, which are fuzzifier, knowledge base, fuzzy inference system, and defuzzified. The fuzzifier acts as a role to convert the crisp input to linguistic information. The database defines the membership functions that will be used in the fuzzy rules. Each input variable possesses a probability distribution function, which is known as the membership function. As for the rule base, it consists of several fuzzy if–then rules, which are also known as fuzzy rules. In short, the process commences with the input being fuzzy by the fuzzifier, and this will be sent to the knowledge base. In the knowledge base, the rule base is applied and provides the user with an answer via defuzzying. Nevertheless, Maier et al. [62] stated some issues that need to be overcome when the FL approach is used, such as the configuration of distributions of membership function, identification of composition operator, and setting the appropriate fuzzy rules in the system.

4 ML Application in Settlement Induced by Tunnelling

The geotechnical engineering field is related to geological materials such as soil and rock where these materials exhibit various properties due to the imprecise physical processes associated with the formation [63, 64]. To solve the complexity of geotechnical behavior and the spatial variability of these materials, traditional forms of engineering design models are simplified [65]. The inherent complexity of geotechnical materials has led to the replacement of tedious theoretical solutions with ML approaches to solve various geotechnical design problems and assessment issues [66]. ML allows computers to learn patterns from existing data, either from field instrumentation or case histories, without being explicitly programmed [67]. Because geotechnical problems are characterized by great uncertainties and involve various factors which cannot be directly determined by engineers, the rapid growth in popularity of ML methods [68, 69]. The concept of machine learning is based on various fields such as artificial intelligence, computer science, and mathematics [70].

There are many types of machine learning algorithms, such as ANN algorithms, Clustering algorithms, Decision Tree algorithms, Ensemble algorithms, and Regression algorithms. ML techniques began in the seventeenth century, and Pascal and Leibniz [71] used this method to create a machine that could add and subtract like a human. As time passed, Samuel [72] from International Business Machines (IBM) coined the term "machine learning" and managed to prove that computers could be programmed to learn.

ML techniques can be categorized into three (3) types, which are supervised, unsupervised, and semi-supervised. Supervised learning is the method of identification for unknown input and output data that is based on known input and output data with an already identified output. In supervised learning, two (2) types of analysis can be carried out, which are classification and regression [73].

As years have passed, tunnel excavation has improved with various types of machines to suit different ground conditions. To achieve ground movement control, these machines comprise several operational parameters such as face pressure, penetration rate, pitching angle, and grouting quality. Other factors, including tunnelling geometry and ground conditions, have caused some complexities in machine-ground interactions. Thus, many researchers have carried out the prediction of ground settlement by using different machine learning methods with various inputs to predict the induced surface settlement due to tunnelling.

Suwansawat and Einstein [29]employed neural networks to predict the settlement by using 10 input parameters, which are depth (m), distance from shaft (m), soil geology at the crown, soil geology at invert, invert to water table (m), average face pressure (kPa), average penetrate (mm/min), pitching (°), tail void grouting pressure (bar), and percentage of tail void grout filling (%). In their analysis, they used three scenarios of the data to predict the settlements, and the performance of the model is measured using RMSE as listed in Table 7. From Table 7, it can be seen that the RMSE for both training and testing data is less than 10 which indicates the prediction can be done with good accuracy.

Other than the research carried out by them, Pourtaghi and Lotfollahi-Yaghin [74] have proposed a new method based on wavenet transform theory and neural network basic concept called a wavenet for predicting the maximum surface settlement caused by tunnelling. This new method has a better performance compared to the ANN method with the testing data (correlation coefficient, R of 0.9562 compared to the R-value for testing data (0.898) in the developed ANN model. This finding proves that other machine learning methods can be used for the prediction of tunnelling-induced surface settlement.

Boubou et al. [75] have carried out additional steps for the ANN method by carrying out the parameters' elimination procedure. From their study, the elimination of some parameters can improve the RMSE value, which can help the prediction performance. However, in their study, the consideration of parameters is divided into two categories which are tunnelling operating parameters and geological parameters.

Table 7 Summary of the ANN performance models in different scenarios

Scenario	Type of data	Training RMSE (mm)	Testing RMSE (mm)
Using data from each tunnel section for predicting surface settlements within the same section	Section A	5.08	7.33
	Section B	2.55	6.22
	Section C	2.31	5.90
	Section D	5.98	7.56
Using initial data for predicting surface settlements	Trained with the first 50% of the data to predict surface settlements in the remaining 50% of section A	2.55	7.24
	Trained with the first 50% of the data to predict surface settlements in the remaining 50% of section B	2.27	7.93
	Trained with the first 50% of the data to predict surface settlements in the remaining 50% of section C	2.59	7.43
	Trained with the first 50% of the data to predict surface settlements in the remaining 50% of section D	5.01	5.24
Using all data for predicting surface settlements of all sections	All data	8.26	8.39

The geometry of the tunnelling is not included in the study where the geometry, such as tunnel depth, could be considered an important factor that affects the tunnelling-induced surface settlement.

Ahangari et al. [76] have carried out the prediction of tunnelling-induced settlement using two techniques, which are the Adaptive Neuro-Fuzzy Inference System (ANFIS) and GEP. However, in their analysis, tunnel operating parameters are not taken into consideration. The input parameters including (cohesion (kPa), friction angle (°), elastic modulus (MPa), tunnel depth (m), and tunnel diameter (m) were used by them. The data was collected from previous research data from 53 tunnels all over the world and the value of the settlement was obtained from numerical modeling (FLAC^{2D} software). Although the final results indicate that GEP has a better prediction than ANFIS, the model can hardly be applied in the real tunnelling project because the data for the settlement is extracted from the numerical analysis instead of the real tunnelling project.

Hasanipanah et al. [77] have carried out the prediction of surface settlement caused by tunnelling using a hybrid model, which is a Particle Swarm Optimisation (PSO) -ANN model. Their developed model was based on the actual data sets obtained from the actual tunnel project. In their analysis, only three parameters are taken into consideration, which is horizontal to vertical stress, cohesion (kPa), and Young's modulus (MPa). The results indicate that the hybrid model has a better result with an R2 of 0.9682 compared to the only ANN model with an R2 of 0.9403. ANN suffers from two disadvantages, such as getting trapped at local minima and a slow rate of learning [78]. As such, several researchers have proven that these disadvantages can be improved by using PSO to further optimize the ANN [79, 80]. PSO is the algorithm that was developed as a bird swarm simulation with the ability to learn from the previous experiences of the swarm and the capability to move towards the optimum goal [81]. This algorithm works with each individual who makes decisions based on the best results of their personal experiences and the experiences of the swarm, achieving the best results for the entire population. At the commencement of the analysis, particles are randomly distributed in the search space in a random pattern. For every particle, it acts as a feasible solution. PSO can be classified into three (3) parts, which are current position, best position, and velocity. The current position represents the current coordinate of the particle, whereas the best position is the best coordinate. Velocity particles are the velocity of particle vectors in D-dimensional space. The main objective of PSO is to determine a termination criterion for terminating the iterative search process. Three (3) termination criteria are frequently used in PSO, which are [82]:

(i) The maximum number of iterations is exceeded
(ii) Solutions are found according to the condition of each problem
(iii) No improvement is achieved after many numbers of iterations.

These criteria are applied to ensure that PSO can converge on a feasible solution. The ANN has the weakness of getting tapped in the local minima. On the other hand, the complementary PSO algorithm plays an important role in determining the global minima. The process of PSO begins with the initialisation of a group of random particles that represent the ANN weights and biases with random assignment. Subsequently, based on initial weights and biases, the hybrid PSO-ANN is trained and the errors between actual and predicted values are calculated. The calculated error is decreased for each iteration by altering the particle position. To update the velocity equation, the best individual and the best group are used. Hence, the value is produced by adjusting the particle positions to the best solution. Next, the updated position with the new error is retrieved. These stages are repeated until the termination criteria are met with the optimisation output is obtained. Nevertheless, the model does not take into consideration tunnel geometry and tunnel operating parameters, where these two categories could have more impact on the surface settlement caused by tunnelling.

Moeinossadat et al. [42] employed two approaches (ANFIS) and multiple regression (MR) for the calculation of the maximum surface settlement induced by EPB shield tunnelling. From their findings, the result shows that ANFIS is more accurate compared to MR. The MR method had poor accuracy because surface settlement depends on many parameters, which have caused interaction among different parameters. Besides, their finding shows that the tunnelling operating factor has the largest effect on the settlement, followed by tunnel geometric and soil strength factors. ANFIS is a hybrid model comprised of Fuzzy Systems (FS) and Neural Networks, and this method was developed by Jang [83]. This technique can also be used in other civil engineering fields [84, 85]. Fuzzy logic is a computing framework that consists of three frameworks, which are fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. ANFIS possesses the ability to map the relationship between input and output data into several constraint sets. This neuro-fuzzy system corresponds to the fuzzy model Takagi-Sugeno, and the weights of the ANN model are the same as the parameters of the fuzzy system [86]. To determine a set of parameters, ANFIS employs a hybrid learning method that combines gradient descent, backpropagation, and a least squares algorithm [87]. This inference system consists of five layers which are the fuzzy layer, product layer, normalized layer, de-fuzzy layer, and total output layer. The layers of each description are as follows:

(a) Layer 1 (Fuzzy layer)—Each node in this layer can be considered an adaptive node.
(b) Layer 2 (Product layer)—Every node output depicts the firing strength of a rule
(c) Layer 3 (Normalized Layer)—Each node in this layer is a fixed neuron and represents the normalized firing strength of each rule.
(d) Layer 4 (de-fuzzy layer)—This layer comprises adaptive neurons which contain consequence parameters.
(e) Layer 5 (Combining Layer)—Comprises a single neuron with summation of all inputs.

Bouayad and Emeriault [88] are utilizing principal component analysis (PCA) to describe the interrelation pattern between the tunnelling boring machine (TBM) parameters and geological profiles. PCA can be known as the proper orthogonal decomposition (POD), which is one of the multivariate statistical methods frequently used in data analysis [89]. The method of PCA is based on the determination of the variances and coefficients of a dataset by finding the eigenvalues and eigenvectors. The covariance matrix is used to measure how much the dimensions vary from the mean value of other parameters. The PCA technique is also used in other tunnelling research studies to reduce the number of irrelevant variables [90, 91]. The main purpose of using PCA is to reduce a large number of interrelated variables and at the same time to retain as much variation as possible in the original data set. In their study, fifteen parameters were reduced to six parameters. With this approach, six (6) variables were used in the Adaptive Neuro-Fuzzy Inference System-Principal Component (ANFIS-PC) for analysis and validation. After carrying out the reduction of the parameter, the results indicated a high correlation between the predicted and measured settlements. Nevertheless, it is good to know that Prasad and Bruce [92]

Table 8 Comparison of the six algorithms for surface settlement prediction due to tunnelling

Technique	Training set, R	Testing set, R
BPNN	0.78	0.46
GRNN	0.83	0.18
ELM	0.53	0.14
SVM	0.74	0.28
RF	1.00	0.70

have highlighted that PCA is not recommended for small sample size problems and that it applies to the linear model.

Zhang et al. [93] have compared five machine learning methods, namely back propagation neural network (BPNN), general regression neural network (GRNN), extreme learning machine (ELM), support vector machine (SVM), and random forest (RF). ELM is part of the neural network which is modified with a single hidden layer feedforward neural network with only one hidden layer. An SVM is built using statistical theory to determine the best hyperplane in N-dimensional space. whereas RF is the average of the decision trees with a large modification of bagging. SVM and RF are both ML techniques. From the researchers' findings, the results show that the RF algorithm outperforms the other machine learning methods for the prediction of tunnelling-induced settlement, and the summary of the training and testing set performance measured using the coefficient of determination, R is shown in Table 8.

All the prediction work on the surface settlement, as stated earlier, is summarized in Table 9. Other than using an ML model for the prediction of surface settlement, some researchers utilize statistical approaches and/or observational methods to identify the importance of the parameters. These methods are summarized in Table 10. It can be seen that most researchers used ML approaches for single tunnelling.

5 Discussion

In summary, it can be stated that there are four (4) typical methods that can be used for the prediction of tunnelling-induced surface settlement. Although empirical formulas are one of the most widely used methods by many practising engineers because of their ease of computation, these empirical approaches are based on simplified assumptions. In empirical equations, all influential factors, such as the tunnelling operating parameters, are not considered. It is worthwhile to know the extent of the numerical analysis model and computation time is limited with model boundaries in contexts where this can be done by redoing the model boundaries by taking the boundaries to be further away from the modeling object and comparing the result. However, this process can be more time-consuming [100]. Furthermore, to simulate the current field condition, several in-situ and laboratory tests are required to retrieve the reflective parameters for the simulation in the numerical analysis.

Table 9 Summary of the AI model for the prediction of the surface settlement due to tunnelling

Reference	Input	No of variables	Analysis model	No data
Suwansawat and Einstein [29]	Tunnel depth, distance from the shaft, soil geology at the crown, soil geology at invert, invert to the water table, average face pressure, average penetrates, pitching, tail void grouting pressure, and percentage of tail void grout filling	10	ANN	49
Pourtaghi and Lotfollahi-Yaghin [74]	Tunnel depth, tunnel distance from shaft, geology at the crown, geology at invert, tunnel invert to water level, average face pressure, average pitching, tail void grouting pressure, and percentage of tail void grout filling	9	Wavenet transform theory and neuro network	49
Boubou et al. [75]	Time required for excavation and installation of one tunnel lining ring, hydraulic pressure used for the cutting wheel, horizontal and vertical guidance parameters (the ability of TBM) to follow the theoretical path), TBM advance rate, confining pressure at the tunnel face, volume of tail void grout filling, total jack thrust, and soil geological profile	8	ANN	95
Ahangari et al. [76]	Cohesion, friction angle, elastic modulus, tunnel depth, and tunnel diameter	5	(1) ANFIS (2) Fuzzy C means clustering (FCM) (3) GEP	53

(continued)

Table 9 (continued)

Reference	Input	No of variables	Analysis model	No data
Moeinossadat et al. [42]	Tunnel depth, tunnel diameter, soil cohesion, soil friction angle, soil modulus of elasticity, penetration rate, thrust force, grouting pressure, fill factor of grouting	9	(1) MLR (2) ANFIS	53
Bouayad and Emeriault [88]	TBM advanced rate, face pressure, tail void grouting pressure, volume of injected tail void grout, pressure of the cutting wheel representing the pressure required to rotate the cutter wheel, total jack thrust, time required for the excavation and installation of one tunnel lining ring, horizontal and vertical guidance parameters, total work representing the energy required for the excavation of 1 m^3 of soil, soil type, soil unit weight, soil cohesion, and soil friction angle	15	PCA with ANFIS	95
Zhang et al. [93]	Tunnel depth below the water level, overburden depth, thrust force, torque, volume of grout filing, penetration rate, chamber pressure, blow counts of the modified standard penetration test, modified dynamic penetration test of soil layers, modified uniaxial compressive strength of weathered rocks ground condition and stoppage	12	(1) BPNN (2) GRNN (3) ELM (4) SVM (5) RF	236

Table 10 Summary of the most impacting parameters for the tunnelling-induced surface settlement

No	Reference	Technique to identify the most affecting parameter that induced surface settlement due to tunnelling	Most affecting parameters
1	Kim et al. [94]	Relative strength of effects (RSE)	(1) Tunnel depth (2) Groundwater inflow rate (3) Rock mass type (4) Tunnel type (5) Velocity of tunnel excavation
2	Kobayashi et al. [95]	Field observation	(1) Passage of shield (2) Tail void closure
3	Hidayat [96]	ANN with Garson's [97] method	(1) Bulk density (2) Earth pressure (3) Advance rate (4) Stiffness (5) Moisture content
5	Santos and Celestino [98]	Sensitivity analysis	(1) Overburden tunnel (2) Tunnel depth below the water table (3) Advanced rate before and after
4	Ocak and Seker [99]	Field observation	(1) Face pressure (2) Penetration rate (3) Amount of excavated material per ring (4) Percent of tail void grout filling (%) for the first and second tunnel
6	Hasanipanah et al. [77]	Cosine amplitude method	(1) Horizontal to vertical stress ratio

On the other hand, the centrifuge model allows actual tunnelling fieldwork to be simulated in laboratory work. However, there are several limitations in the centrifuge model test, which are variations in stress level, radial acceleration field, scaling effects, and boundary effects. The law of scaling of soil models for centrifuge model tests with the acceleration is not constant with depth, and in the centrifuge model, the acceleration field is radial instead of parallel with depth, Taylor [37]. Besides, the similar prototype soils used in the centrifuge model will cause compatibility issues when applying the scaling law to grain sizes as stated by Taylor [37]. Other than that, the side wall in the model has an impact on the test if the model is narrow [39]. Although the traditional statistical model can be used for the prediction of surface settlement, this model is limited to the linear model.

With the presence of ML models, the prediction of surface settlement induced by tunnelling can be carried out at any time with three main input parameters, which are tunnelling geometry, engineering parameters of the ground, and tunnelling operation parameters. This scenario can assist engineers to make quick decisions and understand the surface settlement without any required calculations. Furthermore, the ML model can depict high accuracy in the prediction of settlement. This is indirectly able to help engineers adjust the tunnelling operation parameters during the process of tunnelling. As such, it will significantly reduce the risk of damaging the existing surrounding buildings. Other than the advantages of the ML model, there are also several disadvantages to the ML model.

One of the difficulties of the ML model is establishing a reliable ML model. To develop a good model, a relatively large and reliable set of data is required. The dataset includes relevant input parameters (engineering properties of the ground and tunnelling operation parameters) and measured surface settlement. Therefore, to establish a large database for an ML model, several tunnelling construction projects at various locations are required. Hence, it is highly recommended to collect all the recorded measurements of the settlement as much as possible. Besides, the in-situ and laboratory test results that were carried out at the site with the tunnelling operation parameters shall be kept properly.

Some classical heuristic optimisation methods are not able to provide a good prediction. A hybrid metaheuristic algorithm such as PSO can be considered to improve the prediction of the ML model. For the actual tunnelling construction work, all the design constraints, economic value, and practicality must be taken into consideration. To consider all the aforementioned, an ML model can be considered to be implemented as a future way of method calculation and decision-making. To utilize and develop the ML model for practical purposes, the process of developing this ML model is recommended and illustrated in the flowchart as shown in Fig. 12.

This process can be classified into five (5) main stages. Stage 1 is the data collection of the field record of ground settlement markers, and subsurface investigation including laboratory work data and tunnelling operation parameters for various projects at different locations. Upon receiving all the information, data cleaning and interpretation can be carried out. The first stage is the most time-consuming because the data is not always ready in soft copy format and is dispersed. Next is the development of ML models with training and testing based on all the available data. Stage 3 deals with the assessment of ML models. Two possible scenarios are expected during the assessment of the ML models. If the model performance is low, utilization of optimisation techniques is recommended for improving the ML models. In other scenarios, the good performance of the ML model can proceed to the next stage, which is using the ML model on different projects with various conditions. At this stage 4, if the prediction of the ML model is not good, the user is required to check the input data. After checking the input data, if the prediction remains, the model has to return to Stage 2 and add new data collection to the ML models and restart the whole process. One may suggest the workability of the ML model if the ML model works well enough for many projects at different locations.

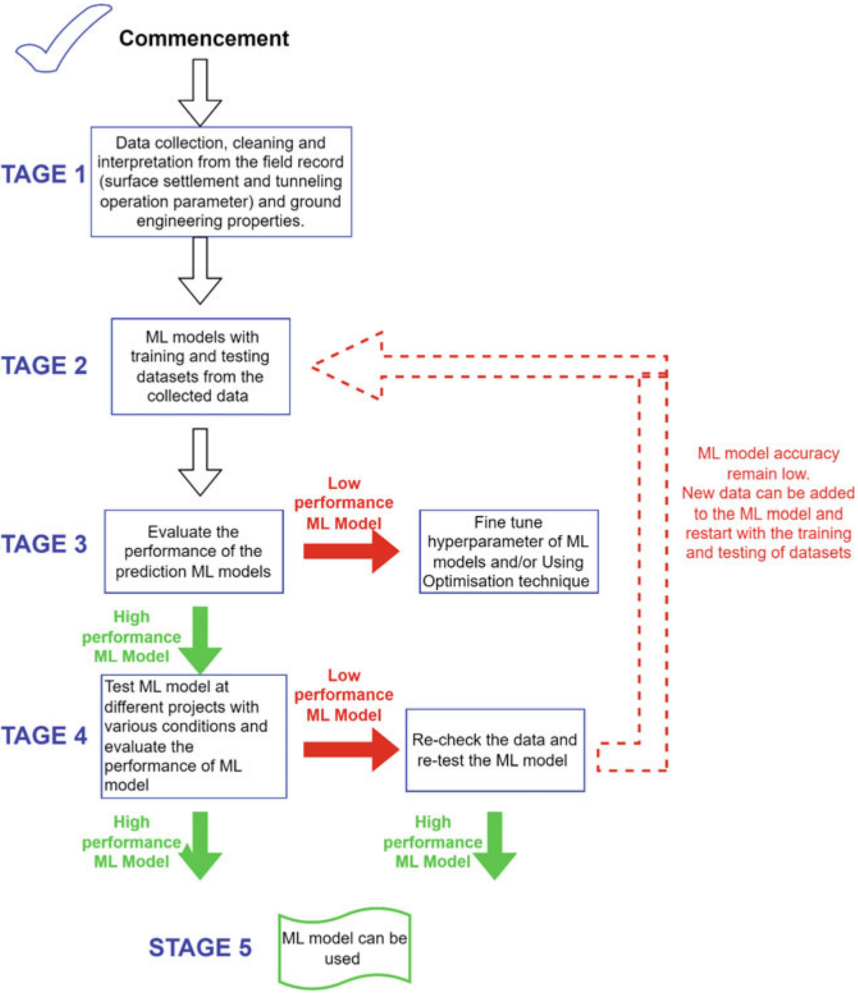


Fig. 12 Flowchart for the process or steps of developing the ML model for the pile capacity

6 Future Perspective

Applications of ML in the geotechnical engineering field have drawn considerable attention in the last decade, which indicates ML methods can be considered to solve complicated geotechnical problems. The ML approach shows concrete proof of better prediction performance compared to the typical statistical calculation. Even though many types of ML methods have been studied in the geotechnical field, this method is rarely used in the actual tunnelling industry. The possible reason is attributed to the lack of good quality data, which is required for establishing a good ML model.

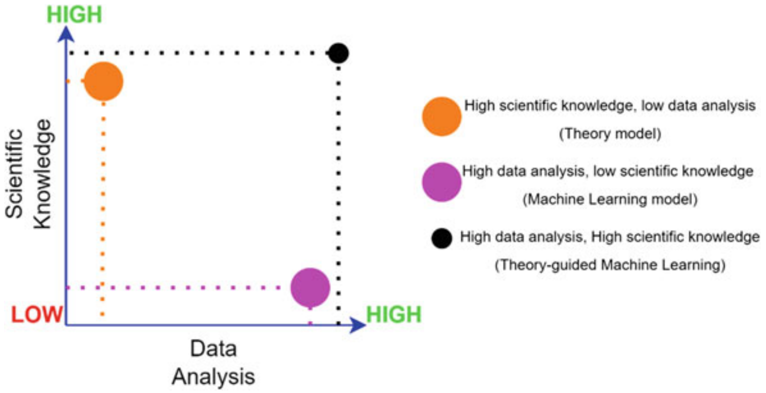


Fig. 13 Illustration of the TGML in the graph format

Besides, practising engineers are not familiar with ML models well enough. Therefore, it is good to further explore this method for its future application in the practical tunnelling field. ML model can be updated from time to time based on the availability of new data to achieve better performance prediction by inserting more training data. Based on the extensive review in this chapter, it can be concluded that ML models have good performance and can be considered a feasible tool for solving tunnelling problems. Data science models comprise high data usage but with limited scientific knowledge. A theory-based model is vice versa of a data science model. The difference between the theory-based model and the data science model can be seen in Fig. 13. Therefore, to further improve AI models, Theory-Guided Machine Learning (TGML) is necessary to provide better predictions compared to the other methods. To improve the usefulness of ML models for scientific discovery, TGML is a new paradigm that uses lots of scientific data. It is the main objective of TGML to ensure that generalizable model learning requires scientific consistency. Based on well-known theories and empirical equations, a comprehensive database that can handle a wide range of effective parameters can be proposed. It will then be used to build a model which can be generalized with a high degree of accuracy. To the best of the authors' knowledge, TGML techniques have not been applied in geotechnical engineering, despite their widespread application in other fields of civil engineering, such as water and hydrology [101, 102].

7 Conclusion

Single tunnels and twin tunnels have different impacts on the surface settlement. The twin tunnels have a larger impact on the surface settlement. Different tunnelling construction methods have a different impact on the surface settlement induced by tunnelling. There is more than one technique that can be used for the prediction of the surface settlement induced by tunnelling. The empirical formulas can be easily used to predict the surface settlement induced by tunnelling. However, these methods have several limitations due to the assumptions made to develop the empirical formulas. In addition, many numerical solutions have been proposed for the prediction of settlement induced by tunnelling using theoretical soil constitutive models. Although the simulation of the numerical models can reflect the closest scenario of the actual tunnelling work, these methods are time-consuming to produce the result for the practical engineer under different ground conditions. Centrifuge modeling can be considered as another category of techniques to stimulate the actual tunnelling work on a small scale, which could lead to a better understanding of the tunnelling work. However, setting up laboratory work for centrifuge modeling for every tunnelling project is high in cost and time-consuming. Other than that, there are many limitations in the scaling model to reflect the actual tunnelling work. The statistical model of MLR can also be used for the prediction of surface settlement. Nevertheless, the statistical model is limited to the certain nature of the project. Furthermore, the soil or rock is a complex material with a wide range of properties, which are not able to reflect linear properties.

Among all the available techniques, ML has the most suitable prediction level for estimating surface settlement caused by tunnelling. This study provided a review of the various approaches used by researchers to predict surface settlement, and the literature review revealed that ML has remarkable accuracy. Nevertheless, the implementation of ML models highly depends on the quality of the input data as these models are data-driven. Therefore, good quality sets of data are required for the successful development of ML models. It is recommended to study the theoretical aspects of the tunnel-induced settlement before using the ML models. The authors suggest that ML methods can be utilized as a complement to conventional computing techniques rather than an alternative to their design. Lastly, in future ML models, the combination of the ML and the theoretical model is strongly recommended to consider the scientific theory in the model.

References

1. Tatiya R (2017) Tunnelling by conventional methods. In: civil excavations and tunnelling: a practical guide. ICE Publishing, pp 137–189
2. King EH, Kuesel TR (1996) An introduction to tunnel engineering. In: Tunnel engineering handbook, pp 1–3. https://doi.org/10.1007/978-1-4613-0449-4_1
3. Abdallah M, Marzouk M (2013) Planning of tunneling projects using computer simulation and fuzzy decision making. J Civ Eng Manag 19:591–607
4. Majedi MR, Afrazi M, Fakhimi A (2021) A micromechanical model for simulation of rock failure under high strain rate loading. Int J Civ Eng 19:501–515. https://doi.org/10.1007/s40999-020-00551-2
5. Afrazi M, Lin Q, Fakhimi A (2022) Physical and numerical evaluation of mode II fracture of quasi-brittle materials. Int J Civ Eng 20:993–1007. https://doi.org/10.1007/s40999-022-00718-z

6. Chakeri H, Ozcelik Y, Unver B (2013) Effects of important factors on surface settlement prediction for metro tunnel excavated by EPB. Tunn Undergr Space Technol 36:14–23
7. Naqa IE, Murphy MJ (2015) Machine learning in radiation oncology, pp 3–11. https://doi.org/10.1007/978-3-319-18305-3
8. Peck RB (1969) Deep excavations and tunneling in soft ground. In: 7th international conference on soil mechanics and foundation engineering, pp 225–290
9. Litwiniszyn J (1956) Application of the equation of stochastic processes to mechanics of loose bodies. Arch Mech Stos 8:393–411
10. Attewell PB, Farmer IW (1974) Ground disturbance caused by shield tunnelling in a stiff, overconsolidated clay. Eng Geol 8:361–381
11. O'Reilly MP, New BM (1982) Settlements above tunnels in the United Kingdom-their magnitude and prediction
12. Kavvadas M, Hewison LR, Laskaratos PG, Seferoglou C, Michalis I (1996) Experiences from the construction of the Athens Metro, pp 1–7
13. Standing JR, Nyren RJ, Burland JB, Longworth TI (1996) The measurement of ground movements due do tunnelling at two control sites along the Jubilee Line extension, pp 751–756
14. Mair RJ, Taylor RN (1999) Bored tunnelling in the urban environments. In: Fourteenth international conference on soil mechanics and foundation engineering. Proceedings international society for soil mechanics and foundation engineering
15. Wan MSP, Standing JR, Potts DM, Burland JB (2017) Measured short-term ground surface response to EPBM tunnelling in London Clay. Geotechnique 67:420–445
16. Fang Q, Zhang D, Li Q, Wong LNY (2015) Effects of twin tunnels construction beneath existing shield-driven twin tunnels. Tunn Undergr Space Technol 45:128–137
17. Gong C, Ding W, Xie D (2020) Twin EPB tunneling-induced deformation and assessment of a historical masonry building on Shanghai soft clay. Tunn Undergr Space Technol 98:103300
18. Islam MS, Iskander M (2021) Twin tunnelling induced ground settlements: a review. Tunn Undergr Space Technol 110:103614
19. Fujita K (1981) On the surface settlements caused by various methods of shield tunnelling, pp 609–610
20. Mair RJ, Taylor RN, Bracegirdle A (1993) Subsurface settlement profiles above tunnels in clays. Geotechnique 43:315–320
21. Loganathan N, Poulos HG (1998) Analytical prediction for tunneling-induced ground movements in clays. J Geotech Geoenviron Eng 124:846–856
22. Terzaghi K (1942) Shield tunnels of the Chicago Subway. Harvard University, Graduate School of Engineering
23. Cording EJ (1975) Displacement around soft ground tunnels. General report: session IV, tunnels in soil
24. Koungelis DK, Augarde CE (2004) Interaction between multiple tunnels in soft ground. University of Durham, UK, School of Engineering
25. Addenbrooke TI, Potts DM, Puzrin AM (1997) The influence of pre-failure soil stiffness on the numerical analysis of tunnel construction. Géotechnique 47:693–712
26. Chapman DN, Rogers CDF, Hunt DVL (2004) Predicting the settlements above twin tunnels constructed in soft ground. Tunn Undergr Space Technol 19:78–380
27. Mirhabibi A, Soroush A (2012) Effects of surface buildings on twin tunnelling-induced ground settlements. Tunn Undergr Space Technol 29:40–51
28. Sterpi D, Cividini A (2004) A physical and numerical investigation on the stability of shallow tunnels in strain softening media. Rock Mech Rock Eng 37:277–298
29. Suwansawat S, Einstein HH (2006) Artificial neural networks for predicting the maximum surface settlement caused by EPB shield tunneling. Tunn Undergr Space Technol 21:133–150. https://doi.org/10.1016/j.tust.2005.06.007
30. Moller S (2006) Tunnel induced settlements and structural forces in linings
31. Do N-A, Dias D, Oreste P, Djeran-Maigre I (2014) 2D tunnel numerical investigation: the influence of the simplified excavation method on tunnel behaviour. Geotech Geol Eng 32:43–58

32. Rukdeechuai T, Jongpradist P, Wonglert A, Kaewsri T (2009) Influence of soil models on numerical simulation of geotechnical works in Bangkok subsoil. Eng J Res Dev 20:17–28
33. Möller SC, Vermeer PA (2008) On numerical simulation of tunnel installation. Tunn Undergr Space Technol 23:461–475
34. Ercelebi SG, Copur H, Ocak I (2011) Surface settlement predictions for Istanbul Metro tunnels excavated by EPB-TBM, Environ. Earth Sci 62:357–365
35. Likitlersuang S, Surarak C, Suwansawat S, Wanatowski D, Oh E, Balasubramaniam A (2014) Simplified finite-element modelling for tunnelling-induced settlements. Geotech Research 1:133–152
36. Hieu NT, Giao PH, Phien-wej N (2020) Tunneling induced ground settlements in the first metro line of Ho Chi Minh City, Vietnam. In: Geotechnics for sustainable infrastructure development. Springer, pp 297–304
37. Taylor RN (2018) Centrifuges in modelling: principles and scale effects. In Geotechnical centrifuge technology. CRC Press, pp 19–33
38. Nomoto T, Imamura S, Hagiwara T, Kusakabe O, Fujii N (1999) Shield tunnel construction in centrifuge. J Geotech Geoenviron Engineering 125:289–300. https://doi.org/10.1061/(asce)1090-0241(1999)125:4(289)
39. Divall S (2013) Ground movements associated with twin-tunnel construction in clay
40. Moussaei N, Khosravi MH, Hossaini MF (2019) Physical modeling of tunnel induced displacement in sandy grounds. Tunn Undergr Space Technol 90:19–27
41. Anato NJ, Chen J, Tang A, Assogba OC (2021) Numerical investigation of ground settlements induced by the construction of Nanjing WeiSanLu tunnel and parametric analysis. Arab J Sci Eng 46:11223–11239
42. Moeinossadat SR, Ahangari K, Shahriar K (2016) Calculation of maximum surface settlement induced by EPB shield tunnelling and introducing most effective parameter. J Cent South Univ 23:3273–3283
43. Moghaddasi MR, Noorian-Bidgoli M (2018) ICA-ANN, ANN and multiple regression models for prediction of surface settlement caused by tunneling. Tunn Undergr Space Technol 79:197–209
44. Ibrahim D (2016) An overview of soft computing. Procedia Comput Sci 102:34–38
45. Chaturvedi DK (2008) Soft computing. Stud Comput Intell 103:509–612
46. Holland John H (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
47. Cramer NL (1985) A representation for the adaptive generation of simple sequential programs. In: Proceedings of the first international conference on genetic algorithms, pp 183–187
48. Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection
49. Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. Cs/0102027
50. Brownlee J (2011) Clever algorithms: nature-inspired programming recipes
51. Steeb WH (2014) The nonlinear workbook: chaos, fractals, cellular automata, genetic algorithms, gene expression programming, support vector machine, wavelets, hidden Markov models, fuzzy logic with C++, Java and Symbolicc++ programs. World Scientific Publishing Company
52. González DM (2005) Discovering unknown equations that describe large data sets using genetic programming techniques
53. Alavi AH, Gandomi AH (2011) A robust data mining approach for formulation of geotechnical engineering systems. Eng Comput (Swansea)
54. Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychol Rev 65:386–408. https://doi.org/10.1037/h0042519
55. Garrett JH (1994) Where and why artificial neural networks are applicable in civil engineering
56. Monjezi M, Ghafurikalajahi M, Bahrami A (2011) Prediction of blast-induced ground vibration using artificial neural networks. Tunn Undergr Space Technol 26:46–50. https://doi.org/10.1016/j.tust.2010.05.002

57. Khandelwal M, Marto A, Fatemi SA, Ghoroqi M, Armaghani DJ, Singh TN, Tabrizi O (2018) Implementing an ANN model optimized by genetic algorithm for estimating cohesion of limestone samples. Eng Comput 34:307–317
58. Momeni E, Nazir R, Armaghani DJ, Maizir H (2014) Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN. Measurement 57:122–131
59. Simpson PK (1991) Artificial neural systems: foundations, paradigms, applications, and implementations. McGraw-Hill Inc.
60. Dreyfus G (2002) Neural networks: methodology and applications. Springer Science & Business Media
61. Zadeh LA (1965) Information and control. Fuzzy Sets 8:338–353
62. Maier HR, Sayed T, Lence BJ (2000) Forecasting cyanobacterial concentrations using B-spline networks. J Comput Civ Eng 14:183–189
63. Mitchell JK, Soga K (2005) Fundamentals of soil behaviour. Wiley & Sons, New York
64. Shahin MA (2013) Artificial intelligence in geotechnical engineering: applications, modeling aspects, and future directions. In: Metaheuristics in water, geotechnical and transport engineering, First edn. Elsevier Inc., pp 169–204. https://doi.org/10.1016/B978-0-12-398296-4.00008-8
65. Shahin MA, Jaksa MB, Maier HR (2001) Artificial neural network applications in geotechnical engineering. Aust Geomech J 36:49–62
66. Zhang N, Shen S-L, Zhou A-N, Lyu H-M (2021) Challenges of earth pressure balance tunnelling in weathered granite with boulders. Proc Inst Civil Eng Geotech Eng 174:372–389
67. Zhang P, Chen RP, Wu HN (2019) Real-time analysis and regulation of EPB shield steering using Random Forest. Autom Constr 106:102860. https://doi.org/10.1016/j.autcon.2019.102860
68. Goh ATC, Zhang WG (2014) An improvement to MLR model for predicting liquefaction-induced lateral spread using multivariate adaptive regression splines. Eng Geol 170:1–10. https://doi.org/10.1016/j.enggeo.2013.12.003
69. Wang L, Wu C, Gu X, Liu H, Mei G, Zhang W (2020) Probabilistic stability analysis of earth dam slope under transient seepage using multivariate adaptive regression splines. Bull Eng Geol Env 79:2763–2775. https://doi.org/10.1007/s10064-020-01730-0
70. Mitchell TM (n.d.) Learning. McGraw-Hill, New York
71. Ifrah G (2001) The universal history of computing: from the abacus to the quantum computer. Wiley, New York. https://doi.org/10.5860/choice.38-5056
72. Samuel AL (2000) Some studies in machine learning using the game of checkers. IBM J Res Dev 44:207–219. https://doi.org/10.1147/rd.441.0206
73. Talabis M, McPherson R, Miyamoto I, Martin J (2014) Information security analytics: finding security insights, patterns, and anomalies in big data. Syngress
74. Pourtaghi A, Lotfollahi-Yaghin MA (2012) Wavenet ability assessment in comparison to ANN for predicting the maximum surface settlement caused by tunneling. Tunn Undergr Space Technol 28:257–271
75. Boubou R, Emeriault F, Kastner R (2010) Artificial neural network application for the prediction of ground surface movements induced by shield tunnelling. Can Geotech J 47:1214–1233. https://doi.org/10.1139/T10-023
76. Ahangari K, Moeinossadat SR, Behnia D (2015) Estimation of tunnelling-induced settlement by modern intelligent methods. Soils Found 55:737–748
77. Hasanipanah M, Noorian-Bidgoli M, Jahed Armaghani D, Khamesi H (2016) Feasibility of PSO-ANN model for predicting surface settlement caused by tunnelling. Eng Comput 32:705–715. https://doi.org/10.1007/s00366-016-0447-0
78. Lee Y, Oh S-H, Kim MW (1991) The effect of initial weights on premature saturation in back-propagation learning. In: IJCNN-91-Seattle international joint conference on neural networks. IEEE, pp 765–770
79. Armaghani DJ, Hajihassani M, Bejarbaneh BY, Marto A, Mohamad ET (2014) Indirect measure of shale shear strength parameters by means of rock index tests through an optimized artificial neural network. Measurement 55:487–498

80. Jahed Armaghani D, Shoib RSNSBR, Faizi K, Rashid ASA (2017) Developing a hybrid PSO–ANN model for estimating the ultimate bearing capacity of rock-socketed piles, Neural Comput Appl 28. https://doi.org/10.1007/s00521-015-2072-z
81. Eberhart R, Kennedy J (1995) Particle swarm optimization. In: Citeseer, pp 1942–1948
82. Hajihassani M, Jahed Armaghani D, Kalatehjari R (2018) Applications of particle swarm optimization in geotechnical engineering: a comprehensive review. Geotech Geol Eng 36:705–722. https://doi.org/10.1007/s10706-017-0356-z
83. Jang J-S (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23:665–685
84. Jahed Armaghani D, Harandizadeh H, Momeni E (2021) Load carrying capacity assessment of thin-walled foundations: an ANFIS–PNN model optimized by genetic algorithm. Eng Comput 1–23
85. Armaghani DJ, Asteris PG (2021) A comparative study of ANN and ANFIS models for the prediction of cement-based mortar materials compressive strength. Neural Comput Appl 33:4501–4532
86. Amirkhani S, Nasirivatan S, Kasaeian AB, Hajinezhad A (2015) ANN and ANFIS models to predict the performance of solar chimney power plants. Renew Energy 83:597–607
87. Mellit A, Kalogirou SA, Shaari S, Salhi H, Arab AH (2008) Methodology for predicting sequences of mean monthly clearness index and daily solar radiation data in remote areas: application for sizing a stand-alone PV system. Renew Energy 33:1570–1590
88. Bouayad D, Emeriault F (2017) Modeling the relationship between ground surface settlements induced by shield tunneling and the operational and geological parameters based on the hybrid PCA/ANFIS method. Tunn Undergr Space Technol 68:142–152
89. Jolliffe IT (2002) Principal component analysis for special types of data. Springer
90. Wang J, Mohammed AS, Macioszek E, Ali M, Ulrikh DV, Fang Q (2022) A novel combination of PCA and machine learning techniques to select the most important factors for predicting tunnel construction performance. Buildings 12:919
91. Yun H-B, Park S-H, Mehdawi N, Mokhtari S, Chopra M, Reddi LN, Park K-T (2014) Monitoring for close proximity tunneling effects on an existing tunnel using principal component analysis technique with limited sensor data. Tunn Undergr Space Technol 43:398–412
92. Prasad S, Bruce LM (2008) Limitations of principal components analysis for hyperspectral target recognition. IEEE Geosci Remote Sens Lett 5:625–629
93. Zhang P, Wu H-N, Chen R-P, Chan THT (2020) Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction: a comparative study. Tunn Undergr Space Technol 99:103383
94. Kim CY, Bae GJ, Hong SW, Park CH, Moon HK, Shin HS (2001) Neural network based prediction of ground surface settlements due to tunnelling. Comput Geotech 28:517–547
95. Kobayashi M, Hagiwara T, Yoshino O, Hayasaka Y, Komiya K (2002) Ground movements due to the Rinkai Higashi-Shinagawa tunnel construction by slurry shield method. In: Geotechnical aspects of underground construction in soft ground, pp 405–410
96. Hidayat DK (2006) Prediction of ground settlement due to tunneling using artificial neural networks
97. Garson DG (1991) Interpreting neural network connection weights
98. Santos OJ Jr, Celestino TB (2008) Artificial neural networks analysis of Sao Paulo subway tunnel settlement data. Tunn Undergr Space Technol 23:481–491
99. Ocak I, Seker SE (2013) Calculation of surface settlements caused by EPBM tunneling using artificial neural network, SVM, and Gaussian processes. Environ Earth Sci 70:1263–1276
100. Brinkgreve RBJ, Engin E (2013) Validation of geotechnical finite element analysis. In: 18th international conference on soil mechanics and geotechnical engineering: challenges and innovations in geotechnics, ICSMGE 2013, vol 1, pp 677–682

101. Zhang D, Wang D, Peng Q, Lin J, Jin T, Yang T, Sorooshian S, Liu Y (2022) Prediction of the outflow temperature of large-scale hydropower using theory-guided machine learning surrogate models of a high-fidelity hydrodynamics model. J Hydrol (Amst) 606:127427
102. Adombi AVDP, Chesnaux R, Boucher M-A (2021) Theory-guided machine learning applied to hydrogeology—state of the art, opportunities and future challenges. Hydrogeol J 29:2671–2683

A Review on the Feasibility of Artificial Intelligence in Mechatronics

Amin Hashemi and Mohammad Bagher Dowlatshahi

1 Introduction

Intelligent control strategies are very flexible for describing from an automation point of view. While the system performs critical functions, this concept is dynamically implemented in real-time. So it is different from traditional feedback. Thus, if the law of control is continuously updated, we can assume the classical adaptive control to be intelligent. This kind of system can be considered borderline according to the classification perspective [1]. If we trace intelligent control approaches from the source for mechatronics analysis, we will face analyzing and processing big data, the evolution of mathematic-based methods, and programming. The exponential growth process of this research area reached the early 2010s and did not stop declining [2].

Various artificial intelligence methods and areas are utilized in mechatronics and robotics, including artificial neural networks (ANNs), machine learning, evolutionary computing algorithms, and fuzzy logic. Machine learning consists of deep learning, reinforcement learning, classical learning (unsupervised and supervised), neural networks (NN), and ensemble methods. Intelligent control algorithms analyze large data sets and exploit beneficial patterns from actions taken by utilizing a variety of probabilistic, statistical, and optimization methods [2]. In the automatic control field, reinforcement learning is practical. Ensemble strategies and classical learning are also used for classifying and processing data sets against neural networks [3–5]. Conversely, neural networks are practical in dealing with unlabeled features and complex data.

A. Hashemi · M. B. Dowlatshahi (✉)
Department of Computer Engineering, Faculty of Engineering, Lorestan University,
Khorramabad, Iran
e-mail: dowlatshahi.mb@lu.ac.ir

A. Hashemi
e-mail: hashemi.am@fe.lu.ac.ir

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*,
Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_3

In autonomous systems that interact with the real world, a critical challenge is the safety and reliability of utilizing intelligent control approaches. This problem is the subject of a review article [6] in which an asymptotic analysis of intelligent control approaches convergence is conducted.

This chapter discusses modern intelligent control approaches in mechatronics to recognize open issues and trends in intelligent control.

The chapter is constructed as follows: Sect. 2 recalls multiple intelligent control approaches. The applications of intelligent approaches in engineering control problems are presented in Sect. 3. The chapter is concluded in Sect. 4.

2 Smart Control Methods

Intelligent control is an apart discipline, but the application of new concepts, such as neural networks to control loops, utilizing different scientific approaches constructed based on automatic control theory, is required. Therefore, intelligent control can improve its performance every time by learning from previous experiences as a combined approach.

In the framework of intellectual approaches, assume the most general modern control theory methods. This chapter pays the most attention to machine learning since some of them are well-known, and there is no further explanation.

2.1 Adaptive Control Methods

Like optimal control, adaptive control is constructed based on a well-developed mathematical and theoretical justification [2]. This method became the initial step for intelligent control, as an integration of adaptive controllers within the framework of the classical automatic control theory provides a quality of operation of the system given the conditions of the parameters of the object and the specification of the external environment are unknown or change indefinitely. The adaptation principle can be considered the heart of intelligent control processes, which evolves from self-optimizing controllers to adaptive learning systems [7].

Adaptive control algorithms for time-discrete systems were applied to artificial intelligence by Yakubovich, who received several algorithms for training linear classification models [2, 8–10]. The Stripe Algorithm (SA) proposed by Yakubovich in a recently published article [11] has shown acceptable performance in machine learning. SA achieves higher performance than traditional linear learning methods by numerical analysis in online machine learning, making it suitable for this field. Lipkovich [12] provides various strategies for the reduction of loss optimization problems in dealing with inequalities systems, considering both regression and classification problems. In reference [11], a comparison analysis is conducted between

SA and modern linear analogs, including logistic and linear regression. Complex non-linear models have the potential to outperform SA. However, the last one includes the common points of interest of linear models, like explainability, development, and implementation [12]. It can be noted that the presence of a hypothesis does not ensure practical application, especially in experimental conditions of control systems [12].

2.2 Optimization Techniques

Optimization techniques emerged before machine learning historically and were utilized to discover the extrema of a function [13]. Most machine learning problems are based on the theory of optimality. This theory can be generally formulated as the minimization of multiple features J regarding multiple parameters $\theta = J(\theta) \rightarrow \min\limits_{\theta \epsilon X}$. The form of the minimized value is determined by the machine learning approach. As an example, the prediction error on the existing sample is minimized in a regression or classification problem, while the greatest advantage from the activities of the plant is discovered in reinforcement learning. This can be accomplished by any search algorithm. As you can see, there are many types, methods, and uses of mathematical optimization.

2.3 ANNs

ANNs inspire biological networks as powerful artificial intelligence tools. ANN is an object that imitates the neural network constituting the human brain so that the computer can learn and make decisions like a human. An input layer, a hidden layer or more, and nodes or neurons as numerous simple computational components as an output layer construct an ANN structure. A simple ANN with two inputs and two hidden layers is presented in Fig. 1. This additionally includes relationships between neurons in consecutive layers through the weights. These weights can change the signal sent from one node to another and increase or decrease the impact of a particular relationship. A weighted input plus one bias from each neuron in the previous layer is received by each hidden layer neuron. The output of neurons is determined by their activation function. An ANN structure is shown below

$$Y = f\left(\sum_{i=1}^{n} w_i v_i - b\right) \tag{1}$$

In Eq. 1, f refers to the activation function, v_i and w_i are shown the input values and the weights of neurons, respectively. Also, y refers to the network's output, b is the bias, and n indicates the neuron's number in the hidden layers. The performance

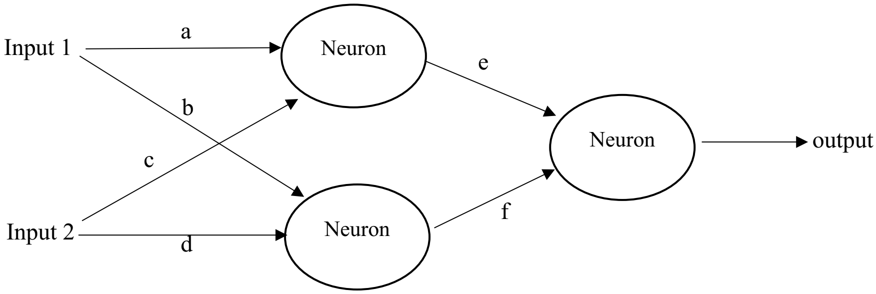


Fig. 1 A simple ANN structure

of the model can be enhanced by updating the network weights during the training phase. ANN's neuron weights determine how the input data affects output data. The primary weights are selected randomly [14].

The network's internal weights are tuned using a learning algorithm. Backpropagation (BP) algorithms are today's most common form of training in ANNs. Also, optimization methods such as genetic algorithm and particle swarm optimization are practical in optimizing the ANN [14].

Using NNs is effective for noisy and non-linear system controls, and adaptability is provided for the system. The NN can work in real time after training. The constant NS advancement in properties and structure aims to overcome the existing shortcomings. The heuristic learning methods for NN can lead to deadlocks and vague solutions, and it needs a training sample to be prepared. Long training time is the principal disadvantage of NN in robotics, increasing the risk of inappropriate control of expensive equipment, the uniformity of training results for predictions, and the current implementation of NN can only be implemented in a very large-scale integration circuit form.

2.4 Fuzzy Logic Method

Zadeh proposed the fuzzy logic as an object with an element membership function in ranges [0, 1] to a set based on the fuzzy set concept [15, 16]. It turns out that fuzzy logic inference can be expressed in the NN form using the membership function to perform the task of neurons and the activation function, considering the neurons' connections as signal transmission. Currently, a lot of fuzzy neural networks roughly explained by the widespread shape of approximators have been developed [17].

2.5 Reinforcement Learning

Reinforcement learning is an approach for handling hybrid optimization problems in machine learning, a structure in which the operator learns how to perform successive decision-making tasks online through interaction with the environment. Reinforcement learning in agent planning is provided by receiving feedback on the outcome of choices made as a reward or punishment without specifying how to achieve the outcome. The reinforcement learning procedure is that the agent first chooses an action from the limited and possible action collection based on observing a situation in the environment and performing that action. Then, the agent receives a predetermined signal from the environment, demonstrating the quality of the operator's action as a reward or punishment. In the next step, the agent transfers to a new environmental status based on the current state [18–20]. In this approach, the agent interacts with the environment by performing a series of actions to find solutions [21, 22]. MDP provides a widely utilized mathematical framework for modeling such problems and consists of four stages [21–23]:

1. A set of states, $S = \{s_1, s_2, ..., s_d\}$
2. A set of actions, $A = \{a_1, a_2, ..., a_m\}$
3. A state transfer function $T = (s'|s, a)$ is a possibility distribution function that a given state s and action into a state s'.
4. A reward function $R = S \times A \times S \rightarrow \mathbb{R}$ gives an instant reward when an agent performs an activity and moves from state to state s'.

Using the Markov chain in reinforcement learning, the agent's choice of action is subject to a policy that determines the probability of choosing the action in a specific status. In other words, it determines the effect of the action in an independent state in such a way that the reinforcing learning agent learns to maximize all future rewards [24, 25].

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \quad = \sum_{k=0}^{\infty} \gamma^k . r_{t+k+1}, \tag{2}$$

where t is the time stage and $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$ is the sequence of rewards after the time stage t, and $\gamma \in [0, 1]$ is a deterrent that handles the significance of instant rewards compared to coming rewards and prevents the reward from going to infinity.

One of the best techniques for solving the Markov decision chain problem is the temporal difference technique, which is notable for its good performance, low computational cost, and plain interpretation. The value of a state or action is estimated using the value of other states or actions [26, 27]. Since the proposed technique basis on temporal difference, we express TD as follows:

$$\mathrm{V}(S_t) = \mathrm{V}(S_t) + \alpha\left[r_{t+1} + \gamma \mathrm{V}(S_{t+1}) - \mathrm{V}(S_t)\right] \tag{3}$$

where parameter α is the learning rate that determines how many errors must accept at every step. Parameter γ is the discount rate that characterizes the influence of the following case. The value inside the bracket is a calculation error in the calculation. It calculates the difference between the worth of case $V(S_t)$ and the estimate of the subsequent step and the subsequent reward $r_{t+1} + \gamma \mathrm{V}(S_{t+1}) - \mathrm{V}(S_t)$ that the operator tries to minimize this time.

3 Application of Intelligent Approaches in Engineering Control Problems

In this section, we will discuss the applications of intelligent approaches in engineering control problems by reviewing a few works in the literature.

3.1 Stabilization and Program Control Problems

Program control and stabilization operations require feedback in the loop. In general, system state vectors are not provided for evaluation, so the available measurement outputs define the control strategy. The robotics and mechatronics standard tasks are similar to speed trajectory tracking and stabilization tasks. These variables can be easily measured at the output of the system. Reference [28] presents a machine learning method for quadcopters. This article presented the approach π_θ using θ as the parameter and is differentiable on parameters. $J(\pi_\theta)$ is the objective function differentiable to θ, for example, the optimization is conducted by the gradient method. For this purpose, probabilistic estimation of the strategy parameters and the mean reward gradient formula is used. The most common method of gradient estimation is formulated as follows:

$$\hat{g} = \widehat{\mathbb{E}_t}\Big[\nabla_\theta \log_{\pi_\theta}(a_t|s_t)\hat{A}_t\Big] \tag{4}$$

where $\widehat{\mathbb{E}_t}$ is the experimental mean for a finite set of instances, $\hat{A}(s_t, a_t) = Q(s_t, a_t) - \hat{V}(s_t)$ represents the advantage function value in time t by changing the sample generation process, and π_θ is the policy enhancement. The dynamic model may be non-differentiable or unknown in this reinforcement learning problem. Thus, the model should be trained, which leads to increasing the gradient estimates variance. For policy optimization in the mentioned article, a solid approach is proposed by incrementally enhancing agent performance. After differentiation of Eq. 4, the objective function is formulated below.

$$J(\theta) = \widehat{\mathbb{E}_\mathrm{t}}\Big[min\Big(r(\theta)\hat{A}_t, clip(r(\theta), 1 - \varepsilon, 1 + \varepsilon)\Big)\Big] \tag{5}$$

where $r(\theta) = \pi_{\theta}/\pi_{\theta old}$ and ε is the hyperparameter.

By differentiation of Eq. 5, gradient $\hat{g}$ is obtained. The reward function is formulated as follows:

$$r_t\left(e_{xt}, e_{yt}, e_{zt}\right) = \alpha - \sqrt{e_{xt}^2 + e_{yt}^2 + e_{zt}^2} \tag{6}$$

where α is a constant to make sure that each quadcopter is assigned a reward, and e_{xt}, e_{yt}, e_{zt} are the coordinates of the quadcopter.

In reference [29], entropy-based reinforcement learning is considered with a soft membrane vibrating drive for ultra-fast tripod robot gait. Data collection for learning and controller development with feedback are needed for this type of problem. A Gaussian normal distribution policy is defined as the controller: $f_{\varphi}(s_t) = (\mu_t, \sigma_t)$ in which φ is the controller parameter, σ_t and μ_t refer to the standard deviation and mean, respectively. The action strategies for starting are defined as $N\left(a_t, f_{\varphi}(s_t)\right)$ and function f_{φ} is constructed as a neural network. The reward function is presented as follows:

$$r(s_t) = -d_t - \delta\theta_t + c \tag{7}$$

where the root mean square error between the current position of the robot and its final position is shown by d_t, c is the coefficient, and the angular difference between the current and desired position is shown by $\delta\theta_t$. To Maximize Entropy Solution, the optimal solutions policy is formulated as follows:

$$\pi_{\alpha}^{*} = \arg\max_{\pi} \mathbb{E}_{\tau\ P,\pi}\left[\sum_{t=0}^{\infty} \gamma^{t}\left(\hat{r}(s_t, a_t)\right) + \alpha H(\pi(.|s_t))\right] \tag{8}$$

$$H\left(\pi_{\varphi}(.|s_t)\right) = \mathbb{E}_{\alpha\sim\pi_{\varphi}}\left[-log\pi_{\varphi}(a|s)\right] \tag{9}$$

where α is the entropy temperature in ranges $[0, \infty)$ and $\hat{r}(s_t, a_t) = \mathbb{E}_{\acute{s}\sim P(\pi(\,.|s,a))}\left[r\left(\acute{s}\right)\right]$. The function value should be minimized by stochastic gradient descent.

If we have control goal changing repeatedly, the mentioned reinforcement learning method is not applicable. To solve this problem, you can use a set of state-action-reward, which can be trained to mimic a specific objective in each set. This solution is presented by Puccetti et al. [30] and is tried on a speed control framework.

Bayesian statistical methods are very effective in intelligent systems [31]. A new hypothesis is achieved by recent data from human brain research led to that in specific types of sensorimotor learning, the brain uses Bayesian internal models to optimize performance on specific tasks. The resulting activity of a particular neuronal population can be modeled as a coordinated Bayesian process. The concept of neural signal processing can be utilized in a variety of applications, from rehabilitation engineering to artificial intelligence.

3.2 Controller Tuning

Utilizing fuzzy and adaptive controllers and PID is common in the industry. In adaptive control schemes, both the controller parameters and structure can be changed in response to parameters alteration of the disturbances or controlled object. An overview provides a historical viewpoint on learning methods and adaptive control [7]. In many cases, the structure of the controller is fixed, and only its parameters need to be tuned. It is known how to tune the controller based on a description of the system dynamics. Therefore, it is not easy to obtain in practice, requiring deep system knowledge and potentially open-loop, large-scale measurements. The first proposed algorithm in this area tries to tune a PID controller with the quick reaction of the system model and the sufficiency and cycle of the closed control-loop natural oscillation [32, 33]. Subsequently, an adaptive PID controller and a discriminative adaptive control algorithm were proposed, and the model parameter estimates were used to adjust coefficients [34, 35]. In some cases, especially if the system is unstable, only feedback measurements are possible. The alteration gets to be cumbersome and wasteful in this connection as the operating conditions of the system change. It, therefore, relies on automated methods that can rapidly decide the parameters of the controller without human intercession based on the task. A self-regulating structure starts work in this area.

In reference [36], a multi-parameter self-tuning controller is proposed to control an injection molding machine. The dynamics of a system are explained by the following probabilistic model of discrete time.

$$A\left(q^{-1}\right)y(t) = B\left(q^{-1}\right)u(t-d-1) + C\left(q^{-1}\right)e(t) \tag{10}$$

where an output vector of dimension p is shown by $y(t)$, an input vector of dimension s is indicated by $u(t)$, q^{-1} is the reverse shift operator, $e(t)$ is white Gaussian noise of dimension p, d is the unit time delay, and $(q^{-1}y(t)) = y(t-1)$. The model presented in Eq. 8 is presented by the self-tuning estimation strategy with recursion in k-step as follows:

$$\acute{y}(t+k|t) = \sum_{i=1}^{n_a} \hat{A}_i \hat{y}(t+k-i|t) + \sum_{i=d}^{n_b} \hat{B}_i u(t+k-i) + \sum_{i=d}^{n_c} \hat{C}_i \hat{e}(t+k-i) \tag{11}$$

where $\hat{A}_i$, $\hat{B}_i$, $\hat{C}_i$ indicates the estimated matrices for Eq. 10 and $k = 1, 2, ..., d$. Thus, the optimization problem is reformulated as follows:

$$J = \left\| D_0 \hat{B}_d u(t) + \hat{L}(t) \right\|_{Q1}^2 + \left\| G_0 u(t) + \sum_{i=1}^{r} G_i u(t-1) \right\|_{Q_1}^2 \tag{12}$$

This turns a stochastic optimization problem into a deterministic problem: $\partial J/\partial u(t)$, in which the output of the self-tuning controller indicates by $u(t)$:

$$u(t) = -\left[\left(D_0\hat{B}_d\right)^T Q_1 D_0 \hat{B}_d + G_0^T Q_2 G_0\right]^{-1} \left[\left(d_0\hat{B}_d\right)^T Q_1 \hat{L}(t) + G_0^T Q_2 \sum_{i=1}^{r} G_i u(t-1)\right] \tag{13}$$

The structure adjustment capabilities and additional control of the learning controller must be utilized to fulfill the needs of more complex machines based on the simulation results.

In a study dedicated to self-tuning controllers [37], algorithms were obtained and analyzed by aggregating the least squares estimation (LSE) and tuning the minimum oscillations achieved by the dynamics model. Two theorems are achieved by the primary results assuming convergence of estimating parameters and defining a closed system. Some cross-covariance and output of the output control variable will vanish from the little presumptions of the registry in the first theorem. The second theorem assumes that the control framework may be a common linear likelihood framework of order *n*. When the parameter estimation process is converged, we show that the resulting control law is the control law of most minor variability that can be computed with the known parameters.

3.3 Identification Problems

In reference [38], a method using the bee swarm algorithm to identify linear systems described by differential equations is proposed. To get a model and parameter set that minimizes the prediction error between the model output and the real object, an optimization problem is defined based on the identification problem. The result of the algorithm operation is displayed on the DC motor model.

In reference [39], to adjust the parameters of the PID controller of an evaporator control system while minimizing the system tracking absolute squared error, a heuristic colony competition algorithm was used. The genetic algorithm and Ziegler–Nichols method demonstrate this algorithm's effectiveness.

To determine the lasting magnet synchronous motor model parameters in real time, Rahimi et al. [40] used a heuristic competition algorithm. For this, a minimization process is conducted based on the mean squared error of the system state vector control.

3.4 Optimization Problems

Gradientless search algorithms are widely utilized for all optimization problems due to their versatility. This also applies to NN because NN does not utilize the gradient of the function and does not consider it is differentiable [41]. Their characteristic is that the optimization problem solution is worthy but not ideal. Recently, various biomimetic solutions that borrow ideas from nature are gaining popularity [42, 43]. These include populations [44], swarm and colony algorithms [45–47], evolutionary, etc. A bat algorithm [44] is also known and is related to echolocation-based swarm intelligence. The cuckoo swarm algorithm tunes the PID controller in thyristor series compensation [48] and DC motor control systems [49]. The former was more efficient compared to the Swarm algorithm with the heuristic algorithm.

In reference [50], using support vector algorithms, an optimal control approach is proposed to minimize the bipedal robot's power consumption under a small data sample size and an unknown system dynamics model. The new controller has been integrated into the optimal controller, constraining the robot's joint angles to minimize the energy-related cost function. The energy functional is

$$J_{EE} = \int_{0}^{T} \frac{1}{2}\tau^{T}\tau dt, \tau = g(\Theta) \tag{14}$$

where $g(.)$ is parameterized by NN and Q is a vector of generalized coordinates. The quadratic form support vector machine quality function is

$$J_{SVM} = \min \frac{1}{2}W^{T}W + \frac{1}{2}C\sum_{i=1}^{N}\xi_i^2 as, \tau_i = w^{T}\varphi(\Theta_i) + \xi_i \tag{15}$$

where ξ_i is a positive variable, w is a vector of weights, C is a penalty factor, N refers to the training instances number, and $\varphi(.)$ refers to the transformation function of the input space to the input space of higher-order features. The resulting functional includes the aggregation of J_{EE} and J_{SVM}.

In Ref. [51], an improved "learn-learn" search algorithm is utilized by multi-objective optimization of PID controller parameters. This prevents function values from getting stuck in local minima. To this end, there are not only two learners in the learning, but it includes an additional state. Also, parameters ear to inconsistent targets is combined with a blocked device phase where they are blocked. This ensures that each objective cannot collide with another [52]. The results of comparative studies on optimizing the parameters of the PID controller of the DC motor control system utilizing the particle swarm method, the honey bee colony algorithm, and the learning-learning. The last one showed the best results.

3.5 Problems of Iterative Learning

Machine learning is known as one form of artificial intelligence, which is that rather than being explicitly programmed, the systems can be trained by data stored in memory [53]. Based on processing the training data set, a more accurate model is constructed. This allows you to train the model before and on an ongoing basis. The iterative model training process continuously improves the types of relationships between data items, no matter how complex or large. You can continue training in real time using models trained offline.

In Ref. [54], a fault-tolerant control approach is proposed according to the iterative current-loop learning control for recovering the execution of polyphase permanent magnet drives under open-circuit conditions. This method does not need diagnostics and troubleshooting as its main advantage, and torque measurements are sufficient. Iterative learning management, therefore, provides comprehensive knowledge on reliability for modeling uncertainty and the system. We developed a flexible trajectory-assisted control scheme using iterative learning control for a cloud-wheeled robot system to move along a given trajectory and transport cargo simultaneously and performed a system stability analysis [55]. In [56], a human-led iterative learning framework is presented for a trajectory-tracking task in which a controller gets input from the activities of a human agent. Hladowski et al. [57] considered the influence of noise to achieve new results for the dynamic enhancement of iterative learning control laws.

An iterative procedure is presented for planning the milling process in reference [3]. For that, it is necessary to know the machine's technical parameters and the parts' geometrical parameters to form the machine tool trajectory. Tool deviation is a severe problem in which the milling process requires constant review and planning. Dittrich et al. [3] presented the following solution that reduces processing errors by up to 50% by predicting the error between a model of machined shape and actual surface measurements using machine learning methods. Thus, a statistical support vector machine uses the previous process dataset as the training dataset.

4 Conclusions

The modern world trend towards organizations of advanced production types is reflected in intelligent control scientific publications methods in electromechanical systems. Using AI methods, it is possible to solve previously impossible problems of controlling mechatronic systems while at the same time increasing ease of implementation and computational efficiency. The complexities of control tasks for multi-agent systems are inherently non-linear, uncertain, or influenced by external environments and require individual approaches to solving specific problems, for which many tools are proposed. Only by actual experiments, the effectiveness of these learning algorithms on complex systems can be measured. The development of the algorithm itself aims not only to increase the accuracy and speed of learning but also to increase independence from adaptation to various goals and learning strategies that humans strictly set. Developers try to recreate the behavior of living organisms by utilizing natural thoughts in algorithms. Future research establishes a task, usually referred to as "learning for learning," when agents need to select learning strategies and tune meta-parameters.

References

1. Vepa R (1993) Review of techniques for machine learning of real-time control strategies. Intell Syst Eng 2. https://doi.org/10.1049/ise.1993.0009
2. Zaitceva I, Andrievsky B (2022) Methods of intelligent control in mechatronics and robotic engineering: a survey. Electronics (Basel) 11:2443. https://doi.org/10.3390/electronics11152443
3. Dittrich MA, Uhlich F, Denkena B (2019) Self-optimizing tool path generation for 5-axis machining processes. CIRP J Manuf Sci Technol 24. https://doi.org/10.1016/j.cirpj.2018.11.005
4. Gurel S, Akturk MS (2008) Scheduling preventive maintenance on a single CNC machine. Int J Prod Res 46. https://doi.org/10.1080/00207540701487833
5. Mosheiov G (2001) Scheduling problems with a learning effect. Eur J Oper Res 132. https://doi.org/10.1016/S0377-2217(00)00175-2
6. Matni N, Proutiere A, Rantzer A, Tu S (2019) From self-tuning regulators to reinforcement learning and back again. In: Proceedings of the IEEE conference on decision and control. https://doi.org/10.1109/CDC40024.2019.9029916
7. Annaswamy AM, Fradkov AL (2021) A historical perspective of adaptive control and learning. Annu Rev Control 52. https://doi.org/10.1016/j.arcontrol.2021.10.014
8. Fradkov AL (2017) Scientific school of Vladimir Yakubovich in the 20th century. IFAC-PapersOnLine. https://doi.org/10.1016/j.ifacol.2017.08.461
9. Bondarko VA, Yakubovich VA (1992) The method of recursive aim inequalities in adaptive control theory. Int J Adapt Control Signal Process 6. https://doi.org/10.1002/acs.4480060303
10. Gusev SV, Bondarko VA (2020) Notes on Yakubovich's method of recursive objective inequalities and its application in adaptive control and robotics. IFAC-PapersOnLine. https://doi.org/10.1016/j.ifacol.2020.12.1885
11. Lipkovich M (2022) Yakubovich's method of recursive objective inequalities in machine learning. IFAC-PapersOnLine. 55:138–143. https://doi.org/10.1016/j.ifacol.2022.07.301
12. Perel'man II (1991) Analysis of modern adaptive control methods from the stand-point of application to automatization of technological processes. Avtomatika i Telemekhanika
13. Andrievsky BR, Fradkov AL (2021) Speed gradient method and its applications. Autom Remote Control 82. https://doi.org/10.1134/S0005117921090010
14. Abdullah AM, Usmani RSA, Pillai TR, Marjani M, Hashem IAT (2021) An optimized artificial neural network model using genetic algorithm for prediction of traffic emission concentrations. Int J Adv Comput Sci Appl 12. https://doi.org/10.14569/IJACSA.2021.0120693
15. Novák V, Perfilieva I, Močkoř J (1999) Mathematical principles of fuzzy logic.https://doi.org/10.1007/978-1-4615-5217-8
16. Gupta MM, Kiszka JB (2003) Fuzzy sets, fuzzy logic, and fuzzy systems. In: Encyclopedia of physical science and technology. https://doi.org/10.1016/b0-12-227410-5/00270-2
17. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2. https://doi.org/10.1007/BF02551274
18. Mazyavkina N, Sviridov S, Ivanov S, Burnaev E (2021) Reinforcement learning for combinatorial optimization: a survey. Comput Oper Res 134. https://doi.org/10.1016/j.cor.2021.105400

19. Joshi DJ, Kale I, Gandewar S, Korate O, Patwari D, Patil S (2021) Reinforcement learning: a survey. Adv Intell Syst Comput AISC 1311:297–308. https://doi.org/10.1007/978-981-33-4859-2_29
20. Kober J, Bagnell JA, Peters J (2013) Reinforcement learning in robotics: a survey. Int J Robot Res 32:1238–1274. https://doi.org/10.1177/0278364913495721
21. Baird G (2020) Optimising darts strategy using Markov decision processes and reinforcement learning. J Oper Res Soc 71:1020–1037. https://doi.org/10.1080/01605682.2019.1610341
22. Sutton RS, Precup D, Singh S (1999) Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif Intell 112:181–211. https://doi.org/10.1016/S0004-3702(99)00052-1
23. Bäuerle N, Rieder U (2010) Markov decision processes. Jahresber Deutsch Math-Verein 112:217–243. https://doi.org/10.1365/s13291-010-0007-2
24. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. Morgan Kaufmann Publishers. https://doi.org/10.1613/jair.301
25. Wang Y, Wang J, Liao H, Chen H (2017) An efficient semi-supervised representatives feature selection algorithm based on information theory. Pattern Recogn 61:511–523. https://doi.org/10.1016/j.patcog.2016.08.011
26. van Seijen H, Mahmood AR, Pilarski PM, Machado MC, Sutton RS (2016) True online temporal-difference learning. J Mach Learn Res 17:1–40
27. Paniri M, Dowlatshahi MB, Nezamabadi-pour H (2021) Ant-TD: ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection. Swarm Evol Comput 64:100892. https://doi.org/10.1016/j.swevo.2021.100892
28. Lopes GC, Ferreira M, da Silva Simoes A, Colombini EL (2018) Intelligent control of a quadrotor with proximal policy optimization reinforcement learning. In: Proceedings—15th Latin American robotics symposium, 6th Brazilian robotics symposium and 9th workshop on robotics in education, LARS/SBR/WRE 2018. https://doi.org/10.1109/LARS/SBR/WRE.2018.00094
29. Kim JI, Hong M, Lee K, Kim D, Park Y-L, Oh S (2020) Learning to walk a tripod mobile robot using nonlinear soft vibration actuators with entropy adaptive reinforcement learning. IEEE Robot Autom Lett 5:2317–2324. https://doi.org/10.1109/LRA.2020.2970945
30. Puccetti L, Kopf F, Rathgeber C, Hohmann S (2020) Speed tracking control using online reinforcement learning in a real car. In: 2020 6th international conference on control, automation and robotics, ICCAR 2020. https://doi.org/10.1109/ICCAR49639.2020.9108051
31. Poon CS (2004) Sensorimotor learning and information processing by Bayesian internal models. In: Annual international conference of the IEEE engineering in medicine and biology—proceedings. https://doi.org/10.1109/iembs.2004.1404245
32. Ziegler JG, Nichols NB (1943) Process lags in automatic control circuits. Trans ASME 65
33. Ziegler JG, Nichols NB (1995) Optimum settings for automatic controllers. InTech 42
34. Astrom KJ, HÄgglund T (2006) Advanced PID control. IEEE Control Syst 26. https://doi.org/10.1109/MCS.2006.1580160
35. Nishikawa Y, Sannomiya N, Ohta T, Tanaka H (1984) A method for auto-tuning of PID control parameters. Automatica 20. https://doi.org/10.1016/0005-1098(84)90047-5
36. Dong CM, Tseng AA (1989) A multivariable self-tuning controller for injection molding machines. Comput Ind 13. https://doi.org/10.1016/0166-3615(89)90042-0
37. Åström KJ, Borisson U, Ljung L, Wittenmark B (1977) Theory and applications of self-tuning regulators. Automatica 13. https://doi.org/10.1016/0005-1098(77)90067-X
38. Erçin O, Çoban R (2012) Identification of linear dynamic systems using the artificial bee colony algorithm. Turk J Electr Eng Comput Sci 20. https://doi.org/10.3906/elk-1012-956
39. Atashpaz Gargari E, Hashemzadeh F, Rajabioun R, Lucas C (2008) Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process. Int J Intell Comput Cybern 1. https://doi.org/10.1108/17563780810893446
40. Rahimi A, Bavafa F, Aghababaei S, Khooban MH, Naghavi SV (2016) The online parameter identification of chaotic behaviour in permanent magnet synchronous motor by Self-Adaptive Learning Bat-inspired algorithm. Int J Electr Power Energy Syst 78. https://doi.org/10.1016/j.ijepes.2015.11.084

41. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80. https://doi.org/10.1007/s11042-020-10139-6
42. Balochian S, Baloochian H (2019) Social mimic optimization algorithm and engineering applications. Expert Syst Appl 134. https://doi.org/10.1016/j.eswa.2019.05.035
43. Kumar S, Kumar A, Shankar G (2018 Crow search algorithm based optimal dynamic performance control of SVC assisted SMIB system. In: 2018 20th national power systems conference, NPSC 2018. https://doi.org/10.1109/NPSC.2018.8771814
44. Sharma P, Sharma K (2022) Fetal state health monitoring using novel Enhanced Binary Bat Algorithm. Comput Electr Eng 101:108035. https://doi.org/10.1016/j.compeleceng.2022.108035
45. Bayati H, Dowlatshahi MB, Hashemi A (2022) MSSL: a memetic-based sparse subspace learning algorithm for multi-label classification. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-022-01616-5
46. Hashemi A, Dowlatshahi MB, Nezamabadi-pour H (2021) Gravitational search algorithm. In: Handbook of AI-based metaheuristics, p 32
47. Hashemi A, Joodaki M, Joodaki NZ, Dowlatshahi MB (2022) Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: a case study in ensemble feature selection. Appl Soft Comput 109046. https://doi.org/10.1016/j.asoc.2022.109046
48. Sethi R, Panda S, Sahoo BP (2015) Cuckoo search algorithm based optimal tuning of PID structured TCSC controller. In: Smart innovation, systems and technologies. https://doi.org/10.1007/978-81-322-2205-7_24
49. Kishnani M, Pareek S, Gupta R (2014) Optimal tuning of PID controller by Cuckoo Search via Lévy flights. In: 2014 international conference on advances in engineering and technology research, ICAETR 2014. https://doi.org/10.1109/ICAETR.2014.7012927
50. Wang L, Liu Z, Chen CLP, Zhang Y, Lee S (2012) Support vector machine based optimal control for minimizing energy consumption of biped walking motions. Int J Precis Eng Manuf 13. https://doi.org/10.1007/s12541-012-0260-7
51. Xiao L, Zhu Q, Li C, Cao Y, Tan Y, Li L (2014) Application of modified teaching-learning algorithm in coordination optimization of TCSC and SVC. Commun Comput Inf Sci. https://doi.org/10.1007/978-3-662-45646-0_5
52. Shouran M, Habil M, Tuning of PID Controller using different optimization algorithms for industrial DC motor. In: 2021 international conference on advance computing and innovative technologies in engineering, ICACITE 2021. https://doi.org/10.1109/ICACITE51222.2021.9404616
53. Judith Hurwitz DK (2018) Machine learning for dummies. IBM Limited Edition
54. Mohammadpour A, Mishra S, Parsa L (2013) Iterative learning control for fault-tolerance in multi-phase permanent-magnet machines. In: 2013 American control conference. IEEE, pp 5929–5934. https://doi.org/10.1109/ACC.2013.6580768
55. Li J, Wang S, Wang J, Li J, Zhao J, Ma L (2021) Iterative learning control for a distributed cloud robot with payload delivery. Assembly Autom 41. https://doi.org/10.1108/AA-11-2020-0179
56. Warrier RB, Devasia S (2016) Iterative learning from novice human demonstrations for output tracking. IEEE Trans Hum Mach Syst 46. https://doi.org/10.1109/THMS.2016.2545243
57. Hladowski L, Galkowski K, Rogers E (2017) Further results on dynamic iterative learning control law design using repetitive process stability theory. In: 2017 10th international workshop on multidimensional (ND) systems, NDS 2017. https://doi.org/10.1109/NDS.2017.8070621

Feasibility of Artificial Intelligence Techniques in Rock Characterization

Mohamad Bagher Dowlatshahi, Amin Hashemi, Masoud Samaei, and Ehsan Momeni

Abstract In recent past years, the implementation of artificial intelligence (AI) techniques in rock characterization is highlighted in the literature. This is attributed to the fact that direct determination of rock engineering properties such as unconfined compressive strength (UCS) is time-consuming, costly, and some times difficult. This study aimed to review the recent works which propose AI techniques, as indirect methods, for assessing the UCS of rock samples. For this reason, first, the well-established AI techniques are discussed. Subsequently, the prediction performances of recent AI-based predictive models are underlined in this book chapter. Based on the reviewed works, preparing a suitable dataset and selecting proper input parameters for an AI-based predictive model of UCS play crucial roles in the reliability of the developed models. According to the reviewed studies, there should be a meaningful relationship between the considered input parameter and UCS. Additionally, it is recommended to consider input parameters that are approximately independent of each other. On the other hand, the reviewed studies suggest utilizing relatively large datasets for developing intelligent models. Apart from that, a word of caution is required for generalizing the prediction performances of AI-based predictive models of UCS, especially when the dataset size is small and the range of future data is beyond the range of the model's dataset. Overall, the findings recommend the feasibility of artificial intelligence techniques in predicting the UCS of rock samples.

1 Introduction

Unconfined compressive strength (UCS) of rocks is an important parameter in designing rock engineering problems. There are different techniques for estimating the UCS of rocks, however, they are mostly categorized into direct and indirect

---

M. B. Dowlatshahi · A. Hashemi · E. Momeni (✉)
Faculty of Engineering, Lorestan University, Khorramabad, Iran
e-mail: Momeni.e@lu.ac.ir

M. Samaei
Department of Civil, Construction and Environmental Engineering, North Dakota State University (NDSU), Fargo, ND, USA

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*,
Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_4

techniques. Direct method for UCS estimation in the laboratory is time-consuming and costly. Hence, the implementation of indirect methods for assessing the UCS of rocks is of advantage. Indirect methods are mostly categorized into regression-based techniques and soft computing techniques. It is well established that rock index tests such as point load test and P-wave velocity test have good mutual correlation with UCS of rocks. Therefore, many researchers prepared a set of related experimental data for developing regression equations for UCS estimation. On the hand, recently workability of soft computing-based techniques is underlined in literature. As will be discussed later, many studies underlined the feasibility of various artificial intelligence techniques in developing predictive models of UCS. This chapter aimed to shed light on the workability of soft computing methods in rock characterization. For this reason, some of the well-established soft computing techniques are discussed in the next section. Section 3 of this chapter deals with a comprehensive review on the workability of intelligent techniques in assessing the unconfined compressive strength of rocks. The last section of this chapter underlines the summary and conclusion remarks.

2 Artificial Intelligence Methods

In this section, we intend to discuss the Artificial Intelligence (AI) methods, including the Neural network, Adaptive Neuro-Fuzzy Inference System (ANFIS), Gene Expression Programming (GEP), Random Forest (RF), and the combination of neural networks with Particle Swarm Optimization (PSO) as well as Genetic Algorithm (GA).

2.1 ANFIS Algorithm

ANFIS [1] is constructed based on a hybrid system consisting of FIS (**F**uzzy **I**nference **S**ystem) and a neural network. FIS is generally recognized for its ability to map prior knowledge to some constraints. The resulting set can be used to optimize the search space at the network topology level. In conjunction with FIS, ANFIS integrates neural networks with Backpropagation (BP) to automate the tuning of fuzzy controller parameters. ANFIS includes the necessary functions for tuning the network configuration using the Takagi–Sugeno (TS) controller to achieve the best tuning. A learning algorithm is generally embedded in ANFIS through a procedure consisting of two steps. The first is the offline learning phase which consists of forward passing with the least square error. The second step uses a gradient descent algorithm with BP [2]. The architecture of ANFIS is presented in Fig. 1, which has two inputs and one output.

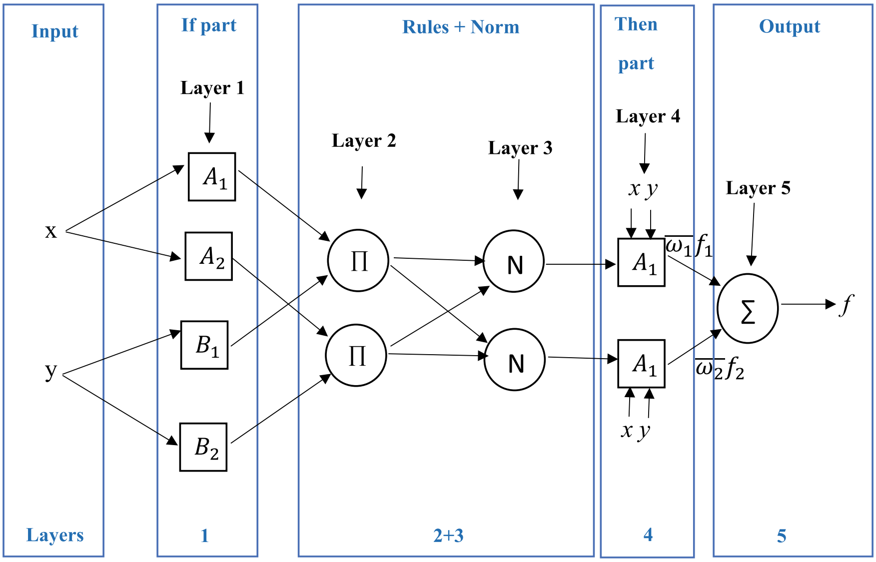


Fig. 1 ANFIS architecture

The structure presented in Fig. 1 has five layers of neurons or perceptrons. Neurons or perceptrons are similar to each other with the same function within a layer, as follows:

Layer 1: This layer is also called the Fuzzyfing layer. Neurons in the first layer contain premise parameters that are considered adaptive nodes.

Layer 2: This layer is also called the Implication layer. Circles present neurons in the second layer including $\prod$ as their label. Thus, the input signals form the output node. The strength of each rule is represented by the output node w_i.

Layer 3: This layer is called the Normalizing layer and includes fixed neurons. Circles present neurons in the third layer, including N as their label. The i-th node in the third layer computes the ratio of firepower of the i-th rule to the sum of all firepower as the output of this layer.

Layer 4: This layer is also called the Defuzzyfing layer. The fourth layer's neurons are actually adaptive and contain the outcome parameters.

Layer 5: This layer is called the Combining layer. The only node in the fifth layer calculates its total output as the sum of all its inputs.

2.2 Artificial Neural Networks

An **A**rtificial **N**eural **N**etwork (ANN) is a system that performs the matching of problematic input and output patterns. An ANN discovers knowledge through several iterations in a learning process. The ANN is ready to evaluate problems with nonlinear functions, predict new behaviors, or classify new information after the learning process is complete. A series of interconnected neurons (represented by functions) construct an ANN and are organized into layers. The ANN's input values are sent through layers, and the information transformation is done using corresponding synaptic weights (values in the range [0, 1]). The subsequent layer neurons then summate this information depending on whether they are connected [3].

Additionally, this sum includes another input called bias, whose value is 1. This bias is denoted by b and indicates the threshold representing the minimum level required for a neuron to activate [3]. An ANN structure is shown in Fig. 2.

$$Y = f\left(\sum_{i=1}^{n} w_i v_i - b\right) \tag{1}$$

In Eq. 1, f refers to the activation function, v_i and w_i are shown the input values and the weights of neurons, respectively. Also, y refers to the network's output, b is the bias, and n indicates the neuron's number in the hidden layers. The performance of the model can be enhanced by updating the network weights during the training phase. ANN's neuron weights determine how the input data affects output data. The primary weights are selected randomly [4].

The network's internal weights are tuned using a learning algorithm. Backpropagation (BP) algorithms are today's most common form of training in ANNs. Also, optimization methods such as GA and PSO are practical in optimizing the ANN [4]. For this optimization, a cost function is required to evaluate the network's performance in each iteration. It means that the network weights should be set so that the predicted output is close to the actual output [5]. The mean squared error (MSE) function is a popular cost function defined as follows for this task:

$$MSE = \frac{1}{2}\sum_{k=1}^{G}\sum_{j=1}^{m}\left[Y_j(k) - T_j(k)\right]^2 \tag{2}$$

where m and G refer to output nodes and the number of training instances, respectively. $T_j(k)$ shows the actual output and $Y_j(k)$ refers to the expected output.

In the rest of this section, we will discuss optimization methods used to tune ANN weights.

2.2.1 Designing ANN Using GA

Genetic algorithm (GA) is a class of collective adaptation algorithms inspired by the evolution theory of Darwin. Each genome or individual in a population represents a specific location in the search space. GA uses the (objective) fitness function to evaluate the objective value of every individual in the population. A favorable solution is randomly selected to improve weak solutions using a selection mechanism (such as a roulette wheel). This mechanism tends to select the best solution because the probability is commensurate to the goodness of the objective value. Setting the initial population, selection, crossover, and mutation are the main steps of the GA algorithm [6, 7]. This section describes these steps individually.

Initializing the population: First, several solutions are accidentally generated. The initial population size depends on the problem's nature and typically ranges from hundreds to thousands of solutions. The initial population can be planted in a specific area with a high likelihood of discovering an optimal solution. The following steps are used to enhance the chromosomes in the initial population.

Selection: A new generation is produced in every iteration of GA from the previous generation. Solutions are evaluated according to their objective values, so the likelihood of selecting better solutions are higher compared to the other solutions. There is still a chance for selection of weak individuals which can help the population diversity. There are multiple selection mechanisms for GA. Tournament selection and roulette wheel selection are two popular methods for this task.

Crossover: After selecting individuals, they should be utilized to produce a new generation. Chromosomes in female and male genes mate to form new chromosomes in nature. This is modeled in the GA algorithm by aggregating two solutions (the parent solutions) to produce two new solutions (the child solutions). There are several strategies to use the crossover operator, such as single-point, double-point, uniform crossover, to name a few.

Mutation: The most straightforward genetic operator is mutation. This operator alters the bits value of a chromosome from 0 to 1 or 1 to 0 by random selection. This operation may lead to new solutions in the search space, which may make the algorithm one step closer to finding the optimal solution. In order to avoid the algorithm does not become a simple random search, the mutation rate should be low. Mutations include gene deletions, gene duplications, gene sequence inversions, and the insertion of part of a chromosome into another one.

The GA algorithm first selects a random population. The three operators above improve the population until the termination criterion is reached. The final population optimal solution is the optimal solution for the given problem. Algorithm 1 presents the pseudo-code of the GA algorithm.

Algorithm 1: GA

```
Population initialization
Repeat
    For all chromosomes in the population, Do
        Compute the fitness value
    End For
    If the termination criterion is reached, Then
        Halt
    Else
        Place copies of the best individuals in the new population
        Repeat
            Perform crossover based on the rate of crossover
            Perform mutation based on the rate of mutation
        Until the new generation is produced
    End If
Until the termination criterion is reached
```

GA algorithms can solve many optimization problems, including ANN weight optimization. In this regard, we can set various random values for the ANN as the initial GA population to achieve the best value. For better understanding, let us consider a simple ANN with two inputs and two hidden layers in Fig. 2. All weights in the network are combined into one chain. GA then uses this row as a chromosome. Each chromosome represents the weight of the entire network. In Fig. 3, a chromosome structure is shown based on the structure in Fig. 2. Thus, multiple chromosomes are generated with random values as the initial population of GA based on the population size. The fitness function is also presented in Eq. 2. To evaluate the individuals, they are fed into the network to predict the output. Then based on Eq. 2, the fitness evaluation is performed. At last, the based weights are determined for the network as the result of GA [8].

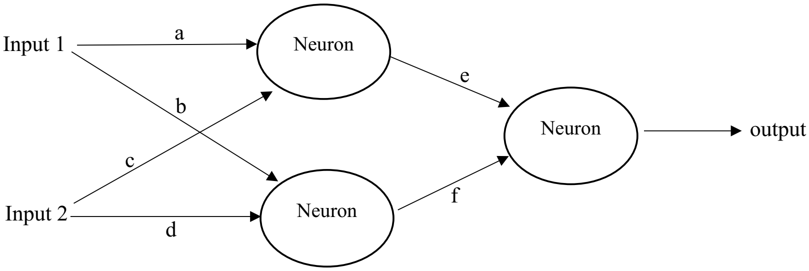


Fig. 2 A simple ANN structure

Fig. 3 A chromosome (A particle)

a	b	c	d	e	f

2.2.2 Designing ANN Using PSO

PSO is a population-based search technique inspired by the social behavior of bird flocks and fish aggregation populations. Each individual is called a particle in the PSO. The particles fly through the search space and adjust their positions using their experience and neighbors. Particles can perform a global search by traveling fast away from the best location to search unknown regions, or they can be very slow and close to a specific location for best results (tuning). PSO is straightforward to implement and includes a few control parameters. The following equations are the two main rules for updating a standard PSO [7]. The following formula updates the particle's speed:

$$v_{id}^{(t+1)} = wv_{id}^{(t)} + u[0, 1]\varphi_1\left(p_{id}^{(t)} + x_{id}^{(t)}\right) + u[0, 1]\varphi_2\left(p_{id}^{(t)} + x_{id}^{(t)}\right) \tag{3}$$

where t is the iteration index, $u[0, 1]$ is a uniform random distribution of particles, and φ_1 and φ_2 indicate the impact rate of the local and global optima on the total speed of particles [7]. The position of each particle is updated according to Eq. 4 as follows:

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t)} \tag{4}$$

The PSO algorithm first generates a random population of articles using random values for particle positions and speeds. Until the end criterion is reached, the position and velocity of articles are improved by the above equations. At first, the fitness value is determined for all particles to discover *G.best*. *G.best* refers to the best solution in the population, and *P.best* indicates the best position of a particle that has reached so far. Then all particles tend to be the *G.best* since it is closer to the optimum solution. This process is repeated until no enhancement is done to the *G.best* particle [3, 9]. Algorithm 2 presents the pseudo-code of the PSO algorithm.

Algorithm 2: PSO

```
Population initialization (initial positions and velocities)
Repeat
    For all particles in the population, Do
        Compute the fitness value
        If the position of particle $i$ achives the best fitness so far for this particle, Then
            $P.best_i = x_i$
            If the position of particle $i$ achives the best overall fitness, Then
                $G.best_i = x_i$
            End If
        End If
    End For
Update the positions and velocities of all particles based on Eqs. 3 and 4
Until the termination criterion is reached
```

PSO algorithm can solve many optimization problems, including optimizing the weights of an ANN. For this matter, different random values are set for the ANN as the initial particles of PSO. To understand better, let us consider a simple ANN with two inputs and two hidden layers in Fig. 2. All weights in the network are combined into one chain. PSO then uses this row as a particle. Each particle represents the weight of the entire network. In Fig. 3, a particle structure is shown based on the structure in Fig. 2. Thus, multiple particles are generated with random values as the initial positions of PSO particles based on the population size. The fitness function is also presented in Eq. 2. To evaluate the particles, they are fed into the network to predict the output. Then based on Eq. 2, the fitness evaluation is performed. At last, the based weights are determined for the network as the *G.best* in PSO. It can be noted that the particle's initial speeds are set randomly in the range $[-1, 1]$.

2.3 Gene Expression Programming (GEP)

The Genetic Programming (GP) method is proposed as a GA alternative using fixed-length binary strings. GP algorithm is considered a promising technique due to the use of non-linear parse tree structures. The initial non-linearity of the data is considered by this algorithm. GP is inappropriate because it ignores independent genomes. The non-linear structure of GP works for both phenotype and genotype. It is impossible to develop a basic and simple model. An evolutionary population-based algorithm is proposed, known as the GEP method, to overcome the contradictions of the GP algorithm. It is a modified version of GP to overcome its weakness [10].

Passing the genome to the next generation is a significant change in GEP concerning GP. Another notable feature is using chromosomes of different genes to create objects. Each gene arises from arithmetic operations, fixed-length parameters, and a finite set of constants used as functions in GEP. There is a stable and fluid interface between the relevant functions and the chromosomal level. Chromosomes record essential information which are needed to build models, and a new language, Carba, is being developed to output this information [10].

A flowchart of the GEP algorithm is shown in Fig. 4. The algorithm starts by randomly generating a fixed-length chromosome for every individual. Then it is similar to ET (expression tree). After that, the suitability of each individual is assessed. Over generations, iterations begin with multiple individuals until the best results are generated. Genetic functions such as mutation, reproduction, and crossover are performed to repeat a population [10].

2.4 Random Forest Regression (RFR)

Random forest regression (RFR) is an improved regression method with flexibility and agility in developing relationships between output and input parameters. Random

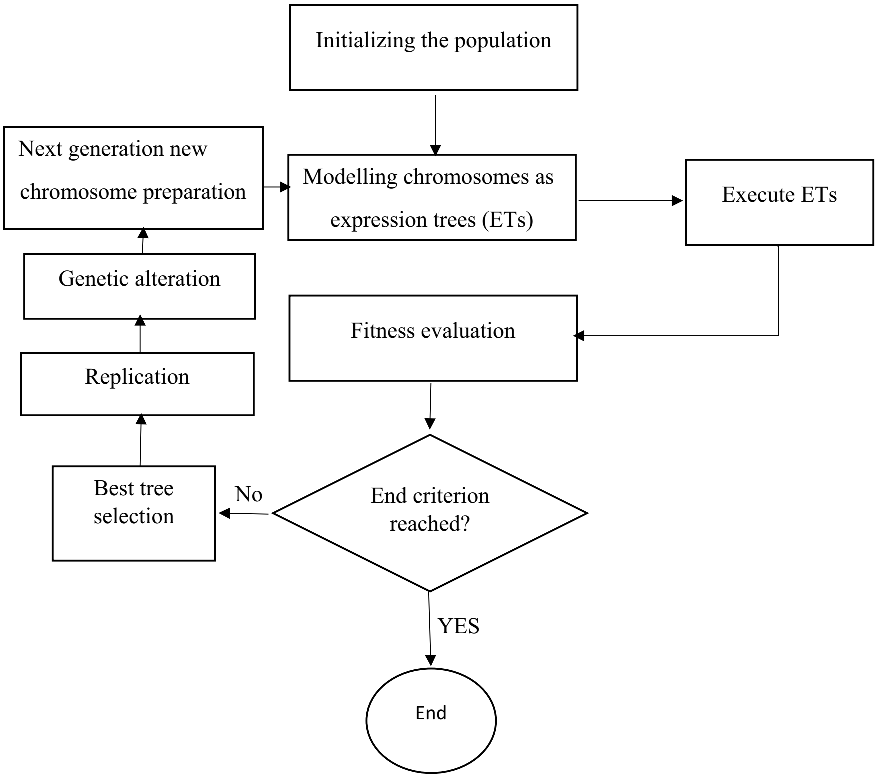


Fig. 4 The flowchart of GEP steps

forests also process large data sets more efficiently than machine learning algorithms. It is used in various fields such as banking, customer response prediction, and stock price direction, to name a few [10].

The RF method consists of three main steps: building a trained regression tree with the training data set, computing the average of the results of one regression tree, and testing the prediction results with the validation data set. The original trained set is used to compute a new trained data set of bootstrap data. In this step, some data points are removed and replaced with the current data points. Eliminated data points collected from other data sets are called out-of-bag data points. The regression function is then evaluated using 2/3 of the data points, and the ones outside the batch are used to validate the model. The process continues until the required accuracy is achieved [10].

RFR is a built-in process that uses out-of-bag data points for validation after their elimination as the characteristic of RFR. Finally, it calculates the total error for each expression tree, showing the efficiency and correctness of each expression tree [10].

3 Application of Artificial Intelligence in Rock Characterization

Laval et al. [11] highlighted the feasibility of soft computing methods in solving rock mechanic problems. According to their study, the limited availability of databases is the major problem in developing intelligent models. Baghbani et al. [12] reported that more than one thousand studies implemented soft computing techniques for solving geotechnical engineering problems. According to their study, ANN is the most popular technique for developing intelligent models. Nevertheless, the following subsections deal with a review on the related AI-based predictive model of UCS.

3.1 ANN-Based Models for UCS Prediction

In a study conducted by Momeni et al. [13], an artificial neural network model which is improved with particle swarm optimization was developed for assessing the UCS of granite and limestone. 66 granite and limestone samples were collected from different states in Malaysia for the purpose of conducting an extensive experimental program. In the experimental program, UCS of samples were measured directly and indirectly. In their study, UCS results were set as the outputs of the network, while laboratory results such as point load index test ($I_{S(50)}$), Schmidt hammer rebound number (SRn), and p-wave velocity test (Vp) were set as model inputs. The proposed hybrid model was compared with a conventional ANN in order to determine its prediction performance. An evaluation of the coefficients of determination, R^2, resulting from conventional ANN and PSO-based ANN techniques revealed the superiority of the PSO-based ANN model. In the case of conventional ANN, the value of R^2 was 0.71, while it was 0.97 for the proposed hybrid predictive model. The sensitivity analysis revealed that *Vp* and *SRn* have a slightly larger impact on the predicted UCS value than other parameters.

A paper published by Zakaria et al. [14] presented the application of Support Vector Machine (SVM) algorithm for UCS prediction. An algorithm was developed based on dry density and velocity parameters and tested on a series of rock data sets. 160 rock samples were used to investigate the relationship between the dry density, the sonic velocity, and the UCS using the commercial software RapidMiner Studio. Based on the study's results, it was determined that SVM can predict missing values with an accuracy of 75%.

GFFN (generalized feedforward neural network) and Imperial Competitive Algorithm (ICA) were used in a study conducted by Asheghi et al. [15] to predict the UCS. 197 sets of data were compiled from almost all of Iran's quarries, including rock class, density, porosity, P-wave velocity, point load index, and water absorption. Based on different error criteria and established confusion matrixes, the efficiency and performance of GFFNs and hybrid ICA-GFFNs were compared to multilayer perceptrons

(MLP) and radial basis functions (RBF) neural network models. Multivariate regressions were also conducted. Their hybrid ICA-GFFN showed superior predictability levels with 11.37, 14.27, and 22.74% improvement in correct classification rate over GFFN, RBF, and MLP. A sensitivity analysis determined that P-wave velocity and rock class had the greatest and lowest influence on the predicted UCS.

Using an indirect modeling approach and machine learning algorithms, Barzegar et al. [16] estimated the UCS of travertine rocks to address the limitations of direct measurements. For the prediction of UCS in travertine rocks from the Azarshahr area of northwest Iran, three standalone tree-based machine learning models were developed and compared (random forest (RF), M5 model tree, and multivariate adaptive regression splines (MARS)). An ensemble committee-based ANN model was developed to achieve further accuracy in UCS prediction. The models were constructed and validated using data from 93 travertine core samples, including P-wave velocity (V_p (Km/s)), Schmidt Hammer (R_n), porosity ($n\%$), and point load index (I_s (MPa)). With a ratio of 70:15:15 (train: validate: test), V_p, R_n, $n\%$, and Is data were incorporated into the ensemble tree-based machine learning model. Based on the results of this study, a standalone MARS model outperformed all other standalone tree-based models in predicting UCS. However, the ANN-committee model showed the best performance, with an *r*-value of approximately 0.890 and a root mean square error (RMSE) of 3.80 MPa, indicating that the ensemble model is more accurate in predicting UCS than standalone models. With a limited set of model-designed datasets, the ensemble committee-based model appears to be a practical approach for predicting the UCS of travertine rocks.

The adaptive neuro-fuzzy inference system model was systematically optimized using stochastic fractal search (SFS) algorithms by Jing et al. [17]. Using three hybrid methodologies based on ANFIS, genetic algorithm, differential evolution (DE), and particle swarm optimization, the efficacy of SFS-ANFIS was assessed. Their study proposed that UCS can be predicted using SFS-ANFIS, GA-ANFIS, DE-ANFIS, PSO-ANFIS, and ANFIS models. The freshwater tunnel in Pahang-Selangor in Malaysia was considered for this purpose, and the required sample were collected. Model evaluations were conducted using different metrics, such as coefficient of determination (R^2) and mean absolute error. According to the effectiveness results of SFS-ANFIS, it was found that SFS-ANFIS (with R^2 of 0.981) was able to predict the UCS better than PSO-ANFIS, DE-ANFIS, GA-ANFIS, and ANFIS models.

Using the long short term memory (LSTM), deep neural networks (DNN), K-nearest neighbor (KNN), Gaussian process regression (GPR), support vector regression (SVR), and decision trees (DT), Mahmoodzadeh et al. [18] attempted to predict the UCS of a variety of rock types acquired from almost every quarry location in Iran, including Claystone, Granite, Schist and Sandstone, Travertine, Limestone, Slate, Dolomite, and Marl. The methods were applied to 170 data sets, including porosity (n), Schmidt hammer (S_H), and P-wave velocity (V_p). To conclude, the results of the prediction methods were compared. Fivefold cross-validation was used to assess the performance ability of the applied methods. Results showed that computational intelligence approaches could be used to predict UCS. Overall, the GPR performed the best, with an R^2 of 0.9955 and an RMSE of 0.52169.

Several soft computing techniques were used by Gül et al. [19] for predicting UCS with the aid of experimental tests such as Brazilian tensile strength, ultrasonic wave velocity, and Shore hardness. The utilized AI methods comprises Multilayer Perceptron Neural Networks (MLPNN), M5 Model Trees (M5MT), and Extreme Learning Machines (ELM). 30 sets of data from six Turkish stones were analyzed using soft computing methods. MLPNN, M5MT, and ELM models were compared in terms of their performance using different performance criteria (RMSE, MAE, VAF, R^2, and a10-index). Compared to other methods, MLPNN performed slightly better ($R^2 = 0.9982$, RMSE = 1.3421, MAE = 0.7985, VAF = 99.7409, a10-index = 1). Using MLPNN, M5MT, and ELM methods, Brazilian tensile strength, ultrasonic wave velocity, and Shore hardness were found to have the strongest influence on the predicted UCS values. Based on the results of the modeling studies, it was found that machine learning algorithms can make high-accuracy predictions in materials with heterogeneous structures, such as rock.

To minimize the impact of outliers, Gupta and Natarajan [20] adopted a new machine learning algorithm called density weighted least squares TSVR (PDWLSTSVR) for predicting the UCS of rock samples. KNN distance was used to determine the weights. Additionally, the performance of the model was compared with random forest (RF), extreme learning machine (ELM), least squares support vector regression (LSSVR), and primal least squares twin support vector regression (PLSTSVR). In their study, the results of R^2 values showed that PDWLSTSVR outperformed RF, ELM, LSSVR, and PLSTSVR.

Based on the serpentinization percentage, physical, dynamic, and mechanical characteristics of serpentinites, Moussas and Diamantis [21] developed an ANN model to predict the UCS indirectly. Earlier experimental data from central Greece were used for this purpose, including 32 block samples. The input parameters included effective porosity (n_e), dry unit weight (γ_d), saturated unit weight (γ_s), water absorption (W_a), P- and S-wave velocities (V_p and V_s), point load index (I_{s50}), and Schmidt hammer number (S_m). In order to select the best ANN model and its optimal structure from many candidate configurations, a Monte Carlo analysis was performed in their study. A comparison was made between the ANN-based results and the experimental results. ANN-based models were shown to be very accurate in predicting UCS (>94%) and were exceptionally efficient at classifying material categories (100%). Overall, ANN-based models were found to be feasible in predicting UCS values.

On the other hand, the use of machine learning methods to predict UCS from geophysical logging data has increased significantly over the past few years. Li et al. [22] developed a Group-based Machine Learning (GML) method that performed better compared to the conventional methods. Using the unsupervised learning (*K-means*) method, the data points were classified into groups; after that, machine learning regression models were built for each group. In comparison with conventional non-grouping machine learning models, the GML performed better.

As a result of rock drilling operations, Kumar et al. [23] developed an ANN model to predict the geomechanical properties of sedimentary rock types using dominant frequencies. This study utilized the train and test data collected during core drilling operations in the laboratory to predict rock properties. Approximately 875 experimental drilling operations were used as input parameters, including drill bit spindle speeds (rpm), drill bit penetration rates (mm/min), drill bit diameters (mm), and dominant acoustic frequencies (Hz). Outputs were set to be the uniaxial compressive strength, the Brazilian tensile strength (BTS), the density, and the abrasivity (%). Based on the training and testing data which were associated with the minimum epochs, the resilient backpropagation algorithm had the highest accuracy, with Variance Accounted For (VAF) value of 96–97%, root mean square error of 0.00013771–1.000840687, and mean absolute percentage error of 0.000674671–3.00774799.

Yesiloglu-Gultekin and Gokceoglu [24], implemented AI techniques with the help of simple and non-destructive test results. They used various non-linear and AI-based prediction models for UCS and elasticity modulus (E_i) estimation of basalt. They implemented ANFIS, non-linear multiple regression (NLMR), and ANN for their non-linear prediction. Their dataset included 137 sets of data which were the results of laboratory tests such as unit weight, porosity, sonic velocity, *E*, and UCS. Various metrics were used to assess the performance of the developed models, including coefficient of determination (R^2), VAF, root mean square error (RMSE), and a 20-index. According to their conclusion, ANFIS performed slightly better compared to other models that predicted UCS. However, in the case of predictive models of *E*, the ANN was the most successful prediction tool.

3.2 Tree-Based Models for UCS Prediction

Briševac et al. [25] performed a study for estimating the UCS of mudstone. For modeling purposes, 30 samples of intact rock materials were collected from six Croatian locations. Their study compared four statistical models for estimating the uniaxial compressive strength, including multiple linear regression, regression tree, and two other regression tree models based on bagging and random forests. A number of properties were calculated for the sample, including density, effective porosity, Schmidt rebound hardness, P-wave velocity, and uniaxial compressive strength. The most efficient estimation of uniaxial compressive strength was obtained using random forests in comparison with multiple linear regression and regression trees using cross-validation.

Ghasemi et al. [26] estimated the UCS and *E* of carbonate rocks with the aid of soft computing approaches, known as model trees. As a data learning tool, model trees are easier to use and, most importantly, represent understandable mathematical rules. WEKA software was used to train and test their developed models. The M5P algorithm was used to build and evaluate model trees (UCS and *E* model trees). According to their study, compared to the commonly used soft computing techniques, M5P model trees are easy to use and train, capable of handling many attributes and dimensions, and robust when missing data is present. Furthermore, M5P generates a simple tree structure and meaningful linear models in leaves, so the relationship between inputs and outputs is clearly explained. A first version of the models was developed without pruning, and then a second version was developed with pruning to avoid overfitting. Model trees were trained and tested using data collected in quarries in southwestern Turkey. Input variables of their models included Schmidt hammer, effective porosity, dry unit weight, P-wave velocity, and slake durability index. The models proved to be accurate tools in predicting UCS and E when unpruned, and pruned trees were tested using a variety of statistical indices (RMSE, MAE, VAF, and R^2). Nevertheless, P-wave velocity and slake durability were the essential parameters for UCS and E predictions, according to the pruned model trees.

On the other hand, in another study, the Brittleness Index (BI) was predicted by Samaei et al. [27]. Their study was based on the classification and regression tree (CART) and a non-linear multivariable regression model. 48 sets of data including rock type, UCS, and Brazilian tensile strength were used for model development. It is worth mentioning that the data were obtained from 30 different tunnel projects, most of which were excavated in the U.S. There were three types of rocks: igneous, metamorphic, and sedimentary. Also, in order to obtain the BI, a punch penetration test was conducted. Overall, after a comparative study, it was found that the CART model with $R^2 = 0.94$ was the best predictor and had the lowest error rate. In addition to the CART model, a non-linear multivariable equation with $R^2 = 0.91$ was proposed after the CART model. Based on a sensitivity analysis, it was determined that the rock type had the greatest influence on BI results.

Wang et al. [28] developed a UCS predictive model based on the random forest (FT) algorithm. Based on the coefficient correlation and the their performed analysis, the proper indirect parameters for estimating UCS were determined (Schmidt hammer rebound value (L-type) and ultrasonic P-wave velocity). A total of 2000 sets of data were collected from over 50 references. In addition, to enhance the diversity of the proposed models different kinds of rocks were considered such as granite, tonalite, marble, chalk, basalt, and limestone. Overall, after implementation of RF algorithm, and verifying the RF-based predicted values of UCS with laboratory tests, it was found that the RF-based predictive model was good enough in capturing the UCS of aforementioned rocks. The latter conclusion was based on the values of R^2 (0.89 and 0.90 for training and testing data, respectively).

Shahani et al. [29] developed four gradient boosting machine learning algorithms to predict the UCS of soft sedimentary rocks in Block-IX at the Ar Coalfield, Pakistan, including gradient boosted regression (GBR), Catboost, Light GBM, and Extreme Gradient Boosting (XGBoost). A total of 106 datasets were used in the study, with parameters including wet density, moisture, dry density, and Brazilian tensile strength. According to their results, XGBoost-based model outperformed GBR, Catboost, and Light GBM models. Their conclusion was based on the value of R^2 which was 0.99, the mean absolute error (MAE) value of 0.00062, the MSE value of 0.0000006, and the RMSE value of 0.00079 in training step as well as the R^2 value of 0.99, the MAE value of 0.00054, the MSE value of 0.0000005, and the RMSE value of 0.00069 during testing. Nevertheless, the sensitivity analysis showed that BTS and wet density, ρ_w, are positively correlated. Also, it was concluded that the moisture content and dry density, ρ_d, are negatively correlated with the UCS.

Although several AI methods have been used to model the UCS and E, explainable AI (XAI) had not been considered before Nasiri et al. [30] study. A model with an XAI is not a black box, and it provides humans with a way to understand its approach to solving problems. In his study, Nasiri et al. [30] demonstrated SHAP (Shapley Additive Explanations) as one of the most recent XAI methods for modeling UCS and E. As a result of integrating SHAP with eXtreme gradient boosting (XGBoost), the intercorrelations between rock properties (porosity, point load index, P-wave velocity, and Schmidt hammer rebound number) and UCS could be demonstrated in each individual record, as well as a combination. From the Hajiabad mine, Iran, ten-block samples of Travertine were collected for this study. According to the sensitivity analysis, the P-wave velocity was the most important factor in predicting UCS and E. Based on statistical analysis (coefficient of determination value of 0.99), XGBoost outperformed random forest and support vector regression in predicting UCS and E.

Khan et al. [31] studied the effects of thermal radiation on marble rock's physical, chemical, and mechanical properties and developed a prediction model for UCS using multi-linear regression (MLR), artificial neural networks, random forests (RFs), and k-nearest neighbors (KNN). MLR, ANN, RF, and KNN models were developed using temperature, P-wave velocity, porosity, density, and dynamic Young's modulus as input variables. Moreover, the model's performance was evaluated using the coefficient of determination (R^2) and mean square error (MSE). In MLR and ANNs, UCS had R^2 value of 0.90, while in KNN and RF, the corresponding R^2 values were 0.97.

Based on the GEP technique, Xue [32] developed an empirical model for predicting the UCS of rocks. For the construction of the GEP model, 44 datasets were collected from literature. There were three types of rock in the dataset: granite, schist, and sandstone. In the GEP model, four main parameters were used as inputs: (i) the block punch index (BPI), (ii) the point load strength ($I_{s(50)}$), (iii) the Schmidt rebound hardness (SRH), and (iv) the ultrasonic p-wave velocity (USV). The output parameter was the UCS of rocks. In terms of three statistical indices, four conventional regression models and an ANN model were used to evaluate the developed GEP model. A comparison between the GEP model, the conventional regression models, and the ANN model revealed that the GEP model had the lowest RMSE and the highest R^2 and R values. According to the training data samples, the proposed GEP model had RMSE values of 10.22, R values of 0.9806, and R^2 values of 0.9651. Overall, based to their findings, the GEP-predictive model of UCS is a capable tool in assessing the UCS of rock samples.

4 Summary and Conclusion

This chapter highlighted the workability of soft computing methods in assessing the UCS of rocks. Overall, findings confirmed that AI-based methods can be utilized as a powerful tool in capturing the UCS of rocks. The performed comprehensive review showed that various soft computing techniques comprising ANN, ANFIS, Tree-based methods, PSO, ICA, and GA can be implemented for constructing intelligent-based predictive models of UCS. Based on the conducted review, different input parameters such as rock index tests and petrographic data can be used for developing soft computing-based predictive models of UCS. Nevertheless, there should be a meaningful relationship between UCS and the considered input parameter. Apart from that, it is suggested to utilize input variables which are almost independent of each other.

In rock engineering problems, collecting a large-enough database is a difficult task to be accomplished. Hence, collecting related data from literature is common. In fact, the review study revealed that the soft computing-based models can be developed using small or large databases. Although, most of the reviewed studies in this chapter confirmed the workability of AI- techniques in rock characterization, in most of these studies, a word of caution is required in generalizing their recommended intelligent models. It should be highlighted that the reliability of the soft computing-based techniques is not more than the reliability of the input data. In fact, the quality of the data plays an important role in the reliability of the proposed soft computing-based models. Additionally, the role of input data is of prime importance. If the range of future data is beyond the range of input data, the models cannot be generalized well enough. Apart from that, the size of the dataset is also an important parameter in designing reliable intelligent models. A large database can avoid model overtraining and overfitting.

References

1. Jang JSR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23
2. Armaghani DJ, Asteris PG (2021) A comparative study of ANN and ANFIS models for the prediction of cement-based mortar materials compressive strength. Neural Comput Appl 33. https://doi.org/10.1007/s00521-020-05244-4
3. Garro BA, Vázquez RA (2015) Designing artificial neural networks using particle swarm optimization algorithms. Comput Intell Neurosci 2015. https://doi.org/10.1155/2015/369298
4. Abdullah AM, Usmani RSA, Pillai TR, Marjani M, Hashem IAT (2021) An optimized artificial neural network model using genetic algorithm for prediction of traffic emission concentrations. Int J Adv Comput Sci Appl 12. https://doi.org/10.14569/IJACSA.2021.0120693
5. Ahmadi MA, Ebadi M, Shokrollahi A, Javad Majidi SM (2013) Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir. Appl Soft Comput J 13. https://doi.org/10.1016/j.asoc.2012.10.009
6. Mirjalili S (2019) Genetic algorithm, pp 43–55. https://doi.org/10.1007/978-3-319-93025-1_4.
7. Rahmani M (2008) Particle swarm optimization of artificial neural networks for autonomous robots. Chalmers University of Technology
8. Mahajan R, Kaur G (2013) Neural networks using genetic algorithms. Int J Comput Appl 77. https://doi.org/10.5120/13549-1153
9. Ahmadzadeh E, Lee J, Moon I (2017) Optimized neural network weights and biases using particle swarm optimization algorithm for prediction applications. J Korea Multimedia Soc 20:1406–1420

10. Khan MA, Memon SA, Farooq F, Javed MF, Aslam F, Alyousef R (2021) Compressive strength of Fly-Ash-based geopolymer concrete by gene expression programming and random forest. Adv Civil Eng 2021. https://doi.org/10.1155/2021/6618407
11. Lawal AI, Kwon S (2021) Application of artificial intelligence to rock mechanics: an overview. J Rock Mech Geotech Eng 13(1):248–266
12. Baghbani A, Choudhury T, Costa S, Reiner J (2022) Application of artificial intelligence in geotechnical engineering: a state-of-the-art review. Earth Sci Rev 228:103991
13. Momeni E, Armaghani DJ, Hajihassani M, Amin MFM (2015) Prediction of uniaxial compressive strength of rock samples using hybrid particle swarm optimization-based artificial neural networks. Measurement 60:50–63
14. Zakaria H, Abdullah RA, Ismail AR (2019) Predicting uniaxial compressive strength using Support Vector Machine algorithm. Warta Geologi 45(1):13–16
15. Asheghi R, Abbaszadeh Shahri A, Khorsand Zak M (2019) Prediction of uniaxial compressive strength of different quarried rocks using metaheuristic algorithm. Arab J Sci Eng 44(10):8645–8659
16. Barzegar R, Sattarpour M, Deo R, Fijani E, Adamowski J (2020) An ensemble tree-based machine learning model for predicting the uniaxial compressive strength of travertine rocks. Neural Comput Appl 32(13):9065–9080
17. Jing H, Nikafshan Rad H, Hasanipanah M, Jahed Armaghani D, Qasem SN (2021) Design and implementation of a new tuned hybrid intelligent model to predict the uniaxial compressive strength of the rock using SFS-ANFIS. Eng Comput 37(4):2717–2734
18. Mahmoodzadeh A et al (2021) Artificial intelligence forecasting models of uniaxial compressive strength. Transp Geotech 27:100499
19. Gül E, Ozdemir E, Sarıcı DE (2021) Modeling uniaxial compressive strength of some rocks from Turkey using soft computing techniques. Measurement 171:108781
20. Gupta D, Natarajan N (2021) Prediction of uniaxial compressive strength of rock samples using density weighted least squares twin support vector regression. Neural Comput Appl 33(22):15843–15850
21. Moussas VC, Diamantis K (2021) Predicting uniaxial compressive strength of serpentinites through physical, dynamic and mechanical properties using neural networks. J Rock Mech Geotech Eng 13(1):167–175
22. Li JX et al (2022) UCS prediction by group-based machine learning method
23. Kumar C, Vardhan H, Murthy CS (2022) Artificial neural network for prediction of rock properties using acoustic frequencies recorded during rock drilling operations. Model Earth Syst Environ 8(1):141–161
24. Yesiloglu-Gultekin N, Gokceoglu C (2022) A comparison among some non-linear prediction tools on indirect determination of uniaxial compressive strength and modulus of elasticity of basalt. J Nondestr Eval 41(1):1–24
25. Briševac Z, Špoljarić D, Gulam V (2014) Estimation of uniaxial compressive strength based on regression tree models. Rudarsko-geološko-naftni zbornik 29(1):39–47
26. Ghasemi E, Kalhori H, Bagherpour R, Yagiz S (2018) Model tree approach for predicting uniaxial compressive strength and Young's modulus of carbonate rocks. Bull Eng Geol Env 77(1):331–343
27. Samaei M, Ranjbarnia M, Zare Naghadehi M (2018) Prediction of the rock brittleness index using nonlinear multivariable regression and the CART regression tree. J Civil Environ Eng 48(92):33–40
28. Wang M, Wan W, Zhao Y (2020) Prediction of the uniaxial compressive strength of rocks from simple index tests using a random forest predictive model. Comptes Rendus Mécanique 348(1):3–32
29. Shahani NM, Kamran M, Zheng X, Liu C, Guo X (2021) Application of gradient boosting machine learning algorithms to predict uniaxial compressive strength of soft sedimentary rocks at Thar Coalfield. Adv Civil Eng 2021
30. Nasiri H, Homafar A, Chelgani SC (2021) Prediction of uniaxial compressive strength and modulus of elasticity for Travertine samples using an explainable artificial intelligence. Results Geophys Sci 8:100034

31. Khan NM et al (2022) Application of machine learning and multivariate statistics to predict uniaxial compressive strength and static Young's modulus using physical properties under different thermal conditions. Sustainability 14(16):9901
32. Xue X (2022) A novel model for prediction of uniaxial compressive strength of rocks. Comptes Rendus Mécanique 350(G1):159–170

A Review on the Application of Soft Computing Techniques in Foundation Engineering

Ehsan Momeni, Masoud Samaei, Amin Hashemi, and Mohamad Bagher Dowlatshahi

Abstract Determining footing design parameters is crucial in designing buildings and other geotechnical structures. Bearing capacity and settlement of foundations are two important design parameters that can be determined with the aid of experimental, numerical, and analytical methods. However, estimating the aforementioned parameters using the above-mentioned methods can be difficult, time-consuming, and costly. On the other hand, recently, the importance of soft computing (SC) methods in solving civil engineering problems is underlined in literature. This study sheds some light on the workability of soft computing techniques in the quick prediction of bearing capacity and settlement of foundations. For this reason, in this book chapter, the first famous soft computing techniques are discussed. Subsequently, the recent studies which highlight the successful application of various SC methods for predicting bearing capacity and settlement of different types of foundations include shallow, skirted, and deep foundations. Overall, the reviewed studies suggest the feasibility of SC methods in predicting bearing capacity and settlement of foundations. However, the predicted values for the aforementioned design parameters should be interpreted with caution as some of the proposed predictive models are based on moderately unsound assumptions.

1 Introduction

Foundations are designed to transfer and distribute the superstructure loads into the ground. These structural elements can be divided into shallow (spread), skirted and deep foundations. Foundation is considered shallow if the embedded depth to width ratio of the foundation is less than four. If the aforementioned ratio is bigger than four, it is called deep foundation. On the other hand, skirted foundations are

---

E. Momeni · A. Hashemi · M. B. Dowlatshahi (✉)
Faculty of Engineering, Lorestan University, Khorramabad, Iran
e-mail: dowlatshahi.mb@lu.ac.ir

M. Samaei
Department of Civil, Construction and Environmental Engineering, North Dakota State University (NDSU), Fargo, ND, USA

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*, Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_5

shallow foundations with skirted walls. Skirted foundations are often utilized in soft soils. Nevertheless, bearing capacity and settlement of foundations are two major parameters that play crucial roles in designing civil engineering structures. There are several methods for estimating the aforementioned parameters including analytical, semi-empirical, empirical, and numerical methods. Details on the relevant equations and their derivations are beyond the scope of this chapter and can be found in classic foundation engineering books.

On the other hand, recently the workability of artificial intelligence methods in solving foundation engineering problems has drawn considerable attention. This chapter aimed to highlight the feasibility of artificial intelligence techniques in foundation engineering. For this reason, some of the well-respected artificial intelligence techniques are discussed in Sect. 2. Section 3 of this chapter deals with a comprehensive review on the feasibility of soft computing and simulation-based techniques in assessing the bearing capacity and settlement of shallow, skirted, and deep foundations. Section 4 of this chapter underlines the summary and conclusion remarks.

2 Fundamental Methods

In this section, we intend to discuss some of the fundamental methods, including the Neural network, **A**daptive-**N**etwork-based **F**uzzy **I**nference **S**ystem (ANFIS), and the combination of neural networks with Imperialist Competitive Algorithm (ICA), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO).

2.1 ANFIS Algorithm

ANFIS [1] is an artificial neural network constructed according to the combination of the Takagi–Sugeno fuzzy inference system and a neural network that adapts through learning. It combines fuzzy logic principles and neural networks, allowing us to leverage both in a single framework. Multiple IF–THEN rules and a learning function are utilized in the ANFIS inference system for approximating non-linear functions. Therefore, we can consider ANFIS a universal estimator [2].

Let us assume a fuzzy inference system with two Takagi–Sugeno type rules for simplicity

$$Rule\,1 \rightarrow x\,is\,A_1\,and\,y\,is\,B_1,\ then\ f_1 = p_1 + p_1 y + r_1$$

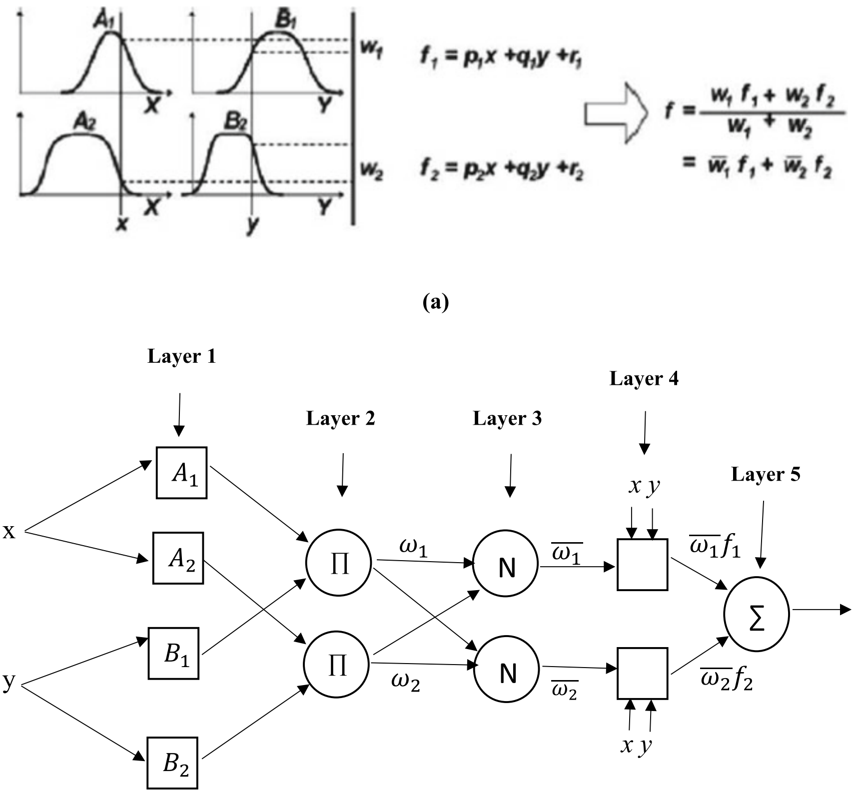
$$Rule\,2 \rightarrow x\,is\,A_2\,and\,y\,is\,B_2,\ then\ f_2 = p_2 + p_2 y + r_2$$

The reasoning mechanism of ANFIS and its architecture are presented in parts (a) and (b) of Fig. 1, respectively. It can be noted that the node's operation belongs to the same functional family for each ANFIS level [3].

Layer 1: There is a set of linguistic labels for first-layer nodes, each associated with one node. Also, the membership value of these labels is considered the output of the nodes. The node's parameters can affect the membership values. As an example, the *i*-th node function is formulated as.

$$O_i^1 = \mu_{A_i}(x) = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{bi}} \tag{1}$$

where the linguistic label is shown by A_i and the parameter set is $\{a_i, b_i, c_i\}$. This layer's parameters are named premise parameters.



(a)

(b)

Fig. 1 Fuzzy model (**a**), ANFIS architecture (**b**)

Layer 2: The nodes in the second layer compute the effectiveness of every rule

$$O_i^2 = \omega_i = \mu_{A_i}(x) \times \mu_{B_i}(y), i = 1, 2. \tag{2}$$

Layer 3: The percentage of firepower of each rule to the sum of all firepowers is computed based on its corresponding node in the third layer

$$O_i^3 = \overline{\omega_i} = \frac{\omega_i}{\omega_1 + \omega_2} i = 1, 2. \tag{3}$$

Layer 4: The following node function is associated with node i in this layer

$$O_i^4 = \overline{\omega_i} f_i = \overline{\omega_i}(p_i x + q_i y + r_i) i = 1, 2. \tag{4}$$

Layer 5: The total output of the ANFIS system is computed as the sum of all its inputs. This is done by the only node in the fifth layer

$$Overall\,output = O_i^5 = \sum \overline{\omega_i} f_i = \frac{\sum \omega_i f_i}{\omega_1 + \omega_2}, i = 1, 2. \tag{5}$$

Therefore, an adaptive network (Fig. 1b) is constructed, and it is identical to an inference system (Fig. 1a) based on its function. Thus, we have a fuzzy inference system according to adaptive networks known as ANFIS. To compute the error rate in ANFIS, gradient descent backpropagation is utilized. In this approach, the error rate of every output node is recursively calculated as the derivative of the squared error to the input nodes, the same as neural networks. The overall output f can be represented based on the values of premise parameters based on the presented architecture of Fig. 1 as a linear combination as follows:

$$\begin{aligned} f &= \overline{\omega_1} f_1 + \overline{\omega_2} f_2 = (\overline{\omega_1} x) p_1 + (\overline{\omega_1} y) q_1 + (\overline{\omega_1}) r_1 \\ &= (\overline{\omega_2} x) p_2 + (\overline{\omega_2} y) q_2 + (\overline{\omega_2}) r_2 \end{aligned} \tag{6}$$

The result is a hybrid learning algorithm that aggregates least squares estimation and gradient descent. More specifically, in the preceding step, the processing of the output nodes is conducted in step 4, and the resulting parameters are set based on the least squares and updated by gradient descent as a backward procedure.

2.2 *Artificial Neural Networks*

Artificial **N**eural **N**etworks (ANNs) inspire biological networks as powerful artificial intelligence tools. ANN is an object that imitates the neural network constituting the human brain so that the computer can learn and make decisions like a human. An input layer, a hidden layer or more, and nodes or neurons as numerous simple computational components as an output layer construct an ANN structure. This additionally includes relationships between neurons in consecutive layers through the weights. These weights can change the signal sent from one node to another and increase or decrease the impact of a particular relationship. A weighted input plus one bias from each neuron in the previous layer is received by each hidden layer neuron. The output of neurons is determined by their activation function. An ANN structure is shown in Fig. 3.

$$Y = f\left(\sum_{i=1}^{n} w_i v_i - b\right) \tag{7}$$

In Eq. 7, f refers to the activation function, v_i and w_i are shown the input values and the weights of neurons, respectively. Also, y refers to the network's output, b is the bias, and n indicates the neuron's number in the hidden layers. The performance of the model can be enhanced by updating the network weights during the training phase. ANN's neuron weights determine how the input data affects output data. The primary weights are selected randomly [4].

The network's internal weights are tuned using a learning algorithm. Backpropagation (BP) algorithms are today's most common form of training in ANNs. Also, optimization methods such as GA, PSO, and ICA are practical in optimizing the ANN [4].

Multi-layer Perception is known as the most famous network architecture among many existing ones. In a feedforward neural network, the number of nodes in the input and output layers equals the number of process inputs and outputs, respectively [5]. The optimal network weights are obtained in the training phase, which minimizes the error function. The mean squared error (MSE) function is used as follows for this task:

$$MSE = \frac{1}{2}\sum_{k=1}^{G}\sum_{j=1}^{m}\left[Y_j(k) - T_j(k)\right]^2 \tag{8}$$

where m and G refer to output nodes and the number of training instances, respectively. $T_j(k)$ shows the actual output and $Y_j(k)$ refers to the expected output.

In the rest of this section, we will discuss optimization methods used to tune ANN weights.

2.2.1 Designing ANN Using GA

Darwin's theory of evolution inspired GA to model the survival of the fittest organisms and their genes. In GA, as a population-based algorithm, the possible solutions are represented by chromosomes, and the gens refer to the solution parameters. GA uses the (objective) fitness function to evaluate the objective value of every individual in the population. The preferable solutions are randomly chosen using a selection mechanism (such as a roulette wheel) to improve the weak ones. This mechanism tends to select the best solution because the probability is commensurate to the goodness of the objective value. The GA algorithm steps are initializing the population, selection, crossover, and mutation [6]. We will discuss these steps individually in this section.

Initializing the population: The initial population is often chosen randomly in Ga-based algorithms. This population contains multiple solutions representing individual chromosomes. Every chromosome contains some variables, and each one represents a gene. The main goal of the initialization phase is to broadcast the solutions evenly in the search space to increase the population's diversity and the likelihood of discovering promising areas. The following steps are used to enhance the initial population chromosomes.

Selection: Natural selection is the fundamental motivation for the GA algorithm. The most suitable individuals in nature are more likely to get food and a mate. Thus, in the production of the next generation of the same species, these genes are more participating. Inspired by this simple idea, using the roulette wheel, the GA algorithm can assign and select probabilities to individuals to generate next-generation commensurate to their objective values. Besides the roulette wheel, other strategies, like Boltzmann selection, tournament selection, rank selection, etc., are utilized for the GA algorithm.

Crossover: After selecting individuals, they should be utilized to produce a new generation. Chromosomes in female and male genes mate to form new chromosomes in nature. This is modeled in the GA algorithm by aggregating two solutions (the parent solutions) to produce two new solutions (the child solutions). There are several strategies to use the crossover operator, like single-point, double-point, uniform crossover, etc.

Mutation: In mutation, one or more genes are modified after the generation of the offspring solutions. A low mutation rate is set for GA because the high rate turns GA into a raw random search. Mutation operators maintain population diversity by introducing a new level of randomness. In essence, the mutation process prevents the similarity of solutions and grows the probability of preventing local solutions in GA algorithms. Some popular mutation strategies are power mutation, uniform, and Gaussian.

An initial population is first accidentally created in the GA algorithm. The above three operators improve the population until the stop condition is fulfilled. The best solution is chosen from the final population and is considered the global best for the given problem. Figure 2 presents the flowchart of the GA algorithm. GA algorithm can be used to deal with numerous optimization problems, including optimizing the weights of an ANN. For this matter, different random values are set for the ANN as the initial population of GA to reach the best values. To understand better, let us consider a simple ANN with two inputs and two hidden layers in Fig. 3. All weights in the

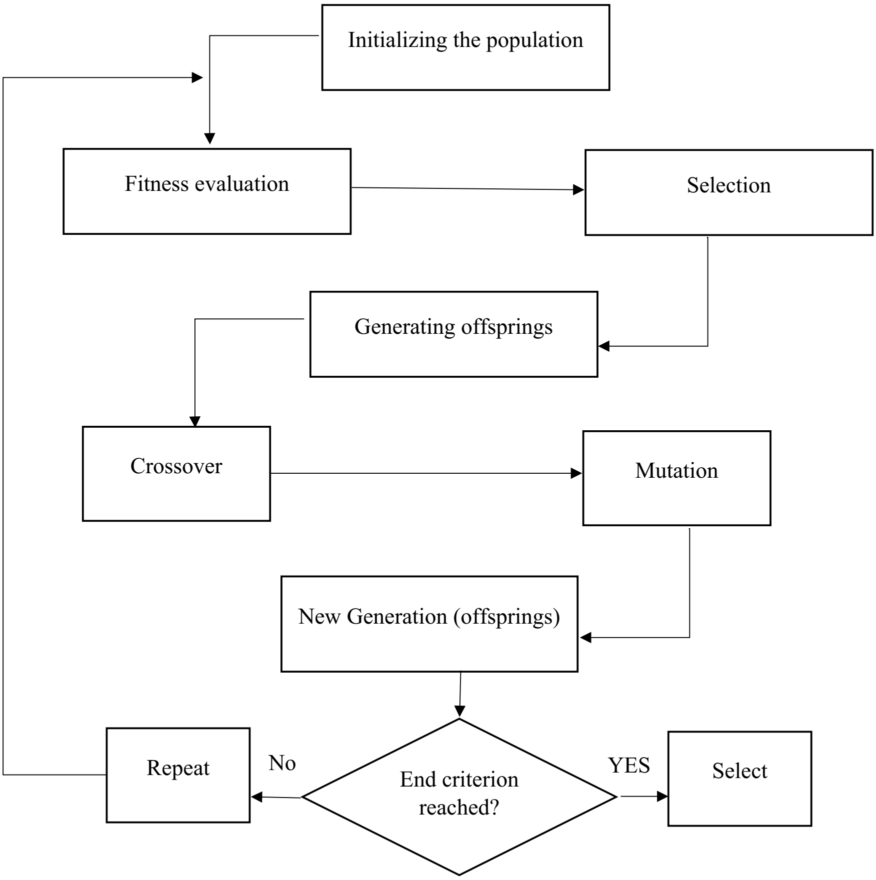


Fig. 2 The flowchart of GA steps

network are combined into one chain. GA then uses this row as a chromosome. Each chromosome represents the weight of the entire network. In Fig. 4, a chromosome structure is shown based on the structure in Fig. 3. Thus, multiple chromosomes are generated with random values as the initial population of GA based on the population size. The fitness function is also presented in Eq. 8. To evaluate the individuals, they are fed into the network to predict the output. Then based on Eq. 8, the fitness evaluation is performed. At last, the based weights are determined for the network as a result of GA [7].

2.2.2 Designing ANN Using PSO

PSO is widely used in asymmetric optimization problems. This algorithm is motivated by the social behavior of bird flocks and fish aggregation populations. This

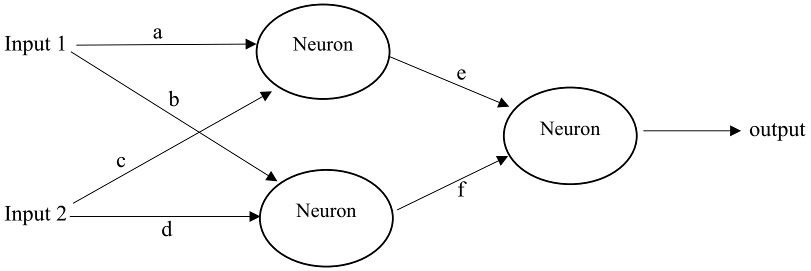


Fig. 3 A simple ANN structure

Fig. 4 A chromosome (A particle or country)

a	b	c	d	e	f

algorithm produces an initial population as a series of random solutions called particles, like a flock of birds. These particles can freely investigate an N-dimensional search space. Every flock member is a solution to the optimization problem, and each particle can be evaluated based on a fitness value. There are two essential parameters in PSO. P.best represents the best fitness value of each particle so far. At the same time, G.best is the fitness value among all particles. The position and speed of all particles are updated by iterating the algorithm (iterations are like a flight) until the algorithm converges to the optimal solution. The following i.e. (Eq. 9) updates the particle speed:

$$v_{id}^{(t+1)} = wv_{id}^{(t)} + u[0,1]\varphi_1\left(p_{id}^{(t)} + x_{id}^{(t)}\right) + u[0,1]\varphi_2\left(p_{id}^{(t)} + x_{id}^{(t)}\right) \tag{9}$$

where t is the iteration index, $u[0, 1]$ is a uniform random distribution of particles, and φ_1 and φ_2 indicate the impact rate of the local and global optima on the total speed of particles. Particles' position are updated according to Eq. 10 as follows

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t)} \tag{10}$$

The PSO algorithm first generates a random population of articles using random values for particle positions and speeds. The position and velocity of articles are improved by the above equations until reaching the desired results. At first, the fitness value is determined for all particles to discover G.best. Then all particles tend to be the G.best since it is closer to the optimum solution. At last, the final G.best represents the solution to the problem of interest [8, 9]. Figure 5 presents the flowchart of the GA algorithm.

PSO algorithm can be used in solving many optimization problems, including optimizing the weights of an ANN. For this matter, different random values are set for the ANN as the initial particles of PSO. To understand better, let us consider a simple ANN with two inputs and two hidden layers in Fig. 3. All weights in the

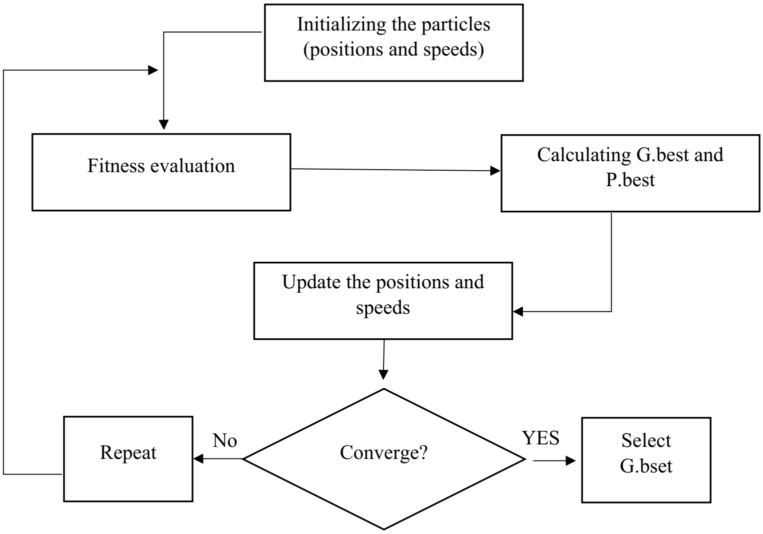


Fig. 5 The flowchart of PSO steps

network are combined into one chain. PSO then uses this row as a particle. Each particle represents the weight of the entire network. In Fig. 4, a particle structure is shown based on the structure in Fig. 3. Thus, multiple particles are generated with random values as the initial positions of PSO particles based on the population size. The fitness function is also presented in Eq. 7. To evaluate the particles, they are fed into the network to predict the output. Then based on Eq. 7, the fitness evaluation is performed. At last, the based weights are determined for the network as the G.best in PSO. It can be noted that the particle's initial speeds are set randomly in the range $[-1, 1]$.

2.2.3 Designing ANN Using ICA

ICA can solve optimization problems based on the inspiration of human socio-political evolution. Starting with the initial population (countries of the world), the best countries (lower cost) are chosen as imperialist countries, and the others construct the colonies. In ICA, every country corresponds with a $1 \times n$ array for an n-dimensional optimization problem that shows the values of that country. Initialization begins with several random solutions as the countries and then sorting them in ascending order according to the cost function values. The imperialists are some lower-cost solutions, and the rest are considered colonies. Then, the colony is distributed among the imperialists in proportion to the imperialist power obtained from the normalized imperialist costs. Assimilation is the colony's movement toward the imperialist, constructed by the vector from colonial to imperialist. In an imperialist race, several fragile colonies (usually one) of the weakest empire will belong to the mightiest empire. If an empire has no remaining colonies, it will fall. The algorithm will repeat until all but the most powerful (cheapest) empire fall. The imperialism of this empire is the solution to the optimization problem [10, 11]. Figure 6 presents the flowchart of the ICA algorithm.

The ICA algorithm can solve many optimization problems, including optimizing the weights of an ANN. For this matter, different random values are set for the ANN as the ICA's initial population (countries). To understand better, let us consider the structure in Fig. 3. All weights in the network are combined into one chain. ICA then uses this row as a country. Each country represents the weight of the entire network. In Fig. 4, a country structure is shown based on the structure in Fig. 3. Thus, multiple countries are generated with random values as the initial population of ICA based on the population size. The fitness function is also presented in Eq. 7. To evaluate the individuals, they are fed into the network to predict the output. Then based on Eq. 7, the fitness evaluation is performed. At last, the based weights are determined for the network as the result of ICA.

3 Application of Artificial Intelligence in Foundation Engineering

Baghbani et al. [12] highlighted the workability of artificial intelligence techniques in geotechnical engineering. They mentioned more than one thousand studies implemented AI techniques for solving geotechnical engineering problems. According to their study, the artificial neural network technique is the most widely used technique in solving geotechnical problems. The effectiveness of the ANNs in robust designing of foundations is underlined in another study [13]. Shahin et al. [14] were among the earliest scholars who attempted to use artificial intelligence methods in order to predict the settlement of shallow foundations. In their study, the predicted settlements obtained by artificial neural networks (ANNs) are compared with estimated settlements using conventional method. According to their results, ANNs are effective techniques for the prediction of settlement of shallow foundations. The following subsections deal with a review on the related AI-based predictive model of either bearing capacity or settlement of various type of foundations.

3.1 ANN-based Predictive Models of Bearing Capacity for Shallow Foundations

To estimate the bearing capacity of strip footings on reinforced non-cohesive soils, Soleimanbeigi and Hataf [15] used two different types of ANNs, feedforward back-propagation (BP) and radial basis function (RBF). Their final model was developed and validated using 351 laboratory and field measurements of footing load tests on

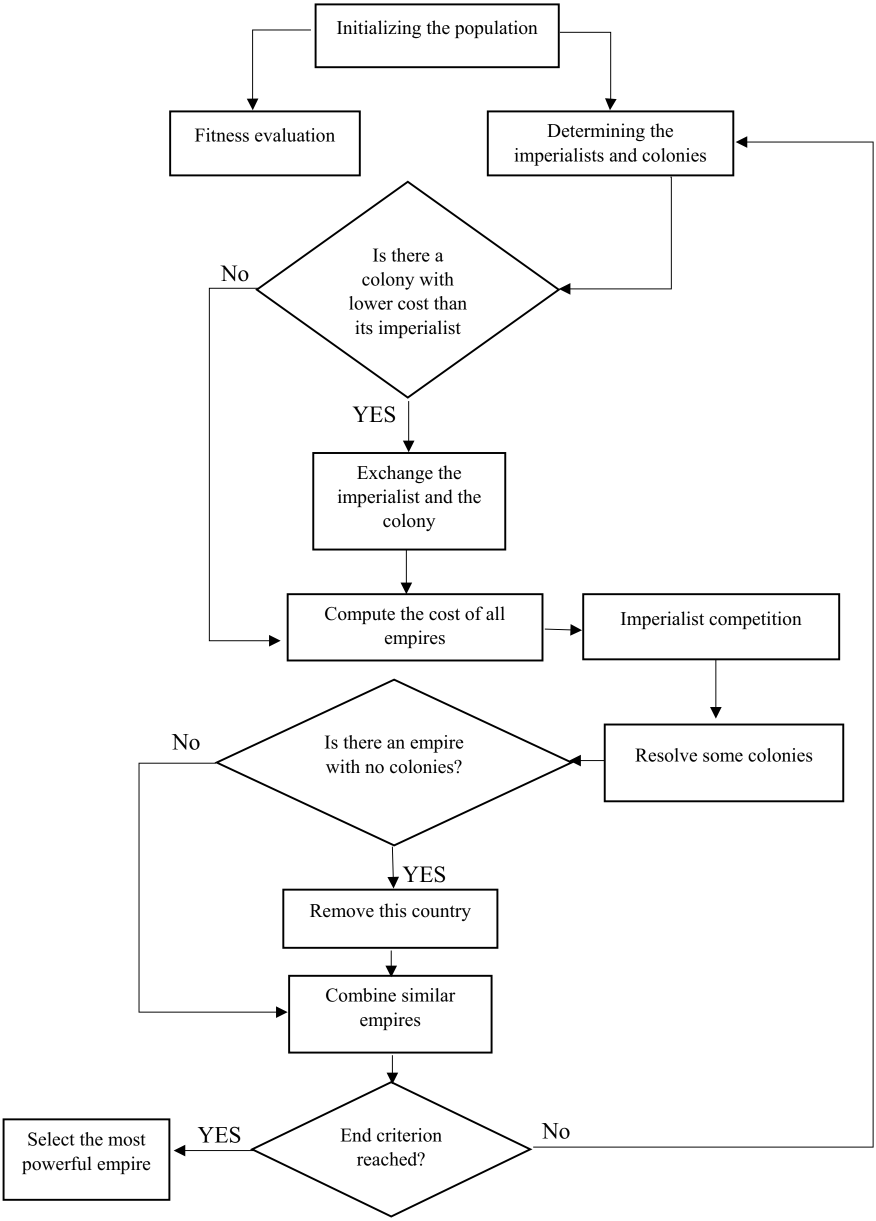


Fig. 6 The flowchart of ICA steps

reinforced cohesionless soils. According to them, theoretical methods often simplify the problem by incorporating several assumptions. Overall, their conclusion showed that the bearing capacity can be predicted quickly with the aid of their proposed model. Padmini et al. [16] examined the potential of neuro-fuzzy computing to predict the ultimate bearing capacity of shallow foundations on cohesionless soils. Neuro-fuzzy models integrate fuzzy inference systems (FIS) with Artificial Neural Networks for learning. Nevertheless, model calibration and testing were performed by using data from 97 load tests on footings. Using the same data, independent fuzzy and ANN models were comprehensively evaluated against the neuro-fuzzy model's performance. According to their results, the ANFIS model showed better prediction performance in comparison with the ANN and FIS models. Ornek et al. [17] analyzed the bearing capacity of circular shallow footings supported by compacted granular fill over natural clay soil using ANNs and multi-linear regression models (MLRs). Field tests were conducted with seven different diameters of footings, up to 0.90 m, as well as three different granular fill thicknesses. For predicting the bearing capacity of circular footings on stabilized natural clay soil, the ANN model proved to be an effective and straightforward tool.

Marto et al. [18] proposed a unified approach based on the particle swarm optimization algorithm to overcome the ANN's disadvantages such as getting trapped in local minima and the slow rate of learning. Models were built using 40 sets of data including full-scale axial compression load tests on shallow foundations with granular soils. To evaluate the ultimate axial bearing capacity, parameters such as footing width and length, embedded footing depth, soil friction angle, and groundwater level were considered. Coefficients of determination (R^2) were used to evaluate model performance. Using the PSO-based ANN model, the predicted bearing capacities were in excellent agreement with the measured bearing capacities with R^2 values equal to 0.997 and 0.991 for training and testing datasets, respectively. Based on the results, it was concluded that the PSO-based ANN predictive models can be used to predict shallow foundation bearing capacities in a feasible and accurate manner.

Researchers at Sharjah, United Arab Emirates, used artificial neural networks to predict shallow foundation bearing capacity and elastic settlement on granular soil [19]. A total of 600 borehole reports were used for training and validating their proposed model. A forward sequential feature selection algorithm [20] is used to choose the subset of parameters that influence output accuracy the most. The factors considered to have the greatest influence on the permissible bearing capacity and elastic settling of strip footings were effective unit weight, foundation width, and SPT blow count. Meanwhile, recent literature reviews have found similar results. Several studies have shown that, for example, Soleimanbeigi and Hataf [15], Padmini et al. [16], and Nazir et al. [21] have demonstrated that a foundation's ability to withstand external loads is significantly influenced by the foundation shape, angle of friction, and unit weight of sand. According to Shahin et al. [14], Nazir et al. [22], and Erzin and Gul [23], the three main and important input variables that have a substantial impact on the elastic settlement of strip footings in granular soils are the SPT blow count, net applied pressure, and geometrical features. SPT is one of the most frequently applied tests to gauge the compressibility of non-cohesive soils, despite not being the most precise in situ test for doing so [19]. The created ANN models may be utilized successfully to estimate the permissible bearing capacity and elastic settlement, according to comparisons of the models' coefficients of determination, root mean square error, and mean absolute error. The proposed models can thus take the place of traditional techniques for calculating shallow foundation bearing capacity and settlements on granular soils in the initial stages [19].

ANN and multivariable regression analysis (MRA) were used by Gnananandarao et al. [24] to forecast the load carrying capacity and settlement of multi-edged skirted footings foundations on sand. They compared the settlement reduction factor (SRF), which is a proportion of the difference between the settlements of unskirted and skirted footings at a fixed pressure, and the bearing capacity ratio (BCR) of skirted over unskirted foundations to assess these parameters. In the case of BCR prediction, two input parameters were used: the angle of internal friction (ϕ) and skirt depth (Ds) to footing width (B). For SRF prediction, one additional input parameter was taken into account: normal stress (σ). The proposed ANN models' finest architectures were 2-2-1 for the BCR and 3-2-1 for the SRF. ANN models for the multi-edged skirted footings had coefficients of determination of 0.940–0.977, and regression models had coefficients of determination of 0.827–0.934. Similarly, the R^2 for SRF prediction was between 0.913–0.985 for ANN and 0.739–0.932 for regression analysis. Overall, in comparison with MRA, ANN worked better. The findings of the sensitivity study also showed that the skirt depth, followed by the sand's friction angle, has the biggest impact on the multi-edged skirted footings' BCR and SRF.

The study by Pham et al. [25] aimed to predict the maximum bearing capacity of shallow foundations in sandy soil using a hybrid model built using Random Search (RS) and Deep Neural Networks (DNN). 97 sets of data were used in their model development process which comprised foundation width (m), foundation depth (D), foundation length ratio (L/B), and unit weight of soil (kN/m^3). As a result of hyperparameter tuning progress, the RS-DNN model with two hidden layers performed the best. As a result of the sensitivity analysis, they found that foundation width (B) and foundation depth (D) play a significant role in predicting the bearing capacity of shallow foundations. To evaluate the performance of the DNN model, R-squared (R^2), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Variance Accounted For (VAF) were calculated. Based on R^2 value of 0.988, RMSE value of 69.563 kPa, MAE value of 45.667 kPa, and VAF value of 99.938% for testing data, the RS-DNN model was found to be more efficient compared to the DNN model with randomly chosen hyperparameters or the traditional MLP model.

3.2 *Tree-Based Predictive Models of Bearing Capacity for Shallow Foundations*

Artificial intelligent systems, such as gene expression programing has been used to solve several geotechnical problems including the axial bearing capacity of piles, soil deformation moduli, rock tensile and compressive strength, and settlement induced by tunnelling. In this regard, a shallow foundation on granular soil was predicted with the M5' Model Tree by Khorrami et al. [26]. 169 experimental data were used for creating this model. The input parameter of their proposed model were foundation width, foundation length, embedment depth, internal friction angle (φ), and specific gravity. It was observed that the developed Tree Model is able to model actual and predicted values accurately due to its high correlation coefficient and low MAE and RMSE errors. Based on the sensitivity analysis, soil friction angle (φ) was the most important parameter.

3.3 AI-Model for Skirted Foundations

While shallow foundations are recommended in good subsurface conditions, geotechnical engineers are concerned about bearing capacity and settlement in unsatisfactory ground conditions [27, 28]. Hence, a number of studies in the past few years have emphasized the use of skirted shallow foundations or thin-walled spread foundations [29–31]. By using PSO-based ANNs, Rezaei et al. [32] developed a predictive model of bearing capacity for thin-walled spread foundation. Additionally, the model was compared with an ANN improved with GA. To do this, four small-scale footing tests were performed in cohesionless soils in the laboratory. The required dataset for creating a prediction model included the results of the lab experiments plus 145 other similar foundation load experiments that were documented in the literature. Their proposed model's input variables include foundation width, soil friction angle, soil unit weight, and the proportion of thin-wall length to foundation width. Based on the testing dataset, their proposed predictive model was found to be superior based on the values of correlation coefficient (R) and MSE. The R and MSE values were 0.98 and 0.005, respectively. Standard ANN models and GA-based ANN models have correlation coefficients of 0.65 and 0.80, respectively. Momeni et al. [33] also examined the feasibility of predicting thin-walled foundation bearing capacities by using an adaptive neuro-fuzzy inference system. A literature-based data set consisting of nearly 150 documented footing load tests was assembled to address this issue. Inputs to the predictive model of bearing capacity were the strip width, wall length-to-footing width ratio, soil friction angle, and unit weight of soil. Furthermore, the bearing capacity of thin-walled foundations was estimated using a pre-developed artificial neural network (ANN) for comparison purposes. According to these findings, ANFIS can be used as a highly effective technique for predicting the bearing capacity of thin-walled footings. In estimating the bearing capacity of thin-walled spread footings, ANFIS had a higher precision and performance levels in comparison with the ANN model ($R^2 = 0.71$, RMSE $= 0.51$ for the training dataset, and $R^2 = 0.420$, RMSE $= 0.529$ for the testing dataset). Based on their findings, the researchers concluded that ANFIS can be used to predict thin-walled spread foundation bearing capacity in a feasible and quick manner.

In order to predict the bearing capacity of thin-walled foundations, Jahed Armaghani et al. [34] developed a hybrid intelligent technique based on the ANFIS-polynomial neural network (PNN) and the GA called ANFIS-PNN-GA. The GA was actually used to optimize the ANFIS–PNN structure in ANFIS–PNN–GA system. Model performance was evaluated using several performance indices, including correlation coefficient (R) and mean square error (MSE). For the testing and training sets of the ANFIS, PNN, and ANFIS-PNN-GA models, R values of 0.9825, 0.9071, and 0.9928; 0.8630, 0.7595, and 0.9241, respectively, were achieved.

3.4 ANN-Based Predictive Models of Bearing Capacity for Piles

Many researchers used soft computing techniques and artificial neural networks in order to estimate the load-bearing capacity of piles based on both static and dynamic data sets [35–39]. In a study by Pal and Deswal [40], AI-based models were used to predict the ultimate bearing capacity of concrete spun pipe piles. They used stress-wave data and pile geometrical properties as their dataset. It was found that ANN performed better in comparison with support vector machine in predicting pile bearing capacity. Their proposed predictive model is highly reliable due to the high coefficient of determination, i.e., $R^2 = 0.98$.

ANN-based predictive models of pile bearing capacity were enhanced by Momeni et al. [41] using a genetic algorithm (GA). In order to develop the model, 50 Pile Driving Analyzer (PDA) tests were conducted on precast prestressed concrete piles. For the GA-based ANN predictive model, pile set, pile cross-section area, pile length, hammer weight, and drop height were used as inputs. A model built with 8 hidden nodes in one hidden layer using the optimum GA parameters and after trial-and-error proved to be highly reliable in predicting the bearing capacity of piles. The GA parameters including single-point crossover, mutation, and recombination probability were 90, 1, 9, respectively. Based on the R^2 value of 0.990, the developed model could be used to predict pile foundation bearing capacities; therefore, it appears to be a viable and efficient tool. A GA-based ANN model shows superior performance in predicting pile bearing capacity in comparison with the utilized conventional ANN model. Based on the sensitivity analysis results, pile geometrical properties and hammer weight have the most significant influence on the GA-based ANN predictive model for pile bearing capacity.

According to Momeni et al. [42] study, 36 PDA tests have been performed on concrete piles of various diameters and lengths to develop an ANN-based predictive model. Most of the tests were conducted on cohesionless soils. In order to construct the model, they incorporated PDA results, pile lengths, cross-sectional areas, and average SPT (N) values along the pile shaft and tip. The output parameters of their proposed model were the tip and skin resistances of piles. After testing several hidden nodes in one hidden layer, they found that a network with five hidden nodes performed best. Test results demonstrated that the suggested ANN models had R^2 values of 0.941, 0.936, and 0.951, respectively, for estimating the shaft, tip, and ultimate bearing capacities of piles. Furthermore, the pile length and cross-sectional area were found to be the most crucial parameters for determining the bearing capacity.

As outlined in Moayedi and Jahed Armaghani [43] article, the objective of their study was to introduce and evaluate an ANN model which is optimized with an Imperialist Competitive Algorithm (ICA) model in order to estimate the bearing capacity of driven piles in cohesionless soils. It has been determined that the in situ studies were used as training data for the optimization of the ICA-ANN structure. For the development of the ICA-ANN model, the authors used 55 input parameters, which included the internal friction angles of soil along the shafts and tips of the piles, pile lengths, effective vertical stresses, and pile areas. The output of their model was the total bearing capacity of driven piles in cohesionless soils. To demonstrate the capability of the hybrid model, the predicted results were compared with a pre-developed ANN model. ANN and ICA-ANN models, respectively, were found to provide correlation R^2 values of 0.885, 0.894, and 0.964, 0.974 for testing and training data, respectively. Moreover, the ANN and ICA-ANN algorithms yielded variance accounts for (VAF) values of (88.212 for training and 89.215 for testing) and (96.369 for training and 97.369 for testing).

A number of intelligent techniques were developed in a study of Shaik et al. [44] to predict pile bearing capacity in cohesionless soil. Two hybrid ANN models, ICA-ANN and ANFIS, were created to show how FIS systems and ICA affect pre-built ANNs. According to the performance indices, the best technique among these techniques was selected. It is worth mentioning that the model inputs were the effective vertical tension at the pile toe, the pile area, the internal friction angle of the soil placed in the column and tip, and the pile length. In the testing and training datasets for ANN, ICA-ANN, and ANFIS models, the coefficient of determination (R^2) values were (0.895, 0.905), (0.945, 0.958), and (0.967, 0.975). The findings showed that both hybrid models can estimate bearing capacity with high levels of accuracy; however, depending on the performance indices employed, the ANFIS-based model was discovered to be more appropriate.

Harandizadeh et al. [45] developed new AI algorithms to predict pile bearing capacity in their research. ANFIS and the group method of data handling (GMDH) structure were integrated in the first model, and the ANFIS-GMDH-PSO model also was refined using the particle swarm optimization (PSO) methodology. A fuzzy polynomial neural network-based group data handling method (FPNN-GMDH) is employed in the second model. The database included pile properties and soil characteristics collected from literature and used in model training and testing. An ANN model was also used as a reference model for comparison purposes. It was found that two models can be employed as new hybrid soft computing tools that display acceptable precision in the field of geotechnical engineering. The new alternative approaches might be a good alternative to in situ field experiments and semi-empirical regression-based equation methods for assessing ultimate pile bearing capacity, which are expensive, time-consuming, unreliable, and uncertain when executive conditions are complex. It was determined that the upgraded ANFIS-GMDH model outperformed the ANN and FPNN-GMDH models when it comes to accuracy and reliability. The aforementioned conclusion was based on common statistical performance indicators like the correlation coefficient (R), mean square error, root mean square error, and error standard deviation.

Harandizadeh [46] investigated the use of cone penetration test (CPT) data to estimate ultimate axial pile bearing capacity (UPBC) in geotechnical engineering applications using hybrid artificial intelligence techniques. They prepared 108 sets of data in order to develop a new hybrid ANFIS network structure. They combined GMDH with ANFIS and they optimized the new hybrid system with PSO. To construct the proposed system, they used input parameters such as the cross-section of the pile toe, the average cone tip resistance along the embedded pile length, and the sleeve frictional resistance along the shaft. Compared to the conventional methods proposed by Schmertmann [47], De Kuiter and Bringen [48], and LPC/LPCT methods [49], their developed ANFIS-GMDH-PSO model predicted the UPBC with acceptable precision. Additionally, CPT-based models were compared to the ANFIS-GMDH-PSO models in terms of statistical criteria. The results of the statistical analysis showed that the ANFIS-GMDH-PSO model outperformed the aforementioned CPT-based empirical methods in predicting pile ultimate bearing capacity.

To estimate the axial bearing capacity of driven piles, Harandizadeh [45] combined neural-fuzzy (NF) with the group method of data handling (GMDH). Additionally, metaheuristic techniques such as particle swarm optimization (PSO) and gravitational search algorithm (GSA) were used to optimize the design of the hybrid (NF-GMDH) network. The input parameters of the predictive model of pile bearing capacity comprises flap number, soil properties, pile geometrical characteristics, and internal friction angles of the pile-soil interface. Based on the sensitivity analysis results, it was concluded that the flap number plays the most significant role in predicting bearing capacity. According to the results, combining the NF-GMDH model structure with the PSO algorithm improved the model performance. The achieved RMSE value of 1375 showed higher levels of accuracy in predicting ultimate pile bearing capacity in comparison with the GSA NF-GMDH model with RMSE value of 1740.7. Additionally, findings of their study also suggested that NF-GMDH networks developed in their study outperformed gene programming and linear regression models.

Momeni et al. [50] introduced the Gaussian process regression (GPR) method in another study to evaluate the pile carrying capacity proposed GPR technique for assessing the pile bearing capacity. The database contains 296 dynamic pile load tests in the field, with input factors such as, pile set, ram weight, pile diameter, drop height of hammer, and pile length. The pile bearing capacity was forecasted using four different covariance types of GPR: squared exponential, exponential, Matérn 5/2, and rational quadratic. Three statistical performance predictions were implemented, including VAF, R^2 and system error. According to the developed method, VAF, R^2, and system error were ranged from 84.07 to 86.41, 0.83 to 0.84, and 0.2006 to 0.2063, respectively for GPR models. They found that among all the covariance types, a rational quadratic model with a VAF value of 86.41%, an R^2 of 0.84, and a system error of 0.2006 showed the best overall performance. Overall, their results showed that all models capture the pile bearing capacity good enough.

Using a new prediction method, Dehghanbanadaki et al. [51] proposed an improved method in order to calculate the single driven piles' ultimate bearing capacity (UBC). The effectiveness of the multilayer perception (MLP) and adaptive neuro-fuzzy inference system (ANFIS) models was improved with the aid of gray wolf optimization (GWO) technique to evaluate the UBC. A total of 100 driven piles were used for training the model. UBC was calculated using the input parameters such as pile cross-sectional area (m^2), pile length (m), flap number, average cohesion (kN/m^2) and friction angle (°), average soil specific weight (kN/m^2), and average pile–soil friction angle (°). They observed that while ANFIS and MLP had good UBC model's performance for piles, the MLP-GWO model produced superior outcomes. According to the test data, the model had RMSE value of 1.86 and R^2 of 0.991. An experimental dataset was used to validate the MLP-GWO model, and the difference between estimated and actual UBC was just 2%, confirming the model's high accuracy.

Pham and Vu [52] study centered on developing a machine learning algorithm known as Ensemble Learning (EL), which used weight voting (WV) and average voting (AV) protocols of three base machine learning algorithms, gradient boosting (GB), random forests (RF), and classic linear regression (LR) to predict the pile's bearing capacity. For training and testing, 108 pile load tests were used. R-square (R^2), RMSE, and MAE were used to evaluate the models' performance in predicting pile bearing capacity. As compared to the base models, AV-EL and WV-EL showed superior performance. Among all models, the WV-EL model achieved the best performance and achieved the best balance.

Hoang et al. [53] proposed a data-driven approach that combines machine learning and metaheuristics to address the bearing capacity of piles. Their analysis was performed using least squares support vector regression (LSSVR). To further enhance the LSSVR model, opposition-based differential flower pollination (ODFP) metaheuristics were also developed. Regarding the mean absolute percentage error, mean absolute error, and coefficient of determination, the empirical findings demonstrated that the ODFP-optimized LSSVR performs good enough in capturing the bearing capacity of piles.

3.5 Tree-Based Predictive Models of Bearing Capacity for Piles

In a study, Pham et al. [54] introduced ANNs and random forests (RFs) for predicting the ultimate axial bearing capacity of driven piles. A database containing the 2314 recorded cases of static load tests was used for their model development. The input parameters of their model include the pile diameter, length of pile segments, natural ground elevation, pile top elevation, pile tip elevation, average standard penetration test (SPT) value along the embedded length of pile, and average SPT blow counts at the tip of pile. Their model output was the ultimate load at pile's tip. The prediction performance of the developed model was checked against five empirical equations and a multivariate regression model. The comparative study showed that in general the RF-based predictive model performs good enough. In a sensitivity analysis, it was revealed that the pile tip elevation and average SPT value were the essential factors in predicting axial bearing capacity.

A study by Huat et al. [55] proposed a series of tree-based forecasting techniques for pile bearing capacity estimation. They introduced a process for choosing the most important parameters. To predict pile friction bearing capacity and selection of influential parameters, decision tree (DT), random forest (RF), and gradient boosted tree (GBT) models were developed. 130 dynamic high strain load tests in a Malaysian town were used to create the models. The input variables included pile diameter, hammer drop height, hammer weight, pile length, and N values of the standard penetration. Results showed that hammer drop height, pile length, and the average SPT-N value were the most effective input variables. Overall, the values of coefficients of determination for training and testing data (0.901 and 0.816) recommend the feasibility of their proposed predictive model.

Amjad et al. [56] studied the feasibility of extreme gradient boosting (XGBoost) model for predicting pile bearing capacity. They used 200 recorded cases of static load tests for their model development. The input parameters of their model were diameter of the pile (D), the depth of soil embedded in the pile (X1), the depth of soil embedded in the second layer of soil (X2), the depth of soil embedded in the third layer of soil (X3), the pile top elevation (Xp), the ground elevation (Xg), the extra pile top elevation (Xt), the pile tip elevation (Xm), the blow count at the pile shaft (NS), and the SPT. The proposed XGBoost model was compared to a number of widely used algorithms such as Adaptive Boosting (AdaBoost), Random Forest, Decision Tree, and Support Vector Machines using different performance indicator metrics such as coefficient of determination, mean absolute error, Nash–Sutcliffe model efficiency coefficient, root mean square error, mean absolute relative error, and relative strength ratio. Additionally, sensitivity analysis was conducted to determine the impact of input parameters on bearing capacity. Overall, it was found that all predictive models can predict the output of interest. However, results showed that XGBoost model outperformed other models. ($R^2 = 0.955$, MAE = 59.92, RMSE = 80.653, MARE = 6.6, NSE = 0.950, and RSR = 0.215). It should be mentioned that the blow counts along the pile shaft had the biggest impact on pile bearing capacity, according the findings of the sensitivity analysis.

3.6 AI-Based Predictive Models of Settlement for Piles

In a study by Armaghani et al. [57], a neural network optimized by particle swarm optimization (neuro-swarm) was used to estimate pile settlements. In order to develop neuro-swarm models, they used datasets of several piles that were socketed into rock masses. In order to determine the most influential parameter on pile settlement, several sensitivity analyses were conducted. At last, five neuro-swarm models were constructed in order to assess the workability of the hybrid model in predicting pile settlement. In the train and test stages, the coefficient of determination (R^2) and system error values for the best neuro-swarm model were (0.851 and 0.079) and (0.892 and 0.099), respectively, indicating this hybrid model is a capable tool in predicting pile settlement.

Armaghani et al. [58] developed a gene expression programming (GEP)-based predictive model of settlement for rock-socketed piles in the Klang Valley Mass Rapid Transit project, Malaysia and compared the results with the MLR-based predictive models. In order to develop new equations using GEP and MLR techniques, 96 datasets were collected. To predict pile settlement, they used the following factors: Ratio between soil layer length and rock layer length, Ratio between total length of pile and its diameter, uniaxial compressive strength, Standard penetration test, and ultimate bearing capacity. In this regard, five GEP equations and five MLR equations were proposed. Based on the new predictive equation, the GEP model showed a higher reliability for predicting the settlement of rock-socketed piles.

4 Summary and Conclusion

This chapter underlined the feasibility of AI techniques in solving foundation engineering problems. Overall, findings showed that soft computing techniques can be implemented as a quick and feasible tool in predicting bearing capacity and settlement of various type of foundations including shallow, skirted, and deep foundations. A comprehensive review showed that different AI techniques such as ANN, ANFIS, Tree-based methods, PSO, ICA, and GA can be utilized for developing intelligent-based predictive models of either settlement or bearing capacity. Based on the conducted review, various input parameters comprising foundation geometrical properties, soil properties, in situ tests such as standard penetration test and cone penetration test can be implemented for developing AI-based predictive models of either bearing capacity or settlement.

In geotechnical engineering, compiling large sets of relevant experimental data is a difficult task to be accomplished. Hence, compiling related data from literature or generating data using numerical modeling techniques for model construction is unavoidable. In fact, the review study showed that the intelligent models can be constructed using small or large sets of data. Although, most of the highlighted studies in this chapter shed some light on the feasibility of soft computing techniques in foundation engineering, in many of these works, a word of caution is required in generalizing their proposed predictive models. It should be underlined that the reliability of the AI-based techniques is not more than the reliability of the models' feeding data. In other words, quality of the data plays a crucial role in the reliability of the developed AI-based models. Apart from that, the role of feeding data is of prime importance. The models cannot be generalized good enough if the range of future data are beyond the range of feeding data. Additionally, size of dataset is also an influencing parameter on the reliability of the developed models. Large dataset (excluding the outliers) can avoid model overtraining and overfitting, and therefore, can guarantee the reliability of the developed model.

References

1. Jang JSR (1993) ANFIS : Adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 23
2. Walia N, Singh H, Sharma A (2015) ANFIS: adaptive neuro-fuzzy inference system—a survey. Int J Comput Appl 123. https://doi.org/10.5120/ijca2015905635
3. Melin P, Soto J, Castillo O, Soria J (2012) A new approach for time series prediction using ensembles of ANFIS models. Expert Syst Appl 39. https://doi.org/10.1016/j.eswa.2011.09.040
4. Abdullah AM, Usmani RSA, Pillai TR et al (2021) An optimized artificial neural network model using genetic algorithm for prediction of traffic emission concentrations. Int J Adv Comput Sci Appl 12. https://doi.org/10.14569/IJACSA.2021.0120693
5. Ahmadi MA, Ebadi M, Shokrollahi A, Javad Majidi SM (2013) Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir. Appl Soft Comput J 13. https://doi.org/10.1016/j.asoc.2012.10.009
6. Mirjalili S (2019) Genetic algorithm, pp 43–55
7. Mahajan R, Kaur G (2013) Neural networks using genetic algorithms. Int J Comput Appl 77. https://doi.org/10.5120/13549-1153
8. Ahmadzadeh E, Lee J, Moon I (2017) Optimized neural network weights and biases using particle swarm optimization algorithm for prediction applications. J Korea Multimedia Soc 20:1406–1420
9. Garro BA, Vázquez RA (2015) Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms. Comput Intell Neurosci 2015. https://doi.org/10.1155/2015/369298
10. Ahmadi MA (2011) Prediction of asphaltene precipitation using artificial neural network optimized by imperialist competitive algorithm. J Pet Explor Prod Technol 1. https://doi.org/10.1007/s13202-011-0013-7
11. Hasanzade-Inallu A, Zarfam P, Nikoo M (2019) Modified imperialist competitive algorithm-based neural network to determine shear strength of concrete beams reinforced with FRP. J Cent South Univ 26. https://doi.org/10.1007/s11771-019-4243-z
12. Baghbani A, Choudhury T, Costa S, Reiner J (2022) Application of artificial intelligence in geotechnical engineering: a state-of-the-art review. Earth Sci Rev 228:103991
13. Khajehzadeh M, Keawsawasvong S, Nehdi ML (2022) Effective hybrid soft computing approach for optimum design of shallow foundations. Sustainability 14(3):1847
14. Shahin MA, Maier HR, Jaksa MB (2002) Predicting settlement of shallow foundations using neural networks. J Geotech Geoenviron Eng 128(9):785–793
15. Soleimanbeigi A, Hataf N (2005) Predicting ultimate bearing capacity of shallow foundations on reinforced cohesionless soils using artificial neural networks. Geosynth Int 12(6):321–332
16. Padmini D, Ilamparuthi K, Sudheer K (2008) Ultimate bearing capacity prediction of shallow foundations on cohesionless soils using neurofuzzy models. Comput Geotech 35(1):33–46
17. Ornek M, Laman M, Demir A, Yildiz A (2012) Prediction of bearing capacity of circular footings on soft clay stabilized with granular soil. Soils Found 52(1):69–80
18. Marto A, Hajihassani M, Momeni E (22014) Bearing capacity of shallow foundation's prediction through hybrid artificial neural networks. In: Applied mechanics and materials, vol 567. Trans Tech Publ., pp 681–686
19. Omar M, Hamad K, Al Suwaidi M, Shanableh A (2018) Developing artificial neural network models to predict allowable bearing capacity and elastic settlement of shallow foundation in Sharjah, United Arab Emirates. Arab J Geosci 11(16):1–11
20. Menshawy ME, Benharref A, Serhani M (2015) An automatic mobile-health based approach for EEG epileptic seizures detection. Expert Syst Appl 42(20):7157–7174

21. Nazir R, Momeni E, Marsono K, Maizir H (2015) An artificial neural network approach for prediction of bearing capacity of spread foundations in sand. Jurnal Teknologi 72(3)
22. Nazir R, Momeni E, Hajihassani M (2014) Prediction of spread foundation's settlement in cohesionless soils using a hybrid particle swarm optimization-based ANN approach. In: International conference on advances in civil, structural and mechanical engineering, London, UK, pp 20–24
23. Erzin Y, Gul TO (2014) The use of neural networks for the prediction of the settlement of one-way footings on cohesionless soils based on standard penetration test. Neural Comput Appl 24(3):891–900
24. Gnananandarao T, Khatri VN, Dutta RK (2020) Bearing capacity and settlement prediction of multi-edge skirted footings resting on sand. Ingeniería e Investigación 40(3):9–21
25. Pham TA, Vu H-LT, Duong H-AT (2021) Improving deep neural network using hyper-parameters tuning in predicting the bearing capacity of shallow foundations. J Appl Sci Eng 25(2):261–273
26. Khorrami R, Derakhshani A, Moayedi H (2020) New explicit formulation for ultimate bearing capacity of shallow foundations on granular soil using M5'model tree. Measurement 163:108032
27. Momeni E, Maizir H, Gofar N, Nazir R (2013) Comparative study on prediction of axial bearing capacity of driven piles in granular materials. Jurnal Teknologi 61(3)
28. Abdolhosseinzadeh A, Samui P, Samaei M, Garousi A (2022) Numerical analysis of bearing capacity of circular footing reinforced with geogrid layers. Arab J Geosci 15(8):1–10
29. Eid HT (2013) Bearing capacity and settlement of skirted shallow foundations on sand. Int J Geomech 13(5):645–652
30. Al-Aghbari MY, Dutta R (2008) Performance of square footing with structural skirt resting on sand. Geomech Geoeng Int J 3(4):271–277
31. Al-Aghbari M, Mohamedzein Y-A (2004) Model testing of strip footings with structural skirts. Proc Inst Civil Eng Ground Improv 8(4):171–177
32. Rezaei H, Nazir R, Momeni E (2016) Bearing capacity of thin-walled shallow foundations: an experimental and artificial intelligence-based study. J Zhejiang Univ Sci A 17(4):273–285
33. Momeni E, Armaghani DJ, Fatemi SA, Nazir R (2018) Prediction of bearing capacity of thin-walled foundation: a simulation approach. Eng Comput 34(2):319–327
34. Jahed Armaghani D, Harandizadeh H, Momeni E (2021) Load carrying capacity assessment of thin-walled foundations: an ANFIS–PNN model optimized by genetic algorithm. Eng Comput 1–23
35. Chan W, Chow Y, Liu L (1995) Neural network: an alternative to pile driving formulas. Comput Geotech 17(2):135–156
36. Chow Y, Chan W, Liu L, Lee S (1995) Prediction of pile capacity from stress-wave measurements: a neural network approach. Int J Numer Anal Meth Geomech 19(2):107–126
37. Goh AT (1996) Pile driving records reanalyzed using neural networks. J Geotech Eng 122(6):492–495
38. Teh C, Wong K, Goh A, Jaritngam S (1997) Prediction of pile capacity using neural networks. J Comput Civ Eng 11(2):129–138
39. Lok T, Che W (2004) Axial capacity prediction for driven piles using ANN: model comparison. In: Geotechnical engineering for transportation projects, pp 697–704
40. Pal M, Deswal S (2008) Modeling pile capacity using support vector machines and generalized regression neural network. J Geotech Geoenviron Eng 134(7):1021–1024
41. Momeni E, Nazir R, Armaghani DJ, Maizir H (2014) Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN. Measurement 57:122–131
42. Momeni E, Nazir R, Armaghani DJ, Maizir H (2015) Application of artificial neural network for predicting shaft and tip resistances of concrete piles. Earth Sci Res J 19(1):85–93
43. Moayedi H, Jahed Armaghani D (2018) Optimizing an ANN model with ICA for estimating bearing capacity of driven pile in cohesionless soil. Eng Comput 34(2):347–356
44. Shaik S, Krishna K, Abbas M, Ahmed M, Mavaluru D (2019) Applying several soft computing techniques for prediction of bearing capacity of driven piles. Eng Comput 35(4):1463–1474

45. Harandizadeh H, Jahed Armaghani D, Khari M (2021) A new development of ANFIS–GMDH optimized by PSO to predict pile bearing capacity based on experimental datasets. Eng Comput 37(1):685–700
46. Harandizadeh H (2020) Developing a new hybrid soft computing technique in predicting ultimate pile bearing capacity using cone penetration test data. AI EDAM 34(1):114–126
47. Schmertmann JH (1978) Guidelines for cone penetration test: performance and design. Federal Highway Administration, United States
48. De Kuiter J, Beringen F (1979) Pile foundations for large North Sea structures. Mar Georesour Geotechnol 3(3):267–314
49. Bustamante M, Gianeselli L (1982) Pile bearing capacity prediction by means of static penetrometer CPT. In: Proceedings of the 2nd European symposium on penetration testing, vol 2. Balkema, Amsterdam, The Netherlands, pp 493–500
50. Momeni E, Dowlatshahi MB, Omidinasab F et al (2020) Gaussian process regression technique to estimate the pile bearing capacity. Arab J Sci Eng 45:8255–8267
51. Dehghanbanadaki A, Khari M, Amiri ST, Armaghani DJ (2021) Estimation of ultimate bearing capacity of driven piles in c-φ soil using MLP-GWO and ANFIS-GWO models: a comparative study. Soft Comput 25(5):4103–4119
52. Pham TA, Vu H-LT (2021) Application of ensemble learning using weight voting protocol in the prediction of pile bearing capacity. Math Prob Eng 2021
53. Hoang N-D, Tran X-L, Huynh T-C (2022) Prediction of pile bearing capacity using opposition-based differential flower pollination-optimized least squares support vector regression (ODFP-LSSVR). Adv Civil Eng 2022
54. Pham TA, Ly H-B, Tran VQ, Giap LV, Vu H-LT, Duong H-AT (2020) Prediction of pile axial bearing capacity using artificial neural network and random forest. Appl Sci 10(5):1871
55. Huat CY et al (2021) Factors influencing pile friction bearing capacity: proposing a novel procedure based on gradient boosted tree technique. Sustainability 13(21):11862
56. Amjad M, Ahmad I, Ahmad M, Wróblewski P, Kamiński P, Amjad U (2022) Prediction of pile bearing capacity using XGBoost algorithm: modeling and performance evaluation. Appl Sci 12(4):2126
57. Armaghani DJ et al (2020) On the use of neuro-swarm system to forecast the pile settlement. Appl Sci 10(6):1904
58. Armaghani DJ, Faradonbeh RS, Rezaei H, Rashid ASA, Amnieh HB (2018) Settlement prediction of the rock-socketed piles through a new technique based on gene expression programming. Neural Comput Appl 29(11):1115–1125

Application of a Data Augmentation Technique on Blast-Induced Fly-Rock Distance Prediction

Biao He, Danial Jahed Armaghani, and Sai Hin Lai

Abstract Fly-rock induced by blasting is an inevitable phenomenon in quarry mining, which can give rise to severe hazards, for example, causing damage to buildings and human life. Thus, successfully estimating fly-rock distance is crucial. Many researchers attempt to develop empirical, statistical, or machine learning models to accurately predict fly-rock distance. However, for most previous research, a worrying drawback is that the amount of data related to fly-rock distance prediction is insufficient. This is because the measurement work of fly-rock distance is costly for manpower and material resources. To deal with the problem of data shortage, we first separated the original data set that was collected from four granite quarry sites in Malaysia into two parts, i.e., the training and testing sets, and then adopted a data augmentation technique termed tabular variational autoencoder (TVAE) to augment the amount of the training (true) data, so as to generate a fresh synthetic data set. Subsequently, we utilized several statistical visualization methods, such as the boxplot, kernel density estimation, cumulative distribution function, and heatmap, to testify to the effectiveness of the synthetic data generated by the TVAE model. Lastly, several commonly used machine learning models were developed to verify whether the mixed data set—which is obtained by merging the training and synthetic data sets—can benefit from the addition of the synthetic data. The verification work is implemented on the testing data set. The results demonstrate that the size of the training data set has increased from the initial 131 to 1000 to obtain a synthetic data set, and the statistical methods proved that the synthetic data set not only preserves

---

B. He
Department of Civil Engineering, Faculty of Engineering, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
e-mail: s2005282@siswa.um.edu.my

D. J. Armaghani (✉)
Department of Urban Planning, Institute of Architecture and Construction, South Ural State University, Engineering Networks and Systems, 454080 Chelyabinsk, Russia
e-mail: danialarmaghani@susu.ru

S. H. Lai
Department of Civil Engineering, Faculty of Engineering, Universiti Malaya, 50603 Kuala Lumpur, Malaysia
e-mail: laish@um.edu.my

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*, Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_6

the inner characteristics of the training data set but also generalizes more diversities compared with the training data set. Further, by comparing the performance of five machine learning models on three data sets (i.e., the training, synthetic, and mixed data sets), it can be concluded that the overall performance of all machine learning models on the mixed data set outperforms that on the training and synthetic data sets. Consequently, it can be asserted that the application of the data augmentation technique on the fly-rock distance issue is fruitful in the present study and has profound engineering application value.

Keywords Fly-rock distance prediction · Data augmentation · Tabular variational autoencoder · Machine learning prediction models

1 Introduction

Blasting is the main technique for fracturing consolidated mineral deposits in the mining and construction sectors. However, the blasting process continues to pose a number of risks to nearby things and people, for example, the fly-rock. The fly-rock is defined by that the excessive random toss of rock pieces from an explosion that can go miles outside the blast protection region. According to the cast type of rock, fly-rock can be classified into three categories: face bursting, rifling, and cratering [1], as shown in Fig. 1. Among them, face bursting occurs when the explosive charges intersect the geological structures or the weak zones. Owing to the release of high-pressure gases, it is inevitable to generate the fly-rock. For this case, an effective way to control the fly-rock distance is to regulate the burden conditions; rifling occurs when stemming material is invalid or even absent, which can cause the stemming ejection and ejection of the collar rock; cratering occurs when the stemming length is insufficient or the collar rock is fragmentized. For this case, fly-rock can be projected in any direction [2].

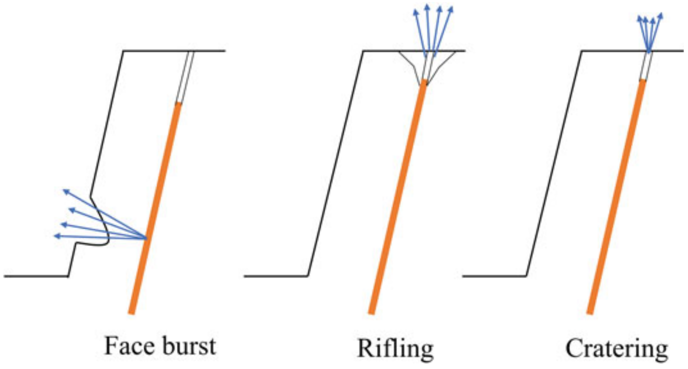


Fig. 1 Categories of fly-rock in open-pit mines

Generally, the fly-rock projection is abysmal since it is a random phenomenon and cannot be allowed for experimental purposes [3]. Therefore, accurate prediction of fly-rock distance, based on the measured data sets, is productive and profitable for practical engineering. The fly-rock issue has been investigated by a large number of researchers to forecast the potential travel distance of the rock fragment. The research methodologies for predicting fly-rock distance include empirical, statistical, and machine learning (ML) methods. Among them, the empirical formula (method) is simple to build and use, but the major shortcoming of the empirical method is that it is site-dependent, which means the given empirical formulas are merely suitable for the site-measured range of data. Moreover, the empirical formulas consider the limited numbers of factors influencing the fly-rock distance, and thus they are not satisfactory for addressing the complicated nature of the fly-rock issue [4]. As for the statistical method, it is established according to the available data set and can generally characterize the association between fly-rock distance and the influential factors. However, the shortcoming of statistical methods, such as linear or non-linear multiple regression for predicting FRD, is when newly available data are different from the original one, which is used to fit the regression function, the established regression function needs to be updated, otherwise, its performance would encounter collapse. At present, the mainstream for the estimation of fly-rock distance is to develop the ML models, as shown in Table 1.

Compared to the empirical and statistical methods, the ML models can well handle high-dimensional and large volume data sets as well as non-linear problems. This is due to the fact that intelligence-based prediction models benefit from their flexibility, which allows for simple model calibration when new data becomes available. Some researchers have proved the superiority of ML models in different areas of science and engineering [18–21, 22, 23, 24, 25, 26, 27, 28]. For instance, Armaghani et al. compared the predictive performance between the machine learning model (i.e., PSO-ANN) and several empirical models, and they concluded that the developed PSO-ANN showed better accuracy than the empirical model. This is because the PSO-ANN model leverages more variables that affect the generation of fly-rock, whereas the empirical models merely consider several of them [7]. Armaghani et al. developed an adaptive neuro-fuzzy inference system (ANFIS) model and an empirical model based on the measured fly-rock data set. The results indicated that the predictive accuracy of the ANFIS model is better than that of the empirical model. The ANFIS model can provide favorable capacity in predicting fly-rock distance [4]. Hasanipanah et al. utilized the particle swarm optimization (PSO) algorithm and multiple linear regression (MLR) to develop the statistical functions for predicting fly-rock distance based on a database containing 76 blasting events. The function obtained by the PSO algorithm showed a good capacity for fly-rock distance prediction compared with the MLR function [29]. Rezaei et al. compared the performance of the fuzzy set theory (FST) model with the MLR model and found that the FST model significantly outperformed the MLR model.

Despite the ML model showing a good ability to predict fly-rock distance, one fact that should be noted is that the ML model is sensitive to the available data set. In other words, if the quality of the data set is good enough, the developed ML model

Table 1 Relative studies on the prediction of fly-rock distance

Literature	Techniques	Input parameters	Dataset samples
Rezaei et al. [5]	FST	B, S, HL, SD, St, MC, RD, PF	490
Manoj and Monjezi [6]	SVM	HL, S, B, St, PF, SD	234
Armaghani et al. [7]	PSO-ANN	HD, HL, MC, S, B, St, PF, RD, Sd, NR	44
Marto et al. [8]	ICA-ANN	HL, BS, St, MC, PF, RD, SN	113
Ghasemi et al. [9]	FL	HL, B, S, St, PF, MC	230
Faradonbeh et al. [10]	GEP	B, S, St, HL, PF	97
Saghatforoush et al. [11]	ACO-ANN	B, S, HL, St, PF	97
Jahed Armaghani et al. [4]	ANFIS	MC, PF	232
Kumar et al. [12]	PSO-ANN	HL, S, RL, B, PF, RD	/
Nguyen et al. [13]	EANNs	MC, PF, St, S, B	210
Nguyen et al. [1]	WOA-SVM-RBF	W, B, PF, St, S	210
Jamei et al. [14]	KELM	S, B, St, PF	73
Murlidhar et al. [15]	HHO-MLP	HD, PF, CPM, St/B, HL, RQD, WI, GSI	152
Bhagat et al. [16]	CART	HL, B, SD, NH, MCPB, St, St/B, W, SC, D, V	61
Shamsi et al. [17]	GEP	B, HL, MC, St, NH, NB	33

SVM: support vector machine, PSO: particle swarm optimization, ANN: artificial neural network, ICA: imperialist competitive algorithm, KELM: kernel extreme learning machine, EANNs: ensemble of ANN models, WOA: whale optimization algorithm, RBF: radius basis function, FL: fuzzy logic, ACO: ant colony optimization, FST: fuzzy set theory, ANFIS: adaptive neuro-fuzzy inference system, GEP: gene expression programming, CART: classification and regression trees. HL: hole length, HD: hole diameter, S: spacing, RL: reducing length, B: burden per delay, PF: powder factor, RD: rock density, BS: burden to spacing, St: stemming, MC: maximum charge per delay, SN: Schmidt hammer rebound number, W: the amount of explosive used per blast, Sd: sub-drilling, NR: number of rows, SD: specific drilling, MCPB: mean charge per blasthole, NH: number of holes, NB: number of boosters, CPM: explosive charge per meter, RQD: rock quality designation, GSI: geological strength index, WI: site-specific weathering index, D: density of rock, SD: specific drilling, SC: specific charge, V: volume of the boulder

can be generalized and robust. In contrast, if the quality of the data set is poor, the developed ML model would be faced with the issue of overfitting, which indicates that the applicability of the developed ML model is poor. The quality of the data set much depends on the size of the data, because sufficient data samples mean that it covers more instances and can provide the ML model with more useful information. As shown in Table 1, different databases, ranging from 33 to 490 data samples, were used by previous researchers. In effect, for these researches, the available data is limited because the measurement work of fly-rock distance is costly manpower and material resources. Therefore, it is not easy for the researcher to collect enough data samples from practical engineering, which means the established ML model cannot meet this requirement of good generalization.

To deal with the problem of data shortage, some data augmentation methods, such as generative adversarial networks (GANs) and variational autoencoders (VAEs), have been proposed in recent years. For example, Ohno used VAEs to generate data samples based on seven benchmark datasets. The results show that the multi-task learning of VAEs relating to data augmentation is effective and improves the generalization performances of models [30]. Huang et al. proposed a boosting resampling technique based on the conditional VAE, and the results show that the conditional VAE model can generate and supply the minority samples in the data set, so as to improve the predictive performance of the minority data samples [31]. The popularity of the VAE model can be gleaned by the fact that it has been applied in many practices, such as image identification, random transformation, pattern mixing, natural language processing, etc.

Based on the above analysis and inspired by previous research, we can find that the superiority of the VAE model is obvious. Meanwhile, as far as the authors are aware, there is no study developing a data augmentation model for the generation of a fly-rock database. Thus, in the present study, we utilized a variant of the vanilla VAE model, namely tabular variational autoencoder (TVAE), to augment the original fly-rock data set. After that, we adopted several statistical indices to evaluate the generated data set by the TVAE model. Finally, we verified the effectiveness of the generated data set using several classical ML models.

The rest of the paper is organized as follows. Section 2 describes the factors affecting the generation of fly-rock projection. Section 3 presents the basic information and source of the data set used in the present study. Section 4 elaborates on the principle of the TVAE model and the demonstration of five used ML models. Section 5 expounds on the effective verification of the generated data set by the TVAE model. In Sect. 6, conclusions are drawn.

2 Factors Influencing Fly-Rock

The main factors mastering the generation of fly-rock can be summarized as two aspects, i.e., the controllable parameters such as blasting design parameters, and uncontrollable parameters such as rock properties. More specifically, controllable parameters resulting in fly-rock projection include insufficient burden, improper delay timing, inadequate stemming, inaccurate drilling, improper blast hole layout, inappropriate delay time, unwarranted powder factor, and so on [9, 32, 33], while uncontrollable parameters causing fly-rock projection include the discontinuity in the rock mass, fracture, localized fault, joint spacing and orientation, aperture, presence of voids, an anomaly in the geology and rock structure, loose rock on the top of the bench, and so on [34–36, 35–37].

For the influence of the blast design parameters, Nayak et al. stated that the case when the burden dimension is less than 25 times the charge diameter, it will give rise to a high specific charge, so as to release much energy which can incur great fly-rock distances [33]. In other words, too large a burden can result in the ejection of stemming materials and the improper fracturing of the rock mass, thereby incurring the cratering effect. Besides, they also stated that the specific charge in a hole is proportional to the fly-rock distance, which indicates that the fly-rock distance will increase with an increase in specific charge. Besides, the size of stemming materials is also of great importance. The role of stemming material is to provide confinement and prevent the escape of high-pressure gases from the blasting holes. For the case that the size of stemming materials is greater than 10 mm, it can result in greater fly-rock distances compared to fine particles. Gomes-Sebastiao and De Graaf pointed out that the exact size of stemming materials should be in compliance with 10–15% of the blast hole diameter [38]. Likewise, Ghasemi et al. reported that an increase in spacing, hole length, stemming, hole diameter, and powder factor can result in more fly-rock projection, whereas an increase in amounts of burden and mean charge per hole can reduce the fly-rock distance [2]. Mohamad et al. found that the closer the joint spacing distance, the higher the potential of the occurrence of fly-rock projection; the larger the aperture gap, the more the fly-rock generates; the existing voids in a rock mass can lead to high consumption of explosives, thereby causing a large amount of fly-rock projection; Rock face with low mean spacing and large aperture can also cause excessive fly-rock projection [34]. Additionally, the orientation of the face angle also plays a significant impact on the throwing and distance of fly-rock. The blasting design parameters in a quarry site are portrayed in Fig. 2.

For the influence of the rock properties, Kecojevic and Radomsky found that loose rock with numerous cracks can cause serious fly-rock hazards and the existing discontinuities in different rock structures may engender a discrepancy between explosive energy and rock resistance [32]. This is because the discontinuity in the

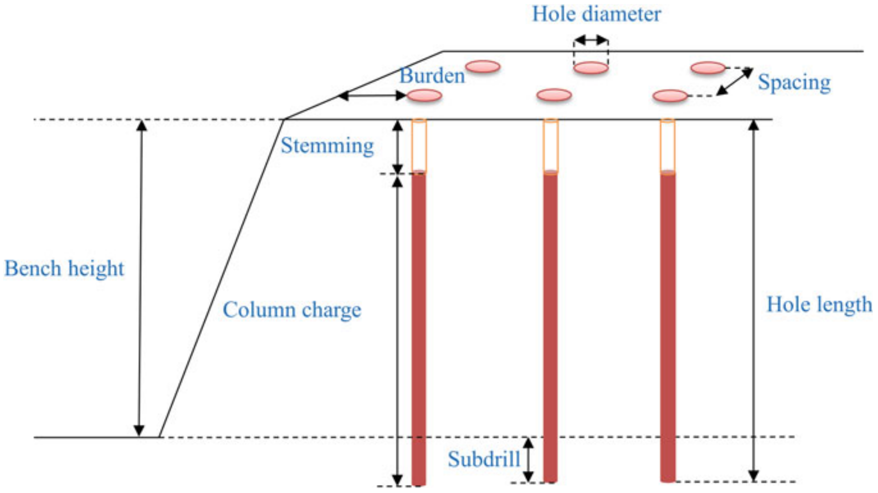


Fig. 2 Blasting design influential parameters on fly-rock

geology and rock structure causes a mismatch between the explosive energy and the resistance of the rock, and thus the released energy induced by blasting would increase due to the existing discontinuities. when an explosive's available energy is more than the energy needed to shatter a rock, fly-rock will be produced. The more discontinuities that exist in the rock mass, the more excessive fly-rock generates [2].

3 Data Source and Pre-Processing

3.1 Study Area

Four granite quarry sites in the Johor area, Malaysia were investigated in the present paper. The goal of these quarries' blasting operations is to generate aggregate with a monthly capacity of 160,000 to 380,000 t. The information on these four quarry sites is shown in Table 2. Specifically, the Taman Bestari quarry has the lowest bench height of 7 m, while the Bukit Indah quarry had the highest bench height of 28 m. In all these quarries, a variety of rock mass weathering zones were discovered, ranging from moderately worn (MW) to fully weathered (CW). To identify the weathering zones, the Schmidt hammer test was implemented to estimate the rock mass strength. The test results revealed that the lowest and maximum uniaxial compressive strength (UCS) was 40.7 and 99.8 MPa, respectively. Besides, the geological discontinuities, i.e., the rock quality designation (RQD) values, were quantified as a percentage of the drill core in lengths of 100 mm or more. The RQD findings had the lowest and highest values of 22.5 and 61.25, respectively.

3.2 Overview of Data

In the present study, a comprehensive data set that consists of 166 data samples were collected from the granite quarry sites. The available data set includes the blasting design parameters, such as the number of holes, hole length, burden to spacing, stemming, powder factor, the maximum charge per delay, and rock property, such as the Schmidt hammer rebound number. According to the prior summary of

Table 2 Description of granite quarry sites [39]

Quarry name	Latitude	Longitude	Bench height (m)
Taman Bestari	1° 60′ 41″ N	103° 78′ 32″ E	7–17
Senai Jaya	1° 36′ 00″ N	103° 39′ 00″ E	13–24
Kulai	1° 39′ 21″ N	103° 36′ 11″ E	10–22
Bukit Indah	1° 93′ 12″ N	103° 35′ 08″ E	15–28

the parameters used by researchers to implement fly-rock distance prediction work, herein, the blasting design parameters, e.g., the number of holes, hole length, the burden to spacing, stemming, powder factor, and maximum charge per delay, were considered in this study, which will be utilized to execute the development of ML models. The statistical information of the selected parameters is tabulated in Table 3. As shown in the table, the range of hole length (HL) is between 7.0 and 28.4 m, the range of burden to spacing (BS) is between 0.486 and 0.913, the range of stemming (St) is between 1.4 and 4.0 m, the range of powder factor (PF) is between 0.24 and 0.98 kg/m^3, the range of maximum charge per delay (MC) is between 69.79 and 309.09 kg, the range of the number of holes is between 22 and 60, and fly-rock distance (FRD) has a range between 39.0 and 258.0 m. Figure 3 depicts the distribution of each parameter. It can be found that, for most of the blasting events, the values of HL are clustered between 20 and 25 m; the values of BS are commonly greater than 0.6; the values of St length are commonly greater than 2.0 m; the values of PF are clustered between 0.6 and 0.9 kg/m^3; the values of MC are commonly greater than 150 kg; the values of NH are commonly less than 50, and the values of FRD are clustered between 125 and 200 m.

Before developing the ML models, one important work is to conduct data cleaning, which contributes to eliminating the considerable influence of outliers. In the present study, a boxplot was used to detect the outliers of the available data set. Boxplot can show an obvious visualization of the five-number summary which includes the extreme lower (Min), the extreme upper (Max), the first quartile (Q1), the third quartile (Q3), and the median as well as the mean [40]. As shown in Fig. 4, the box extends from the first quartile (Q1) to the third quartile (Q3) of the data, with a red line at the median and a rhombus point at the mean, and flier points (circle points), define as outliers in given data set, are those past the end of the whiskers [41]. As a result, it can be seen that only the parameter BS has two outliers that need to be removed from the original data set. After removing the two samples with outliers from the original data, there are a total of 164 data samples remaining and they will be used for the subsequent analysis.

Table 3 Characteristics of the collected data set

Variables	Symbol	Unit	Range	Mean	Std. Dev
Hole length	HL	m	(7.0, 28.4)	19.066	5.59
Burden to spacing	BS	–	(0.486, 0.913)	0.757	0.096
Stemming	St	m	(1.4, 4.0)	2.874	0.617
Powder factor	PF	kg/m^3	(0.24, 0.98)	0.688	0.197
Maximum charge per delay	MC	kg	(69.79, 309.09)	201.805	64.25
Number of holes	NH	–	(22, 60)	39.933	11.071
Fly-rock distance	FRD	m	(39.0, 258.0)	139.78	48.51

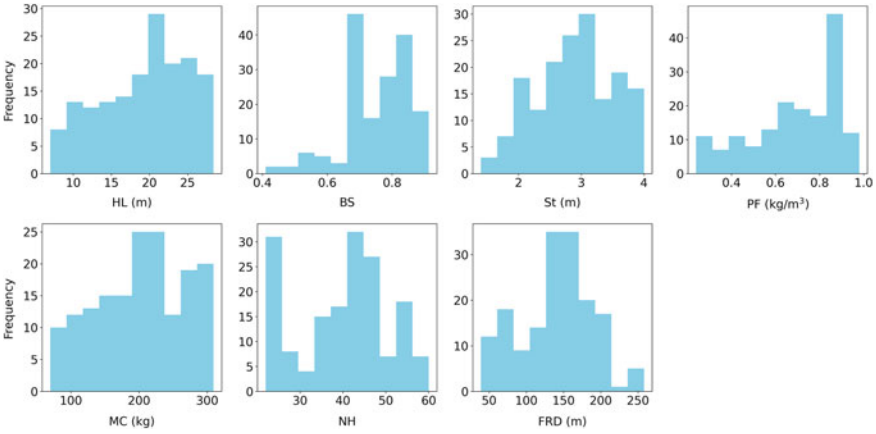


Fig. 3 Histogram of the individual variables

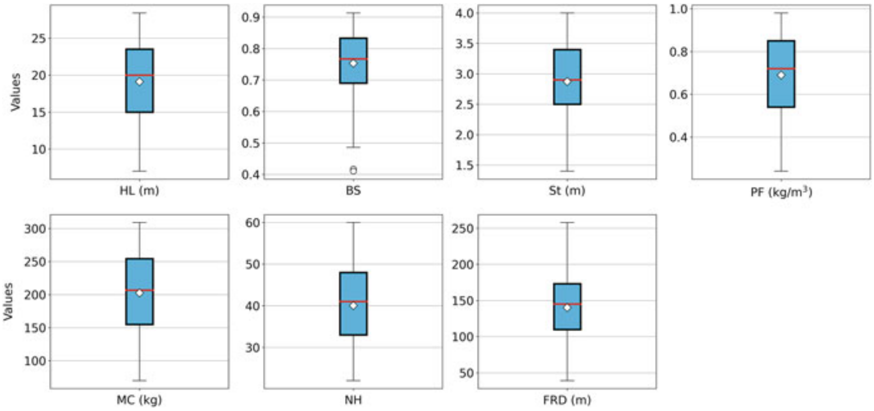


Fig. 4 Boxplot of the available data set

3.3 Data Classification

For the development of ML models, one pivotal task is to ensure the generalization capacity of the models. To achieve this aim, we selected a portion of data samples from the original data set to train the ML models and use the remaining data samples to validate the developed ML models. In the present study, 80% (131 data samples) of the entire data is assigned as the training data set and the remaining 20% (33 data samples) of the entire data are assigned as the testing data set. Table 4 shows some statistical indices of the training and test sets. It can be found that the statistical indices like the range, mean, and standard deviation of the training set are approximate to those of the testing sets. To have an intuitive visualization of the data distribution of the training and testing sets, a boxplot is used herein to achieve this aim. As shown in Fig. 5, it can be observed that the discrepancies between the training and testing sets are the Q1 and Q3 values, for example, the Q1 and Q3 values of variables HL, MC, NH, and FRD, which indicates the data distribution of the training and testing sets is subtly different. Further, a kernel density estimate (KDE) plot is utilized to visualize the distribution of observations in the two data sets. KDE represents the data through a continuous probability density function or curve in one or more dimensions [42]. As can be seen in Fig. 6, for variables HL, MC, NH, and FRD, the shape of their probability density curves of the training and testing sets are similar. Nevertheless, for variables BS, St, and PF, the shape of their probability density curves of the training and testing sets is diverse, for example, although the probability density curves of variables BS and PF of both training and testing sets are bimodal distribution, the positions of the crest differ considerably. Besides, regarding the variable St, the peak intensity of the training set is higher than that of the testing set.

Table 4 Characteristics of the training and testing data sets

Symbol	Unit	Training set			Testing set		
		Range	Mean	Std. Dev	Range	Mean	Std. Dev
HL	m	(8.0, 28.4)	19.282	5.514	(7.0, 28.0)	18.209	5.803
BS	–	(0.486, 0.913)	0.754	0.097	(0.5, 0.909)	0.769	0.09
St	M	(1.5, 4.0)	2.884	0.604	(1.4, 3.9)	2.835	0.662
PF	kg/m^3	(0.24, 0.98)	0.692	0.196	(0.27, 0.95)	0.674	0.202
MC	kg	(74.78, 309.09)	204.373	63.655	(69.79, 301.61)	191.612	65.578
NH	–	(22, 60)	39.786	10.95	(23, 60)	40.515	11.521
FRD	m	(39.0, 258.0)	141.744	48.817	(44.8, 239.8)	131.982	46.458

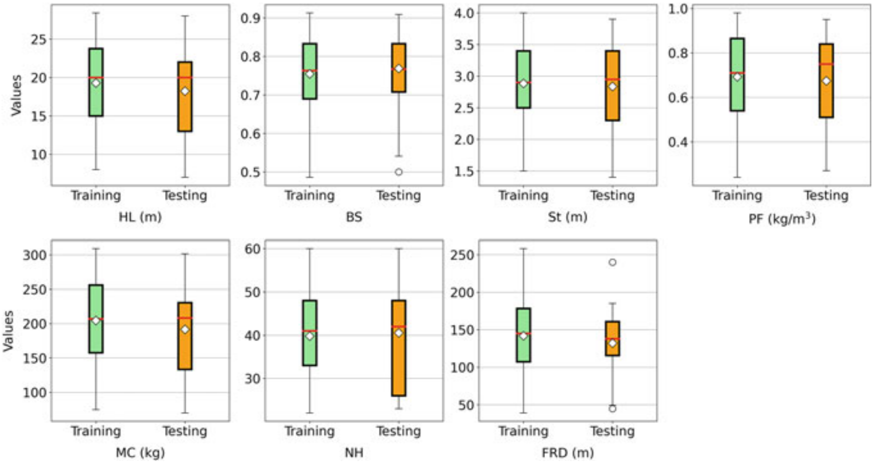


Fig. 5 Boxplot of the training and testing data sets

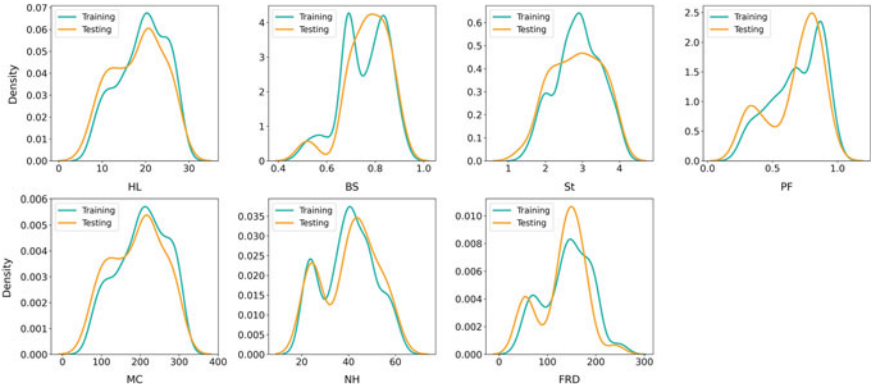


Fig. 6 KDE plot of the training and testing data sets

According to the above analysis, we can sum up that there are both similarities and slight differences between the training and test sets, which is beneficial for the development of the ML models. This is because we can use the similarities between the data sets to verify the predictive performance of the ML models, while the differences between the data sets can provide an evaluation of the generalization capacity.

4 Methodology

4.1 Study Step

After the training and testing data are successfully prepared, they will be used to implement the modeling and validation tasks, respectively. The research methodology in the present study is mainly composed of two parts. More specifically, the first part is that a data augmentation technique termed tabular variational autoencoder (TVAE) will be used to synthesize the new data samples for the prediction of fly-rock distance, while the second part is to test the validity of the generated data set using some famous ML models, i.e., support vector regression (SVR), light gradient boosting machine (LightGBM), extreme learning machine (ELM), group method of data handling (GMDH), and multilayer perceptron (MLP). The flowchart of methodology in this study is illustrated in Fig. 7 and a detailed description of these techniques will be given hereinafter.

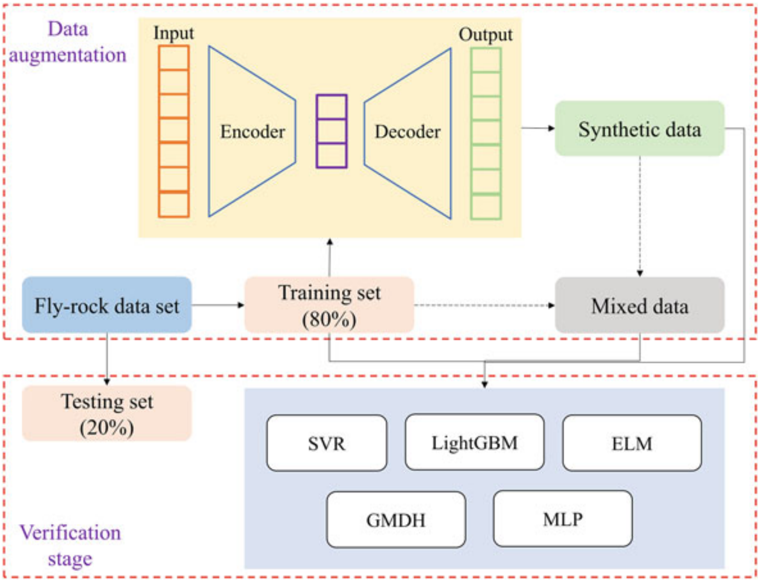


Fig. 7 Flowchart of the methodology used in this research

4.2 Tabular Variational Autoencoder (TVAE)

Variational autoencoder (VAE) is a kind of neural network generative model [43, 44]. VAE utilizes two neural networks to model two probability density distributions. One, called the inferential network, is used for variational inference of the original input data to generate the variational probability distribution of the latent variable z, the other, called the generative network, restores to generate an approximate probability distribution of the original data based on the variational probability distribution of the generated latent variable [45]. The VAE can be regarded as a hybrid of a neural network and a Bayesian network [31]. In the VAE, the nodes corresponding to the latent encoding can be regarded as random variables and the other nodes are considered ordinary neurons. In this way, the encoder becomes a variational inferential network, while the decoder can be seen as a generative network that maps latent variables to observed variables. Compared with traditional deep generative networks, VAE has two most prominent advantages: (1) by introducing a variational lower bound, it avoids the direct calculation of complex marginal likelihood probabilities; (2) The complex Markov chain sampling process is avoided by parameter transformation.

The TVAE model is specially designed for tabular data. It uses two neural networks to model $p_\theta\left(r_j|z_j\right)$ and $q_\phi\left(z_j|r_j\right)$ [46]. $p_\theta\left(r_j|z_j\right)$ is the prior distribution of the latent variable z and $q_\phi\left(z_j|r_j\right)$ is the approximate posterior. Compared to VAE, TVAE uses the same pre-processing method as VAE, but the loss function of TVAE is modified on the basis of the loss function of VAE. The loss in TVAE is evidence lower-bound

(ELBO) loss [43, 47], shown as follows.

$$\begin{aligned} logp_\theta\left(X_j\right) &\geq \mathcal{L}\left(\theta, \phi; X_j\right) \\ &= \mathbb{E}_{q_\phi\left(z_j|X_j\right)}\left[logp_\theta\left(X_j|z_j\right)\right] - \mathbb{KL}\left[q_\phi\left(z_j|X_j\right)||p\left(z_j\right)\right] \end{aligned} \quad (1)$$

where $logp_\theta(X) = logp_\theta(X_1, X_2, \ldots, X_n) = \sum_{j=1}^{n} logp_\theta\left(X_j\right)$

The second term of the right-hand side of Eq. (1) is the KL divergence of the approximate posterior and prior. Generally, $p\left(z_j\right)$ is multivariate Gaussian distribution, and $p_\theta\left(X_j|z_j\right)$ and $q_\phi\left(z_j|X_j\right)$ are parameterized by neural networks and optimized by gradient descent algorithm. A more detailed description of the TVAE model can be found in [47].

4.3 Prediction Models

To verify the effectiveness of the applied data augmentation method (i.e., the TVAE model), several common ML models are adopted to develop fly-rock distance prediction models. The given data sets used for the development of ML models are comprised of three parts, i.e., the true (training) data set, the synthetic data set obtained by the TVAE model, and the mixed data set which consists of both true (training) and synthetic data sets. The validity of the synthetic data is demonstrated by comparing the predictive performances of the mentioned ML models on these three kinds of data sets. The description and structure of these five ML models are given hereinafter.

4.3.1 Support Vector Regression (SVR)

SVR, one of the traditional ML techniques, can solve big data regression problems, maximize predictive accuracy, and avoid overfitting simultaneously [48]. Essentially, SVR is built on the basis of target values and aims to find a function that can map data to a flat space. In SVR, linear and non-linear regression are used to solve complex problems. More specifically, linear regression problems can be solved by means of a convex optimization with solutions and constraints, while non-linear regression problems can be solved by a convex optimization with a kernel function that is capable of transforming the data into a high-dimensional space. The crucial concepts of SVR include four points: (1) the separating hyperplane, (2) the maximum-margin hyperplane, (3) the soft margin, and (4) the kernel function [49]. The notion of a separating hyperplane is a specific straight line in a high-dimensional space that separate the data sets according to their respective characteristics. The maximum-margin hyperplane refers to selecting a particular hyperplane that maximizes the SVR's ability to obtain the correct results of previously unseen instances as many as possible. The soft margin is a user-specified parameter that can allow some data points to push their way through the margin of the separating hyperplane without affecting the final results. In essence, the soft margin represents a trade-off between hyperplane violations and the size of the margin. As for the kernel function, its role is to project data from a low-dimensional space to a high-dimension space. In order to obtain the separable data in the resulting higher dimensional space, the important step is to select the suitable kernel function. The most commonly used kernel functions are polynomial and radial basis functions.

4.3.2 Light Gradient Boosting Machine (LightGBM)

LightGBM is a fast and efficient boosting model based on the framework of the GBDT algorithm [50]. Although the traditional boosting algorithm already has better efficiency, in today's situations of large samples and high dimensionality, the traditional boosting algorithm cannot meet the current demand in terms of efficiency and scalability. Its main drawback is that it needs to scan all the features of each sample when selecting the optimal splitting point, which is a very time-consuming and memory-consuming process. To amend this problem, LightGBM, on the one hand, adopts the histogram algorithm, which discretizes the floating-point feature values in the samples into K integers, forming a histogram of width K [51]. When LightGBM traverses all samples, the discrete values are counted as cumulative indexes, and then the best splitting nodes are captured based on the discrete values. At the same time, the LightGBM model uses a histogram for the discretization to accelerate the model computation. During the differencing process, only K time calculations are required, which results in a significant increase in computational speed. On the other hand, LightGBM uses a more efficient leaf growth strategy (Leaf-wise), i.e., find the one with the largest splitting gain to split from all the current leaves, and then find another one that produces the largest splitting gain, and so on in a continuous loop. At the same number of splits, the Leaf-wise strategy can obtain better accuracy compared to Level-wise [52]. Also, to prevent overfitting, a maximum depth limit parameter is added to the Leaf-wise strategy.

4.3.3 Extreme Learning Machine (ELM)

Extreme learning machine (ELM), as a new single hidden layer feedforward neural network (SLFN) learning scheme, has obtained extensive attention and has been widely used in many applications [53]. ELM can be extended to generalized multi-layer feedforward neural networks in which a hidden node could be a subnetwork of nodes. The learning theories of ELM show that when learning parameters of hidden layer nodes are generated independently of training samples, as long as the activation function of the feedforward neural network is non-linear and continuous, it can approach any continuous objective function. In ELM, the input weights and hidden biases connecting the input layer and the hidden layer can be independent and randomly generated from any continuous probability distribution. The output weight matrix between the hidden layer and the output layer is obtained by minimizing the squared loss function and solving the Moore–Penrose generalized inverse operation. The parameter that only needs to be optimized is the number of hidden layer nodes. Different from traditional gradient-based neural network learning algorithms, which are sensitive to the combination of parameters and easy to trap in local optimum, ELM has a faster learning speed, least human intervention, and is easy to implement [54]. The mathematical model of ELM is shown as follows:

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} \tag{2}$$

where $f_L(x)$ denotes the output function, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_L]^{\mathrm{T}}$ denotes the vector of the output weights, $\boldsymbol{h}(\boldsymbol{x}) = [h_1(x), \ldots, h_L(x)]^{\mathrm{T}}$ denotes the output vector of the hidden layer with respect to the input x, which is also called a feature mapping because it maps the data from the d-dimensional input space to the L-dimensional hidden-layer feature space (i.e., the ELM feature space) [55].

4.3.4 Group Method of Data Handling (GMDH)

GMDH is a self-organizing data mining algorithm based on inductive learning algorithms. GMDH can find the optimal structure of the mathematical description of a complex object by sorting many variants according to a certain ensemble of external criteria [56]. Its main idea is to start from a partial model (or function) composed of reference functions; then, according to certain laws, it generates the first generation of the candidate models by inheritance and mutation; next, it selects the optimal number of items from the first generation of the candidate models, and then generates the second generation of the candidate models, so as to make the generated models keep evolving. Repeating such a process, that is, inheritance, mutation, selection, and evolution, GMDH can make the complexity of the intermediate models increase until the optimal complexity model is obtained [57]. Essentially, GMDH can establish a higher-order polynomial relationship between the independent and dependent variables to obtain a polynomial model with a good explanation for the dependent variable. GMDH generally uses the Kolmogorov–Gabor polynomial function to establish the functional relationship between the input variables and the target output. The mathematical model of GMDH is shown as follows:

$$y = a_0 + \sum_{i=1}^{M} a_i x_i + \sum_{i=1}^{M}\sum_{j=1}^{M} a_{ij} x_i x_j + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} a_{ijk} x_i x_j x_k + \cdots \tag{3}$$

where $(x_1, x_2, \ldots, x_M)$ represent the input variables, $(a_1, a_2, \ldots, a_M)$ represents the weight vector or matrix, and y is the target output. GMDH model typically uses a multilayer iterative algorithm for the selection of neurons in the modeling process, thereby achieving a non-linear mapping between input variables and output. Besides, the GMDH model uses a minimum deviation criterion to select the optimal model.

4.3.5 Multilayer perceptron (MLP)

MLP, as one type of neural network, can be trained to approximate virtually any smooth function [58]. MLP consists of a system of simple interconnected nodes and can be considered as a model representing a non-linear mapping between input variables and target output. The weights and output signals connecting the nodes in an MLP are a function of the nodes' total input variables, as adjusted by a straightforward non-linear transfer or activation function. The nodes in the MLP model are fully connected, with each node connected to every node in the next and previous layer. A commonly applied activation function is the logistic function because of its easily computed derivative. The architecture of an MLP is generally composed of several layers of neurons [59].

By selecting a suitable set of weights and activation functions, an MLP can fit any measurable functions between the input and output variables. Training an MLP is the process to capture the connection weights to obtain minimal differences between the true target and the network output [60]. To achieve this aim, the backpropagation (BP) algorithm is the most used technique. BP algorithm transfers the mapping problem between the input and output variables into a non-linear optimization problem. The network starts training by setting small random interconnection weights, and by repeatedly loading training samples and adjusting the weights until the loss function drops to an acceptable threshold. The learning process of the BP algorithm consists of forwarding propagation and backward propagation. In the forward propagation process, the input information is passed from the input layer through the hidden layer and then to the output layer, with the state value of each layer only affecting the state value of the neuron in the next layer; if the desired output value cannot be obtained in the output layer, it is transferred to the backward propagation, where the error signal is returned along the reverse path, and the total error of the network is made to converge to a minimum by correcting the weights of the neurons in each layer [61].

5 Results and Analysis

5.1 Data Synthesis

This section demonstrates the generation of the synthetic data, as well as the correlations and discrepancies between the true data and the synthetic data. The training data set with 131 samples, also known as the true data set, is adapted to train the TVAE model. It should be noted that data augmentation is an unsupervised learning technique so that all features in the training data set are used to fit the TVAE model, which indicates that there is no labeled data in this stage. In addition, the TVAE model applied in the present study is provided by one open-source Python library, namely the synthetic data generation ecosystem SDV-The Synthetic Data Vault [62]. The configuration of the TVAE model is mainly composed of an encoder and a decoder. The encoder has three hidden layers and the layer sizes are 128, 128, and 128, respectively. The decoder has the same architecture as the encoder. Besides, the size of the random sample passed to the generator of the TVAE model is 128, and the value of the regularization term is 1e−5. The TVAE model is trained using Adam (adaptive moment estimation) with a batch size of 60, epochs of 5000, and a learning rate of 1e−3. More detailed information regarding the TVAE model is demonstrated in [46].

After data augmentation, the sample size of the synthetic data is 1000 samples, which is around 10 times the size of the true data set. The features/variables of true and synthetic data sets consist of HL, BS, St, PF, MC, NH, and FRD. As the data dimension is seven, it is difficult to visualize the sample space. Herein, we applied t-SNE (t-distributed stochastic neighbor embedding [63]) to have an insight into the distribution of true and synthetic data samples in an equivalent two dimensions domain, as shown in Fig. 8. It can be observed that the green points (true data samples) are almost fused with the orange points (synthetic data samples). Additionally, we can also find some orange points which are not intersected with the red points, which means the TVAE model fitted by true data is capable of generalizing the true data, so as to introduce some unknown data instances. Consequently, a vital task arising from this is to judge the reasonableness and veracity of the synthetic data compared to the true data.

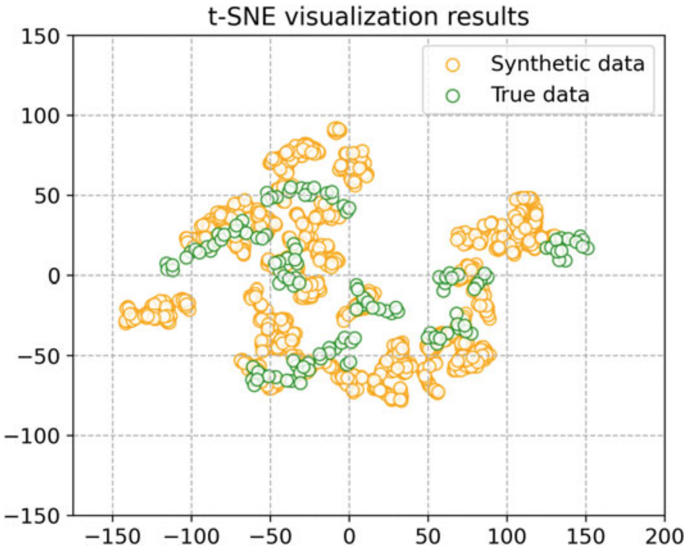


Fig. 8 Visualization of data samples of different data sets (i.e., the true and synthetic data sets)

Table 5 Characteristics of the true (training) and synthetic data sets

Symbol	Unit	True data set			Synthetic data set		
		Range	Mean	Std. Dev	Range	Mean	Std. Dev
HL	m	(8.0, 28.4)	19.282	5.514	(7.731, 28.898)	19.181	4.809
BS	–	(0.486, 0.913)	0.754	0.097	(0.525, 0.918)	0.763	0.088
St	m	(1.5, 4.0)	2.884	0.604	(1.62, 3.89)	2.837	0.477
PF	kg/m^3	(0.24, 0.98)	0.692	0.196	(0.24, 0.98)	0.683	0.169
MC	kg	(74.78, 309.09)	204.373	63.655	(74.78, 309.09)	203.462	56.849
NH	–	(22, 60)	39.786	10.95	(22, 60)	38.899	9.726
FRD	m	(39.0, 258.0)	141.744	48.817	(39.0, 255.8)	146.296	42.451

To clearly elucidate the latent correlation between the true and synthetic data sets, some statistical metrics or methods, such as the data characteristics, boxplot method, data distribution, cumulative distribution, and data relationship, are utilized herein. The characteristics of the true and synthetic data sets are summarized in Table 5. Obviously, the statistical indices like range, mean, and standard deviation of the true data are approximate to those of the synthetic data, which indicates there exist pronounced similarities between these two data sets. Of course, we can also observe some discrepancies between them. For instance, the HL of the synthetic data set has a wider range than that of the true data set; the upper bound of BS of the synthetic data set is expanded but its lower bound is shrunken; both upper and lower bounds of St of synthetic data set are covered by that of true data set; while for PF, MC, NH, and FRD, their ranges in true and synthetic data sets are identical but with different mean values and standard deviations.

Boxplot results of true and synthetic data sets are illustrated in Fig. 9. It can be seen that Q1 and Q3 of true and synthetic data sets are similar, which indicates that their data structures are approximative. Further, the KDE plot is applied to illustrate the conditions of data distribution of true and synthetic data sets, as shown in Fig. 10. Intuitively, for variables BS, St, NH, and FRD, the shape of their probability density curves of the true and synthetic data sets are similar. Nevertheless, for variables HL, PF, and MC, the shape of their probability density curves is diverse, for example, the probability density curves of variables HL, PF, and MC of the synthetic data set are bimodal distribution while their probability density curves of the true data set are unimodal distribution, which implies that the TVAE model generates different data structures for these variables. Apart from that, we can also observe that the peak intensities of variables of the synthetic data set are all higher than those of the true data set. This is because more data samples are generated by the TVAE model around the peak.

Further, the cumulative distribution of variables in true and synthetic data sets is given in Fig. 11. The cumulative distribution function (CDF) essentially allows plotting a variable of the given data in order from least to greatest and seeing the whole feature as if is distributed across the data set. CDF plots are useful for comparing

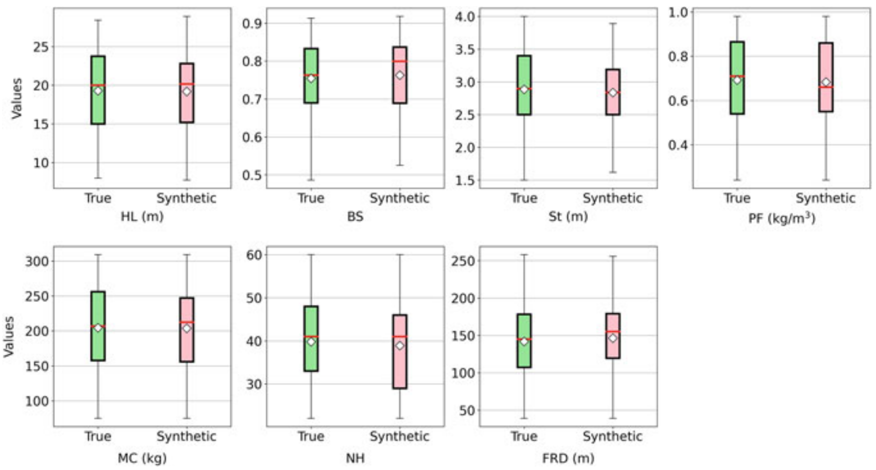


Fig. 9 Boxplot of the true (training) and synthetic data sets

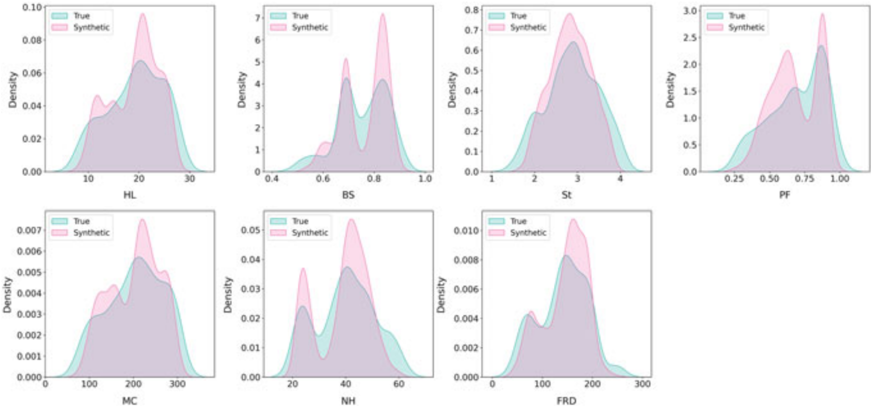


Fig. 10 KDE plot of the true and synthetic data sets

the distribution of different sets of data [64]. As a result, one quick insight is that the cumulative distribution curves of true and synthetic data sets are generally similar. For variables like BS, St, and NH, the cumulative distribution curve of the synthetic data set is smoother compared to that of the true data set, which is attributed to the TVAE model filling some of the gaps between the data points. Overall speaking, the cumulative distribution of the synthetic data set conforms to the cumulative distribution of the true data set.

Eventually, in addition to comparing the differences of single variables of the synthetic and true data sets, we also analyze the column-wise correlation of variables using the heatmap method, as shown in Fig. 12. The values in the box represent the Pearson correlation coefficient (PCC) [65, 66]. When the value of PCC is greater

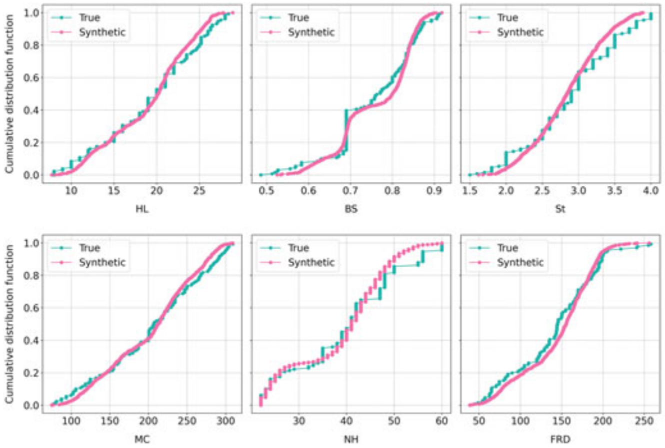
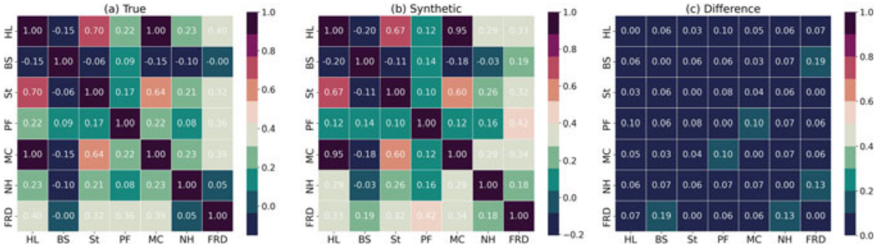


Fig. 11 Cumulative distribution plots of the true (training) and synthetic data sets

than 0 but less than 1, it represents positive linear correlations between variables; when the value of PCC is greater than −1 but less than 0, it represents negative linear correlations between variables. Further, the closer the value of PCC is to 1/−1, the stronger the positive/negative correlation is. Figure 12a and b illustrate the column-wise correlation of true and synthetic data sets, respectively, and Fig. 12c shows the absolute differences between the true data set and the synthetic data set. As a result, it is copiously clear that, for the majority of variables, the column-wise correlation of the synthetic data set approximates that of the true data set. While for PCC between BS and FRD, PCC between MC and PF, as well as PCC between NH and FRD, although their absolute differences are greater compared to others, these values are all smaller than 0.20, which is acceptable owing to the indistinctive correlations between these variables.

To sum up, based on the evaluation results of the aforementioned statistical methods, one can assert that the TVAE model performed well in generating the synthetic data. Synthetic data complies with the inherent characteristics of true data.


(a) True

	HL	BS	St	PF	MC	NH	FRD
HL	1.00	-0.15	0.70	0.22	1.00	0.23	0.40
BS	-0.15	1.00	-0.06	0.09	-0.15	-0.10	-0.00
St	0.70	-0.06	1.00	0.17	0.64	0.21	0.32
PF	0.22	0.09	0.17	1.00	0.22	0.08	0.36
MC	1.00	-0.15	0.64	0.22	1.00	0.23	0.39
NH	0.23	-0.10	0.21	0.08	0.23	1.00	0.05
FRD	0.40	-0.00	0.32	0.36	0.39	0.05	1.00

(b) Synthetic

	HL	BS	St	PF	MC	NH	FRD
HL	1.00	-0.20	0.67	0.12	0.95	0.29	0.33
BS	-0.20	1.00	-0.11	0.14	-0.18	-0.03	0.19
St	0.67	-0.11	1.00	0.10	0.60	0.26	0.32
PF	0.12	0.14	0.10	1.00	0.12	0.16	0.42
MC	0.95	-0.18	0.60	0.12	1.00	0.29	0.34
NH	0.29	-0.03	0.26	0.16	0.29	1.00	0.18
FRD	0.33	0.19	0.32	0.42	0.34	0.18	1.00

(c) Difference

	HL	BS	St	PF	MC	NH	FRD
HL	0.00	0.06	0.03	0.10	0.05	0.06	0.07
BS	0.06	0.00	0.06	0.06	0.03	0.07	0.19
St	0.03	0.06	0.00	0.08	0.04	0.06	0.00
PF	0.10	0.06	0.08	0.00	0.10	0.07	0.06
MC	0.05	0.03	0.04	0.10	0.00	0.07	0.06
NH	0.06	0.07	0.06	0.07	0.07	0.00	0.13
FRD	0.07	0.19	0.00	0.06	0.06	0.13	0.00

Fig. 12 Correlations between the variables of the true (training) and synthetic data sets

Next, we will apply several ML models to verify the effectiveness of the synthetic data on the task of predicting fly-rock distance.

5.2 Validation and Analysis

In this section, five commonly used ML models, i.e., SVR, LightGBM, ELM, GMDH, and MLP, are employed to evaluate the quality of the synthetic data. To train the ML models, three data sets are prepared, that is, the true data which has 131 data samples, the synthetic data which has 1000 data samples, and the mixed data which is consisted of the true and synthetic data sets and thus has 1131 data samples. In these three data sets, the input parameters include HL, BS, St, PF, MC, and NH, while the output is FRD. As a result, SVR, LightGBM, ELM, GMDH, and MLP are used to establish the fly-rock distance prediction models. The modeling steps are conducted as follows: first, three data sets (i.e., true, synthetic, and mixed data sets) are used to fit five ML models, respectively. Here, the Grid-search method was applied to determine the optimal hyperparameters or architectures of the ML models and the hyperparameters of the ML models are shown in Table 6; then, the fitted ML models are assessed on the testing data set which has 33 data samples; finally, through comparing the assessment results, we can clarify the effectiveness of the synthetic data set and further determine whether the synthetic data can be used as a complement to, or even more so, as an alternative for the true data.

To precisely quantify the accuracy of these five ML models on the testing data set, three widely used evaluation metrics for regression tasks are applied herein, i.e., the coefficient of determination (R^2), the root mean squared error (RMSE), and mean absolute percentage error (MAPE). The following formulas are utilized to calculate these metrics:

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_i - \widehat{y_i}\right)^2}{\sum_{i=1}^{N} \left(y_i - \overline{y}\right)^2} \tag{4}$$

Table 6 Hyperparameters of the ML models

Model	Key hyper-parameters or architecture
SVR	kernel = ‘rbf’, C = 80,000, gamma = ‘scale’, epsilon = 0.1
LightGBM	n_estimators = 500, subsample = 0.5, boosting_type = ‘gbdt’
ELM	hidden_units = 16, activation_function = ‘sigmoid’, C = 0
GMDH	ref_functions = ‘quadratic’, criterion_minimum_width = 5, L2 = 0.5
MLP	Input-FC(256)-FC(256)-FC(256)-Dropout(0.20)-Output, learning_rate = 1e−4, weight_decay = 1e−5

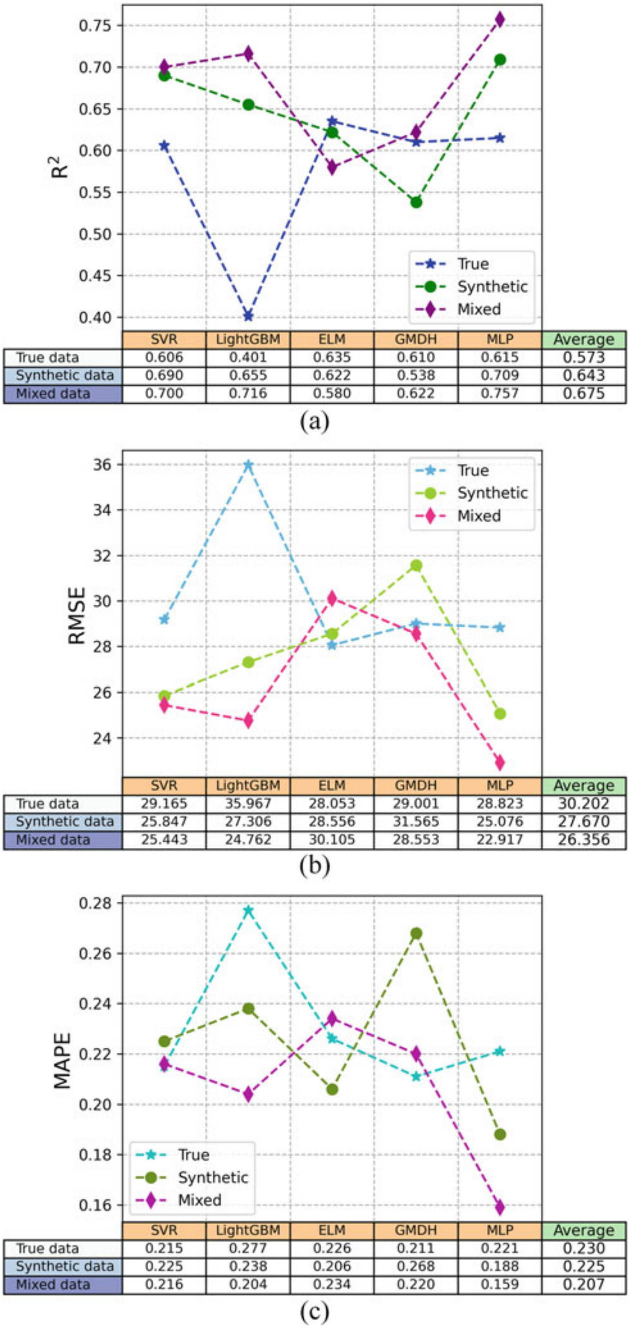
$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{N}\left(y_i - \widehat{y_i}\right)^2} \tag{5}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y_i}}{y_i}\right| \tag{6}$$

where N denotes the sample size, y_i represents the true FRD, $\widehat{y_i}$ represents the predicted FRD, $\overline{y}$ represents the average of y_i. For an excellent model, the values of R^2 should be closer to 1 and the value of the RMSE and MAPE should be closer to 0.

As shown in Fig. 13a, for SVR, LightGBM, and MLP models, the same pattern is that with the increase of input data samples (i.e., from true data to synthetic data to mixed data), the R^2 between the models' prediction results and the true fly-rock distance on testing set is gradually increased. More specifically, the R^2 of SVR models fitted by true, synthetic, and mixed data sets is 0.606, 0.690, and 0.700, respectively; the R^2 of LightGBM models fitted by true, synthetic, and mixed data sets is 0.401, 0.655, and 0.716, respectively; the R^2 of MLP models fitted by true, synthetic, and mixed data sets is 0.615, 0.709, and 0.757, respectively. For the ELM model, the performance ranking is: ELM model fitted by true data > ELM model fitted by synthetic data > ELM model fitted by mixed data, which can be proved by that their R^2 values are 0.635, 0.622, and 0.580, respectively. For the GMDH model, the performance ranking is: GMDH model fitted by mixed data > GMDH model fitted by true data > GMDH model fitted by synthetic data, which can be proved by that their R^2 values are 0.622, 0.610, and 0.538, respectively. Besides, we can also find that the best-fitted model for the true data set is ELM, and the best-fitted model for synthetic and mixed data sets is MLP. In the end, we compute the average R^2 values of these models on the testing data set, which are 0.573, 0.643, and 0.675, respectively. Compared to the ML models fitted by the true data set, the R^2 values of the ML models fitted by the synthetic and mixed data sets obtain an improvement of 12.216% and 17.801%, respectively.

As shown in Fig. 13b, for SVR, LightGBM, and MLP models, with the increase of input data samples (i.e., from true data to synthetic data to mixed data), the RMSE between the models' prediction results and the true fly-rock distance on testing set is gradually decreased. In particular, the RMSE of SVR models fitted by true, synthetic, and mixed data sets are 29.165, 25.847, and 25.443, respectively; the RMSE of LightGBM models fitted by true, synthetic, and mixed data sets are 35.967, 27.306, and 24.762, respectively; the RMSE of MLP models fitted by true, synthetic, and mixed data sets are 28.823, 25.076, and 22.917, respectively. While for ELM and GMDH models, different cases appeared. For example, for the ELM model, the performance ranking is: ELM model fitted by true data > ELM model fitted by synthetic data > ELM model fitted by mixed data, which can be proved by that their RMSE values are 28.053, 28.556, and 30.105, respectively; for GMDH model, the performance ranking is: GMDH model fitted by mixed data > GMDH model fitted



(a)

	SVR	LightGBM	ELM	GMDH	MLP	Average
True data	0.606	0.401	0.635	0.610	0.615	0.573
Synthetic data	0.690	0.655	0.622	0.538	0.709	0.643
Mixed data	0.700	0.716	0.580	0.622	0.757	0.675

(b)

	SVR	LightGBM	ELM	GMDH	MLP	Average
True data	29.165	35.967	28.053	29.001	28.823	30.202
Synthetic data	25.847	27.306	28.556	31.565	25.076	27.670
Mixed data	25.443	24.762	30.105	28.553	22.917	26.356

(c)

	SVR	LightGBM	ELM	GMDH	MLP	Average
True data	0.215	0.277	0.226	0.211	0.221	0.230
Synthetic data	0.225	0.238	0.206	0.268	0.188	0.225
Mixed data	0.216	0.204	0.234	0.220	0.159	0.207

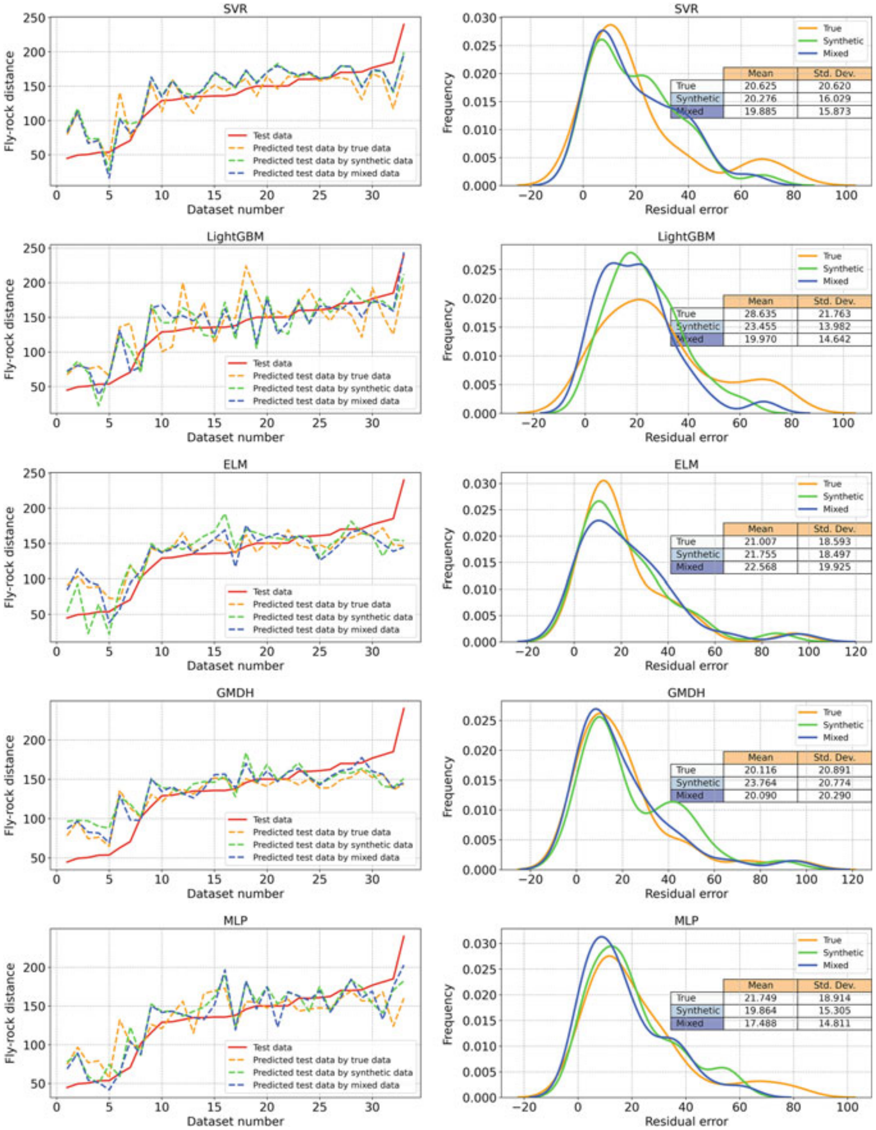
Fig. 13 Comparison results of ML models on testing set: **a** R^2; **b** RMSE; **c** MAPE

by true data > GMDH model fitted by synthetic data, which can be proved by that their RMSE values are 28.553, 29.001, and 31.565, respectively. Additionally, we can also find that the best-fitted model for the true data set is ELM, and the best-fitted model for synthetic and mixed data sets is MLP. In the end, we also compute the average RMSE values of these models on the testing data set, which are 30.202, 27.670, and 26.356, respectively. Compared to the ML models fitted by the true data set, the RMSE values of the ML models fitted by the synthetic and mixed data sets obtain reduced errors of 9.151% and 14.593%, respectively.

As shown in Fig. 13c, for LightGBM and MLP models, with the increase of input data samples (i.e., from true data to synthetic data to mixed data), the MAPE between the models' prediction results and the true fly-rock distance on testing set is gradually decreased. In particular, the MAPE of LightGBM models fitted by true, synthetic, and mixed data sets are 0.277, 0.238, and 0.204, respectively; the MAPE of MLP models fitted by true, synthetic, and mixed data sets are 0.221, 0.188, and 0.159, respectively. While for SVR, ELM, and GMDH models, different cases appeared. For example, for the SVR model, the performance ranking is: ELM model fitted by true data > ELM model fitted by mixed data > ELM model fitted by synthetic data, which can be proved by that their MAPE values are 0.215, 0.216, and 0.225, respectively; for ELM model, the performance ranking is: ELM model fitted by synthetic data > ELM model fitted by true data > ELM model fitted by mixed data, which can be proved by that their MAPE values are 0.206, 0.226, and 0.234, respectively; for GMDH model, the performance ranking is: GMDH model fitted by true data > GMDH model fitted by mixed data > GMDH model fitted by synthetic data, which can be proved by that their MAPE values are 0.211, 0.220, and 0.268, respectively. Additionally, we can also find that the best-fitted model for the true data set is GMDH, and the best-fitted model for synthetic and mixed data sets is MLP. In the end, we also compute the average MAPE values of these models on the testing data set, which are 0.230, 0.225, and 0.207, respectively. Compared to the ML models fitted by the true data set, the MAPE values of the ML models fitted by the synthetic and mixed data sets obtain reduced errors of 2.222% and 11.111%, respectively.

Consequently, it can be inferred that the mixed data set shows the best performance and applicability in predicting fly-rock distance, followed by the synthetic data set, but the true data set, has the worst performance. From this point, the fly-rock distance prediction work can benefit the superiority of the mixed data which is mainly generated by the TVAE model. This type of data combines the characteristics of the original data with the information added by the synthetic data, which has a promising prospect when implementing similar prediction works.

Here, we also provide the detailed prediction results of these five ML models, as shown in Fig. 14. The plot on the left of each row represents the measured fly-rock distance (i.e., the testing set) versus the predicted fly-rock distance. It should be noted that the measured fly-rock distance is sorted in an ascending form and the corresponding predicted fly-rock distance is given simultaneously, as suggested by Sadrossadat et al. [67]. Intuitively, for all ML models fitted by three data sets, the



SVR

	Mean	Std. Dev.
True	20.625	20.620
Synthetic	20.276	16.029
Mixed	19.885	15.873

LightGBM

	Mean	Std. Dev.
True	28.635	21.763
Synthetic	23.455	13.982
Mixed	19.970	14.642

ELM

	Mean	Std. Dev.
True	21.007	18.593
Synthetic	21.755	18.497
Mixed	22.568	19.925

GMDH

	Mean	Std. Dev.
True	20.116	20.891
Synthetic	23.764	20.774
Mixed	20.090	20.290

MLP

	Mean	Std. Dev.
True	21.749	18.914
Synthetic	19.864	15.305
Mixed	17.488	14.811

Fig. 14 Detailed prediction results and residual error of all methods on the testing set

predicted fly-rock distance values are remarkably different from the measured fly-rock distance values. It is difficult to distinguish the advantages and disadvantages of these models. Given this point, we provide the KDE plot of the absolute residual errors between the measured fly-rock distance and predicted fly-rock distance of each ML model, which aims to compare the model's performance, as shown in the plots on the right of each row. Specifically, for SVR models, the optimal SVR model is obtained by fitting the mixed data set, followed by the SVR model fitted by the synthetic data, and the SVR model fitted by the true data has the worst performance, which can be proved by that the average of the corresponding residual errors are 19.885, 20.276, and 20.625, respectively. Likewise, LightGBM and MLP models, show the same patterns as the SVR model, that is, the models fitted by the mixed data show the best performance while the models fitted by the true data show the worst performance. It should be noted that, for SVR, LightGBM, and MLP models fitted by the true data, their KDE plots show that there are individually large values of absolute residual errors, which are close to around 100. This could be the reason why the ML models fitted by the true data do not show good accuracy on fly-rock distance prediction.

For ELM models, it shows the inverse result, that is, the ELM model fitted by the true data shows the best performance, followed by the ELM model fitted by the synthetic data, while the ELM model fitted by the mixed data shows the worst performance. For GMDH models, the GMDH model fitted with the mixed data shows the best performance, followed by the GMDH model fitted with the true data, while the GMDH model fitted with the synthetic data shows the worst performance.

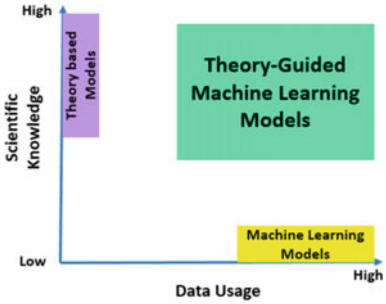
In summary, apart from the ELM models, the remaining four ML models fitted with the mixed data all show the best performance, which indicates that the mixed data is effective and robust to conduct the prediction task of fly-distance prediction.

From the above analysis, it can be concluded that the data augmentation method (i.e., the TVAE model used in the present study) is quite effective, and is capable of enhancing the diversity of the original data. It can greatly enlarge the size of the data set and improve the performance of the fly-rock distance prediction models. Thus, we highly recommend using the data augmentation technique to remedy the drawbacks of the original data set, so as to gain a comprehensive data set.

6 Future Direction

According to [68], theory-guided machine learning (TGML) is a new paradigm that aims to leverage the wealth of scientific knowledge in order to enhance the efficacy of ML models in enabling scientific discovery (Fig. 15). Whenever a component of scientific theories and/or well-known empirical equations is added to the data samples or model output, TGML is built. Commonly, in geotechnical engineering, the databases required for ML techniques are collected from pure laboratory tests or field investigations. On the other hand, there is an imperative need to have a substantial number of data samples for ML techniques, and providing these numbers of reliable data samples is not typically possible in the laboratory/field because it would be expensive and time-consuming. Therefore, using the previous empirical equations in the area of blasting environmental issues such as fly-rock, the number of data samples can be increased. Then, the new database which is actually constructed based on previous relevant theories/empirical equations is ready to be applied in suitable ML techniques for prediction or classification purposes. The developed

Fig. 15 A representation of knowledge discovery methods in scientific applications by TGML



models in this way are more familiar to civil and geotechnical engineers and they can apply these models in real-life problems. The goal of TGML in this situation is to make scientific consistency a necessary component of learning generalizable models with the utilization of well-known theories or empirical equations to establish a comprehensive database which is able to cover a wide range of effective parameters.

7 Conclusion

In this work, a data augmentation technique has been applied to tackle the blast-induced fly-rock issue in the quarry. A fly-rock data set, consisting of seven variables, i.e., hole length, the burden to spacing, stemming, powder factor, the maximum charge per delay, number of holes, and fly-rock distance, was prepared and split into two parts. One, i.e., a training data set containing 131 samples, is used to implement data augmentation, and the other, i.e., a testing data set containing 33 samples, is used to test the model performance. At the data augmentation stage, the TVAE model was used to generate the synthetic data set. Specifically, the synthetic data is synthesized on the basis of the training data set and consists of 1000 samples. Through the verification of some statistical indices, such as boxplot, kernel density estimation, cumulative distribution function, and heatmap, it is proved that the synthetic data set is suitable and effective. Subsequently, at the model testing stage, five classical ML models are trained using the training, synthetic, and mixed data sets, and then tested on the testing data set. The evaluation results show that the R^2 values of the ML models fitted by the synthetic and mixed data sets obtain an improvement of 12.216% and 17.801% compared with the models trained by the training data set, respectively; the RMSE values of the ML models fitted by the synthetic and mixed data sets obtain the reduced errors of 9.151% and 14.593% compared with the models trained by the training data set, respectively; the MAPE values of the ML models fitted by the synthetic and mixed data sets obtain the reduced errors of 2.222% and 11.111% compared with the models trained by the training data set, respectively. Likewise, the evaluation results of absolute residual show that the ML models trained by the mixed data set have the greatest predictive performance compared with other data sets. Thus, we can conclude that the mixed data is effective and robust to conduct the prediction task of fly-distance prediction and the data augmentation technique is highly recommended when encountering similar cases.

References

1. Nguyen H, Bui X-N, Choi Y, Lee CW, Armaghani DJ (2021) A novel combination of whale optimization algorithm and support vector machine with different kernel functions for prediction of blasting-induced fly-rock in quarry mines. Nat Resour Res 30(1):191–207. https://doi.org/10.1007/s11053-020-09710-7
2. Ghasemi E, Sari M, Ataei M (2012) Development of an empirical model for predicting the effects of controllable blasting parameters on flyrock distance in surface mines. Int J Rock Mech Min Sci 52:163–170. https://doi.org/10.1016/j.ijrmms.2012.03.011
3. Raina AK, Murthy VMSR, Soni AK (2015) Estimating flyrock distance in bench blasting through blast induced pressure measurements in rock. Int J Rock Mech Min Sci 76:209–216. https://doi.org/10.1016/j.ijrmms.2015.03.002
4. Jahed Armaghani D, Tonnizam Mohamad E, Hajihassani M, Alavi Nezhad Khalil Abad SV, Marto A, Moghaddam MR (2016) Evaluation and prediction of flyrock resulting from blasting operations using empirical and computational methods. Eng Comput 32(1). https://doi.org/10.1007/s00366-015-0402-5
5. Rezaei M, Monjezi M, Varjani A (2011) Development of a fuzzy model to predict flyrock in surface mining. Saf Sci
6. Manoj K, Monjezi M (2013) Prediction of flyrock in open pit blasting operation using machine learning method. Int J Min Sci Technol 23(3):313–316
7. Armaghani DJ, Hajihassani M, Mohamad ET, Marto A, Noorani SA (2014) Blasting-induced flyrock and ground vibration prediction through an expert artificial neural network based on particle swarm optimization. Arab J Geosci 7(12):5383–5396
8. Marto A, Hajihassani M, Jahed Armaghani D, Tonnizam Mohamad E, Makhtar AM (2014) A novel approach for blast-induced flyrock prediction based on imperialist competitive algorithm and artificial neural network. Sci World J
9. Ghasemi E, Amini H, Ataei M, Khalokakaei R (2014) Application of artificial intelligence techniques for predicting the flyrock distance caused by blasting operation. Arab J Geosci 7(1):193–202
10. Faradonbeh RS, Armaghani DJ, Monjezi M, Mohamad ET (2016) Genetic programming and gene expression programming for flyrock assessment due to mine blasting. Int J Rock Mech Min Sci 88:254–264
11. Saghatforoush A, Monjezi M, Faradonbeh RS, Armaghani DJ (2016) Combination of neural network and ant colony optimization algorithms for prediction and optimization of flyrock and back-break induced by blasting. Eng Comput 32(2):255–266
12. Kumar N, Mishra B, Bali V (2018) A novel approach for blast-induced fly rock prediction based on particle swarm optimization and artificial neural network. In: Proceedings of international conference on recent advancement on computer and communication. Springer, pp 19–27
13. Nguyen H, Bui X-N, Nguyen-Thoi T, Ragam P, Moayedi H (2019) Toward a state-of-the-art of fly-rock prediction technology in open-pit mines using EANNs model. Appl Sci 9(21):4554
14. Jamei M, Hasanipanah M, Karbasi M, Ahmadianfar I, Taherifar S (2021) Prediction of flyrock induced by mine blasting using a novel kernel-based extreme learning machine. J Rock Mech Geotech Eng 13(6):1438–1451
15. Murlidhar BR, Nguyen H, Rostami J, Bui X, Armaghani DJ, Ragam P, Mohamad ET (2021) Prediction of flyrock distance induced by mine blasting using a novel Harris Hawks optimization-based multi-layer perceptron neural network. J Rock Mech Geotech Eng 13(6):1413–1427

16. Bhagat NK, Rana A, Mishra AK, Singh MM, Singh A, Singh PK (2021) Prediction of fly-rock during boulder blasting on infrastructure slopes using CART technique. Geomat Nat Haz Risk 12(1):1715–1740. https://doi.org/10.1080/19475705.2021.1944917
17. Shamsi R, Amini MS, Dehghani H, Bascompta M, Jodeiri Shokri B, Entezam S (2022) Prediction of Fly-rock using gene expression programming and teaching—learning-based optimization algorithm. J Min Environ 13(2):391–406. https://doi.org/10.22044/jme.2022.11825.2171
18. Asteris PG, Argyropoulos I, Cavaleri L, Rodrigues H, Varum H, Thomas J, Lourenço PB (2018) Masonry compressive strength prediction using artificial neural networks. In: International conference on transdisciplinary multispectral modeling and cooperation for the preservation of cultural heritage. Springer, pp 200–224
19. Asteris PG, Lourenço PB, Roussis PC, Adami CE, Armaghani DJ, Cavaleri L, Mohammed AS et al (2022) Revealing the nature of metakaolin-based concrete materials using artificial intelligence techniques. Constr Build Mater 322:126500
20. Asteris PG, Rizal FIM, Koopialipoor M, Roussis PC, Ferentinou M, Armaghani DJ, Gordan B (2022) Slope stability classification under seismic conditions using several tree-based intelligent techniques. Appl Sci 12(3):1753
21. Barkhordari M, Armaghani D, Asteris P (2022) Structural damage identification using ensemble deep convolutional neural network models. CMES Comput Model Eng Sci. https://doi.org/10.32604/cmes.2022.020840
22. He B, Armaghani DJ, Lai SH (2022) A short overview of soft computing techniques in tunnel construction. Open Constr Build Technol J 16(1):1–6. https://doi.org/10.2174/18748368-v16-e2201120
23. Koopialipoor M, Asteris PG, Mohammed AS, Alexakis DE, Mamou A, Armaghani DJ (2022) Introducing stacking machine learning approaches for the prediction of rock deformation. Transp Geotech 34:100756
24. Li C, Zhou J, Tao M, Du K, Wang S, Armaghani DJ, Mohamad ET (2022) Developing hybrid ELM-ALO, ELM-LSO and ELM-SOA models for predicting advance rate of TBM. Transp Geotech 36:100819
25. Liu Z, Armaghani DJ, Fakharian P, Li D, Ulrikh DV, Orekhova NN, Khedher KM (2022) Rock strength estimation using several tree-based ML techniques. CMES Comput Model Eng Sci. https://doi.org/10.32604/cmes.2022.021165
26. Shan F, He X, Armaghani DJ, Zhang P, Sheng D (2022) Success and challenges in predicting TBM penetration rate using recurrent neural networks. Tunn Undergr Space Technol 130:104728
27. Zeng J, Asteris PG, Mamou AP, Mohammed AS, Golias EA, Armaghani DJ, Hasanipanah M et al (2021) The effectiveness of ensemble-neural network techniques to predict peak uplift resistance of buried pipes in reinforced sand. Appl Sci 11(3):908
28. Zhou J, Zhu S, Qiu Y, Armaghani DJ, Zhou A, Yong W (2022) Predicting tunnel squeezing using support vector machine optimized by whale optimization algorithm. Acta Geotech 7. https://doi.org/10.1007/s11440-022-01450-7
29. Hasanipanah M, Armaghani DJ, Amnieh HB, Majid MZA, Tahir MMD (2017) Application of PSO to develop a powerful equation for prediction of flyrock due to blasting. Neural Comput Appl 28(1):1043–1050
30. Ohno H (2020) Auto-encoder-based generative models for data augmentation on regression problems. Soft Comput 24(11):7999–8009. https://doi.org/10.1007/s00500-019-04094-0
31. Huang Y, Liu DR, Lee SJ, Hsu CH, Liu YG (2022) A boosting resampling method for regression based on a conditional variational autoencoder. Inf Sci 590:90–105. https://doi.org/10.1016/j.ins.2021.12.100
32. Kecojevic V, Radomsky M (2005) Flyrock phenomena and area security in blasting-related accidents. Saf Sci 43(9):739–750
33. Nayak NP, Jain A, Ranjan Mahapatra S (2021) Application of mine excellence software in flyrock prediction & mitigation. Mater Today Proc 48:1271–1276. https://doi.org/10.1016/j.matpr.2021.08.282

34. Mohamad ET, Yi CS, Murlidhar BR, Saad R (2018) Effect of geological structure on flyrock prediction in construction blasting. Geotech Geol Eng 36(4):2217–2235
35. Pour AE, Afrazi M, Golshani A (2022) Experimental study of the effect of length and angle of cross-cracks on tensile strength of rock-like material. Iran J Sci Technol Trans Civ Eng 46:4543–4556. https://doi.org/10.1007/s40996-022-00891-0
36. Afrazi M, Lin Q, Fakhimi A (2022) Physical and numerical evaluation of mode II fracture of quasi-brittle materials. Int J Civ Eng 20:993–1007. https://doi.org/10.1007/s40999-022-00718-z
37. Majedi MR, Afrazi M, Fakhimi A (2021) A micromechanical model for simulation of rock failure under high strain rate loading. Int J Civ Eng 19:501–515. https://doi.org/10.1007/s40999-020-00551-2
38. Gomes-Sebastiao GL, De Graaf WW (2017) An investigation into the fragmentation of blasted rock at Gomes Sand. J South Afr Inst Min Metall 117(4):321–328. https://doi.org/10.17159/2411-9717/2017/v117n4a2
39. Jahed Armaghani D, Hajihassani M, Monjezi M, Mohamad ET, Marto A, Moghaddam MR (2015) Application of two intelligent systems in predicting environmental impacts of quarry blasting. Arab J Geosci 8(11):9647–9665. https://doi.org/10.1007/s12517-015-1908-2
40. Smiti A (2020) A critical overview of outlier detection methods. Comput Sci Rev 38:100306
41. Wickham H, Stryjewski L (2011) 40 years of boxplots, pp 1–17. Had.Co.Nz
42. Leonard S, Carroll RJ (1990) Deconvoluting kernel density estimators. Statistics 21(2):169–184. https://doi.org/10.1080/02331889008802238
43. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: 2nd international conference on learning representations, ICLR 2014—conference track proceedings, (Ml), pp 1–14
44. Kingma DP, Welling M (2019) An introduction to variational autoencoders. Found Trends Mach Learn 12(4):307–392. https://doi.org/10.1561/2200000056
45. Miles C, Carbone MR, Sturm EJ, Lu D, Weichselbaum A, Barros K, Konik RM (2021) Machine-learning Kondo physics using variational autoencoders, pp 1–18
46. Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K (2019) Modeling tabular data using conditional GAN. Adv Neural Inf Process Syst 32(NeurIPS)
47. Lei X (2020) Synthesizing tabular data using conditional GAN. Massachusetts Institute of Technology
48. Cortes C, Vapnik V (1995) Support vector machine. Mach Learn 20(3):273–297
49. Noble WS (2006) What is a support vector machine? Nat Biotechnol 24(12):1565–1567. https://doi.org/10.1038/nbt1206-1565
50. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Liu TY et al (2017) LightGBM: a highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 3147–3155
51. Li K, Xu H, Liu X (2022) Analysis and visualization of accidents severity based on LightGBM-TPE. Chaos Solitons Fractals 157:111987. https://doi.org/10.1016/j.chaos.2022.111987
52. Wen X, Xie Y, Wu L, Jiang L (2021) Quantifying and comparing the effects of key risk factors on various types of roadway segment crashes with LightGBM and SHAP. Accid Anal Prev 159(June):106261. https://doi.org/10.1016/j.aap.2021.106261
53. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501. https://doi.org/10.1016/j.neucom.2005.12.126
54. Huang G-B, Zhu Q-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE international conference on neural networks—conference proceedings, vol 2, pp 985–990. https://doi.org/10.1109/IJCNN.2004.1380068
55. Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 42(2):513–529. https://doi.org/10.1109/TSMCB.2011.2168604
56. Madala HR, Ivakhnenko AG (2019) Inductive learning algorithms for complex systems modeling. In: Inductive learning algorithms for complex systems modeling. https://doi.org/10.1201/9781351073493

57. Lemke F, Mueller JA (2003) Medical data analysis using self-organizing data mining technologies. Syst Anal Model Simul 43(10):1399–1408. https://doi.org/10.1080/02329290290027337
58. Noriega L (2005) Multilayer perceptron tutorial. In: School of computing. Staffordshire University, pp 1–12
59. Gardner MW, Dorling SR (1998) Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmos Environ 32(14–15):2627–2636
60. Ramchoun H, Amine M, Idrissi J, Ghanou Y, Ettaouil M (2016) Multilayer perceptron: architecture optimization and training. Int J Interact Multimedia Artif Intell 4(1):26. https://doi.org/10.9781/ijimai.2016.415
61. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
62. Patki N, Wedge R, Veeramachaneni K (2016) The synthetic data vault. In: Proceedings—3rd IEEE international conference on data science and advanced analytics, DSAA 2016, pp 399–410. https://doi.org/10.1109/DSAA.2016.49
63. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(11)
64. Drew JH, Glen AG, Leemis LM (2000) Computing the cumulative distribution function of the Kolmogorov-Smirnov statistic. Comput Stat Data Anal 34(1):1–15. https://doi.org/10.1016/S0167-9473(99)00069-9
65. Sarkhani Benemaran R, Esmaeili-Falak M, Javadi A (2021) Predicting resilient modulus of flexible pavement foundation using extreme gradient boosting based optimized models. SSRN Electron J 1–20. https://doi.org/10.2139/ssrn.3986942
66. Taylor KE (2001) Summarizing multiple aspects of model performance in a single diagram. J Geophys Res Atmos 106(D7):7183–7192. https://doi.org/10.1029/2000JD900719
67. Sadrossadat E, Heidaripanah A, Osouli S (2016) Prediction of the resilient modulus of flexible pavement subgrade soils using adaptive neuro-fuzzy inference systems. Constr Build Mater 123:235–247. https://doi.org/10.1016/j.conbuildmat.2016.07.008
68. Karpatne A, Atluri G, Faghmous J, Steinbach M, Banerjee A, Ganguly A, Kumar V et al (2017) Theory-guided data science: a new paradigm for scientific discovery. IEEE Trans Knowl Data Eng 29(10):2318–2331
69. Mohamad ET, Armaghani DJ, Motaghedi H (2013) The effect of geological structure and powder factor in flyrock accident, Masai, Johor, Malaysia. Electron J Geotech Eng 18:5561–5572

Forecast of Modern Concrete Properties Using Machine Learning Methods

Yashar Asghari, Golnaz Sadeghian, Seyed Esmaeil Mohammadyan-Yasouj, and Elahe Mirzaei

1 Introduction

In past decades, investigations have been conducted about the mechanical, fresh, and durability properties of concrete in terms of predictions by using ML methods. Researchers have extensively considered Self-consolidating concrete (SCC), ultra-high performance concrete (UHPC), alkali-activated concrete (AAC), recycled aggregate concrete (RAC), and geopolymer concrete (GC) in recent years. Machine learning algorithms have been broadly applied in different fields to evaluate predictive outcomes that are closely related to experiments. The results of a test could, however, be affected by a complex matrix of parameters, of which the majority contribute little to the test results. Consequently, computer scientists are required to develop novel selection algorithms based on data-driven models in order to identify the most relevant independent variables and reduce the dimensionality of the input matrix as quickly as possible. There is increasing use of soft computing tools in predicting engineering components, systems, and materials, with ANNs being one of the most popular soft computing paradigms that have been successfully applied in many areas of engineering [1].

Y. Asghari
Department of Civil Engineering, Sahand University of Technology, Tabriz, Iran
e-mail: y_asghari@sut.ac.ir

G. Sadeghian
Independent Researcher, Isfahan, Iran
e-mail: g.sadeghian@alumni.iut.ac.ir

S. E. Mohammadyan-Yasouj (✉)
Department of Civil Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran
e-mail: semy2016@pci.iaun.ac.ir

E. Mirzaei
Department of Civil Engineering, Tarbiat Modares University, Tehran, Iran

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*, Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_7

2 Machine Learning Concept

The concept of ML refers to a group of methods for discovering affinities between quantities of interest from data. ML approaches are used in different fields to understand how a system works and make predictions about unobserved quantities by using relationships between inputs and outputs [2].

Four examples of using ML in civil engineering.

- In geotechnical field: It is crucial to forecast the resistance of the sites that have not been observed using discrete observations of soil resistances.
- In the environment: To comprehend the influences of temperature and fertilizers on the prevalence of cyanobacteria, fish mortality, and the color of the water.
- In the transportation field: Surveys and transit card use are used to predict the demand for public transportation.
- In structures: It is essential to detect anomalous behavior in a structure based on observations of its displacement over time [2].

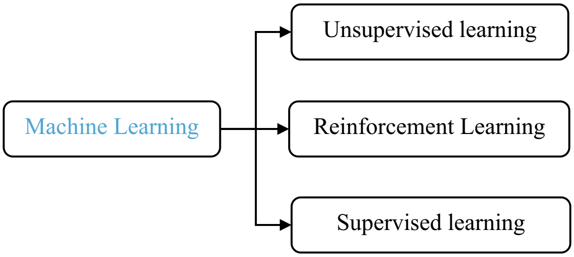
There are some different standards that can be used to categorize ML techniques. Methods of ML can be categorized based on the need for human supervision throughout the training period. With regard to .

Figure 1, based on the type of learning being performed, ML techniques are generally categorized into three groups including supervised, unsupervised, and reinforcement learning [3].

A supervised learning method trains on a labeled dataset before generating forecasts on an unlabeled dataset. The usage of unsupervised learning consists of training a model on the unlabeled dataset in a way that it can learn automatically from it by extracting characteristics and trends. Reinforcement learning involves training an agent on the environment, which allows the agent to identify the most suitable solution in a complicated case [4, 5].

In response to the constantly changing demands of the construction industry, the increased development of novel concrete types has encouraged further investigation into forecasting models that are able to foresee different properties of concrete. In order to meet the requirements of various design codes and guidelines, researchers have been attempting to anticipate the mechanical properties of concrete. In prior

Fig. 1 The major classification of ML



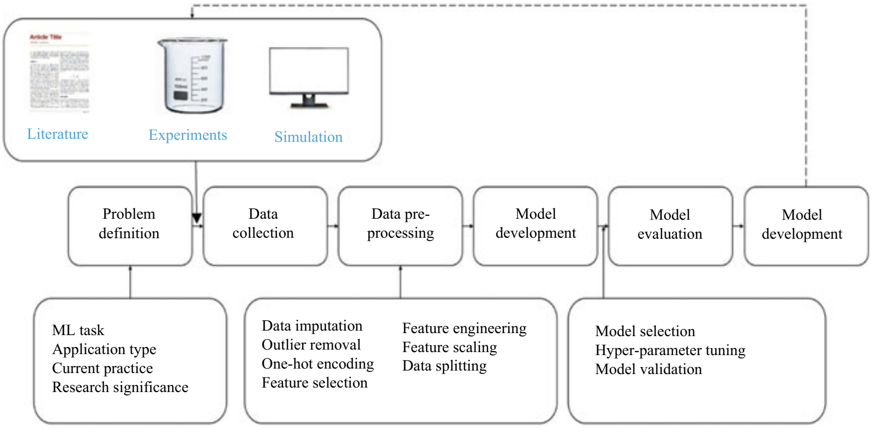


Fig. 2 An overview of the ML process in concrete science [9–11]

techniques, linear and non-linear regression approaches were utilized to predict different properties of concrete through analytical analysis of laboratory data [6, 7].

The use of ML models has been extensively utilized in recent years for forecasting the mechanical properties of concrete as an effective tool. The models are usually applied to large datasets separated into training, validation, and testing groups. While the training datasets are used for model training, the validation datasets provide an unbiased assessment of the model concerning the training data and avoid overfitting by preventing the training phase as error levels rise. As a final step, the model is used for the testing dataset in order to assess its predictive power [8].

As shown in Fig. 2 a typical ML study consists of six steps:

- Problem description
- Data collection
- Data pre-processing
- Model development
- Model assessment
- Model deployment.

Several statistical methods have been applied to evaluate the performance of ML methods. Based on Table 1, ML models can be evaluated using statistical indices that indicate how closely predicted values match actual values [8].

Recently, ML methods have been developed to forecast a wide range of concrete properties. The most commonly used ML methods are artificial neural network (ANN), fuzzy logic (FL), decision tree (DT), support vector machine (SVM), gene expression programming (GEP), and Bagging and Boosting.

Figure 3 shows the different ML techniques subjected to investigation in terms of predicting different properties of concrete in this study. As Fig. 3 indicates, ML methods can be divided into three major types: methods that can be categorized as

Table 1 Statistical index

Statistical index	Formula
Correlation coefficient (R)	$R = \frac{n\sum_{i=1}^{n} y_i^{'} y_i - \left(\sum_{i=1}^{n} y_i^{'}\right)\left(\sum_{i=1}^{n} y_i\right)^2}{\sqrt{n\left(\sum_{i-1}^{n} {y_i^{'}}^2\right) - \left(\sum_{i-1}^{n} y_i^{'}\right)^2}\sqrt{n\left(\sum_{i-1}^{n} y_i^2\right) - \left(\sum_{i-1}^{n} y_i\right)^2}}$
Mean square error (MSE)	$\mathrm{MSE} = \frac{\sum_{i-1}^{n}\left(y_i^{'} - y_i\right)^2}{n}$
Coefficient of determination (R^2)	$R^2 = 1 - \frac{\sum_{i-1}^{n}\left(y_i^{'} - y_i\right)^2}{\sum_{i}^{n}(y_i - \overline{y})^2}$
Root mean square error (RMSE)	$\mathrm{RMSE} = \sqrt{\frac{\sum_{i-1}^{n}\left(y_i^{'} - y_i\right)^2}{n}}$
Mean absolute error (MAE)	$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left\lvert y_i^{'} - y_i\right\rvert$
Mean absolute percentage error (MAPE)	$\mathrm{MAPE}\ (\%) = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i^{'} - y_i}{y_i}\right\rvert \times 100$

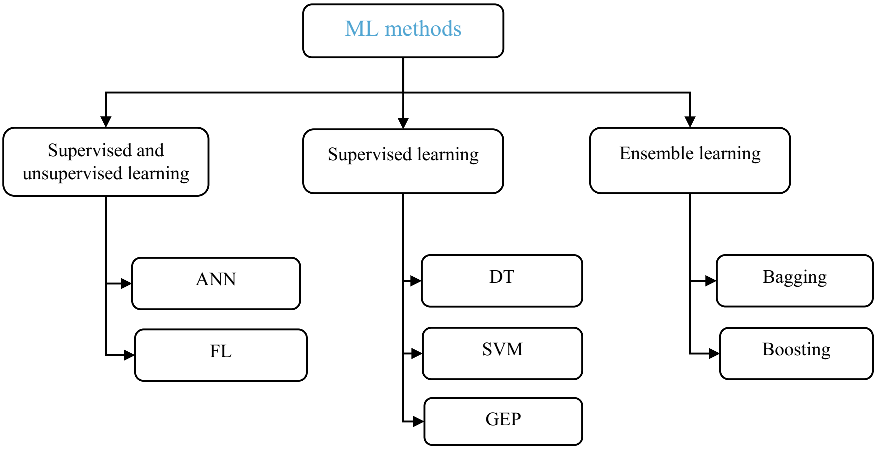


Fig. 3 Machine learning models

both supervised and unsupervised learning, methods based on supervised learning, and ensemble learning.

2.1 *Supervised Learning Methods*

Machine learning algorithms consider every dataset instance as a collection of features. These features can be categorical, binary, or continuous. In supervised

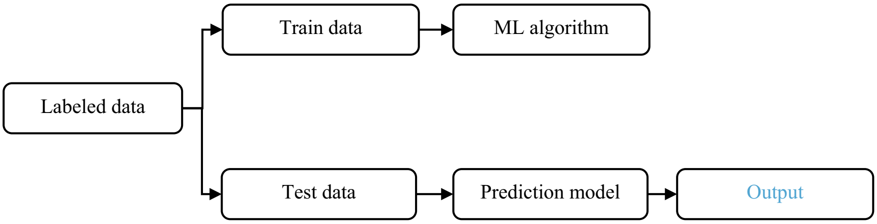


Fig. 4 An overview of the fundamental structure of supervised learning

learning, instances are labeled in the process of learning [12]. Supervised learning involves training the model on labeled data and testing it on unlabeled data. A fundamental aspect of its architecture is the collection of datasets; the datasets are divided into testing and training data, and then the data are pre-processed. To learn the features associated with each label, an algorithm is fed the extracted features and then trained with the data. As shown in Fig. 4, once the model is provided with the test data, it makes predictions based on the data by providing the expected labels.

2.2 Support Vector Machine

A SVM is a supervised learning method popular for performing classification and regression analyses based on data analysis and pattern recognition. Various methods are available depending on the structure and attributes of the classifier. A linear classifier is the most commonly known SVM, predicting the class of each input between two possible classifications.

Figure 5 illustrates that SVMs are constructed by building a hyperplane or set of hyperplanes to categorize all input data in a high-dimensional or infinite space. A support vector is a set of values closest to the classification margin. The SVM aims to maximize the margin between the hyperplane and the support vectors. Among many off-the-shelf classifiers, SVMs are top-rated. Additionally, SVM can be implemented in a variety of environments and toolboxes. These reasons led us to choose SVM as the method of classification of infeasible test cases [13].

It should be noted, however, that some classes are unable to be divided using a linear hyperplane, as shown in Fig. 6. To achieve linear class separation, the input space must be mapped into a higher-dimensional feature space [14].

An essential characteristic of the non-linear mapping technique is that it is typically conducted through non-linear functions. A kernel function is used to determine the output of the algorithm from non-linear space [15, 16].

It is possible to categorize these functions into five groups:

- Polynomial
- exponential radial basis
- radial basis

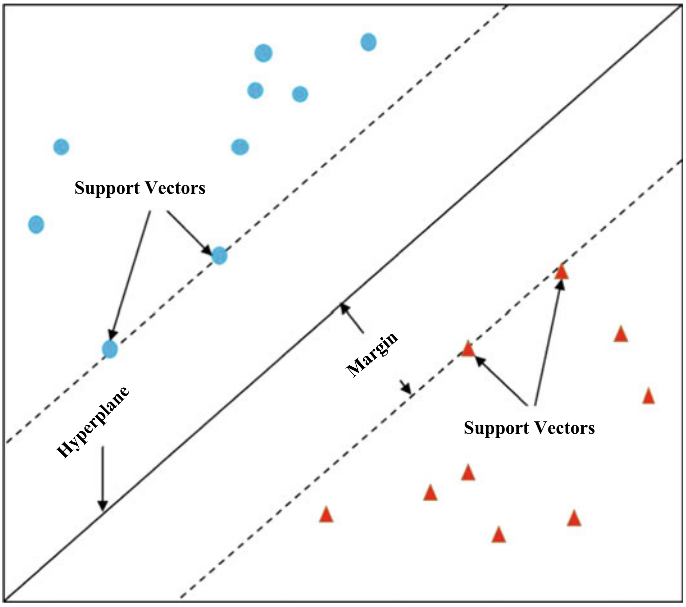


Fig. 5 Hyperplane classification

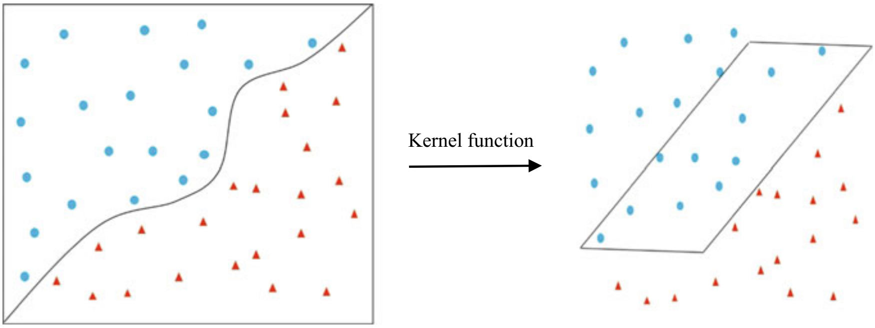


Fig. 6 Non-linear mapping in SVM

- Sigmoid
- linear [17].

This technique enables the determination of a non-linear decision boundary without calculating the optimum hyperplane parameters within the feature space. As a result, the solution can be represented as a combination of the weighted values of kernel functions at a support vector [15].

In general, support vector regression (SVR) is referred to as SVM when it is applied primarily for regression analysis.

2.3 Hybrid SVM-Based Models

The modified firefly algorithm (MFA), which is based on the flashing characteristics and behavior of tropical fireflies, is a recently developed nature-inspired metaheuristic method [18].

There are three main idealized rules that MFA follows:

- A firefly, because of its unisex gender, can be attracted by another firefly of any gender.
- Brightness and attractiveness are correlated, so the less bright fireflies will be attracted to the more brilliant ones, and the more distance a firefly is from another, the less attractive it becomes.
- An objective function is used to measure the brightness of a firefly [19].

By incorporating SVM into hybrid approaches, it is possible to enhance the performance and efficiency of individual SVM techniques. A number of investigations have employed MFA, for example, as an optimization method to assess compressive strength and the shear strength of concrete [8].

Bui et al. [19] utilized the MFA-ANN model to predict the mechanical properties of High-performance concrete (HPC). The authors compared the ANN hybrid model with the smart firefly algorithm-based Least Square Support Vector Regression (SFA-LSSVR) developed by Chou et al. [20]. The MFA-ANN hybrid system can better predict HPC performance concrete properties and solve problems more quickly based on the results.

The response surface methodology (RSM) involves optimizing factorial variables so that the output achieves the desired peak or lowest value. Keshtegar et al. [21] generated an RSM-SVM hybrid model based on SEM and SVM. The model was used to predict the shear strength of steel fiber-unconfined reinforced concrete beam. The hybrid RSM-SVM model was compared with different individual ML techniques, including RSM, SVR, and classical neural networks. Compared to the other models, the RSM-SVR model was more accurate.

2.4 Decision Tree Model

Decision trees consist of models that integrate basic tests, individually evaluating a numerical attribute with a threshold value or a nominal metric with a range of probable values. Unlike the numerical weights of the links between the nodes of a neural network, the logical rules followed by DT are much more straightforward to understand. It is more comfortable for decision-makers to use models they are familiar with. A DT classifies data points within a partitioned region according to the most prevalent class within that region. An error rate is calculated by dividing the number of misclassified data points by the total number of data points, and an accuracy rate is calculated by subtracting the error rate from one. Several programs have been developed that perform automatic induction of DT. Typically, these programs need labeled instances, that is, a collection of previously acquired data characterized by a class label. Algorithms are designed to describe or find patterns in data. As a result, it determines which tests (questions) best divide the instances into separate classes, resulting in a tree-like structure [22].

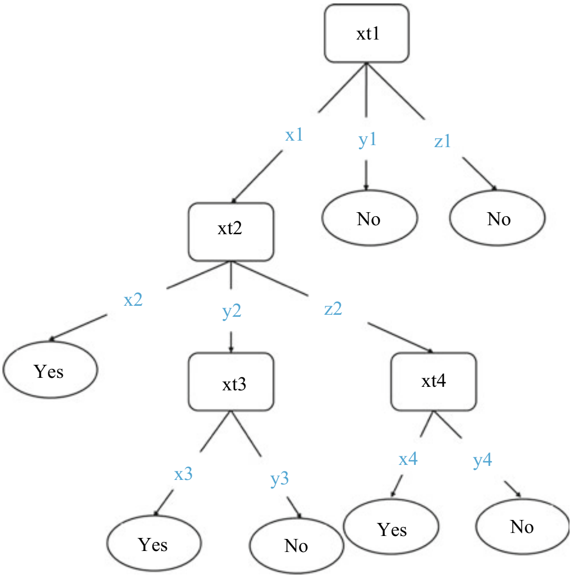
Table 2 is an instance of the training set to generate DT. Developing the DT illustrated in Fig. 7 as an example, it is assumed that xt1 = x1, xt2 = y2, xt3 = x3, and xt4 = y4 would sort to the nodes: xt1 to xt3, which would categorize the instance as being positive (shown by Yes).

The DT induction process has two significant steps:

Table 2 An example of training set

xt1	xt2	xt3	xt4	Class
x1	x2	x3	x4	Yes
x1	a2	x3	b4	Yes
x1	b2	x3	x4	Yes
x1	b2	y3	y4	No
x1	z2	x3	x4	Yes
x1	z2	x3	y4	No
y1	y2	y3	y4	No
z1	y2	y3	y4	No

Fig. 7 A general DT



- Growth phase
- Pruning phase.

The growth step involves a recursive division of the training dataset resulting in a DT such that either each leaf node is connected with a single class or further division of the given leaf results in at least child nodes being below some specified threshold. The pruning step seeks to develop the DT that was developed in the growth step by creating a sub-tree that sidesteps overfitting to the training data. As opposed to the pre-pruning that happens during the growth phase and seeks to control splits that do not meet certain thresholds, the pruning phase is usually a post-pruning phase.

2.5 Gene-Expression Programming (GEP)

Gene Expression Programming is a well-established evolutionary algorithm for automatic generation computer programs. In GEP, chromosomes/expression trees form a truly functional, indivisible system. It should be noted that in GEP there is no such thing as an invalid expression tree or program. It is evident that the interaction between GEP chromosomes and expression trees requires a system that can translate the language of chromosomes into the language of expression trees without ambiguity. Additionally, the structural organization of GEP chromosomes allows for unconstrained genome modification, which makes evolution possible. Gene expression programming is an uncomplicated artificial life system that has been developed beyond the replicator threshold due to the diverse set of genetic operators designed to familiarize genetic modification into populations [23]. As a result of rapid advancements in GEP over the past few decades, the method has been widely adopted for forecasting the mechanical properties of concrete, so different researchers utilized GEP to forecast the compressive strength of concrete [19, 24–27].

2.6 Artificial Neural Network Methods

Artificial neural network methods are extensively employed in the field of artificial intelligence as well as for the solving of engineering problems. It is mainly used as a forecasting model because it requires no prior knowledge and has high accuracy. Artificial neural network models predict an output based on a set of input data from a given problem domain after learning from past data the patterns of an underlying process and generalizing the knowledge gained (or the mathematical relationships between the input and output data) [28, 29].

Parallel-interconnected neurons have been used to design ANNs. Weights have been assigned to the connections. As Fig. 8 illustrates in general, the ANN has three layers, namely input, hidden, and output. Input patterns are fetched from an external environment by the input layer. The hidden layer separates input and output. There may be more than one hidden layer. The number of hidden layers has been calculated by various methods, but no exact formula has been published. The output layer is responsible for gathering and transferring information as modeled [30].

In the input layer, neurons receive inputs and process them with the chosen weights before passing the weighted sum through some activation function to produce the final ANN output. Table 3 reviews different activation functions which frequently used to develop ANN models [31]. Furthermore, the error term is calculated based on the output of the ANN and the target. The weights are updated based on the error term after it has been calculated. Further, the procedure continues until the desired result is achieved or the error has been minimized [30].

According to Fig. 9, the bias b_k raises or decreases the net input of the activation function, depending on its positive or negative, respectively. Equations 1 and 2 illustrate the mathematical representation of the neuron k, which Fig. 9 represented.

Fig. 8 A simple architecture of ANN

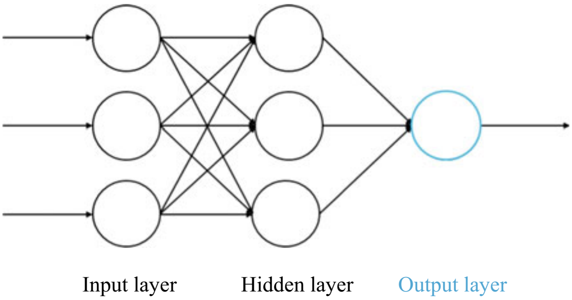


Table 3 Summery of activation functions in ANN

Activation function	$\phi(x)$	$\phi'(x)$	Values
Signum function	$sgn(x) = \begin{cases} -1\ for\ x < 0 \\ 0\ for\ x = 0 \\ 1\ for\ x > 0 \end{cases}$	$\delta 2(x)$	$[-1, 1]$
ReLu	$R(x) = \left\{ \begin{matrix} 0\ for\ x < 0 \\ x\ for\ x \geq 0 \end{matrix} \right\}$	$\left\{ \begin{matrix} 0\ for\ x < 0 \\ 1\ for\ x \geq 0 \end{matrix} \right\}$	$[0, \infty)$
Sigmoid	$S(x) = \frac{1}{1+e^{-x}}$	$\phi(x)(1 - \phi(x))$	$(0, 1)$
Heaviside function	$H(x) = \left\{ \begin{matrix} 0\ for\ x < 0 \\ 1\ for\ x \geq 0 \end{matrix} \right\}$	$\delta(x)$	$[0, 1]$
Hyperbolic tangent	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - \phi(x)^2$	$(-1, 1)$
Softmax	$y_i = \frac{e^{x_i}}{\sum_j^n e^{x_j}}$	$\frac{\partial y_i}{\partial j} = y_i\left(\delta_{ij} - y_j\right)$	$(0, 1)$

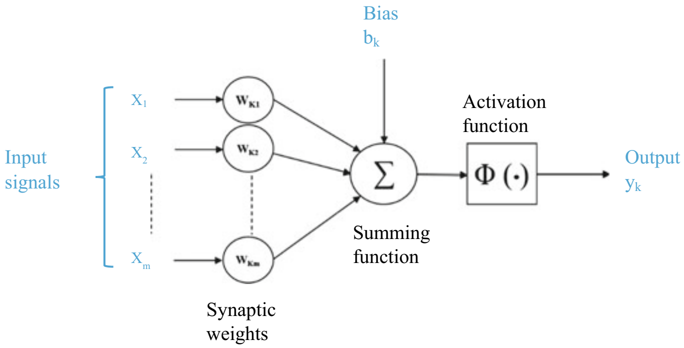


Fig. 9 An illustration of a mathematical artificial neuron named *k* [32]

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

and

$$y_k = \emptyset(u_k + b_k) \tag{2}$$

An Overview of Network Architectures

Different network architectures are used in different situations as ANN models, such as:

- Convolutional Neural Networks (CNNs)
- Deep Belief Networks (DBNs)
- Deep Feedforward Neural Networks (D-FFNN)
- Auto encoders (AEs)
- Long Short-Term Memory networks (LSTMs) [31].

Learning Rules

It is the main objective of ML to make a machine intelligent. For a machine to be intelligent, it must be able to learn from its experiences or from examples. Learning in ANN is generally divided into three types:

- Supervised
- Unsupervised
- Reinforcement learning [30].

It has been reported in several studies that ANNs have demonstrated superior abilities for modeling and solving complex civil engineering problems after the publication of the first article that applied ANNs to civil engineering in 1989. For instance, ANNs have been successfully applied to investigate hardened, fresh, and durability properties such as compressive [25], flexural [33], tensile strengths [34], elastic modulus [35], electrical resistivity [36], chloride penetration resistance [37], mass-loss and volume-loss under HCl attack [38], rheological behavior of SCC [39], and self-healing capacity [40].

2.7 Hybrid ANN-Based Models

Hybrid approaches are based on the idea of combining several algorithms so that the performance and efficiency of the model can be significantly improved by combining these algorithms. Researchers are increasingly interested in hybrid approaches because they combine the advantages of several models [8].

Adaptive Neuro-Fuzzy Inference System (ANFIS) is a well-known Hybrid ANN-based model in which ANN and FL are combined to give a universal approximation. In this model, ANN is utilized to increase membership capacities to reduce error rates in the output, while FL rules are applied to provide expert knowledge [41]. ANFIS models are widely used in concrete science to predict the compressive strength of Ordinary Concrete (OC) [7], SCC [42], HPC [43], and GC [41].

It is possible to optimize the weights and thresholds of ANN by using genetic algorithms (GA). GA is a metaheuristic technique concerning the concept of natural evolution and selection [7, 44, 45].

The GN-ANN model was used by Yuan et al. [7] to forecast the compressive strength of concrete. The results indicated that the GN-ANN model performed well regarding desirable accuracy and applicability in real-life engineering practice, demonstrating its potential to replace conventional regression models. An expert system generated by Bui et al. [19] used the ANN model in conjunction with MFA to predict the mechanical properties of HPC. In conclusion, the hybrid system combining MFA and ANN could better predict HPC concrete properties. Additionally, the MFA-ANN solves problems much faster. Consequently, the proposed approach could be used to predict and design HPC in an efficient and accurate manner.

2.8 Fuzzy Logic (FL)

The FL theory is a relatively new study and is widely applied to the classification of remotely sensed images, as well as the classification of multiple land use and land cover classes. It is possible to apply a fuzzy approach in the context of supervised and unsupervised classification.

A broad fuzzy system is illustrated in.

Figure 10 comprises four main components, including fuzzification, fuzzy rule base, fuzzy output engine, and defuzzification. By using one or more membership functions, fuzzification converts each piece of input data into degrees of membership. In fact, FL is based on the idea that objects can belong to different subsets of a

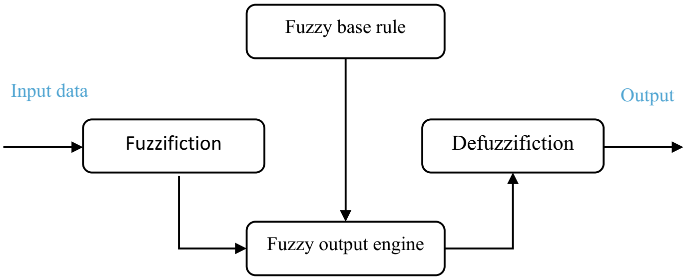


Fig. 10 General fuzzy system

universal set instead of belonging to that single entirely. It is possible to quantify partial belonging to a set numerically by a membership function, which is a function that assumes values between 0 and 1. Among many techniques, intuition, inference, rank ordering, angular fuzzy sets, neural networks, genetic algorithms, and inductive reasoning can be ways to assign membership values or functions to fuzzy variables. It is possible to construct fuzzy membership functions in various ways, but simple linear functions, such as triangular ones, are preferred for practical applications [46].

2.9 Ensemble Learning Methods

Typically, ensemble methods consist of a set of learning machines that combine their decisions, their learning algorithms, different views of data, or any other unique feature that results in more accurate and reliable predictions for supervised and unsupervised learning problems [47, 48]. Ensemble learning methods are divided into three main groups, including bagging, boosting, and stacked generalization. An ensemble-based algorithm called bagging is one of the earliest and simplest algorithms on the market. Despite its simplicity, the bagging algorithm is highly efficient and is best suited to problems with limited training datasets [49].

A boosting algorithm is an iterative process for generating a strong classifier that can achieve arbitrarily low training error from an ensemble of weak classifiers, each of which is barely better than random guessing. A significant difference between boosting and bagging is the way it differs from combining weak classifiers with simple majority voting. As a result of bagging, each instance selected for classifier training has an equal chance of being in any of the training datasets. It should be noted, however, that in boosting, the training dataset for each successive classification algorithm focuses increasingly on instances that the previous classification algorithm had misclassified [49, 50]. Wang et al. [33] used several ensembled ML techniques to predict UHSC flexural strength. In this study DT was compared to ensembled ML models include bagging, gradient boosting (GB), extreme gradient boosting (XGBoost), and Adaboost. The authors reported that when compared to the individual DT model, ensemble ML techniques had better performance.

3 Using ML Methods in Concrete Science

As shown in Fig. 11, based on the review in this study, OC is the most common concrete and SCC, UHPC, GC, AAC, and RAC as the modern concretes are the concrete types noticed by using ML methods.

3.1 Ordinary Concrete

A common form of concrete is OC which is regularly used in building pavements and buildings that do not need very high tensile strength. Many studies have been conducted on the investigation of its properties using different ML techniques. Regarding Table 4, some researchers investigated the potential of ML algorithms such as ANN, DT, GEP, Multi Logistic Regression (MLR), Gaussian Process Regression (GPR), and Full Quadratic (FQ) methods to predict different properties of OC.

3.2 Self-Consolidation Concrete

In the late 1980s, SCC emerged in Japan as a material capable of flowing and compacting under its own weight without vibrating. In complex formwork, reinforced structural elements with congested reinforcement, and hard-to-reach areas, SCC can be placed without additional mechanical compaction.

Compared to conventional concrete, unit SCC has three main features:

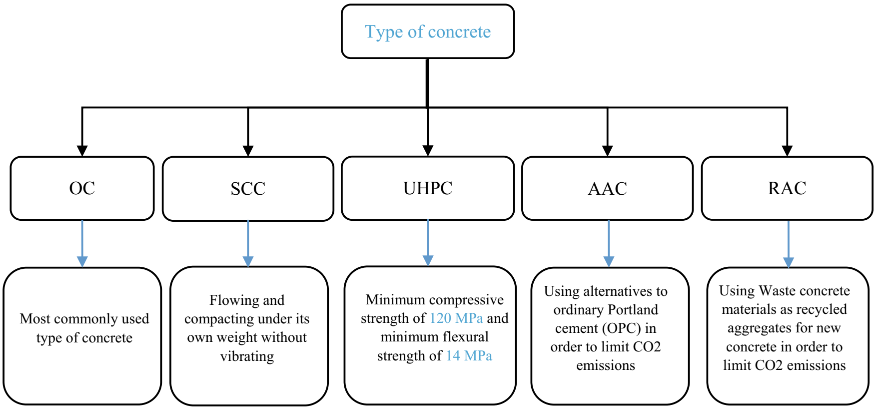


Fig. 11 Concrete types reviewed in this study

Table 4 Summary of using ML in OC

Refs.	Year	Input	Output	Prediction methods
Penido et al. [51]	2022	Cement, fine steel slag, coarse steel slag, fine aggregate, coarse aggregate, water, superplasticizer (SP), pozzolanic admixtures, supplementary cementitious materials (SCMs), filler, age	Compressive strength	GPR, SVR, XGBoost, ANN
Piro et al. [36]	2022	Water to cement ratio (w/c), steel slag, ground granulated blast-furnace slag (GGBFS), age, cement, fine aggregate, coarse aggregate, SP	Electrical current, compressive strength	MLR, FQ, ANN
Song et al. [24]	2021	Cement, fly ash, SP, water, Fine aggregate, coarse aggregate, age	Compressive strength	ANN, Boosting regressor, GEP, DT
Ahmad et al. [26]	2021	Cement, fine aggregate, coarse aggregate, water, waste material, age, water to binder ratio (w/b), SP	Compressive strength	GEP, DT, Bagging
Kandiri et al. [52]	2020	Cement, granulated blast furnace slag (GBFS), water, GBFS grade, coarse aggregate, fine aggregate, age	Compressive strength	ANN, Salp swarm algorithm
Ling et al. [53]	2019	Cement, fly ash, slag, magnesium-ion, sulfate-ion, chloride-ion	Compressive strength	SVM
Hendi et al. [38]	2018	SP, micro silica, fine glass powder, 28-day compressive strength, water absorption ratio, voids of permeable pores, days of exposure, an intercept (constant value)	Mass-loss and volume–loss under HCl attack	ANN
Özcan et al. [54]	2009	Cement, silica fume, water, plasticizer, aggregate, age	Long-term compressive strength	FL, ANN
Bilim et al. [55]	2009	Cement, blast furnace slag, SP, aggregate, water, age	Compressive strength	ANN

- Flowing under its own weight without vibrating.
- Passing through formwork sections that are narrow and crowded with reinforcement.
- Having the ability to become homogeneous and highly resistant to segregation [56, 57].

In order to achieve the required characteristics, SCC usually requires a high powder content. Using only cement will result in a high cost for SCC, as well as its vulnerability to attack and thermal cracking. To achieve an effective SCC, it is essential to choose and use suitable supplementary cementitious and filler materials, such as fly ash, silica fumes, metakaolin, and limestone. [58]

Regarding Table 5, some researchers investigated the potential of ML algorithms such as ANN, RSM, ANFIS, and GEP to predict mechanical and rheological properties of SCC.

3.3 Ultra-High-Performance Concrete

Ultra-high-performance concrete is a relatively novel composite material of HPC, which is characterized by ultra-high strength, outstanding toughness, and excellent durability due to low w/b of around 0.2 and the use of SP and steel fibers. Standards and specifications on the design, testing, and applications of UHPC have been formulated by France, China, the United States (USA), Japan, and South Korea [65–68].

According to the Asian Concrete Federation (ACF), UHPC is a cementitious composite that consists of discrete fibers and has a minimum compressive strength of 120 MPa, a minimum flexural strength of 14 MPa, and a minimum direct tensile strength of 5 MPa. According to Japan and France criteria, UHPC has a compressive strength of 150 MPa, and its tensile strength is 5 and 8 MPa. The USA recommends a minimum compressive strength of 120 and 180 MPa is recommended by South Korea for UHPC. UHPC can, however, reach a compressive strength of up to 250 MPa in practice and 400 MPa in laboratory conditions [69]. Regarding Table 6, some researchers investigated the potential of ML algorithms such as ANN, Random forest (RF), bagging, and boosting methods to predict different properties of UHPC.

3.4 Alkali-Activated Concert

Alkali-activated materials (AAM) are recognized as potential alternatives to ordinary Portland cement (OPC) in order to limit CO_2 emissions as well as beneficiate several wastes into useful products [73]. Over the last several decades, many different alkali-activated types of cement have been created. Alkaline cement may be divided into

Table 5 Summary of using ML in SCC

Refs.	Year	Input	Output	Prediction methods
Ben Aicha et al. [39]	2022	Slump flow diameter, V-funnel flow time, L-box ratio	Yield stress, viscosity	Multivariable regression, ANN
Ofuyatan et al. [34]	2022	Silica fume, plastic waste, cement, Sand, Granite, SP, water	Compressive strength, tensile strength, impact strength	RSM, ANN
Mohamed et al. [37]	2021	Cement, fly ash, silica fume, Slag, age, water, SP, Fine aggregate, coarse aggregate, w/b	Compressive strength, chloride penetration resistance	ANN
Farooq et al. [27]	2021	Cement, w/b, coarse aggregate, fine aggregate, fly ash, SP	Compressive strength	ANN, SVM, GEP
Al-Mughanam et al. [59]	2020	Cement, water, w/b, palm oil fuel ash, fine aggregate, coarse aggregate, SP	Compressive strength	ANFIS
Elemam et al. [60]	2020	Total binder content, fly ash, silica fume, limestone powder, w/b, SP	L-box test, Compressive strength, slump flow test, segregation test	ANN
Vakhshouri and Nejadi [42]	2018	Aggregate volume, aggregate maximum size, powder volume, paste volume, slump flow, w/b, volume of powder to volume of mortar	Compressive strength	ANFIS
Hendi et al. [38]	2018	SP, micro silica, fine glass powder, 28-day compressive strength, water absorption ratio, voids of permeable pores, days of exposure, an intercept (constant value)	Mass-loss and volume–loss under HCl attack	ANN
Safiuddin et al. [61]	2016	Cement, water, coarse aggregate, fine aggregate, palm oil fuel ash, SP, viscosity modifying admixture	Compressive Strength	ANN

(continued)

Table 5 (continued)

Refs.	Year	Input	Output	Prediction methods
Uysal and Tanyildizi [62]	2012	Cement, natural aggregate, aggregate type 1, aggregate type 2 fly ash, GBFS, zeolite, limestone powder (LP), basalt powder (BP) and marble powder, polypropylene fibers, heating degree	Compressive strength	ANN
Uysal and Tanyildizi [63]	2011	Cement, fly ash, Limestone powder, marble powder, natural powder, natural aggregate, aggregate type 1, aggregate type 2, SP, unit weight, water absorption	Core compressive strength	ANN
Siddique et al. [64]	2011	Cement, sand, coarse aggregate, w/b, SP, water, bottom ash, fly ash	Compressive strength	ANN

two major groups based on the characteristics of their cementitious components ($CaO\text{-}SiO_2\text{-}Al_2O_3$ system):

- High calcium cement
- Low calcium cement

Each group has a different pattern of activation. In high calcium cement system, materials that are high in calcium and silicon, like blast furnace slag, are activated in conditions that are not too alkaline. In this case, the main product of the reaction is a gel made of C–S–H (calcium silicate hydrate), which is similar to the gel made when Portland cement hydrates and takes up Al. In low calcium cement systems, mostly aluminum and silicon are used to activate the materials. Metakaolin or type F fly ash from coal-fired steam power plants are some of the materials used in this second alkali activation process. These materials have low amounts of CaO. In this case, the reactions need to be kicked off with more aggressive working conditions, such as highly alkaline media and curing temperatures of 60–200 °C. In this case, the main reaction product is a 3D inorganic alkaline polymer called N–A–S–H gel, which stands for alkaline aluminosilicate hydrate and can be thought of as a zeolite precursor. This gel has a lot of different names, such as geo- or inorganic polymer [74].

Table 6 Summary of using ML in UHPC

Refs.	Year	Input	Output	Prediction methods
Wang et al. [33]	2022	Cement, fly ash, slag, silica fume, nano silica, limestone powder, sand, coarse aggregate, quartz powder, water, SP, steel fiber diameter, steel fiber length, age	Flexural strength	Adaboost, XGBoost, GB, Bagging
Han et al. [70]	2019	Fly ash, blast furnace slag, coarse aggregate, water, Age, fine aggregate, cement, SP	Compressive strength	RF
Chou et al. [32]	2011	Water, age, cement, SP, blast furnace slag, fly ash, fine aggregate, coarse aggregate	Compressive strength	ANN, Bagging regression trees, Multiple additive regression trees, SVM, Multiple regression
Öztaş et al. [71]	2005	Silica fume, fly ash, air-entraining, water, w/b, SP	Slump, compressive strength	ANN
Gupta et al. [72]	2006	Water, cement, fine aggregate, coarse aggregate, average workability, average slump, w/c	Compressive strength	ANN

Due to its early compressive strength, low permeability, high chemical resistance, and exceptional fire-resistant behavior, geopolymer has garnered a lot of interest recently among binders. Due to these beneficial qualities, the geopolymer is a promising candidate to replace ordinary Portland cement in the development of various sustainable products for the construction industry, including concrete, fire-resistant coatings, fiber-reinforced composites, waste immobilization solutions, and building materials [75]. It should be considered that because these materials have different chemical features and characteristics, the products of their reactions will be different. Therefore, the mechanical properties of materials with low calcium and alkali activation, like alkali-activated fly ash, and materials with high calcium and alkali activation, like alkali-activated GGBFS, are very different. For example, the drying shrinkage of alkali-activated fly ash is lower than that of Portland cement concrete, and high calcium AAC [76].

Table 7 summarize the input and output that some researchers used to develop ML algorithms such as ANN, RF, ANFIS, and RSM to predict various properties of AAC.

Table 7 Summary of using ML in AAC

Ref.	Year	Input	Output	Prediction methods
Upreti et al. [77]	2022	Fly ash, extra water, slump, density, coarse aggregate, fine aggregate, NaOH solution, sodium silicate, SP, GGBFS	Compressive strength, splitting tensile strength, flexural strength	ANN, RF
Tang et al. [78]	2022	Na_2SiO_3, NaOH, water, curing types, w/b, ground granulated blast-furnace slag, blaine fineness, Na_2O, silica modulus of the activator, fine aggregate to total aggregate ratio	Compressive strength	ANN
Qin et al. [79]	2022	Alkali concentration of activator ($Na_2O\%$), modulus of activator, w/b, surface area of slag, basicity index of slag	Compressive strength	ANN, alternating conditional expectation
Ahmad et al. [26]	2021	Fly ash, coarse aggregate, fine aggregate, NaOH, sodium silicate, silicon dioxide, sodium oxide, NaOH molarity, curing time	Compressive strength	Adaboost, ANN, boosting
Ibrahim et al. [80]	2021	Nano silica, time	Strength and weight loss under acid resistance	ANN. RSM
Van Dao et al. [41]	2019	Fly ash, Na_2SiO_3, NaOH, water	Compressive strength	ANFIS, ANN
Nagajothi and Elavenil[81]	2019	Manufactured sand to natural river sand ratio, GGBFS, fly ash	Compressive strength, splitting tensile strength, Flexural strength	ANN

3.5 Recycled Aggregate Concrete

People have been concerned with keeping the environment clean for many years. In order to maintain a clean environment, solid waste must also be recycled. From the standpoint of environmental preservation and effective resource utilization, recycling waste concrete is beneficial and necessary. Waste concrete has to be used as recycled aggregates for new concrete in order to be effectively utilized.

In some cases, adding recycled material to concrete might result in a 40% reduction in its compressive strength [82–85]. As a result of replacing 25–30% [86] or 100% [87] of the natural aggregate with recycled aggregate, compressive strength can be reduced by 12–25%. By replacing 30% of coarse or fine natural aggregate with recycled aggregate, the impact is minimal. It was observed that the compressive strength of recycled aggregate decreased due to the replacement of recycled aggregate (due to the old mortar content) and a poor interfacial transition zone [88, 89].

Regarding Table 8, some researchers investigated the potential of ML algorithms to predict various properties of RAC.

In recent years, there have been numerous investigations on the role of ML in the prediction of different properties of concrete. Penido et al. [51] used four ANN, SVR, XGBoost, and GPR models to forecast the compressive strength of concrete

Table 8 Summary of using ML in RAC

Refs.	Year	Input	Output	Prediction methods
Zeng et al. [90]	2022	Cement strength class, w/b, paste to aggregate ratio, recycled coarse aggregate replacement proportion, fly ash replacement proportion, silica fume replacement proportion, slag replacement proportion, slump, sand to aggregate ratio	Compressive strength	ANN
Hammoudi et al. [91]	2019	Cement, recycled aggregate, slump	Compressive strength	ANN, RSM
Golafshani and Behnood [92]	2018	28-day cube compressive strength, volume replacement of natural aggregate by recycled aggregate, coarse aggregate to cement ratio, saturated surface dry specific gravity, water absorption, fine aggregate to total aggregate ratio, w/c	Elastic modulus	ANN, SVR, FL
Topçu and Saridemir [93]	2008	Silica fume, cement, water, sand, recycled aggregate, SP, aggregate, age	Compressive strength, splitting tensile strength	FL, ANN

containing steelmaking slag at 1–360 days. The R^2 values related to all ages were 0.89, 0.91, 0.93, and 0.83 for GPR, ANN, XGBoost, and SVR, respectively. In addition, the R^2 values were reduced for only 28 days which were 0.68, 0.79, 0.73, and 0.73 for GPR, ANN, XGBoost, and SVR, respectively. It is significant to note that the R^2 values for validation were not satisfactory due to negative values for GPR, XGBoost, and SVR. These unsatisfactory R^2 values for validation highlighted some crucial aspects of ML usage:

- Homogeneity and size of the dataset
- Input parameter selection
- Adjusting models with cross-validation.

The compressive strength of concrete containing silica fume was predicted to develop ANN and FL by Özcan et al. [54]. As demonstrated by the R^2 analysis, both the ANN and FL models significantly predicted compressive strength. However, the ANN model supplied better results than the FL model. According to Wang et al. [33], ensembled ML methods and DT were compared in terms of their performance in predicting UHSC flexural strength. For the purpose of evaluating models in their study, MAE, RSME, and R^2 were used. Bagging was found to perform better with R^2, RSME, and MAE equal to 0.95, 8.26, and 2.05, respectively, followed by GB, Adaboost, and XGBoost. The performance of ensemble ML techniques was superior to that of individual DT models.

Song et al. [24] utilized GEP, ANN, and DT to predict the compressive strength of concrete incorporating fly ash. In their study R^2, RME, and RMSE were evaluated to compare ANN with DT methods also cement, fly ash, SP, water, fine aggregate, coarse aggregate, and age were introduced as inputs. The R^2 values of the GEP, ANN and, DT were 0.86, 0.81 and, 0.75, respectively. It is significant to note that bagging regressor as an ensemble algorithm had a higher R^2 value equal to 0.95 compared to individual ML techniques. The lesser values of the errors, MAE (3.69 MPa), MSE (24.76), and RMSE (4.97), also confirmed the high accuracy of the bagging regressor, while other algorithms show higher values for these errors. Amiri and. Hatami [94] investigated compressive strengths and the durability test of rapid chloride migration test (RCMT) of concrete containing GGBFS and recycled concrete aggregate using ANN. A three-layer multilayer perceptron (MLP) was utilized in their study with the structure of an input layer, a hidden layer, and an output layer. The authors reported that R^2 values were 0.9958 and 0.9912 for compressive strength and RCMT, which showed that provided ANN model could strongly predict the properties of concrete.

Uysal and Tanyildizi [62] investigated the potential of different ANN algorithms to forecast the compressive strength of SCC containing polypropylene fiber and mineral additives such as fly ash, GBFS, zeolite, limestone powder, basalt powder, and marble powder in various proportioning rates exposed to high temperature. For predicting the compressive strength of SCC, the authors used some ANN algorithms, including Levenberg–Marquardt backpropagation, BFGS quasi-Newton backpropagation, Powell-Beale conjugate gradient backpropagation, Fletcher–Powell conjugate gradient backpropagation, and Polak–Ribiere conjugate gradient backpropagation, all with R^2 values of 0.9587, 0.9757, 0.9652, 0.9674, 0.9616, and 0.9559,

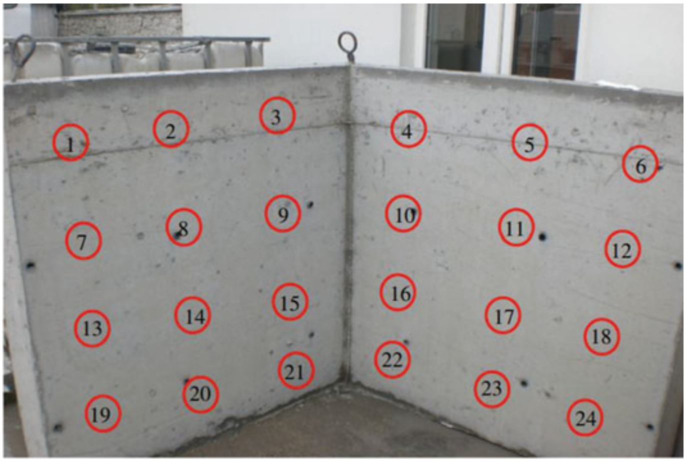


Fig. 12 Core areas on the SCC wall

respectively. Due to the good correlation between experimental results and the ANN model, BFGS quasi-Newton backpropagation algorithm was the best algorithm for calculating the compressive strength of SCC exposed to high temperatures. In another study, Uysal and Tanyildizi [63] developed an ANN model to predict the core compressive strength of SCC incorporating fly ash, limestone powder, marble powder, and natural powder. The core spots on the wall and the architecture of the ANN model are present in Figs. 12 and 13. To develop the ANN model, the authors used the Fletcher-Powell conjugate gradient backpropagation algorithm as well as the Levenberg–Marquardt backpropagation algorithm. As a result of the analysis, the fletcher powell conjugate gradient backpropagation algorithm was found to have a better performance in terms of prediction than the Marquardt backpropagation algorithm, with R^2 values of 0.95 and 0.92. A high correlation coefficient was found in the ANN models when predicting SCC core compressive strength.

Ben Aicha [39] predicted the rheological behavior of SCC utilizing multivariable regression and ANNs. In their study slump flow diameter (SFD), V-funnel flow time (VFT), and L-box ratio (LBR) were introduced as inputs, and yield stress and viscosity were considered as outputs. The correlation between viscosity and VFT, SFD, and LBR was 0.967, 0.955, and 0.916 also, the correlation between yield stress and VFT, SFD, and LBR were 0.963, 0.86, and 0.0.914. The results showed that in the case of data with only one output or small datasets, multivariable linear regression can be utilized instead of ANN. The usage of the multivariable linear regression is not suitable for big datasets, particularly those with multiple outputs.

Hendi et al. [38] utilized ANN to study the effect of glass powder and micro silica usage in SCC and OC under HCl attack. In their study mass-loss and volume-loss were investigated to identify the beneficial or harmful effects of input parameters on the outputs and even to minimize the ANN function; the particle swarm optimization (PSO) method was implemented with 20 iterations and 10,000 particles. The R^2

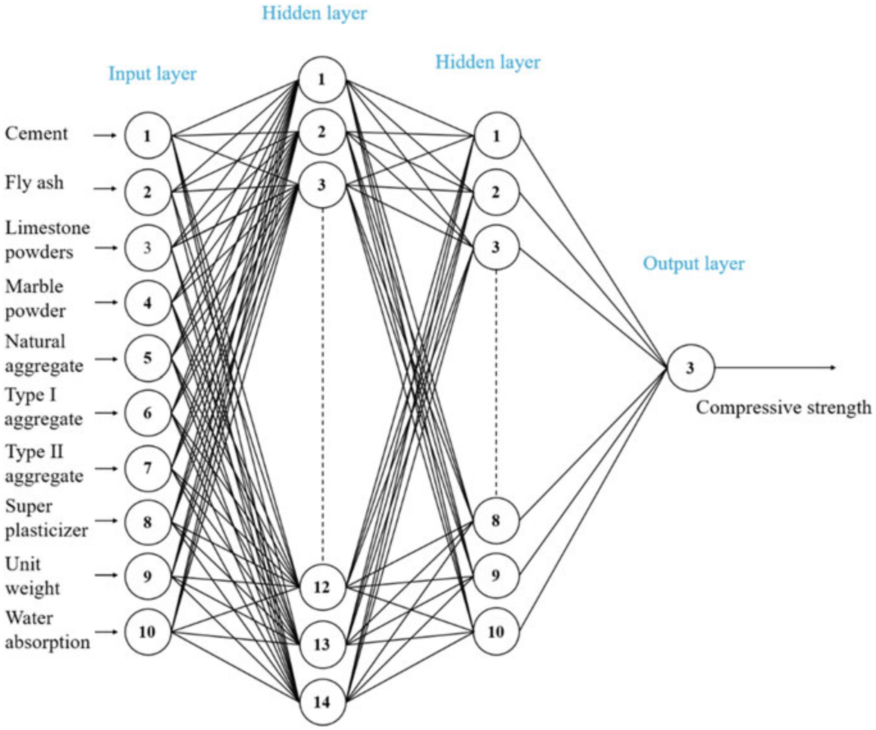


Fig. 13 ANN architecture

values were 0.9982 and 0.9957 for Mass-loss and Volume-loss, which indicated a high actuary of ANN in terms of prediction. An analysis of the average mass-loss of related mixtures in OC was conducted to determine their effects on glass powder and micro silica. It should be noted that adding more SCMs raised mass-loss values, and that micro silica had a more significant impact than glass powder. Concerning PSO results, the optimum compressive strength value equaled 33.84 MPa to minimize Mass-Loss.

The Compressive strength and chloride penetration resistance of SCC were investigated by Mohamed et al. [37] using ANNs. To forecast chloride penetration resistance and compressive strength, the authors used data between 249 and 1031 datasets. Chloride penetration was determined at 7, 14, 28, and 40 days, and compression strength of SCC specimens was determined at 3, 7, and 28 days. Comparing the ANN predicted values with compressive strength values of SCC specimens, most specimens demonstrated greater than 80% prediction accuracy, while a few specimens demonstrated greater than 90% accuracy. In the case of chloride penetration prediction, the ANN model with a learning rate of 0.3, a momentum of 0.5, and 5000 epochs demonstrated the highest overall prediction accuracy of 95%.

The Convolutional Neural Network (CNN) is a type of deep learning neural network that has been developed over the past two decades. This technology significantly contributed computer vision to the field of artificial intelligence. Zeng et al. [90] used CNN to predict compressive strength OC, high strength concrete (HSC), and RAC with data set of 380 groups of concrete mixtures. Figure 14 compares the CNN model with SVM, ANN, and Adaboost methods based on their statistical parameters values. It is clear that CNN with a higher value of R^2 has the best performance among the four ML methods.

Ahmad et al. [25] investigated the potential of the ANN method to forecast the compressive strength of GC incorporating natural zeolite and silica fume. Table 9 indicates the Values of various parameters in the developed ANN. To develop the ANN model, 117 concrete mixtures were used as outputs, and specimen age, NaOH concentration, and content of natural zeolite, silica fume, and GGBS were analyzed as inputs. According to the predicted results, the proposed model was highly accurate and capable of making accurate predictions. Also, the results of the developed ANN model were compared with the GEP model developed by Shahmansouri et al. [95]

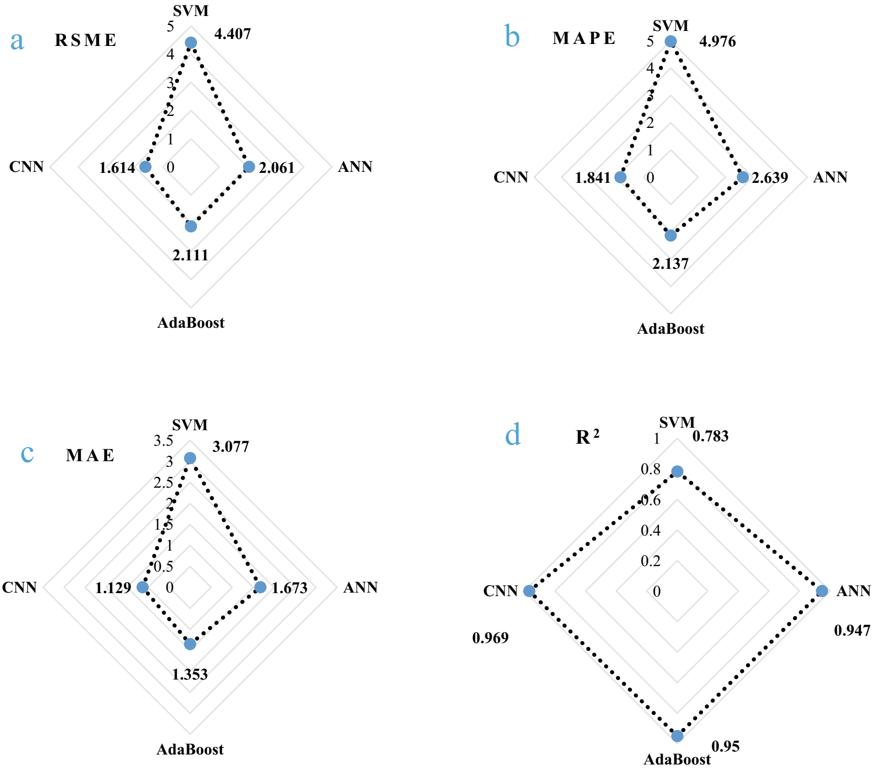


Fig. 14 Compression of statistical indexes for the four models, RSME (**a**), MAPE (**b**), MAE (**c**), and R^2 (**d**)

Table 9 The values of various parameters in the developed ANN

Parameters	Number
Neurons (input layer)	5
Hidden layers	2
Neurons (hidden layer)	6
Neurons (second hidden layer)	5
Neurons (output layer)	1

in order to examine the performance of the model versus another soft computing method. In terms of statistical indexes, the R values for the proposed ANN and the GEP were 0.980 and 0.959, respectively, while the MSE values were 4.7769 and 9.5294, illustrating the high performance of the proposed ANN in comparison with the GEP to predict the strength of pozzolanic GC with silica fume and natural zeolite.

In recent years, RSM has become one of the most typical optimization methods, with high importance among researchers. Recent years have seen a substantial increase in the use of numerical optimization tools in civil engineering, particularly to optimize the mechanical and durability properties of concrete using RSM. The resistance of AAC against acid attack was investigated by Ibrahim et al. [80]. An ANN model was generated in order to predict the weight and strength loss of AAC as a result of acid attacks in this study. Additionally, RSM models were developed to determine the minimum weight and strength loss of $nSiO_2$. High correlation and minimum error were obtained considering the predicted and experimental values showed that Models based on ANN and RSM could effectively predict the weight and strength loss of AAC. The elastic modulus of concrete is one of the essential properties of the material and is widely used to calculate the deformation of structures due to earthquakes [96]. Yan and Shi et al. [97] used SVM to predict elastic modulus of ordinary Portland concrete and HSC. Since in previous studies by Demir [98, 99] mathematical regression models, ANN, and the FL model have been used for the prediction of elastic modulus for OPC and HSC, the authors compared the SVM results with regression, ANN, And Fuzzy model based on RSEM and MAPE for OPC and HSC. Figures 15 and 16 Show MAPE and RMSE values of different models for training and testing. It was evident that SVM is responsible for the lowest values in both OPC and SCC which shows that SVM has better performance in terms of prediction.

Ahmad et al. [26] used ML algorithms to forecast compressive strength of high calcium fly-ash-based GC. In their study ANN, boosting, and Adaboost were utilized based on python coding. Regarding the R^2, MSE, RMSE, and MAE values, boosting indicated the higher value of R^2 equals 0.96, while Adaboost and ANN were less accurate. In addition, the boosting method had lesser error values than the other two methods, which proved to have a better performance in compressive strength prediction. Topçu and Saridemir [93] used ANN and FL to predict compressive strength and splitting tensile strength of RAC containing silica fume at 3, 7, 14, 28, 56, and 90 days. Their study introduced age, cement, sand, aggregate, recycled aggregate, water, SP, and silica fume as inputs, while compressive strength and splitting tensile

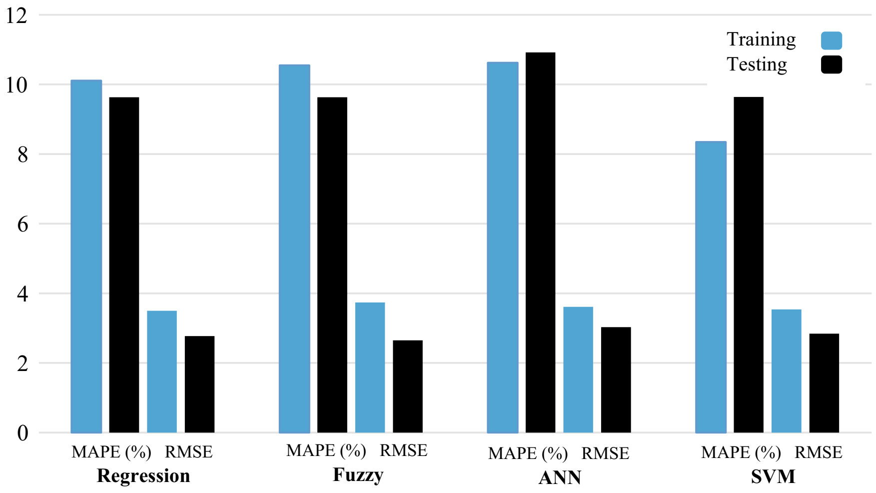


Fig. 15 MAPE and RMSE comparison between SVM and other models for OPC

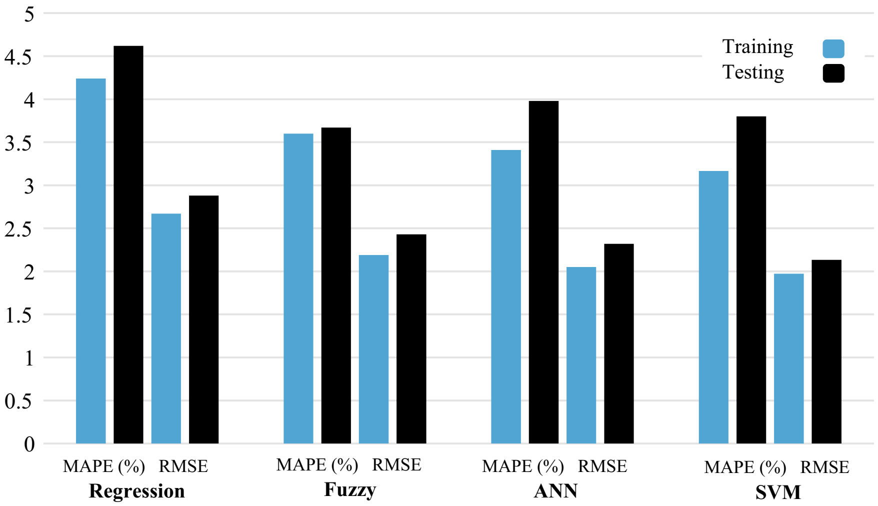


Fig. 16 MAPE and RMSE comparison between SVM and other models for HSC

strength values were used as outputs. In addition, 140 data of experiment results were used for training whereas 70 were employed for testing. The results in Table 10 compare the performance of ANN and FL using RMSE, R^2, and MAPE values. Although both methods significantly predicted RAC mechanical properties, ANN performed significantly better than FL due to more acceptable statistical index values.

Tang et al. [78] developed different ANN models to forecast the compressive strength of alkaline-activated slag Concretes. The authors used four ANN models

Table 10 RMSE, R^2, and MAPE values of ANN and FL for compressive and splitting tensile strength

Statistical parameters	Compressive strength				Splitting tensile strength			
	ANN		FL		ANN		FL	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
RMSE	2.0465	2.3948	2.6557	3.8655	0.1406	0.1968	0.2020	0.2955
R^2	0.9989	0.9984	0.9981	0.9959	0.9989	0.9979	0.9977	0.9952
MAPE	2.8302	3.3699	3.5973	5.5245	2.6287	3.5516	4.2165	5.7306

with 8-8-1, 9-10-1, 9-16-1, and 10-14-1 structures. In addition, 181 datasets were used to develop ANN models: 80%, 15%, and 5% were utilized for training, validation, and testing, respectively. The values of R, R^2, RMSE, MAE, and MAPE indicated that the model with an 8-18-1 structure had better performance in terms of compressive strength prediction.

Mhaya et al. [100] used the GA-ANN model to forecast the mechanical properties and impact resistance of concrete incorporating GBFS and discarded rubber tire crumbs. In terms of the mechanical properties of the modified rubberized concretes, the GA-ANN provided satisfactory results in terms of their mechanical properties. As is evident from the results of this paper, the GA-ANN could also be applied as a solid tool to optimize the weights in the ANN. Hendi et al. [101] used ANN to predict mass-loss and volume-loss of OC and SCC containing glass beads and micro silica which were replaced with cement due to sulfuric acid (H_2SO_4) attack. The authors utilized the feed-forward ANN method and backpropagation algorithm for predictions and error assessment, respectively. The structure of the ANN model was 9-8-6-1 using 60%, 20%, and 20% of all data as the training set, 20, validation dataset, and tasting dataset, respectively. The R^2 values between experimental mass-loss and predicted mass-loss using ANN were 0.8782 and 0.8212 for OPC and SCC, while considering volume-loss as an output, the R^2 values were 0.7988 and 0.7415 for OPC and SCC. According to ANNs analysis, higher micro silica and glass powder contents, as well as concretes with lower compressive strength, performed better in the H_2SO_4 acid medium. As a result, higher compressive strengths did not necessarily translate into better durability.

Suleiman and Nehdi [102] used the GN-ANN model to investigate Self-Healing cracks in concrete by introducing cement content, w/c, type, and dosage of SCMs, bio-healing materials, and expansive and crystalline additives as inputs. The authors reported that the GA–ANN model could provide an alternative solution for modeling the highly complex self-healing phenomena in cement-based materials as a powerful computational tool with high efficiency. Genetic algorithms were successfully applied to the ANN model to identify the most appropriate weights and biases. In addition, the proposed technique provided a reliable prediction for the self-healing ability of a cementitious material, which can be utilized to improve the design of concrete that is more durable and sustainable.

Pham et al. [103] used a Hybrid SVM-based model to predict the compressive strength of HPC. In their study, 215 and 24 samples were used for trading and testing sets to develop the FA-LSVR model, the combination of the firefly algorithm (FA) and the least squares support vector regression (LS-SVR). The authors compared the hybrid model with ANN and SVM models based on RMSE, R^2, and MAPE values. With respect to statistical indexes, the FA-LSVR model could predict compressive strength compared to SVM and ANN. It is significant to note that, following the first experiment, a tenfold cross-validation process was performed. Since all subsamples were mutually exclusive, this method could provide a more accurate evaluation of the proposed FA-LSVR and other analysis techniques. The results of the second experiment showed that the hybrid models achieved the highest R^2 values of 0.93 and 0.87 for both training and testing datasets and the lowest RMSE and MAPE values for both training and testing datasets, which indicated that the hybrid model was the most accurate and SVM was the second best method.

Golafshani and Behnood [92] utilized ML methods for predicting the 28-day elastic modulus of concrete containing recycled aggregate. The authors used ANN, FL, SVR, and radial basis function neural network (RBFNN) in this regard. In addition, nine different training algorithms in other to provide a reliable ANN were investigated. Table 11 indicates compression of statistical indexes related to referent training algorithms of ANN, Levenberg–Marquardt had a better performance than the other eight training algorithms. To compare the performance of ANN with other techniques, they used the Levenberg-Marquard algorithm as the most efficient algorithm.

Regarding Table 12, which compares different ML methods used in this study based on their statistical indexes, by comparing RMSE and r-value statistical parameters, ANN is found to perform better than the other techniques, while SVR shows superior performance on MAE and MAPE. As a result, all four statistical parameters of the developed RBFNN model outperformed the FL. The OBJ value was calculated by taking into account the RMSE, MAE, and r-value statistical parameters of

Table 11 Statistical indexes of nine training algorithms

Training algorithms	Statistical indexes		
	RSME	r-value	OBJ
BFGS Quasi-Newton	2.7790	0.8804	2.5645
Resilient backpropagation	2.7222	0.8854	2.5203
Levenberg–Marquardt	1.7143	0.9561	1.5957
Scaled conjugate gradient	3.0153	0.8572	2.8089
Variable learning rate backpropagation	3.0076	0.8585	2.8281
One step secant	3.1175	0.8476	2.9648
Conjugate gradient with powell/Beale restarts	2.8656	0.8721	2.6610
Polak-Ribiere conjugate gradient	2.8809	0.8707	2.6278
Fletcher-Powell conjugate gradient	2.9844	0.8605	2.7767

Table 12 Statistical indexes of different ML methods

ML methods	Statistical indexes				
	RMSE	MAE	MAPE	r-value	OBJ
ANN	1.7143	1.1591	4.5759	0.9561	1.5957
FL	2.5189	1.6748	6.6711	0.9028	2.2550
SVR	1.7941	1.1403	4.5586	0.9548	1.6920
RBFNN	1.9655	1.4668	5.9627	0.9423	1.8966

the training and testing datasets. The developed ANN model showed the highest efficiency, followed by the developed SVR model, after comparing the OBJ values of four developed soft computing models.

Different researchers utilized different ML methods to compare their performance with respect to statistical indexes. Table 13 summarizes some studies in which different algorithms of ML were compared.

4 Conclusion

A review on the use of machine learning as a strong tool by various researchers to predict concrete properties was conducted. Through this study, the following conclusions are drawn:

- Different researchers used different ML methods to predict the properties of concrete and compressive strength is the most frequent property which is investigated as output.
- Numerous ML models could be used as prediction tools in concrete science, yet ANNs were more frequently used.
- Previous researchers show that ML techniques have better performance than classical regression models.
- Researchers widely used hybrid SVM-based and Hybrid ANN-based models for prediction, and generally, they performed better than individual methods.

Table 13 A comparison of ML models based on statistical indexes

Refs.	Type of concrete	ML method	Data set size	Output	(Statistical index)				
					R	R^2	MSE	RMSE	MAE
Wang et al. [33]	UHSC	Adaboost	317	FS	–	0.93	–	8.69	2.12
		Bagging			–	0.95	–	8.26	2.05
		GB			–	0.93	–	8.25	2.10
		XGBoost			–	0.85	–	4.25	3.12
Özcan et al. [54]	OC	ANN	240	CS	–	0.9944	–	–	–
		FL			–	0.9274	–	–	–
Penido et al. [51]	OC	GPR	406	CS	–	0.68	–	7.09	4.98
		ANN			–	0.79	–	6.52	4.73
		XGBoost			–	0.73	–	6.76	5.02
		SVR			–	0.7	–	7.44	5.51
Ahmad et al. [26]	GC	ANN	154	CS	–	–	20.16	4.49	3.86
		Boosting			–	–	4.16	2.04	1.69
		Adaboost			–	–	6.84	2.62	2.16
Algaifi et al. [104]	Bacteria-based self-healing concrete	RSM	58	CS	0.973	0.972	0.704	0.839	–
		ANN			0.986	0.985	0.628	0.793	–
		ANFIS			0.986	0.986	0.616	0.785	–
Asadi Shamsabadi et al. [105]	OC	Multiple Linear Regression	630	CS	–	0.44	92.49	9.62	7.71
		SVR			–	0.91	17.45	4.18	2.48
		RF			–	0.94	13.81	3.72	2.73

(continued)

Table 13 (continued)

Refs.	Type of concrete	ML method	Data set size	Output	(Statistical index)				
					R	R^2	MSE	RMSE	MAE
		GB			–	0.97	6.48	2.55	1.81
		XGB			–	0.98	4.6	2.15	1.49
		ANN			–	0.97	5.26	2.29	1.35
Bui et al. [19]	HPC	GEP	1133	CS	0.91	–	–	–	5.2
		FA-LSSVR			0.94	–	–	–	3.86
		MFA-ANN			0.95	–	–	–	3.41
Feng et al. [106]	OC	Adaboost	1030	CS	–	0.982	–	2.2	1.64
		ANN			–	0.903	–	5.14	3.41
		SVM			–	0.855	–	6.28	4.44
Chou et al. [32]	HPC	ANN	1030	CS	–	0.909	–	5.03	–
		SVM			–	0.885	–	5.619	–
Yu et al. [107]	HPC	ANN	1761	CS	–	0.96	–	–	5.038
		ANFIS			–	0.906	–	–	4.183
		SVM			–	0.793	–	–	5.95
Yuan et al. [7]	OC	ANN	180	CS	–	0.68	–	3.21	–
		GA-ANN			–	0.813	–	2.22	–

(continued)

Table 13 (continued)

Refs.	Type of concrete	ML method	Data set size	Output	(Statistical index)				
					R	R^2	MSE	RMSE	MAE
		ANFIS			–	0.95	–	1.46	–
Behnood et al. [108]	Reinforcement concrete	ANN	980	TS	–	0.874	–	0.526	0.408
		SVM			–	0.89	–	0.524	0.4
Keshtegar et al. [21]	Reinforcement concrete	ANN	139	SS	–	–	–	0.461	0.322
		SVR			–	–	–	1.01	0.622
		RSM-SVR			–	–	–	0.233	0.186

CS: Compressive strength SS: Shear strength TS: Tensile strength

References

1. Haykin S (2008) Neural networks and learning machines, vol 3. 978-0131471399
2. Goulet J-A (2020) Probabilistic machine learning for civil engineers, vol 1. MIT Press, pp 1–25. https://www.cambridge.org/core/product/identifier/CBO9781107415324A009/type/book_part
3. Karthikeyan A, Priyakumar UD (2022) Artificial intelligence: machine learning for chemical sciences. J Chem Sci 134(1). https://doi.org/10.1007/s12039-021-01995-2
4. Kotsiantis SB, Zaharakis I, Pintelas P et al (2007) Supervised machine learning: a review of classification techniques. Emerg Artif Intell Appl Comput Eng 160(1):3–24
5. Sutton RS, Barto AG (1999) Reinforcement learning: an introduction. Robotica 17(2):229–235
6. Cheng M-Y, Chou J-S, Roy AFV, Wu Y-W (2012) High-performance concrete compressive strength prediction using time-weighted evolutionary fuzzy support vector machines inference model. Autom Constr 28:106–115
7. Yuan Z, Wang LN, Ji X (2014) Prediction of concrete compressive strength: research on hybrid models genetic based algorithms and ANFIS. Adv Eng Softw 67:156–163. https://doi.org/10.1016/j.advengsoft.2013.09.004
8. Ben Chaabene W, Flah M, Nehdi ML (2020) Machine learning prediction of mechanical properties of concrete: Critical review. Constr Build Mater 260:119889. https://doi.org/10.1016/j.conbuildmat.2020.119889
9. Li Z et al (2022) Machine learning in concrete science: applications, challenges, and best practices. NPJ Comput Mater 8(1). https://doi.org/10.1038/s41524-022-00810-x
10. Tao Q, Xu P, Li M, Lu W (2021) Machine learning for perovskite materials design and discovery. NPJ Comput Mater 7(1):1–18
11. Schmidt J, Marques MRG, Botti S, Marques MAL (2019) Recent advances and applications of machine learning in solid-state materials science. NPJ Comput Mater 5(1):1–36
12. Muhammad I, Yan Z (2015) Supervised machine learning approaches: a survey. ICTACT J Soft Comput 5(3)
13. Gove R, Faytong J (2012) Machine learning and event-based software testing: classifiers for identifying infeasible GUI event sequences, vol 86. Elsevier Inc. https://doi.org/10.1016/B978-0-12-396535-6.00004-1
14. Zheng B, Myint SW, Thenkabail PS, Aggarwal RM (2015) A support vector machine to identify irrigated crop types using time-series Landsat NDVI data. Int J Appl Earth Obs Geoinf 34:103–112
15. Moraes R, Valiati JF, Neto WPG (2013) Document-level sentiment classification: an empirical comparison between SVM and ANN. Expert Syst Appl 40(2):621–633
16. Deka PC et al (2014) Support vector machine applications in the field of hydrology: a review. Appl Soft Comput 19:372–386
17. Zendehboudi A, Baseer MA, Saidur R (2018) Application of support vector machine models for forecasting solar and wind energy resources: a review. J Clean Prod 199:272–285
18. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Lecture notes computer sciences (including Subseries Lecture notes artificial intelligence and Lecture notes in bioinformatics), vol 5792. LNCS, pp 169–178. https://doi.org/10.1007/978-3-642-04944-6_14
19. Bui DK, Nguyen T, Chou JS, Nguyen-Xuan H, Ngo TD (2018) A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete. Constr Build Mater 180:320–333. https://doi.org/10.1016/j.conbuildmat.2018.05.201
20. Chou J-S, Chong WK, Bui D-K (2016) Nature-inspired metaheuristic regression system: programming and implementation for civil engineering applications. J Comput Civ Eng 30(5):4016007

21. Keshtegar B, Bagheri M, Yaseen ZM (2019) Shear strength of steel fiber-unconfined reinforced concrete beam simulation: application of novel intelligent model. Compos Struct 212(December):230–242. https://doi.org/10.1016/j.compstruct.2019.01.004
22. Kotsiantis SB (2013) Decision trees: a recent overview. Artif Intell Rev 39(4):261–283. https://doi.org/10.1007/s10462-011-9272-4
23. Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. cs/0102027.
24. Song H et al (2021) Predicting the compressive strength of concrete with fly ash admixture using machine learning algorithms. Constr Build Mater 308. https://doi.org/10.1016/j.conbuildmat.2021.125021
25. Shahmansouri AA, Yazdani M, Ghanbari S, Akbarzadeh Bengar H, Jafari A, Farrokh Ghatte H (2021) Artificial neural network model to predict the compressive strength of eco-friendly geopolymer concrete incorporating silica fume and natural zeolite. J Clean Prod 279:123697. https://doi.org/10.1016/j.jclepro.2020.123697
26. Ahmad A et al (2021) Prediction of geopolymer concrete compressive strength using novel machine learning algorithms. Materials (Basel) 14(4):1–21. https://doi.org/10.3390/ma14040794
27. Farooq SC, Farooq F, Czarnecki S, Niewiadomski P, Aslam F (2021) A comparative study for the prediction of the compressive strength of self-compacting concrete modified with fly ash
28. Zavrtanik N, Prosen J, Tušar M, Turk G (2016) The use of artificial neural networks for modeling air void content in aggregate mixture. Autom Constr 63:155–161
29. Jiang G, Keller J, Bond PL, Yuan Z (2016) Predicting concrete corrosion of sewers using artificial neural network. Water Res 92:52–60
30. Chakraverty S, Jeswal SK (2021) Applied artificial neural network methods for engineers and scientists
31. Emmert-Streib F, Yang Z, Feng H, Tripathi S, Dehmer M (2020) An introductory review of deep learning for prediction models with big data. Front Artif Intell 3(February):1–23. https://doi.org/10.3389/frai.2020.00004
32. Chou J-S, Chiu C-K, Farfoura M, Al-Taharwa I (2011) Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques. J Comput Civ Eng 25(3):242–253. https://doi.org/10.1061/(asce)cp.1943-5487.0000088
33. Wang Q, Hussain A, Farooqi MU, Deifalla AF (2022) Artificial intelligence-based estimation of ultra-high-strength concrete's flexural property. Case Stud Constr Mater 17(April):e01243. https://doi.org/10.1016/j.cscm.2022.e01243
34. Ofuyatan OM, Agbawhe OB, Omole DO, Igwegbe CA, Ighalo JO (2022) RSM and ANN modelling of the mechanical properties of self-compacting concrete with silica fume and plastic waste as partial constituent replacement. Clean Mater 4(February):100065. https://doi.org/10.1016/j.clema.2022.100065
35. Duan ZH, Kou SC, Poon CS (2013) Using artificial neural networks for predicting the elastic modulus of recycled aggregate concrete. Constr Build Mater 44:524–532. https://doi.org/10.1016/j.conbuildmat.2013.02.064
36. Piro NS, Mohammed AS, Hamad SM (2022) The impact of GGBS and ferrous on the flow of electrical current and compressive strength of concrete. Constr Build Mater 349(July):128639. https://doi.org/10.1016/j.conbuildmat.2022.128639
37. Mohamed O, Kewalramani M, Ati M, Al Hawat W (2021) Application of ANN for prediction of chloride penetration resistance and concrete compressive strength. Materialia 17(May):101123. https://doi.org/10.1016/j.mtla.2021.101123
38. Hendi A, Behravan A, Mostofinejad D, Sedaghatdoost A, Amini M (2018) A step towards green concrete: effect of waste silica powder usage under HCl attack. J Clean Prod 188:278–289. https://doi.org/10.1016/j.jclepro.2018.03.288
39. Ben Aicha M, Al Asri Y, Zaher M, Alaoui AH, Burtschell Y (2022) Prediction of rheological behavior of self-compacting concrete by multi-variable regression and artificial neural networks. Powder Technol 401. https://doi.org/10.1016/j.powtec.2022.117345

40. Zhuang X, Zhou S (2019) The prediction of self-healing capacity of bacteria-based concrete using machine learning approaches. Comput Mater Contin 59(1):57–77. https://doi.org/10.32604/cmc.2019.04589
41. Van Dao D, Ly HB, Trinh SH, Le TT, Pham BT (2019) Artificial intelligence approaches for prediction of compressive strength of geopolymer concrete. Materials (Basel) 12(6). https://doi.org/10.3390/ma12060983
42. Vakhshouri B, Nejadi S (2018) Prediction of compressive strength of self-compacting concrete by ANFIS models. Neurocomputing 280:13–22. https://doi.org/10.1016/j.neucom.2017.09.099
43. Golafshani EM, Behnood A, Arashpour M (2020) Predicting the compressive strength of normal and high-performance concretes using ANN and ANFIS hybridized with grey wolf optimizer. Constr Build Mater 232:117266. https://doi.org/10.1016/j.conbuildmat.2019.117266
44. Kramer O (2017) Genetic algorithm essentials. Springer International, Cham (Switzerland)
45. Tsai C-F, Hsu Y-F, Lin C-Y, Lin W-Y (2009) Intrusion detection by machine learning: a review. Expert Syst Appl 36(10):11994–12000
46. Lin YH, Lin CC, Tyan YY (2011) An integrated quantitative risk analysis method for major construction accidents using fuzzy concepts and influence diagram. J Mar Sci Technol 19(4):383–391. https://doi.org/10.51400/2709-6998.2179
47. Dietterich TG (2000) An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn 40(2):139–157
48. Kuncheva LI (2014) Combining pattern classifiers: methods and algorithms. Wiley & Sons
49. Zhang (2012) Ensemble machine learning. https://doi.org/10.1007/978-1-4419-9326-7
50. Bao L, Zhou M, Cui Y (2005) nsSNPAnalyzer: identifying disease-associated nonsynonymous single nucleotide polymorphisms. Nucleic Acids Res 33(suppl_2):W480–W482
51. Penido REK, da Paixão RCF, Costa LCB, Peixoto RAF, Cury AA, Mendes JC (2022) Predicting the compressive strength of steelmaking slag concrete with machine learning—considerations on developing a mix design tool. Constr Build Mater 341(May). https://doi.org/10.1016/j.conbuildmat.2022.127896
52. Kandiri A, Mohammadi Golafshani E, Behnood A (2020) Estimation of the compressive strength of concretes containing ground granulated blast furnace slag using hybridized multi-objective ANN and salp swarm algorithm. Constr Build Mater 248:118676. https://doi.org/10.1016/j.conbuildmat.2020.118676
53. Ling H, Qian C, Kang W, Liang C, Chen H (2019) Combination of Support Vector Machine and K-Fold cross validation to predict compressive strength of concrete in marine environment. Constr Build Mater 206:355–363. https://doi.org/10.1016/j.conbuildmat.2019.02.071
54. Özcan F, Atiş CD, Karahan O, Uncuoğlu E, Tanyildizi H (2009) Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete. Adv Eng Softw 40(9):856–863. https://doi.org/10.1016/j.advengsoft.2009.01.005
55. Bilim C, Atiş CD, Tanyildizi H, Karahan O (2009) Predicting the compressive strength of ground granulated blast furnace slag concrete using artificial neural network. Adv Eng Softw 40(5):334–340. https://doi.org/10.1016/j.advengsoft.2008.05.005
56. Sua-iam G, Makul N (2017) Incorporation of high-volume fly ash waste and high-volume recycled alumina waste in the production of self-consolidating concrete. J Clean Prod 159:194–206
57. Abd El-Mohsen M, Anwar AM, Adam IA (2015) Mechanical properties of self-consolidating concrete Incorporating Cement kiln dust. HBRC J 11(1):1–6
58. Kannan V (2018) Strength and durability performance of self compacting concrete containing self-combusted rice husk ash and metakaolin. Constr Build Mater 160:169–179. https://doi.org/10.1016/j.conbuildmat.2017.11.043
59. Al-Mughanam T, Aldhyani THH, Alsubari B, Al-Yaari M (2020) Modeling of compressive strength of sustainable self-compacting concrete incorporating treated palm oil fuel ash using artificial neural network. Sustain 12(22):1–13. https://doi.org/10.3390/su12229322

60. Elemam WE, Abdelraheem AH, Mahdy MG, Tahwia AM (2020) Optimizing fresh properties and compressive strength of self-consolidating concrete. Constr Build Mater 249:118781. https://doi.org/10.1016/j.conbuildmat.2020.118781
61. Safiuddin M, Raman SN, Salam MA, Jumaat MZ (2016) Modeling of compressive strength for self-consolidating high-strength concrete incorporating palm oil fuel ash. Materials (Basel) 9(5). https://doi.org/10.3390/ma9050396
62. Uysal M, Tanyildizi H (2012) Estimation of compressive strength of self compacting concrete containing polypropylene fiber and mineral additives exposed to high temperature using artificial neural network. Constr Build Mater 27(1):404–414. https://doi.org/10.1016/j.conbuildmat.2011.07.028
63. Uysal M, Tanyildizi H (2011) Predicting the core compressive strength of self-compacting concrete (SCC) mixtures with mineral additives using artificial neural network. Constr Build Mater 25(11):4105–4111. https://doi.org/10.1016/j.conbuildmat.2010.11.108
64. Siddique R, Aggarwal P, Aggarwal Y (2011) Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks. Adv Eng Softw 42(10):780–786. https://doi.org/10.1016/j.advengsoft.2011.05.016
65. Shi C, Wu Z, Xiao J, Wang D, Huang Z, Fang Z (2015) A review on ultra high performance concrete: Part I. Raw materials and mixture design. Constr Build Mater 101:741–751
66. Yoo D-Y, Kang S-T, Yoon Y-S (2016) Enhancing the flexural performance of ultra-high-performance concrete using long steel fibers. Compos Struct 147:220–230
67. ASTM (2017) Standard practice for fabricating and testing specimens of ultra-high performance concrete. ASTM C1856/C1586-17
68. Line JG (2010) Recommendations for design and construction of ultra high strength fiber reinforced concrete structures. JSCE Draft Version-Appendix 5:1–5
69. Zhou M, Wu Z, Ouyang X, Hu X, Shi C (2021) Mixture design methods for ultra-high-performance concrete—a review. Cem Concr Compos 124(September):104242. https://doi.org/10.1016/j.cemconcomp.2021.104242
70. Han Q, Gui C, Xu J, Lacidogna G (2019) A generalized method to predict the compressive strength of high-performance concrete by improved random forest algorithm. Constr Build Mater 226:734–742. https://doi.org/10.1016/j.conbuildmat.2019.07.315
71. Öztaş A, Pala M, Özbay E, Kanca E, Çağlar N, Bhatti MA (2006) Predicting the compressive strength and slump of high strength concrete using neural network. Constr Build Mater 20(9):769–775. https://doi.org/10.1016/j.conbuildmat.2005.01.054
72. Gupta R, Kewalramani MA, Goel A (2006) Prediction of concrete strength using neural-expert system. J Mater Civ Eng 18(3):462–466. https://doi.org/10.1061/(asce)0899-1561(2006)18:3(462)
73. Luukkonen T, Abdollahnejad Z, Yliniemi J, Kinnunen P, Illikainen M (2018) One-part alkali-activated materials: a review. Cem Concr Res 103:21–34
74. Pacheco-Torgal F, Labrincha J, Leonelli C, Palomo A, Chindaprasit P (2014) Handbook of alkali-activated cements, mortars and concretes. Elsevier
75. Singh B, Ishwarya G, Gupta M, Bhattacharyya SK (2015) Geopolymer concrete: a review of some recent developments. Constr Build Mater 85:78–90
76. Sadeghian G, Behfarnia K, Teymouri M (2022) Drying shrinkage of one-part alkali-activated slag concrete. J Build Eng 51:104263
77. Upreti K et al (2022) Prediction of mechanical strength by using an artificial neural network and random forest algorithm. J Nanomater 2022. https://doi.org/10.1155/2022/7791582
78. Tang YX et al (2022) Artificial neural network-forecasted compression strength of alkaline-activated slag concretes. Sustain 14(9). https://doi.org/10.3390/su14095214
79. Qin X, Ma Q, Guo R, Song Z, Lin Z, Zhou H (2022) Compressive strength prediction of alkali-activated slag concretes by using artificial neural network (ANN) and alternating conditional expectation (ACE). Adv Civ Eng 2022. https://doi.org/10.1155/2022/8214859
80. Ibrahim M, Salami BA, Amer Algaifi H, Kalimur Rahman M, Nasir M, Ewebajo AO (2021) Assessment of acid resistance of natural pozzolan-based alkali-activated concrete: Experimental and optimization modelling. Constr Build Mater 304(September):124657. https://doi.org/10.1016/j.conbuildmat.2021.124657

81. Nagajothi S, Elavenil S (2020) Influence of aluminosilicate for the prediction of mechanical properties of geopolymer concrete—artificial neural network. SILICON 12(5):1011–1021. https://doi.org/10.1007/s12633-019-00203-8
82. Chen H-J, Yen T, Chen K-H (2003) Use of building rubbles as recycled aggregates. Cem Concr Res 33(1):125–132
83. Jian S-M, Wu B (2021) Compressive behavior of compound concrete containing demolished concrete lumps and recycled aggregate concrete. Constr Build Mater 272:121624
84. Kazemi M, Madandoust R, de Brito J (2019) Compressive strength assessment of recycled aggregate concrete using Schmidt rebound hammer and core testing. Constr Build Mater 224:630–638
85. Pacheco JN, De Brito J, Chastre C, Evangelista L (2019) Probabilistic conversion of the compressive strength of cubes to cylinders of natural and recycled aggregate concrete specimens. Materials (Basel) 12(2):280
86. Corinaldesi V (2011) Structural concrete prepared with coarse recycled concrete aggregate: from investigation to design. Adv Civ Eng 2011.
87. Safiuddin UJ, Salam AA, Jumaat MZ, Jaafar FF, Saad HB et al (2011) Properties of high-workability concrete with recycled concrete aggregate. Mater Res 14:248–255
88. Kwan WH, Ramli M, Kam KJ, Sulieman MZ (2012) Influence of the amount of recycled coarse aggregate in concrete design and durability properties. Constr Build Mater 26(1):565–573
89. Chakradhara Rao M, Bhattacharyya SK, Barai SV (2011) Influence of field recycled coarse aggregate on properties of concrete. Mater Struct 44(1):205–220
90. Zeng Z et al (2021) Accurate prediction of concrete compressive strength based on explainable features using deep learning. Constr Build Mater 329(September):127082. https://doi.org/10.1016/j.conbuildmat.2022.127082
91. Hammoudi A, Moussaceb K, Belebchouche C, Dahmoune F (2019) Comparison of artificial neural network (ANN) and response surface methodology (RSM) prediction in compressive strength of recycled concrete aggregates. Constr Build Mater 209:425–436. https://doi.org/10.1016/j.conbuildmat.2019.03.119
92. Golafshani EM, Behnood A (2018) Application of soft computing methods for predicting the elastic modulus of recycled aggregate concrete. J Clean Prod 176:1163–1176. https://doi.org/10.1016/j.jclepro.2017.11.186
93. Topçu IB, Saridemir M (2008) Prediction of mechanical properties of recycled aggregate concretes containing silica fume using artificial neural networks and fuzzy logic. Comput Mater Sci 42(1):74–82. https://doi.org/10.1016/j.commatsci.2007.06.011
94. Amiri M, Hatami F (2022) Prediction of mechanical and durability characteristics of concrete including slag and recycled aggregate concrete with artificial neural networks (ANNs). Constr Build Mater 325(October):126839. https://doi.org/10.1016/j.conbuildmat.2022.126839
95. Shahmansouri AA, Bengar HA, Ghanbari S (2020) Compressive strength prediction of eco-efficient GGBS-based geopolymer concrete using GEP method. J Build Eng 31:101326
96. Mesbah HA, Lachemi M, Aitcin P-C (2002) Determination of elastic properties of high-performance concrete at early ages. Mater J 99(1):37–41
97. Yan K, Shi C (2010) Prediction of elastic modulus of normal and high strength concrete by support vector machine. Constr Build Mater 24(8):1479–1485. https://doi.org/10.1016/j.conbuildmat.2010.01.006
98. Demir F (2008) Prediction of elastic modulus of normal and high strength concrete by artificial neural networks. Constr Build Mater 22(7):1428–1435
99. Demir F (2005) A new way of prediction elastic modulus of normal and high strength concrete—fuzzy logic. Cem Concr Res 35(8):1531–1538
100. Mhaya AM, Fahim Huseien G, Faridmehr I, Razin Zainal Abidin A, Alyousef R, Ismail M (2021) Evaluating mechanical properties and impact resistance of modified concrete containing ground Blast Furnace slag and discarded rubber tire crumbs. Constr Build Mater 295:123603. https://doi.org/10.1016/j.conbuildmat.2021.123603
101. Hendi A, Behravan A, Mostofinejad D, Moshtaghi SM, Rezayi K (2017) Implementing ANN to minimize sewage systems concrete corrosion with glass beads substitution. Constr Build Mater 138:441–454. https://doi.org/10.1016/j.conbuildmat.2017.02.034

102. Suleiman AR, Nehdi ML (2017) Modeling self-healing of concrete using hybrid genetic algorithm-artificial neural network. Materials (Basel) 10(2). https://doi.org/10.3390/ma10020135
103. Pham A-D, Hoang N-D, Nguyen Q-T (2016) Predicting compressive strength of high-performance concrete using metaheuristic-optimized least squares support vector regression. J Comput Civ Eng 30(3):1–4. https://doi.org/10.1061/(asce)cp.1943-5487.0000506
104. Algaifi HA et al (2021) Machine learning and RSM models for prediction of compressive strength of smart bio-concrete. Smart Struct Syst 28(4):535–551. https://doi.org/10.12989/sss.2021.28.4.535
105. Asadi Shamsabadi E, Roshan N, Hadigheh SA, Nehdi ML, Khodabakhshian A, Ghalehnovi M (2022) Machine learning-based compressive strength modelling of concrete incorporating waste marble powder. Constr Build Mater 324(September 2021):126592. https://doi.org/10.1016/j.conbuildmat.2022.126592
106. Feng DC et al (2020) Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach. Constr Build Mater 230:117000. https://doi.org/10.1016/j.conbuildmat.2019.117000
107. Yu Y, Li W, Li J, Nguyen TN (2018) A novel optimised self-learning method for compressive strength prediction of high performance concrete. Constr Build Mater 184:229–247. https://doi.org/10.1016/j.conbuildmat.2018.06.219
108. Behnood A, Verian KP, Modiri Gharehveran M (2015) Evaluation of the splitting tensile strength in plain and steel fiber-reinforced concrete based on the compressive strength. Constr Build Mater 98:519–529. https://doi.org/10.1016/j.conbuildmat.2015.08.124

Reliability-Based Design Optimization of Detention Rockfill Dams and Investigation of the Effect of Uncertainty on Their Performance Using Meta-Heuristic Algorithm

Mohammad Mehdi Riyahi, Hossien Riahi-Madvar, and Iman Bahrami Chegeni

Abstract Flood is one of the natural disasters which is of particular importance due to the financial, human, and environmental damages which directly and indirectly inflicts on human societies. For this reason, researchers today have turned to appropriate solutions for flood management to reduce the effects of floods. One of the most suitable structural solutions is the construction of detention rockfill dams to control and mitigate flood damage. Such dams are very popular due to their rapid construction and easy operation. At first, for designing detention rockfill dams, one must select suitable locations for dams. In the second step, the preliminary design of the dam is done to obtain the height and length of the dam, and in the last step, the final design and optimization of the dam are done. In this research, the second and third design steps, i.e., the preliminary and final designs, are performed to obtain the initial height and length of the dam. Then the optimization of the dams is done to provide structural safety factors. For the preliminary design, the input hydrograph equations, the reservoir's volume-height relationship, the dam's stage-discharge equation, and the flow routing equation in the detention rockfill dams and their combination with each other are used. Metaheuristic algorithms are also used for the final design and optimization of the detention rockfill dam. In this research, a self-adaptive genetic algorithm has been used to optimize the dimensions of the detention rockfill dam. Then, using the Monte Carlo simulation method, the effects of uncertainty of design parameters on the hydraulic and structural performance of detention rockfill dam are investigated. It has been shown how uncertainty can change hydraulic performance by studing the dam storage volume and flow through the dam. The structural

---

M. M. Riyahi
Faculty of Civil Engineering and Architecture, Shahid Chamran University of Ahvaz, Ahvaz, Iran
e-mail: mo_riyahi@yahoo.com

H. Riahi-Madvar
Department of Water Engineering, Faculty of Agriculture, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran
e-mail: h.riahi@vru.ac.ir

I. Bahrami Chegeni (✉)
Faculty of Engineering, Lorestan University, Khorramabad, Iran
e-mail: bahrami.i@lu.ac.ir

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*, Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_8

implementation is also evaluated due to the uncertainty propagation on the safety factors. At the end of this chapter, a reliability-based design optimization (RBDO) of the detention rockfill dam was carried out using self-adaptive NSGA-II.

Keywords Detention rockfill dams · Optimization · Uncertainty · Flood management · Reliability-based design optimization

1 Introduction

Today, floods are one of the extreme events that cause several financial and life threats worldwide. Climate change is one of the leading causes that create the flood crisis. There are two main methods for flood control and management, structural and non-structural. The non-structural methods include developing flood warning systems, decision support systems, flood forecasting models, and integrated watershed management methods [1–6]. The structural methods use physical structures to control and mitigate flood peaks, including detention rockfill dams. Due to their easy construction, environmentally friendly, and flexibility in operation, detention rockfill dams have high acceptability. The detention dams in flood events attenuate the flood peak and increase the time to flood peak. The detention rockfill dams are constructed from rock pieces, pebbles, and permeable dam bodies. The coarse porous media of these dams deplete the stored water automatically without any operator action required. Also, dam break risk in these structures is smaller than in earthfall dams [7–9].

The design procedure for these dams has three main steps:

1. In the first stage, the dam location finding determines the dam storage needed to reduce the flood peak with a predetermined return period based on the safe conditions downstream.
2. Designing the detention rockfill dam to derive the preliminary dimensions and sizing of the dam.
3. In the third stage, the optimum dam design based on the preliminary design is done and single or multi-objective optimization algorithms can be used in this stage.

Based on these steps, there are several graphical or mathematical methods for the preliminary design of detention rockfill dams. The graphical methods were developed and used in early studies [10–16]. In these studies, the designs use single or double -orifice outputs without infiltration from the dam body. In these studies, the depth-storage relation of the dam is linear assumed, while in the detention rockfill dams, this relation is non-linear. Another graphical method provided by the United States Soil Conservation Service (SCS) uses two curves for the preliminary design of a detention rockfill dam. One of the curves is the relationship of the storage, which is obtained from the ratio of peak storage to the total volume of the flood. The other curve is the coefficient of peak flood, which shows the ratio of the maximum output flow to the maximum input flow [12]. In the 1990s, Akan [10] presented a graphical method that was used to obtain the structure's size using the input hydrograph obtained from the SCS method. Another graphical method of SCS uses two curves of peak storage ratio to total flood volume and proportion of peak flood discharge to peak inlet discharge [12]. In addition to the graph-based methods mentioned above, other equations have been developed to determine the reservoir volume according to the downstream conditions and the type of hydrograph. Baker [17] presents one of these equations suitable for where the inlet and outlet hydrographs are triangular. This equation is mentioned below [11].

$$\frac{S_f}{V_f} = 1 - \frac{Q_p}{I_p} \tag{1}$$

where V_f is the flood volume, S_f is the required dam storage, I_p is the peak of the input hydrograph, Q_p is the discharge peak of the outlet. Abt and Grigg [12] proposed the following equation where the inlet hydrograph is triangular, and the outlet hydrograph is trapezoidal:

$$\frac{S_f}{V_f} = \left(1 - \frac{Q_p}{I_p}\right)^2 \tag{2}$$

Wycoff and Singh [13], based on numerical simulation of floods and some parametric analysis, developed the following equation:

$$\frac{S_f}{V_f} = \frac{129\left(1 - \frac{Q_p}{I_p}\right)^{0.753}}{\left(\frac{t_b}{T}\right)^{0.411}} \tag{3}$$

where t_b and T are the base time and time to peak in flood hydrograph. It is worth mentioning that in Eq. (3), the effect of the type of output structure is not considered.

McEnrro [11] developed Eqs. (4) and (5) for dams with an overflow weir and detentions dams with bottom orifice outlet, respectively:

$$\frac{S_f}{V_f} = 0.97 - 1.42\frac{Q_p}{I_p} + 0.82\left(\frac{Q_p}{I_p}\right)^2 - 0.34\left(\frac{Q_p}{I_p}\right)^3 \tag{4}$$

$$\frac{S_f}{V_f} = 0.97 - 1.17\frac{Q_p}{I_p} + 0.77\left(\frac{Q_p}{I_p}\right)^2 - 0.46\left(\frac{Q_p}{I_p}\right)^3 \tag{5}$$

These equations are derived based on a non-permeable body of dams, and the depletion is done using single or double-output orifices or wires. In contrast, in the detention rockfill dams, the body of the dam is porous, and depletion is automatically done based on the non-Darcian flow behavior of these structures. Riahi et al. [18] developed a simple preliminary design procedure for detention rockfill dams by combining non-linear flood routing with the non-Darcian non-linear flow in rockfill media, and based on a parametric study, some design equations are developed. While in the previous studies, the multi-objective design and multi-purpose optimizations in detention rockfill dams are neglected; therefore , based on the metaheuristic optimization methods and uncertainty analysis in dam stability and reliability of hydraulic performance, a new design framework is developed. And a Reliability-based design optimization (RBDO) procedure is developed. The aims of the models are the minimization of dam costs and an increase in the reliability of the dam. The Monte Carlo Simulation method has been used to calculate the reliability index. Also, the Latin hypercube sampling (LHS) coupling with the rejection rule is used for generating samples. It should also note that single and multi-objective self-adaptation GA is used to perform single and multi-objective optimization.

2 Material and Methods

The developed model uses single-objective optimization for preliminary design and multi-objective optimization for reliability-based design optimization and investigating the effects of uncertainty on the hydraulic performance of the dam. In the first stage, a preliminary design is done using the developed method by Riahi et al. [18]; then, the preliminary dimensions of the dam are used as lower bounds in the optimization algorithm, which is based on self-adaptive optimization. This novelty causes increasing the convergence speed of the optimization model. The effects of uncertainty on designs and hydraulic performance are investigated using Monte-Carlo simulation, LHS, and rejection rule. Finally, the detention rockfill dam is designed based on RBDO using multi-objective optimization. Figure 1 shows the steps of this research.

2.1 Governing Equations in the First Design Step

This section presents the governing equations in the preliminary hydraulic design of detention rockfill dams.

2.1.1 Inflow Flood Hydrograph

The gamma distribution function has been used in many studies for the inflow flood hydrograph [19–24]. The equation of the gamma probability distribution function is written below [20].

$$I = I_p \left(\frac{t}{t_p} \right)^m \exp\left(-m \left(\frac{t}{t_p} - 1 \right) \right) \tag{6}$$

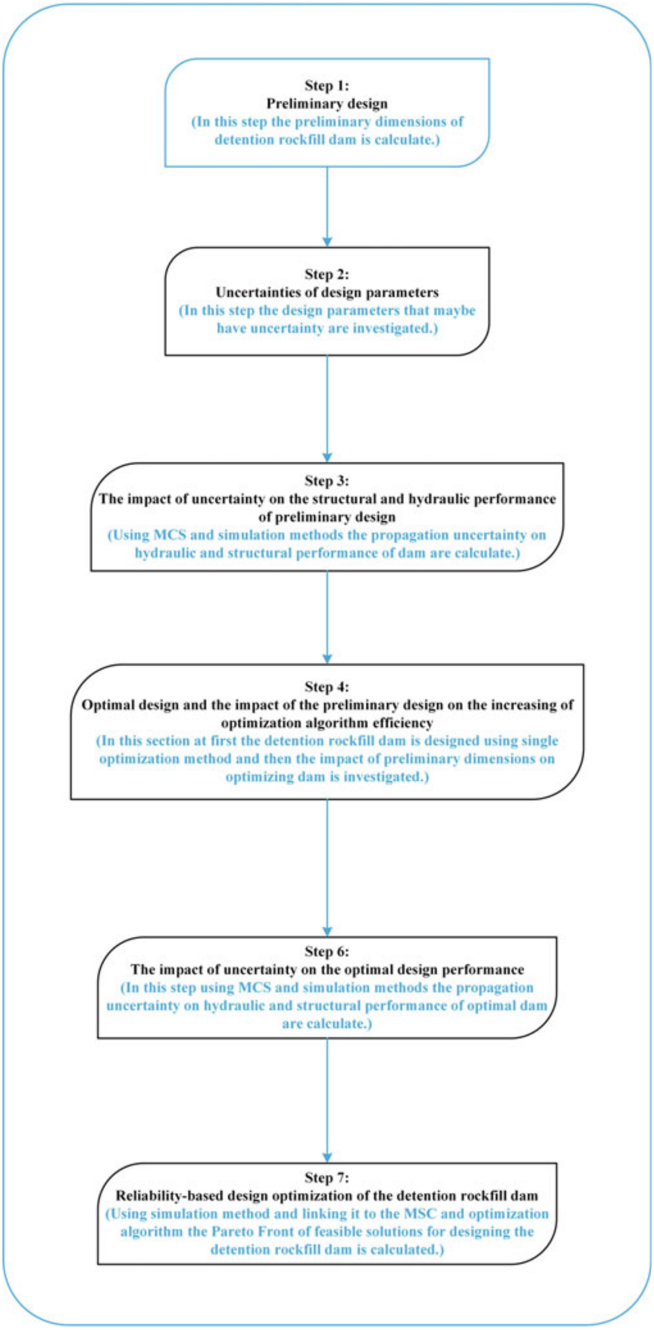


Fig. 1 Research steps

where t_p is the time when the inflow flood hydrograph reaches the inflow flood peak discharge, m is the dimensionless coefficient of the hydrograph shape, I represent the inlet hydrograph discharge, and I_p indicates the inlet hydrograph peak. By integrating Eq. (1) over time the flood volume derived is:

$$V_f = I_p t_p m^{-(m+1)} \exp(m) \Gamma(m+1) \tag{7}$$

where V_f is the flood volume and Γ is the gamma function.

2.1.2 Depth-Volume Relationship in the Reservoir

The depth-volume equation of the dam derived from the depth-area equations ($A = k(H + z_0)^n$) and $A = dS/dh$ is follows:

$$S = \frac{k}{n+1}\left[(H + z_0)^{n+1} - z_0^{n+1}\right] \tag{8}$$

where S is the dam volume, H is the water depth, A is the area, and Z_0, k, and n are constants.

2.1.3 Stage-Discharge in Detention Rockfill Dam

The analytical non-linear stage-discharge equation of detention rockfill dams derived by Samani et al. [25] is:

$$Q = W\left[\frac{H_1^{b+3} - H_2^{b+3}}{L - 0.7 H_1 \cot\theta} \times \frac{1}{\alpha(b+3)}\right]^{\frac{1}{b+2}} \tag{9}$$

In which, α is:

$$\alpha = \frac{a(d - \sigma)^{b-1}}{2 g v^b n_p^{b+1}} \tag{10}$$

where α and β are the constant coefficients, d shows the dam material size (grain size), σ is the standard deviation of the dam material size, g is the gravity acceleration, and n_p indicates the material porosity. L demonstrates the dam length in the flow direction, W is the dam width perpendicular to the flow direction, H_1 and H_2 are the water depths upstream and downstream, and θ is the upstream and downstream slope angle. In a study by Samani et al. [25], the optimal values of α and β using optimization were obtained as 54 and -0.077, respectively.

2.1.4 Non-Linear Flow Routing in Rockfill Dam

The non-linear flood routing in rockfill dams was performed using a continuity equation hybridized (*dS/dt* = *I-Q*) with Eqs. (1–5) is:

$$\frac{dS}{dt} = I_p\left(\frac{t}{t_p}\right)^m exp\left(-m\left(\frac{t}{t_p} - 1\right)\right) - \left(W\left(\frac{\alpha}{b+3}\right)^{\frac{1}{b+2}}\right)\left(\frac{\left((A_0S + A_0)^{\frac{1}{n+1}} - Z_0\right)^{b+3} - H_2^{b+3}}{L - A_2\left((A_0S + A_1)^{\frac{1}{n+1}} - Z_0\right)}\right)^{\frac{1}{b+2}} \quad (11)$$

Riahi et al. [18] developed a numerical model with the parametric study based on this non-linear flood routing and derived the following equation for preliminary designs:

$$\frac{S_f}{V_f} = 1.0166 - 0.231\frac{Q_p}{I_p} - 2.2433\left(\frac{Q_p}{I_p}\right)^2 + 1.4661\left(\frac{Q_p}{I_p}\right)^3 \quad (12)$$

where S_f is the required reservoir volume and Q_p is the outlet discharge peak.

2.2 The Structural Stability of the Dam

In this section, based on the acting forces on the dam body as a rigid structure, the stability and safety factors of the dam are determined [26–30].

2.2.1 The Acting Forces

As shown in Fig. 2, the acting forces over the dam can be summarized in Table 1. The acting forces and moments, with their components in vertical or horizontal directions, affect the dam's stability, as presented in the following sections.

2.2.2 Safety Factors in Detention Rockfill Dams

In this section, based on the acting forces and moment in Table 1, the safety factors against overturning, sliding, and friction are calculated as follows.

- **Safety factor against overturning**

The total resistant moments of dam toe toward the total overturning moments of dam toe are equal to the safety factor against overturning for the rockfill dam as follows:

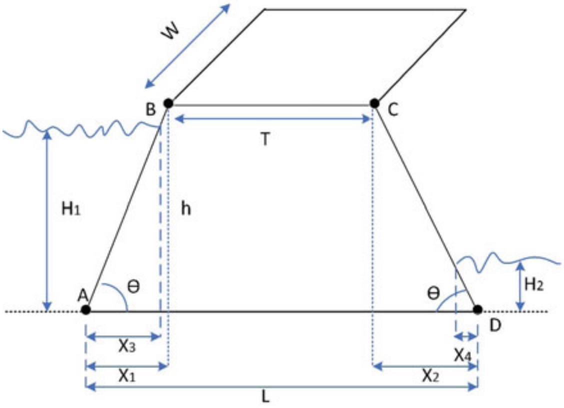


Fig. 2 Detention rockfill dam cross-section

$$SFO = \frac{Me(R)}{Me(O)} \tag{13}$$

$$Me(R) = M(W_1) + M(W_2) + M(W_3) + M(F_2) + M(F_3) + M(F_4) \tag{14}$$

$$Me(O) = M(F_1) + M(U) + M(F_{e1}) + M(F_{e2}) + M(F_{e3}) + Me \tag{15}$$

When SFO exceeds 1.5, the detention rockfill dam is resistant to overturning.

- **Safety factor against sliding**

The total vertical forces toward the total horizontal are equal to the safety factor against the slide of the rockfill dam as follows:

$$SFS = \frac{\mu F_V}{F_H} \tag{16}$$

$$F_V = F_2 + F_4 + W_1 + W_2 + W_3 - U \tag{17}$$

$$F_H = F_1 - F_3 + F_{1e} + F_{2e} + F_{3e} + F_e \tag{18}$$

When SFS exceeds 1.5, the detention rockfill dam is resistant to sliding.

- **Safety factor against friction**

The total ratio of vertical force and allowed shear stress of rockfill dam foundation toward total horizontal force equals the safety factor against friction.

$$SFF = \frac{\mu F_V + qL}{F_H} \tag{19}$$

Table 1 The forces acting on the dam and their moment

No	Equation	Explanation	Force type based on the direction		Force type based on resistance	
			Vertical	Horizontal	Tipper	Resistant
1	$F_1 = 1/2 * \Upsilon_w * H_1^2$	Horizontal pressure force of water in upstream		*	*	
2	$M(F_1) = F_1 * H_1/3$	Horizontal moment of upstream water toward dam toe (D)				
3	$F_2 = 1/2 * \Upsilon_w * H_1 * X_3$	Vertical force of upstream water	*			*
4	$M(F_2) = F_2 * (L - (X_3/3))$	Vertical moment of upstream water toward dam toe (D)				
5	$F_3 = 1/2 * \Upsilon_w * H_2^2$	Horizontal force of downstream water		*		*
6	$M(F_3) = F_3 * H_2/3$	Horizontal moment of downstream water toward dam toe (D)				
7	$F_4 = 1/2 * \Upsilon_w * H_2 * X_4$	Vertical force of downstream water	*			*
8	$M(F_4) = F_4 * X_4/3$	Vertical moment of downstream water toward dam toe (D)				
9	$W_1 = 1/2 * \Upsilon_s * X_1 * h$	Section 1 weight of dam *(W_1)*	*			*
10	$M(W_1) = W_1 * (L - (2 * X_1/3))$	Moment weight of W_1 toward dam toe				
11	$W_2 = 1/2 * \Upsilon_s * T * h$	Section 2 weight of dam *(W_2)*	*			*
12	$M(W_2) = W_2 * (L - -T/2 - X_1)$	Moment weight of W_2 toward dam toe				
13	$W_3 = 1/2 * \Upsilon_C * X_2 * h$	Section 3 weight of dam *(W_3)*	*			*
14	$M(W_3) = W_3 * (2 * X_2/3)$	Moment weight of W_3 toward dam toe				
15	$U = 1/2 * \Upsilon_w * L * (H_1 + H_2)$	Uplift force U	*		*	
16	$M(U) = U * (2 * L/3)$	Uplift force moment toward dam toe				

(continued)

Table 1 (continued)

No	Equation	Explanation	Force type based on the direction		Force type based on resistance	
			Vertical	Horizontal	Tipper	Resistant
17	$F_{e1} = s * W_1$	Static force of W_1 earthquake		*	*	
18	$M(F_{e1}) = F_{e1} * 1/3 * h$	The static force moment of Earthquake W_1				
19	$F_{e2} = s * W_2$	The static force of earthquake W_2		*	*	
20	$M(F_{e2}) = F_{e2} * 1/2 * h$	The static force moment of Earthquake W_2				
21	$F_{e3} = s * W_3$	The static force of Earthquake W_3		*	*	
22	$M(F_{e3}) = F_{e3} * 1/2 * h * (L - X_1 - T)(L - X_1)$	The static force moment of Earthquake W_3				
23	$\beta = atn(X_1/h)$	The angle of the side leading to dam heel (A) with a vertical line in degrees				
24	$Cm = 0.73 * (90 - \beta)/90$	Cm coefficient for calculating the dynamic force of the dam				
25	$Pe = \Upsilon_w * Cm * H_1 * a$	Pe coefficient for calculating the dynamic force of the dam				
26	$Fe = 0.726 * pe * H_1$	Earthquake dynamic force		*	*	
27	$Me = 0.299 * pe * H_1^2$	Moment of dynamic earthquake force toward toe dam D				
28	$W = W_1 + W_2 + W_3$	Total dam weight	*			*

where q is the allowable shear stress of the material, the detention rockfill dam is resistant to friction when SFF is greater than 3.

2.3 Reliability-Based Design Optimization of Detention Rockfill Dams

In this section, the design procedure based on the concept of reliability is developed. According to Fig. 3, three computing steps should be taken for reliability-based design optimization, which is mentioned below.

1. The simulator engine for simulating the hydraulic performance of the dam using Eqs. 6 up to 15.
2. Determine the reliability based on the Monte Carlo Simulation engine. In this step, LHS is used for generating samples, and the rejection rule is implied for deleting some samples which are greater than 15% uncertainty.
3. Multi-objective self-adaptive NSGA-II to find the possible optimum solutions.

2.4 Optimization of Detention Rockfill Dam

In this section, the objective functions of the model are presented. The first objective function is the minimization of the dam cost:

$$Minimize\, C = A_{DRD} \times \gamma_S \tag{20}$$

where A_{DRD} is the dam area and γ_S is the specific height of materials and the decision variable is as follows:

$$X = [x_1, x_2, T, h] \tag{21}$$

The x1, x2, T, and h parameters are shown in Fig. 1. The constraints are safety factors, and the initial sizing of the dam are mentioned below.

$$SFO \geq 1.5, SFS \geq 1.5, SFF \geq 3 \tag{22}$$

$$H_1 \geq H_1', L \geq L' \tag{23}$$

The values of our initial length and height of the dam were derived from the preliminary design step. The second objective function is the reliability objective, which must be maximized, as presented in the Sect. 2.5.

2.4.1 Multi-Purpose Optimization of Rockfill Dams

In this study, the NSGA-II is used for multi-objective optimization, as presented in Fig. 4 [31, 32].

The following steps are used for the multi-objective self-adaptive genetic algorithm [33].

1. Adjusting the GA parameters such as initial population size, number of iterations, mutation rate, etc.
2. Generation of the initial population randomly based on initial predetermined bounds.
3. Evaluation of the objective functions for all populations.
4. Determination of the rate of violation of each member of the population from the constraints.
5. Ranking the populations using non-dominance sorting.
6. Determining the crowding distances for the population.
7. Parent selection.
8. Offspring production.
9. Mutation.
10. Selection of the population using the non-dominance sorting and crowding distances.
11. Repeat the algorithm from step 3 until convergence.

The Self-adaptive NSGA-II algorithm is used for optimization, which uses a self-adaptive strategy for constraints instead of a penalty function. In this algorithm, parent selection is made using Roulette Wheel Selection. The Roulette Wheel Selection is used for the parent selection of chromosomes x and y. The x chromosome is selected as the parent chromosome when the x is feasible and y is infeasible, or x and y are infeasible but the violation of x is lower, or when x and y are feasible and the x cost is lower, else the y is selected.

2.5 Reliability Determination

In a hydraulic system, reliability is the probability of resistance of the system against the loadings presented by Mays [34]:

$$R = P(G(X) > 0) = P(r > l) \tag{24}$$

where R is the reliability, P is the probability, $G(X)$ is the representing function between safety and failure conditions, r and l represent resistance and loadings, X is $(X_1, X_2, .., X_n)^T$ as a random variable. Using the mass density function, reliability can be written as [35, 36]:

$$R = \int\limits_{G(X)>0} f_x(X)dx \tag{25}$$

where f_x is the probability density function of random variables, and the Monte-Carlo Simulation is used for the sample-based determination of R.

2.5.1 Monte-Carlo Simulation

The Monte-Carlo Simulation is one of the most efficient methods in reliability calculation, and its simple-to-use procedure caused frequent applications [37, 38]. This method determiness the probability density function of input parameters, and the sample generation engine is used to produce random values for model parameters. The MCS equation based on the safety probability can be written as follows:

$$p_R = \int \ldots \int I[G(X) > 0] f_x(X)dx \approx \frac{1}{N}\sum_{i=1}^{n} I[G(X) > 0] \approx \frac{n_s}{N} \tag{26}$$

where n_s is the number of samples located in the safe region, N is the total sample numbers, and I is the counter function with one value for the safe region and zero value for the failure region.

3 Results

This part examines the results of numerous analyses on the detention rockfill dam. Following is a description of an example extracted from the literature. Next, the preliminary design of detention rockfill dams is evaluated. The uncertainty in the design parameters and their effects on the preliminary design are addressed. Then the case study is optimized utilizing the dimensions derived from the preliminary design. In the end, the reliability-based design optimization of the detention rockfill dam will be done.

3.1 Case Study

The case study in all of the steps discussed in this research is extracted from Riahi et al. [18]. The details of the case study are shown in Table 2.

Table 2 The case study parameters

Parameters	b	a	n	k	z_0	m	t_p	Q_p	I_p
Value	−0.077	54	2	11,000	0	10	3600	11	28
Parameters	θ	w	n_p	σ	d	ν			
Value	90	2	2	0	0.25	0.000001			

3.2 Preliminary Design

The preliminary design of the detention rockfill dam is obtained using Eqs. 6–15. The initial dam height (H) and initial dam length (L) are determined using these equations. Using the case study parameters, listed in Table 2 and designing the detention rockfill dam as preliminary, H and L are equal to 3.76 and 3.44 m, respectively. Using Eqs. 16 through 20 can determine the values of the dam's safety factors. For the preliminary design, the SFS, SFO, and SFF safety factors have corresponding values of 0.7211, 0.9473, and 3.3801. The values of the safety factors of the preliminary design show that the safety factors are not satisfied for this case, except for SFF, which demonstrates the importance of optimization in reaching an acceptable and safe design.

3.3 Uncertainties of Design Parameters

In order to evaluate the impacts of uncertainty on the structural and hydraulic performance of the detention rockfill dam, it is necessary first to identify the design parameters that are subject to uncertainty and assign them the appropriate uncertainty value. To this end, Table 3 lists all design parameters that are susceptible to uncertainty and affect the structural and hydraulic performance of the rockfill dam. The input parameters are included in this table with their categories, units, standard deviation values, distribution patterns, and average values for each parameter. This study employs three probability distribution functions: the uniform distribution function, the Gaussian distribution function, and the Generalized extreme value distribution. The Generalized extreme value distribution function was applied to variables such as the input hydrograph's maximum flow rate, the time the hydrograph reaches its peak, and the maximum value of the output hydrograph flow. The Gaussian distribution function is used for constant coefficients, whereas the uniform distribution function is applied to the rockfill dam's materials.

Table 3 The uncertainty of input parameters

Number	Type	Parameter	Definition	Unit (SI)	Type of Probability Distribution	Standard Deviation	Crisp value (Known as mean)	Formula
1	Discharge	I_p	inlet hydrograph peak	m^3/s	GEV	5%	28	2
2		Q_p	outlet hydrograph peak		GEV	5%	11	7
3	Time	t_p	The time when the inflow flood hydrograph reaches the inflow flood peak discharge	s	GEV	5%	3600	2
4	Coefficients	m	The dimensionless coefficient of the hydrograph shape	–	Gaussian	5%	10	2
5		z_0	Constant coefficients of the reservoir		Gaussian	5%	0	3
6		k	Constant coefficients of the reservoir		Gaussian	5%	11,000	3
7		n	Constant coefficients of the reservoir		Gaussian	5%	2	3
8		a	Constants from optimization		Gaussian	5%	54	4
9		b	Constants from optimization		Gaussian	5%	– 0.077	4
10		s	Earthquake coefficient		Gaussian	5%	0.1	8
11		μ	Static friction coefficient		Gaussian	5%	0.7	9
12	Material	d	the dam material size	m	Uniform	5%	0.25	4

(continued)

Table 3 (continued)

Number	Type	Parameter	Definition	Unit (SI)	Type of Probability Distribution	Standard Deviation	Crisp value (Known as mean)	Formula
13		σ	the standard deviation of the dam material size		Gaussian	5%	0	4
14		n_p	The material porosity		Uniform	5%	0.2	4
15		γ_S	Specific gravity of materials (stone)	N/m^3	Gaussian	5%	27,000	
16	Length	H_2	water depths downstream	m	Gaussian	5%	1	4
17		w	Dam width		Uniform	5%	2	4
18	Angle	θ	The slope angle of the upstream and downstream	degrees	Uniform	5%	90	4
19	Stress	q	The allowed shear stress of materials at the shear surface	KN/m^2	Gaussian	5%	70	10

3.4 The Impact of Uncertainty on the Structural and Hydraulic Performance of Preliminary Design

This section examines the implications of design parameter uncertainty on the structural and hydraulic performance of a preliminary design detention rockfill dam. To achieve this goal and take advantage of the Monte Carlo simulation method, we should generate a significant number of input parameters of the detention rockfill dam. In this study, 100,000 samples were generated for the input parameters based on their probability distribution using the LHS method. Then based on the sampling rejection principle, values with more than 15% uncertainty were ignored, and the data set was formed. The dataset is input into the simulation model to generate the desired outcomes.

First, applying Eqs. 7, 10, and 13–15, the outputs relevant to the hydraulic performance of the rockfill dam are derived. Figure 5 depicts the histogram values V_f, H_1, and S_f. As shown in Fig. 5, the upper and lower bounds for V_f are 143,650 and 63,048, for H_1 are 6.31 and 2.23, and for S_f are 113,170 and 25,025, respectively. Furthermore, the average and standard deviation for V_f are 85,391 and 7891, respectively; for H_1 are 3.92 and 0.3859, respectively, and for S_f are 57,044 and 7926 0.7606, respectively. Uncertainty in the input parameters propagates uncertainty in the detention rockfill dam's hydraulic output and also produces uncertainty in the detention rockfill dam's hydraulic performance. The associated uncertainty values for V_f, S_f, and H_1 are (+78.29, −21.75), (+110.11, −53.54), and (+67.82, −40.22), respectively.

The effects of uncertainty on the structural performance of the detention rockfill dam are calculated using Eqs. 16–20. Figure 6 shows the histogram of safety factors. According to the histogram obtained from the effects of uncertainty on SFO values, it can be concluded that the presence of uncertainty in the design parameters has caused an impact on the safety factor of SFO to the extent of + 47.61 and −21.78% (compared to the average value of 1.1276). Moreover, the presence of uncertainty in the design parameters has resulted in the propagation of uncertainty on the SFS and SFF values of (+87.81, −32.63), and (+65.07, −28.05), respectively (respectively, compared to 1.0274 and 3.5567). As a result, the safety factors with the most significant uncertainty are +87.81 and −32.63, which are both associated with SFS.

3.5 Optimal Design and the Impact of the Preliminary Design on the Increasing of Optimization Algorithm Efficiency

This section examines the optimal design of the detention rockfill dam. Additionally, the implications of the preliminary design on the optimization algorithm speed are investigated. A self-adaptive GA algorithm is applied for the detention rockfill dam's optimal design. Equation 20 represents the objective function of the optimization algorithm and Eqs. 21–23 provide the optimization problem's constraints that must be satisfied. The optimization procedure is divided into two approaches: (1) without using the preliminary design dimensions and (2) using the preliminary design dimensions as the lower bound of the decision variables (Eq. 23). In both approaches, the optimization algorithm is run ten times for this purpose. After ten runs, the best run, which has the best cost function, is determined for both approaches. Figure 7 depicts the results of 10 runs for both approaches. The cost value in the optimal solution for the first approach (without considering the optimal design's dimensions) is 691009, and the h and L are 5.8358 and 4.3855 m, respectively. Additionally, the SFS, SFO, and SFF for this optimal solution are 1.5668, 1.5000, and 4.4163, respectively. The cost value in the optimal solution for the second approach (taking the dimensions of the preliminary design into account) is 690970, and the h and L are 5.8575 and 4.3690 m, respectively. Additionally, the SFS, SFO, and SFF for this optimal solution are 1.5692, 1.5000, and 4.4081, respectively. The optimal dimensions of this design are very similar to the dimensions of the preliminary design. So, the preliminary design can be considered a preliminary plan to identify the initial dam dimensions, eventually leading to the optimal design.

Comparing the results of the two approaches reveals that the first approach requires 2.1677 s to achieve the optimal value in the optimization algorithm, whereas the second approach requires only 1.9037 s. This result demonstrates a 13% reduction in runtime by utilizing the preliminary design dimensions in the second optimization procedure. In addition, the most optimal cost achieved by the first approach is 691009. In contrast, the most optimal cost achieved by the second approach is 690970.13, demonstrating that the speed of the second optimization procedure is enhanced when the preliminary design is considered in the optimization algorithm. Comparing Fig. 5a and b can be seen that in the second approach, the value of the objective function in the first iteration is considerably smaller than in the first approach. As a result, using preliminary design to generate preliminary dimensions and then using these dimensions as the Lower Bound in the optimization algorithm has sped up and boosted the algorithm's efficiency in finding the best feasible solution.

3.6 The Impact of Uncertainty on the Structural Performance of Optimal Design

The effects of uncertainty on the preliminary design of the detention rockfill dam were addressed in earlier sections. It was stated that the safety factors were not completely satisfied in the preliminary design of the detention rockfill dam. As a result, to avoid accidents during operation and obtain the safety of the detention rockfill dam, we are attempting to achieve an optimal and safe design from single-objective optimization by using the safety factors as constraints of the optimization problem. The effects of design parameter uncertainty on the structural and hydraulic performance of the optimal detention rockfill dam design are examined in this section using the Monte Carlo simulation method. To this end, Table 3 in Sect. 3.3 is used here to assign uncertainties to the design parameters. In this section, sampling data were generated using the LHS method, and the rejection rule was applied to any data with an uncertainty of more than 15%. Equations 13–19 can be used to determine the detention rockfill dam's structural performance under uncertain conditions. These equations can be used to determine the histogram values for SFO, SFS, and SFF, which are displayed in Fig. 8.

Figure 8 shows the upper and lower bounds of SFO, SFS, and SFF values, which are equal to (0.7988, 2.4062), (0.5890, 3.2807), and (1.8905, 8.8715), respectively. The propagation of uncertainty on the dam's hydraulic performance may be obtained using the upper and lower bounds of the values of SFO, SFS, and SFF. The uncertainty propagation for SFO, SFS, and SFF are (+60.41, −46.75), (+109.07, −62.46), and (+101.25, −57.11), respectively. In addition, the values of SFO, SFS, and SFF are depicted using the Violin plot in Fig. 9.

3.7 Reliability-Based Design Optimization of the Detention Rockfill Dam

In the preceding sections, the shortcomings of the preliminary design and the optimal design approaches in the presence of uncertainty in the design parameters, as well as the propagation of uncertainty on the hydraulic and structural performance of the detention rockfill dam, were demonstrated. By linking the simulation model to the optimization model and employing the Monte Carlo Simulation method, an attempt has been made in this section to perform the optimal design of the detention rockfill dam with different degrees of reliability. A multi-objective optimization algorithm known as self-adaptive NSGA-II was employed for this purpose; it offers the constraints of the structure's safety factors for the new method. In addition, the first objective function, the construction cost of a detention rockfill dam, is equal to Eq. 20, and the second objective function, which is the reliability of the dam, is equivalent to Eq. 26. The flowchart for this process is depicted in Fig. 3.

To implement the reliability-based design optimization, a dam with the characteristics listed in Table 2 was utilized. This section has calculated four scenarios with different dam angles, corresponding to 90, 80, 75, and 70 degrees, respectively. The parameters of the multi-objective optimization algorithm for all scenarios of the detention rockfill dam are as follows: initial population size is 50, the number of iterations is 1000, the mutation rate of 5%, and the crossover rate of 90%. The Pareto front generated by this method is depicted in Fig. 10.

Figure 10 illustrates the output of the optimization algorithm, a Pareto Front consisting of a wide range of feasible answers. An answer with a reliability level of around 70% was selected for each scenario of the detention rockfill dam to evaluate its performance and compare the best scenario between the four scenarios for designing a detention rockfill dam based on RBDO. The values of these selected solutions for different scenarios are shown in Table 4.

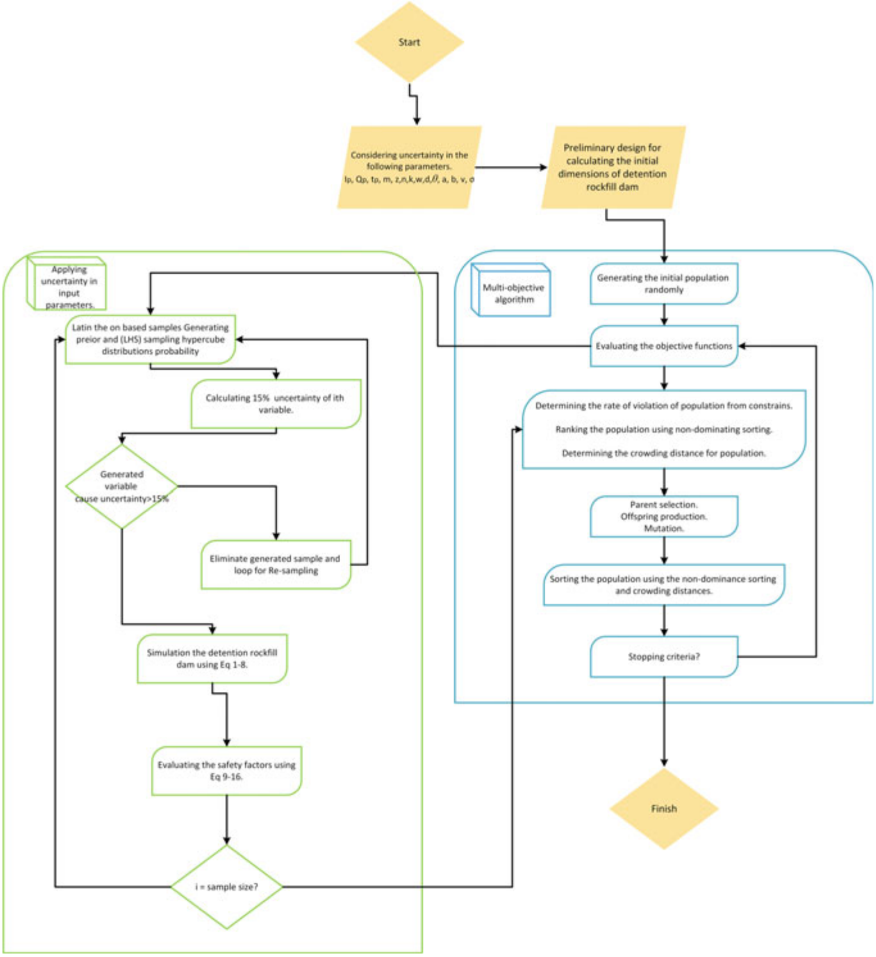


Fig. 3 The flowchart for reliability-based design optimization

Table 4 shows when the detention rockfill dam angle falls, the amount of material utilized, which is regarded as the construction cost of the detention rockfill dam, increases dramatically. For instance, when the angle of the dam is 70°, the cost of constructing a rockfill dam is approximately 40% higher than in the first case, where the dam angle is 90°. On the other hand, by decreasing the dam angle, the safety factors increase. For instance, when the angle is 90°, the corresponding SFO, SFF, and SFS values in the detention rockfill dam are 1.6736, 4.7441, and 1.8598. These values change to 1.7079, 2.2947, and 7.4947, respectively, when the dam angle reaches 70°. Moreover, as the angle of the dam lowers, the length of the dam body increases, and its height drops.

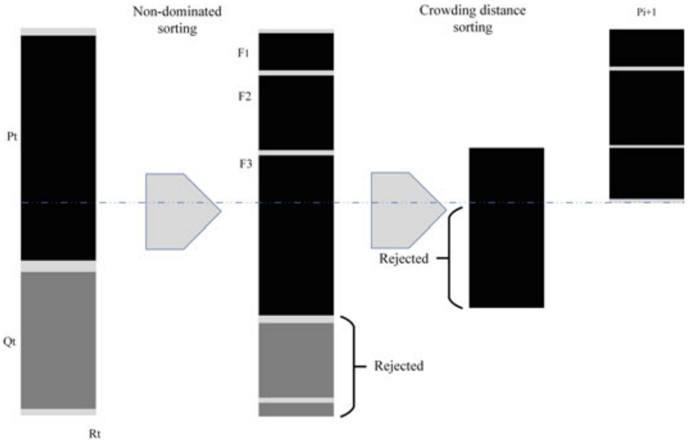
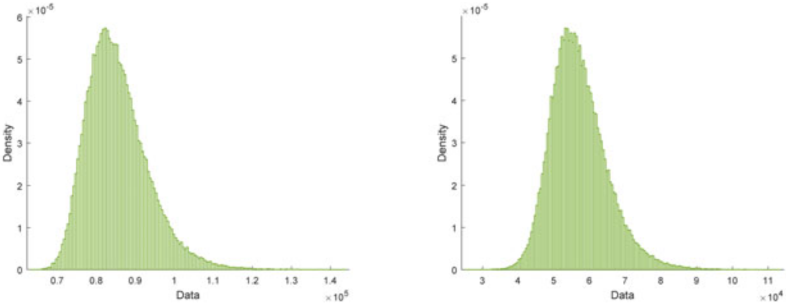
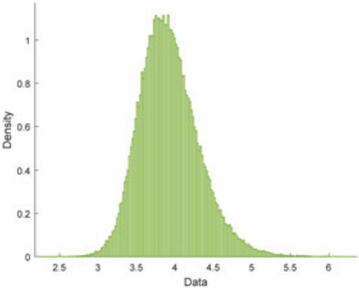


Fig. 4 The optimization algorithm of NSGA-Ⅱ



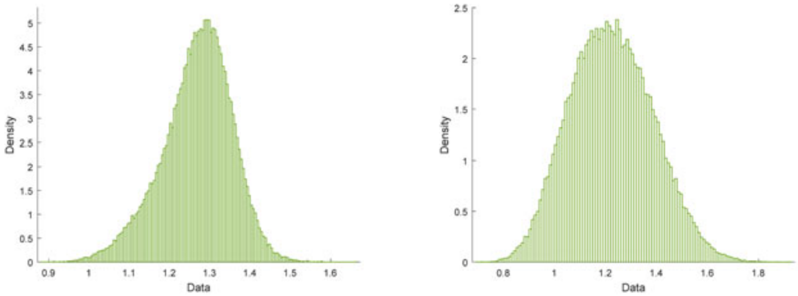
a) V_f *histogram*

b) S_f *histogram*

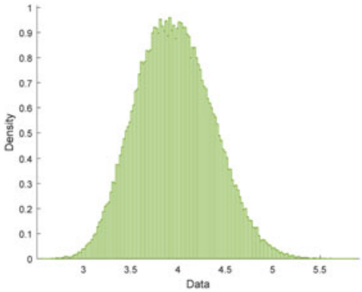


c) H_1 *histogram*

Fig. 5 The histograms of V_f, S_f, and H_1 in the MCS model



a) The histogram of SFO *b) The histogram of SFS*



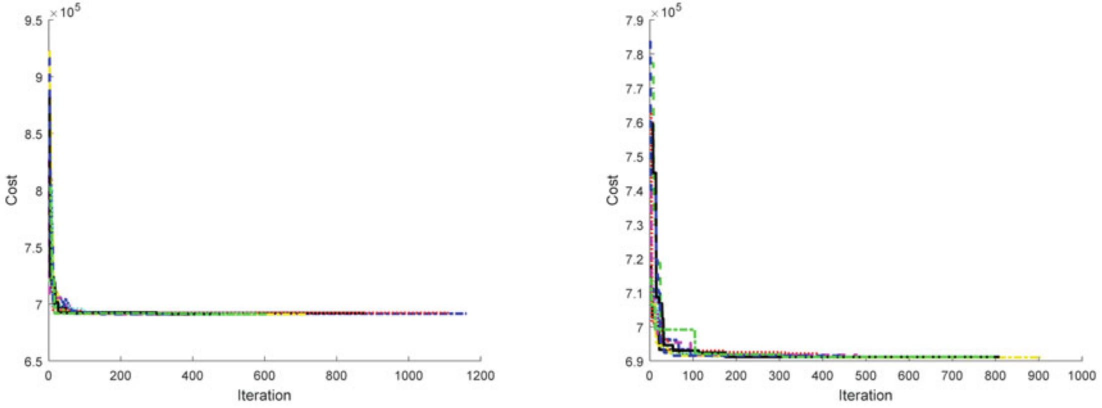
c)The histogram of SFF

Fig. 6 The histogram of SFO, SFS, and SFF

4 Discussion and Conclusion

Detention rockfill dams are one of the most popular utilized engineering structures for flood control and decreasing and delaying flood peaks due to their ease of operation. Unlike scientific literature, which considers the body of rockfill dams to be impermeable, this research employs the stage-discharge equation of non-Darcian flow in the detention rockfill dam, which incorporates leakage from the dam body in the simulation model. In this study, some equations of detention rockfill dams, such as the input hydrograph equations, the reservoir depth-volume relationship, the stage-discharge relationship of the non-Darcian flow, and the non-linear flow routing, were applied in the simulation. In a general outlook, there are three primary steps for designing the detention rockfill dams: (1) identifying the most suitable location for constructing a detention rockfill dam, (2) preliminary design, and (3) optimal design of the dam. In this work, the second and third steps of the design of a rockfill dam are examined, as well as the impact of design parameter uncertainty on the preliminary and optimal designs is studied.

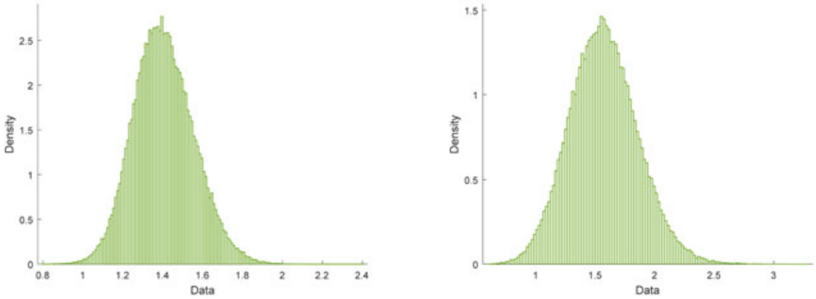
Initially, it was determined that if the dam is designed in a preliminary form, the safety factors of the dam are not provided, and the stability of the dam is at risk. In addition, it has been demonstrated that if there is uncertainty in the design parameters, it may lead to the propagation of uncertainty in the safety factor SFF to



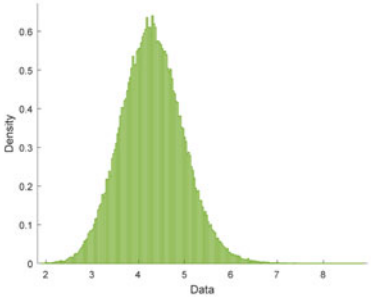
(a) the output of optimal design of detention rockfill dam without using the preliminary design output

(b) the output of optimal design of detention rockfill dam using the preliminary design output

Fig. 7 The output of optimal design of detention rockfill dam



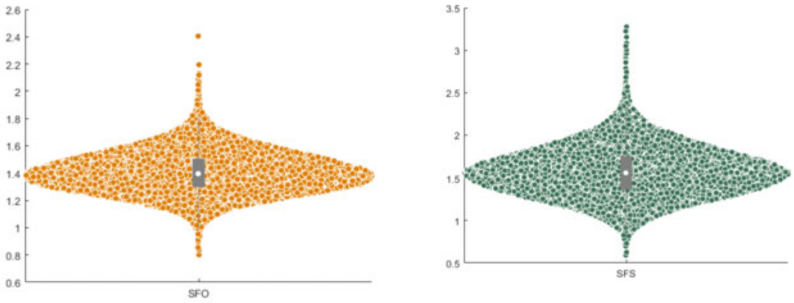
a) SFO histogram *b) SFS histogram*



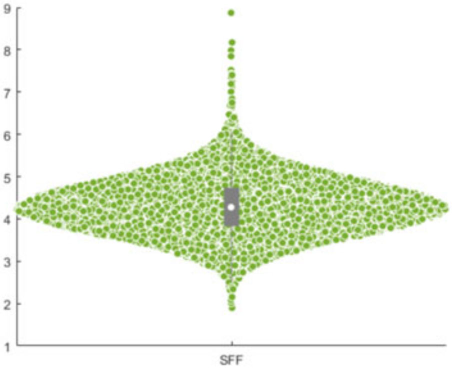
c) SFF histogram

Fig. 8 The histograms of SFO, SFS, and SFF in the MCS model

the amount of (+87.81, −32.63) percent, which is the most significant propagation of uncertainty between safety factors. The second step in designing the detention rockfill dam was to employ optimization to minimize the dam's construction cost and satisfy all safety factors. Two approaches were employed for this purpose: (1) not utilizing the dimensions of the preliminary design in the optimization algorithm and (2) using the dimensions of the preliminary design in the optimization algorithm. The outcome of this comparison revealed that if the second approach is utilized for designing the detention rockfill dam, the cost will be reduced, and the time to get the optimal solution will be cut by roughly 13%. After the optimal design was determined, the impacts of design parameter uncertainty on the hydraulic and structural performance of the detention rockfill dam were explored. It was proved that the presence of uncertainty in the design parameters leads to the propagation of uncertainty to the level of (+78.29, −21.75) percent in the X and Y, respectively. Finally, the multi-objective optimization of the detention rockfill dam was accomplished by coupling the simulator model with the Monte Carlo Simulation method and employing Self-adaptive NSGA-II. The output of this step was the Pareto Front, which displayed the construction cost of a detention rockfill dam against its reliability. The findings of this step indicate that the construction cost of a detention rockfill dam rises as the angle decreases. In addition, as the dam body's angle reduces, the dam body's length



a)The violin plot for SFO *b) The violin pot for SFS*



c)The violin plot for SFF

Fig. 9 The violin plots for SFO, SFF, and SFS

increases, and its height falls. In the case where the angle of the detention rockfill dam is 80°, it was also demonstrated that this is the most economically optimal design.

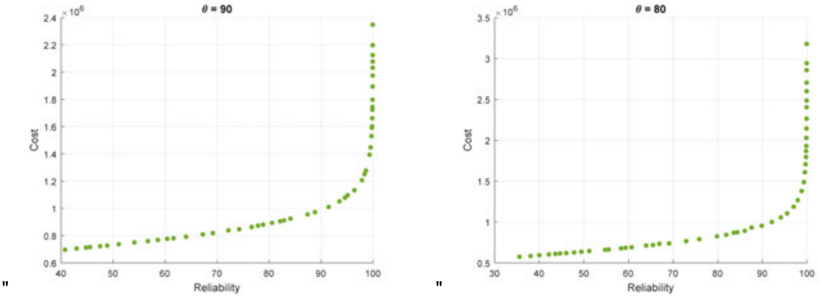


Fig. 10-1 Optimal Pareto Front for Dam with 90 degrees

Fig. 10-2 Optimal Pareto Front for Dam with 80 "degrees

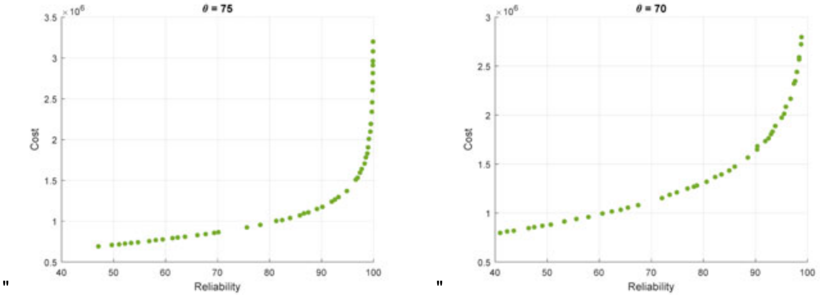


Fig. 10-3 Optimal Pareto Front for Dam with 75 "degrees

Fig. 10-4 Optimal Pareto Front for Dam with 70 "degrees

Fig. 10 Optimal Pareto Front for four scenarios with different dam angles

Table 4. Selected solutions for different scenarios of detention rockfill dam

	L(m)	H(m)	Cost	Reliability (%)	SFS	SFF	SFO
θ_1	4.7393	6.5393	836,780	72.2450	1.8598	4.7441	1.6736
θ_2	5.7140	5.7964	734,290	69.5782	1.9403	5.3572	1.6300
θ_3	7.3661	5.3126	852,410	69.4231	2.0943	6.6227	1.6905
θ_4	10.4824	4.8885	1,148,700	72.1150	2.2947	7.4947	1.7079

References

1. Chuntian C, Chau KW (2002) Three-person multiobjective conflict decision in reservoir flood control. Eur J Oper Res 142(3):625–631
2. Li XY, Chau KW, Cheng CT, Li YS (2006) A web-based flood forecasting system for Shuangpai region. Adv Eng Softw 37(3):146–158
3. Wu CL, Chau KW (2006) A flood forecasting neural network model with genetic algorithm. Int J Environ Pollut 28(3–4):261–273
4. Wang WC, Chau KW, Xu DM, Qiu L, Liu CC (2017) The annual maximum flood peak discharge forecasting using Hermite projection pursuit regression with SSO and LS method. Water Resour Manage 31(1):461–477
5. Mosavi A, Ozturk P, Chau KW (2018) Flood prediction using machine learning models: literature review. Water 10(11):1536

6. Yaseen ZM, Sulaiman SO, Deo RC, Chau KW (2019) An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. J Hydrol 569:387–408
7. Samani HM, Samani JM, Shaiannejad M (2003) Reservoir routing using steady and unsteady flow through rockfill dams. J Hydraul Eng 129(6):448–454
8. Hooshyaripor F, Tahershamsi A (2015) Effect of reservoir side slopes on dam-break flood waves. Eng Appl Comput Fluid Mech 9(1):458–468
9. Hooshyaripor F, Tahershamsi A, Razi S (2017) Dam break flood wave under different reservoir's capacities and lengths. Sādhanā 42(9):1557–1569
10. Akan AO (1990) Single-outlet detention-pond analysis and design. J Irrig Drain Eng 116(4):527–536
11. McEnroe BM (1992) Preliminary sizing of detention reservoirs to reduce peak discharges. J Hydraul Eng 118(11):1540–1549
12. Abt SR, Grigg NS (1978) An approximate method for sizing detention reservoirs 1. JAWRA J Am Water Resour Assoc 14(4):956–965
13. Wycoff RL, Singh UP (1976) Preliminary hydrologic design of small flood detention reservoirs 1. JAWRA J Am Water Resour Assoc 12(2):337–349
14. Akan AO (1989) Detention pond sizing for multiple return periods. J Hydraul Eng 115(5):650–664
15. Akan AO, Al-Muttair FF and Al-Turbak AS (1987) Design aid for detention basins. Design of hydraulic structures. Proceedings international symposium, Fort Collins, Colorado, pp 177–182
16. Horn DR (1987) Graphic estimation of peak flow reduction in reservoirs. J Hydr Engrg ASCE 113(11):1441–1450
17. Baker WR (1977) Stormwater detention basin design for small drainage areas
18. Riahi-Madvar H, Dehghani M, Akib S, Shamshirband S, Chau KW (2019) Developing a mathematical framework in preliminary designing of detention rockfill dams for flood peak reduction. Eng Appl Comput Fluid Mech 13(1):1119–1129
19. Aksoy H (2000) Use of gamma distribution in hydrological analysis. Turk J Eng Environ Sci 24(6):419–428
20. Gray DM (1961) Synthetic unit hydrographs for small watersheds. J Hydraul Div 87(4):33–54
21. Machajski J, Kostecki S (2018) Hydrological analysis of a dyke pumping station for the purpose of improving its functioning conditions. Water 10(6):737
22. Nash JE (1959) Systematic determination of unit hydrograph parameters. J Geophys Res 64(1):111–115
23. Bhunya PK, Ghosh NC, Mishra SK, Ojha CS, Berndtsson R (2005) Hybrid model for derivation of synthetic unit hydrograph. J Hydrol Eng 10(6):458–467
24. Singh PK, Mishra SK, Jain MK (2014) A review of the synthetic unit hydrograph: from the empirical UH to advanced geomorphological methods. Hydrol Sci J 59(2):239–261
25. Samani HMV, Samani JMV, Shaiannejad M (2003) Reservoir routing using steady and unsteady flow through rockfill dams. J Hydraul Eng ASCE 129(6):448–454
26. Kshirsagar DY (2014) Effect of variation of earthquake intensity on stability of gravity dam. J Indian Water Resour Soc. 34(3):1–6
27. Deepika R, Suribabu CR (2015) Optimal design of gravity dam using differential evolution algorithm. Iran Univ Sci Technol 5(3):255–266
28. Punmia BC (1992) Irrigation and water power engineering. Laxmi Publications Pvt Limited
29. Moradi Kia F, Ghafouri HR, Riyahi MM (2022) Uncertainty analysis and risk identification of the gravity dam stability using fuzzy set theory. J Hydraul Struct 7(4):76–92
30. Riyahi MM, Bahrami Chegeni I (2022) Gravity retaining wall stability risk analysis based on reliability using fuzzy set theory. J Struct Constr Eng
31. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
32. Lin YK, Yeh CT (2012) Multi-objective optimization for stochastic computer networks using NSGA-II and TOPSIS. Eur J Oper Res 218(3):735–746

33. Riyahi MM, Riahi-Madvar H (2022) Uncertainty analysis in probabilistic design of detention rockfill dams using Monte-Carlo simulation model and probabilistic frequency analysis of stability factors. Environ Sci Pollut Res. https://doi.org/10.1007/s11356-022-24037-x
34. Mays LW (2010) Water resources engineering. Wiley
35. Nowak AS, Collins KR (2012) Reliability of structures. CRC Press
36. Rashki M (2018) Hybrid control variates-based simulation method for structural reliability analysis of some problems with low failure probability. Appl Math Model 60:220–234
37. Metropolis N, Ulam S (1949) The Monte Carlo method. J Am Stat Assoc 44:335–341
38. Rajabi MM, Ataie-Ashtiani B, Janssen H (2015) Efficiency enhancement of optimized Latin hypercube sampling strategies: application to Monte Carlo uncertainty analysis and meta-modeling. Adv Water Resour 76:127–139

Machine Learning in Mechatronics and Robotics and Its Application in Face-Related Projects

Saeed Najafi Khanbebin and Vahid Mehrdad

1 Introduction: Machine Learning

Making machines think and make decisions similar to humans is a problem that creates the core of a concept, namely Machine Learning (ML). ML is a keyword that accounts for a considerable part of the research these days. Since this concept is used in solving issues in many different tasks and contexts, there is a separate field of study for ML-based research. The ML tools, all the subtests of a global concept, namely artificial intelligence (AI), work as helpful aids with various economic and scientific advantages. The problems that can be solved using machine learning include a broad spectrum. In such a way, from separating images of different fruits to fundamental problems such as astronomical data and calculations of spaceships can be done with the help of a machine learning concept.

Machine learning and machine vision are incorporated with mechatronics and robotics fields for system design and other industry challenges. Many aspects of mentioned industries take advantage of machine learning and machine vision algorithms to address the issues in different sections, such as using machine vision in fault diagnosis and designing a robot navigation system [1].

Many books and articles have been written on machine learning, and the mathematical relationships of these concepts are repeatedly and in full detail in these writings. However, the reasons for writing such a book chapter in this context can explain the motivation for writing the present text. In this chapter, after receiving general information about machine learning and its basics and concepts, the main focus is on the applications of this beneficial concept. The application-based nature

S. N. Khanbebin · V. Mehrdad (✉)
Department of Electrical and Electronics Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran
e-mail: mehrdad.v@lu.ac.ir

S. N. Khanbebin
e-mail: najafi.sa@fe.lu.ac.ir

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*, Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_9

of this writing is the reason that proves the need for this writing. In addition, the main focus is on one of the critical and beneficial applications of this field, i.e., face images in machine learning and mechatronics fields, which can explain the critical issues of this field well, along with the concepts of machine learning.

The following sections contain an explanation of face-related studies in different types of research.

2 Face Related Tasks

One of the most famous and valuable issues being completed in the machine learning world is the face-related problem. Facial images consist of information that can be used in many different applications. Many challenges in this field are addressed by using machine learning in each part and section of the face-related systems. The face-related tasks can be separated into different and identical subjects, which can be a significant issue for researchers in the ML field. These subjects are face detection, facial recognition, facial expression recognition, gender recognition, etc. In this chapter, we will describe these face-related tasks within the research done in each field.

2.1 Face Detection

A face recognition system works in a procedure that, before recognizing the person from facial images, it is needed to detect the face region from images for later uses. As video and image databases have overgrown, intelligent systems are becoming increasingly crucial for auto-interpreting and analyzing information.

As a primary means of expressing identity and feelings, the face plays an essential role in social interactions. Humans are not very good at identifying different faces compared to machines. So, Face Detection is one of the critical steps of mane face-related systems such as face recognition, facial expression recognition, head-pose estimation, human–computer interaction, etc. [2]. Video and imaging databases have grown dramatically, requiring automatic understanding and analysis of data. In terms of social communication, one crucial factor that has a key role is the changes that happen in the faces, such as feelings and attitudes. Differentiating between various faces is not that easy for humans with their vision and recognition system compared to machines. [3]. Human–Computer Interaction (HCI) is made possible by face detection, one of the most fundamental techniques. It is necessary to detect faces to perform all facial analysis algorithms, such as face alignment, face modeling, face relighting, face recognition, face verification/authentication, head pose tracking, facial expression tracking, gender identification, and so on [2]. Face detection technology begins by identifying an image area where a face or face can be found. Face detection is hampered by occlusions, illuminations, and complex backgrounds

[4]. Detecting faces is generally referred to as face localization. During the localization of a face, the object's location and size are primarily determined [3]. The solutions to these challenges have been proposed by a wide variety of algorithms. In general, existing algorithms are classified into two categories: those based on features and those based on images. Unlike feature-based approaches, which use windows or subframes for locating features (image edges, corners, and other structures), image-based approaches depend largely on image scanning [4].

2.1.1 Feature-based Approaches

Active Shape Model (ASM)

The ASM illustrates the actual substantial appearance of features at a higher level. A facial image is linked with facial features, such as the nose, mouth, etc., as soon as the system finds proximity to these features. In order to generate a mask, the coordinates of these parts are used as a map. It is possible to change the mask manually. User adjustments are possible even if the system determines the shape. An improved map can be created by training with more images. Four types of ASMs exist snakes, deformable template models (DTM), deformable part models (DPM), and point distribution models (PDM) [4].

Point Distribution Model

An essential part of The Point Distribution Model (PDM) is the recognition of vectors' shapes and the representation of those shapes. A standard statistical approach can be applied, like the multivariate object. In general, basic components are used for the construction of these models in order to learn the aforementioned constellations from training for the shape points. These are known as PDM or Point Distribution Model [3].

Deformable Templates

It is challenging to locate a facial feature boundary using generic contours since the local evidence of facial edges cannot be consolidated into a sensible global entity. It is also challenging to detect edges around some of these features because of their low contrast. As a result of the snake concept, the extraction process is more reliable since it incorporates global information about the eye. The solution to this problem is to use deformable templates. There are four types of deformation: narrow valleys, narrow edges, narrow peaks, and bright peaks. It is a great challenge for face recognition to extract salient features from faces (eyes, nose, mouth, eyebrows) in addition to the face boundary [2].

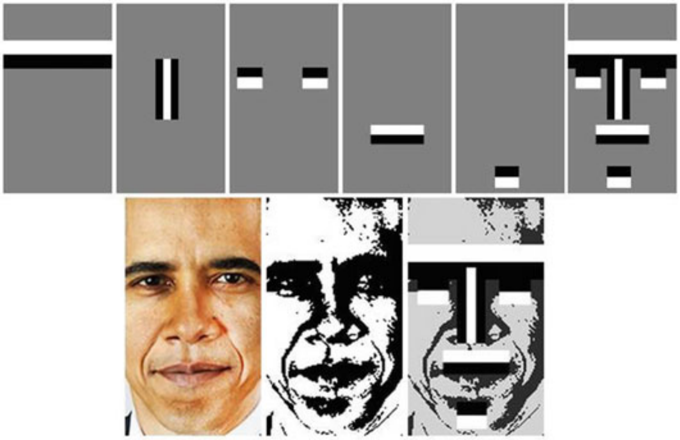


Fig. 1 Haar-like features in face detection [5]

2.1.2 Image-based Approaches

Neural Network

When an image is analyzed by a rationally attached neural network, each window is examined to determine whether it contains a face [2]. Models of neural networks are used to determine whether faces are in each window by analyzing the entire image [3]. There was an earlier suggestion for a hierarchical neural network. As part of the first stage, two parallel sub-networks are used to filter the inputs and obtain intensity values from the images. The extracted feature values create the input of the second stage and the outputs gained from subnetworks. The second stage's output obtains whether there is a face in the desired region of the image. [3]

One of the most famous and well-known face detection algorithms is the Viola-Jones [5] face detection algorithm which uses haar-like features besides the concept of the integral image for rapid detection. This algorithm has been a part of many face-detection approaches during the last decade. When the subject is face detection algorithms, it is necessary to mention a deep learning-based one in this field: the Multi-Task cascaded CNN (MTCNN) [6]. Its power and accuracy made it famous and very useful in recent projects (Fig. 1).

The face detection issue and related research are conducted based on well-studied datasets. Some face detection datasets; FDDB [7], ALFW, and WIDER FACE. In the next section, face recognition uses the result of the face detection step for recognizing an individual based on features and deep networks.

2.2 Face Recognition

The technology of face recognition is a system in which the use of faces does biological characteristics-based identity recognition. The advantages of face images as a biometric system are based on their reliability, social acceptance, and security, which overcomes the well-known biometric systems such as DNA, fingerprint, etc. [8]. Accurate individual detection with high efficiency is applied in facial recognition systems where facial features are used as principal information. Such a system works well in different environments and data types, like images, videos, and real-time [9] (Fig. 2).

Representation of the faces is performed in one or two layers using traditional approaches. The histogram of feature codes, distribution of dictionary atoms, and filtering responses can be considered these traditional strategies. The desired routine transformation and feature extraction steps are performed in a cascade of multiple layers in a CNN-based deep learning method [9]. As explained before, an exciting field such as face recognition attracts researchers' attention to solve its challenges.

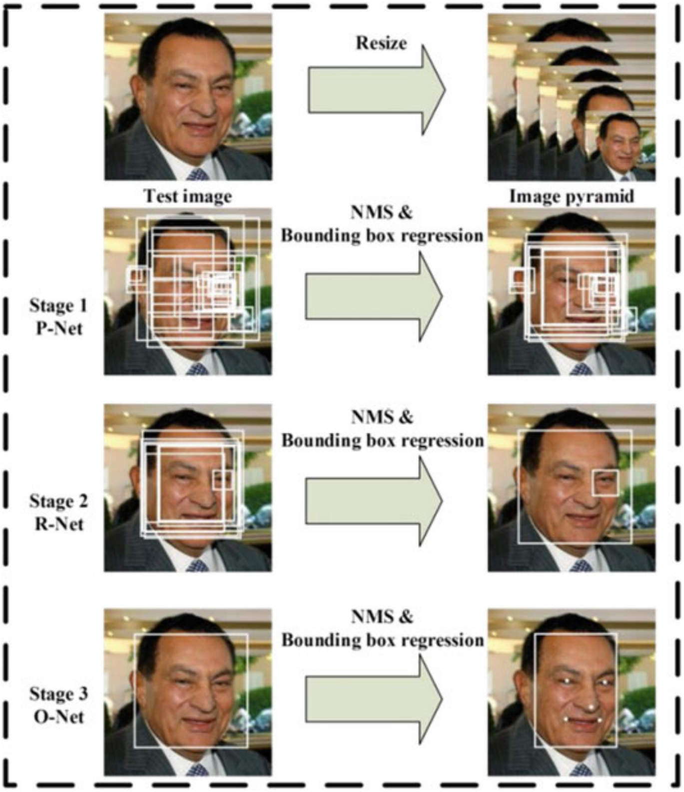
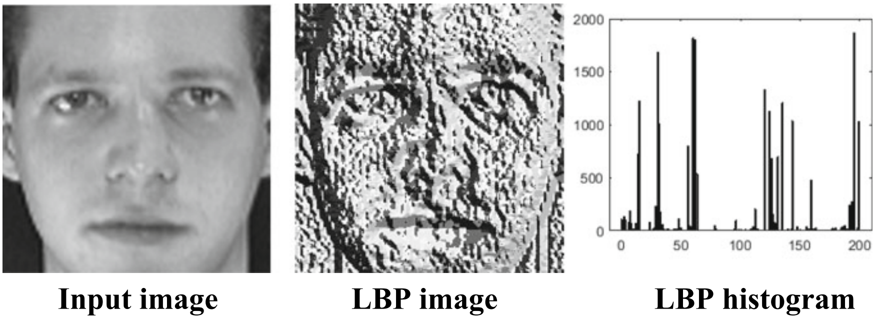


Fig. 2 Steps of MTCNN face detection algorithm [6]

Earlier efforts in this field were based on feature extraction techniques to which a vast number of researches have been applied. One of the most valuable approaches was the Local Binary Patterns (LBP) [10] which extracts the textural information of the facial images to be used in the classification part of the desired model. The LBP was also a basis for many LBP-based approaches, such as [11–14] each one tried to modify the LBP and make their model more accurate. Calculating the difference between neighboring pixels and the central one for each block of the image is the basis of LBP (Figs. 3 and 4).

As mentioned before, lots of approaches have been proposed based on the LBP method. The DR-LBP [15] used LBP operator in a different calculation to achieve a more discriminative feature vector. Its different calculation for the neighborhood pixels is shown in Fig. 5.

In the age of deep learning, the Convolutional Neural Network (CNN) had a massive effect on the direction of research in this field. This kind of network worked well with image data and attracted researchers to use it as a basis for later improvements. CNN-based approaches can be gathered in two types; one type is that proposed algorithms result in a new network with more options and more accurate results. The other type is a fusion of traditional feature extraction-based approaches with CNN-based ones.



Input image **LBP image** **LBP histogram**

Fig. 3 Example of LBP image and its histogram [15]

Fig. 4 Neighborhood pixels used in the LBP in two different values

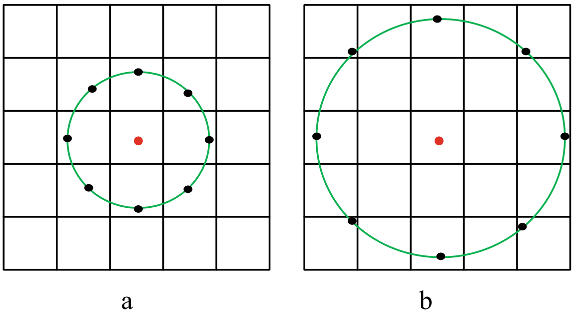
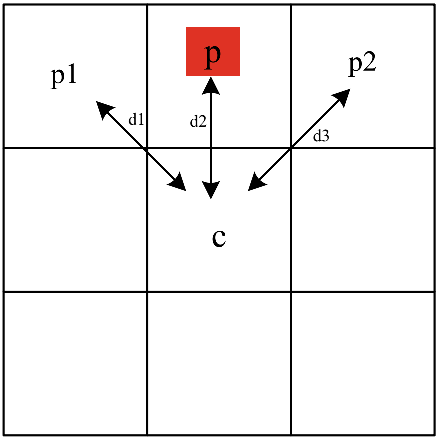


Fig. 5 The pixel difference calculation in DR-LBP [15]



Like any other fields in pattern recognition and machine learning, face recognition tasks also have special databases. The face databases consist of numerous facial images from many subjects in different situations. These data are used to train the model in deep learning-based models and extract information from traditional approaches. Datasets such as ORL [16], Yale [17], VGGFace [18], LFW [19] etc.

Numerous efforts have been made in the 2D face recognition domain. The sensitivity of the 2D face recognition approaches to pose, illumination, expression, and other challenges make it clear that 3D face recognition systems must be presented to overcome these challenges. Such limitations are addressed by reliable geometry information obtained in the 3D face recognition strategies, and this advantage attracted researchers to this subject [8]. There are particular datasets for 3D face recognition tasks such as Bosphorus, BU-3DFE [20], etc.

The spectrum of face recognition methods exists, from traditional to deep learning to 3D methods. The LBP face descriptor [21] Facenet [22], and Led3d [23] can be presented as different face recognition methods from different types of strategies.

2.3 Facial Expression Recognition (FER)

Non-verbal information plays an essential role in daily communication between people. Facial expression accounts for the most considerable portion of these non-verbal communications compared to other types, such as hand gestures, body gestures, and texts [24]. Each person can appear his feelings by changing his facial expression, which results from changes in facial muscles. The FER is one of the most interesting face-related issues. Its application is proven necessary in various fields such as security, advertisement, E-learning, etc. There are seven basic facial expressions in literature: anger, happiness, surprise, fear, sadness, disgust, and neutral.

In facial expression recognition problems, since analyzing people's emotions shows the influence of psychological factors in this type of problem, this aspect of the problem justifies many applications for facial expression recognition.

As we know, in most pattern recognition and machine learning problems, a general process is considered to solve such problems, which consists of three main parts; Receiving the input, which should be cropped images of the face area. After that, the feature extraction stage from the face images begins. Finally, the classification stage identifies different states and assigns each image to the corresponding category. In order to obtain images of the face area, the methods explained in the face detection section are used as the initial stages of a FER process. In the feature extraction section, similar to the issues discussed in face recognition, in FER, the feature extraction process has started with traditional methods based on extracting textural, spatial, frequency, geometric, etc., features from the image. These methods naturally work without the need to train the model and provide vital information about the images based solely on mathematical definitions. The feature vector obtained from this step is used for classification instead of the image itself. If more details about these methods are needed, other parts of these strategies that work with the feature extraction stage can be described. Dimension reduction is one of these parts that complement the feature extraction stage. Since the smaller the dimensions of the feature vector given to the classification part, the speed and sometimes the accuracy of the recognition model increases, so after the feature extraction stage, dimension reduction algorithms are used. Based on the existing relationships between parts of the data and some based on mathematical definitions, these algorithms keep those parts of the data that are more important and carry more critical and helpful information and remove the less important parts. This work increases the model's speed and avoids confusing the classification model by removing the extraneous and unimportant data parts.

With the emergence of new methods, especially in the period when deep learning methods entered the field, the explained process changes. Convolutional networks and their success in working with image data led to the presentation of many methods based on these networks.

CNN networks operate based on convolutional layers instead of fully connected ones. In general, a convolutional network has convolutional layers and fully connected ones simultaneously. In this way, the convolutional layers play the role of feature extraction, and the fully connected layers play the role of the classification part in the explained pipeline. Meanwhile, Pooling layers are also used to reduce the cost of calculations. Also, to improve the training and testing processes, methods are used in addition to the mentioned layers, one of which is the batch normalization layer.

Many methods have been presented based on CNNs, with reasonable accuracy. Nevertheless, as the power of algorithms has increased with the help of deep learning methods and CNN networks, due to the significant increase in computing cost in these models, more processing power and processing resources are needed to implement these algorithms.

The methods have gradually moved towards revealing the characteristics of these networks, i.e., the need for more processing power and more powerful hardware.

Since the core of CNN-based methods is to make changes in the number and order of the layers explained above, many types of research are presented to perform the convolution operation in a more optimal mode and increase the model's speed. This category of methods is called microstructure. This category includes DWS[1] convolutions, 1*1 convolutions, etc., which are the basis of famous structures such as Mobilenet [25], GoogleNet [26], etc.

Since small changes in the face must be detected to detect facial expressions correctly, the effort to increase the ability to detect these changes in CNNs is considered the basis of another part of the methods in this field.

One of these methods combines some traditional methods (as described earlier) with CNN methods. In this case, instead of raw images being given as input to the network, the feature space produced by feature extraction methods is entered into the network as input. This feature space includes images in which details and small changes are highlighted.

Another way to combine deep learning-based and manual feature methods is to input the input image to the feature extraction part and the network in parallel. Then the features extracted from manual methods are combined with features based on deep learning in one of the fully connected layers and form a unit feature vector. This feature vector enters the classification part of the model after going through several fully connected layers. In this way, the model's accuracy can be increased without the double training process, and the computational costs of the model can be adjusted. Also, in all the described CNN-based methods, the Softmax part is responsible for calculating the probabilities related to the category and performing the classification process. Different classification methods instead of Softmax can produce a model with higher accuracy. SVM [27], Random Forest, KNN, etc., can be mentioned among these classifiers.

The facial expression also has its related datasets used in the FER problems. The FER2013 [28] is one of the most famous and well-studied datasets in this field that contains challenges of real-world situations. The CK+ [29] is the other one in which its images are in more controlled conditions. Some other known datasets are affectnet [30], MMI [31], etc.

Figure 6 shows the types of methods that work based on combining traditional methods and deep learning.

In recent years, many efforts have been made to identify facial expressions and present new models. Some of these methods focus on increasing accuracy, while others increase the model's resistance to scene changes and environmental conditions. Another part of the efforts in this field is also focused on reducing computational costs to make it possible to run the models on less powerful hardware.

One of the recent exciting research papers in this field is the DeepEmotion [32] method. In this method, with a small number of convolutional layers, using a mechanism called spatial transformer network [33], the focus of the network is directed to the crucial areas of the image so that the model can obtain proper accuracy. It has

---

[1] Depth-Wise separable.

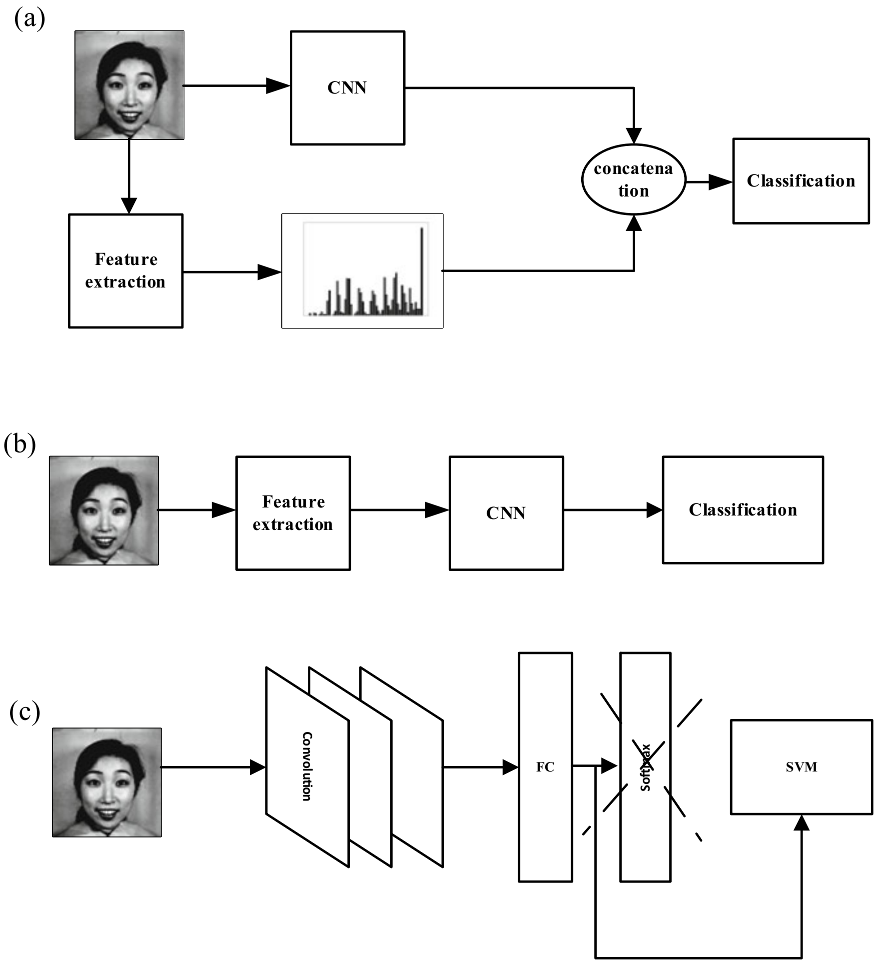


Fig. 6 Different strategies for fusing traditional and deep learning-based approaches are discussed in this section

obtained good accuracy in different data sets in the field of FER, which shows the validity of this method (Fig. 7).

Besides strategies used for creating models with lower computational costs by changing the network structures, there is another way for this purpose. Transfer learning is the other way of lowering computations for training a model to achieve enough accuracy. This strategy works in a way that the CNN is trained before with a dataset consisting of many images. The network layers carry accurate features earned by the training phase. When using these layers and their features, some layers stay unchanged, and some new layers are added, and then a new training phase is started

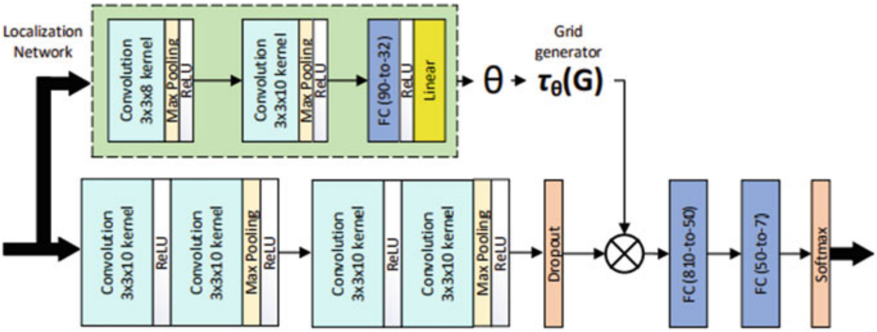


Fig. 7 The structure of the DeepEmotion network [32]

based on gathering old features and newly added layers simultaneously. Freezing a layer means that no changes are made in this layer during the training phases.

After reviewing various efforts in the field of face issues and introducing challenges and data for this field, a new need is raised. Solving all the previously mentioned problems in a single model can be an attractive goal to present in studies. Research [34] presents a model in which face recognition, facial expressions, gender recognition, etc. are done simultaneously.

Figure 8 shows the different parts of this exciting network.

After examining all aspects of face-related issues in machine learning and its application in mechatronics and robotics, we finish the discussion by examining some of these issues in the real world. The use of face-related methods can be implemented

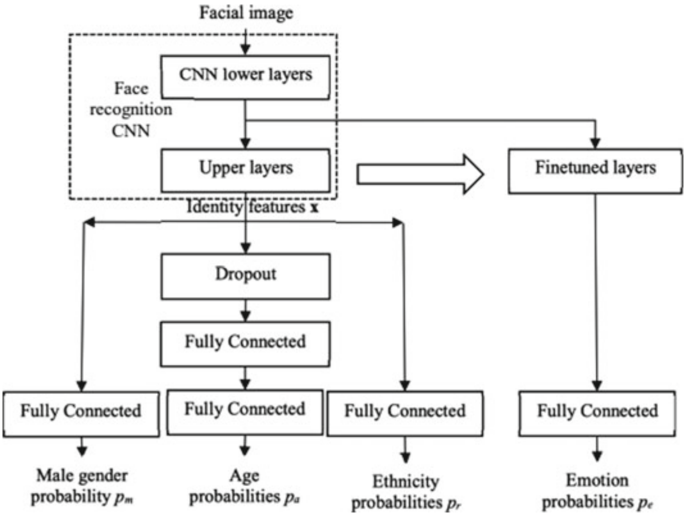


Fig. 8 Multi-task facial attributes recognition [34]

and justified in various fields. In the automotive industry, facial images can be used to increase car security, insurance trends, etc. Also, by analyzing facial emotions and details, driving safety can also be increased, and one of the ways is to detect the driver's fatigue by face-related algorithms. In these years and with the COVID19, it is important for people to wear masks, especially in closed environments. Using face-related projects in face mask detection can improve people's mask-wearing and give us more accurate control. In the mechatronics and robotics field, facial recognition algorithms have applications. One of these applications is to design a robot-controlling system using facial expressions [35] that takes advantage of the explained routines.

References

1. Choi JM, Lee SJ, Won M (2011) Self-learning navigation algorithm for vision-based mobile robots using machine learning algorithms. J Mech Sci Technol 25(1). https://doi.org/10.1007/s12206-010-1023-y
2. Kumar A, Kaur A, Kumar M (2019) Face detection techniques: a review. Artif Intell Rev 52(2). https://doi.org/10.1007/s10462-018-9650-2
3. Razzaq AN, Ghazali R, el Abbadi NK, Dosh M (2022) A comprehensive survey on face detection techniques. Webology 19(1). https://doi.org/10.14704/web/v19i1/web19044
4. Hasan MK, Ahsan MS, Mamun AA, Newaz SHS, Lee GM (2021) Human face detection techniques: a comprehensive review and future research directions. Electronics (Switzerland) 10(19). https://doi.org/10.3390/electronics10192354
5. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vis 57(2):137–154
6. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. Accessed Oct17, 2022. https://ieeexplore.ieee.org/abstract/document/7553523/
7. Jain V, Learned-Miller E (2010) FDDB: a benchmark for face detection in unconstrained settings. Accessed Oct 18, 2022. https://asset-pdf.scinapse.io/prod/182571476/182571476.pdf
8. Li M, Huang B, Tian G (2022) A comprehensive survey on 3D face recognition methods. Eng Appl Artif Intell 110. https://doi.org/10.1016/j.engappai.2022.104669
9. Dalvi J, Bafna S, Bagaria D, Virnodkar S (2022) A survey on face recognition systems. Accessed Oct 17, 2022. https://arxiv.org/abs/2201.02991
10. Ahonen T, Hadid A, Pietikäinen M (2004) Face recognition with local binary patterns. In: European conference on computer vision, pp 469–481
11. Khanbebin SN, Mehrdad V (2020) Genetic-based feature fusion in face recognition using arithmetic coded local binary patterns. IET Image Process 14(15):3742–3750
12. Khanbebin SN, Mehrdad V (2020) Local improvement approach and linear discriminant analysis-based local binary pattern for face recognition. Neural Comput Appl 1–17
13. Truong HP, Kim Y-G (2018) Enhanced line local binary patterns (EL-LBP): an efficient image representation for face recognition. In: International conference on advanced concepts for intelligent vision systems, pp 285–296
14. Silwal R, Alsadoon A, Prasad PWC, Alsadoon OH, Al-Qaraghuli A (2020) A novel deep learning system for facial feature extraction by fusing CNN and MB-LBP and using enhanced loss function. Multimed Tools Appl 79(41):31027–31047
15. Najafi Khanbebin S, Mehrdad V (2021) Local improvement approach and linear discriminant analysis-based local binary pattern for face recognition. Neural Comput Appl 33(13):7691–7707

16. AT & T Lab (1994) ORL database of faces. Cambrage University. Accessed Apr 14, 2020 from http://cam-orl.co.uk/facedatabase.html
17. Georghiades A (2002) Yale face database. Center for Computational Vision and Control at Yale University. http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html
18. Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. Accessed Oct 17, 2022. https://ora.ox.ac.uk/objects/uuid:a5f2e93f-2768-45bb-8508-74747f85cad1
19. Huang GB, Mattar M, Berg T, Learned-Miller E (2008) Labeled faces in the wild: a database forstudying face recognition in unconstrained environments
20. Yin L, Wei X, Sun Y, Wang J, Rosato MJ (2006) A 3D facial expression database for facial behavior research. In: FGR 2006: Proceedings of the 7th international conference on automatic face and gesture recognition, vol 2006. https://doi.org/10.1109/FGR.2006.6
21. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
22. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
23. Mu G, Huang D, Hu G, Sun J, Wang Y (2019) Led3d: a lightweight and efficient deep approach to recognizing low-quality 3d faces. Accessed Oct 18, 2022. http://openaccess.thecvf.com/content_CVPR_2019/html/Mu_Led3D_A_Lightweight_and_Efficient_Deep_Approach_to_Recognizing_Low-Quality_CVPR_2019_paper.html
24. Ekundayo OS, Viriri S (2021) Facial expression recognition: a review of trends and techniques. IEEE Access 9. https://doi.org/10.1109/ACCESS.2021.3113464
25. Howard AG et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
26. Szegedy C et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, vol 07–12 June 2015. https://doi.org/10.1109/CVPR.2015.7298594
27. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2(2):121–167
28. Goodfellow IJ et al (2013) Challenges in representation learning: a report on three machine learning contests. In: International conference on neural information processing, pp 117–124
29. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops, pp 94–101
30. Mollahosseini A, Hasani B, Mahoor MH (2019) AffectNet: a database for facial expression, valence, and arousal computing in the wild. IEEE Trans Affect Comput 10(1):18–31. https://doi.org/10.1109/TAFFC.2017.2740923
31. Pantic M, Valstar M, Rademaker R, Maat L (2005) Web-based database for facial expression analysis. IEEE international conference on multimedia and expo, ICME 2005, vol 2005, pp 317–321. https://doi.org/10.1109/ICME.2005.1521424
32. Minaee S, Minaei M, Abdolrashidi A (2021) Deep-emotion: facial expression recognition using attentional convolutional network. Sensors 21(9):3046
33. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K (2015) Spatial transformer networks. Adv Neural Inf Process Syst 2015:2017–2025
34. Savchenko AV (2021) Facial expression and attributes recognition based on multi-task learning of lightweight neural networks. In: SISY 2021–IEEE 19th international symposium on intelligent systems and informatics, proceedings. https://doi.org/10.1109/SISY52375.2021.9582508
35. Hasegawa N, Takahashi Y (2019) How recognition of human facial expression can be incoporated in robot control. In: Proceedings of the 2019 20th international conference on research and education in mechatronics, REM 2019. https://doi.org/10.1109/REM.2019.8744094