



# Comparative analysis of machine learning models for brain tumor detection: a meta-analysis

Tangsen Huang<sup>1</sup> · Xiangdong Yin<sup>1</sup> · Ensong Jiang<sup>1</sup>

Received: 30 July 2024 / Revised: 3 February 2025 / Accepted: 25 February 2025 / Published online: 10 September 2025  
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2025

## Abstract

Because brain tumors proliferate uncontrollably inside the cranial cavity and cause serious neurological consequences, they present severe hurdles for both diagnosis and treatment. This research delves into the interdisciplinary approach necessary for efficient treatment planning, entailing cooperation between neurosurgeons, oncologists, and additional experts. With further improvements in advanced imaging technologies and molecular diagnostics, the key to better patient outcomes is still detection and diagnosis. Even then, brain tumors represent a complex entity, the benefits from which can accrue only after incessant studies and development. In this work, subjects were categorized by type of brain tumor with the application of 2 very popular machine learning (ML) models: Decision Trees Classifier (DTC) and Support Vector Classifier (SVC). Further, optimization of these models was performed with the use of sophisticated optimization methods, namely the Differential Squirrel Search Algorithm (DSSA) and the Escaping Bird Search for Constrained Optimization (EBSO), so that these models could work out their best regarding accuracy and dependability. The best models identified were SVDS, SVEB, DTDS, and DTEB, which were used in the analysis. The model that achieved the highest accuracy is the DTEB model, in which all the data analyzed had an accuracy of 0.969. It is also the most robust model in this study because it showed the highest accuracy in both phases, training, and testing, which were 0.970 and 0.969, respectively. The application of advanced optimization techniques and ML models is indicative of the commitment to the utilization of the latest techniques for medical diagnosis and decision-making. This project tries to increase the accuracy and efficiency of diagnosing brain tumors by incorporating novel optimization approaches combined with well-established classification algorithms. Ultimately, this would translate to better patient outcomes and more personalized healthcare interventions.

**Keywords** Machine learning · Brain tumors · Artificial intelligence · Classification · Disease prediction · Decision tree classifier · Support vector classification

## 1 Introduction

Brain tumors pose significant challenges in both diagnosis and treatment due to the uncontrolled growth of brain tissues. This unchecked expansion exerts pressure within the skull, disrupting normal brain function and potentially leading to severe neurological complications (Robles et al. 2015). These tumors are broadly sorted as benign (non-cancerous) or malignant (cancerous), with malignant tumors characterized by rapid growth and the potential to metastasize to

other parts of the body (Lakshmi and Nagaraja Rao 2022; Jun and Liyuan 2022; Rehman et al. 2020). The nature of brain tumors is complex, and hence their management needs a team of neurosurgeons, oncologists, and other professionals to put up a treatment plan. Early detection and proper diagnosis are important for better outcomes for patients. Imaging technologies and molecular diagnostics have improved to identify and characterize brain tumors. With such advances, brain tumors continue to be one of the most serious and complex entities in this field, and much future research and endeavors are awaited.

Brain tumors, based on behavior and characteristics, are divided into 4 grades. Grade I tumors are usually slow-growing, have a low possibility of spreading, and can be completely surgically removed. An example is the pilocytic astrocytoma. Grade II tumors grow slowly and may

✉ Xiangdong Yin  
yinxiangdong@huse.edu.cn

<sup>1</sup> School of Information Engineering, Hunan University of Science and Engineering, Yongzhou 425199, hunan, China

infiltrate surrounding tissues; sometimes they become more aggressive over time, turning into higher-grade tumors. A tumor of oligodendroglioma is an example of this. Grade III tumors grow faster than Grade II and may infiltrate surrounding tissue, thus requiring additional treatment such as chemotherapy or radiotherapy. A Grade III tumor is an Aden squamous Astrocytoma. The most severe and aggressive cancers, grade IV tumors often utilize the blood arteries to accelerate their growth. A Grade IV tumor is glioblastoma multiforme. Knowledge of the various grades of brain tumors helps in informing the best course of action regarding therapy and management of the potential risks associated with each grade (Rehman et al. 2020; Fernando et al. 2021; Lundervold and Lundervold 2019).

To accomplish this goal, a blend of traditional and cutting-edge approaches is being used. Conventional tactics, like Leksell Gamma Knife, Gamma Knife (GK), and radioactive beams, are essential for lesion detection but are labor-intensive and dependent on human intervention (Rundo et al. 2016). Brain tumors are identified using a variety of medical imaging modalities, including computer tomography (CT), magnetic resonance imaging (MRI) scans, and Positron Emission Tomography (PET). Furthermore, chemical exchange saturation transfer (CEST), a specific MRI method, makes it possible to image chemicals at concentrations that might not be visible utilizing conventional MRI or MRS tactics. Of these modalities, magnetic resonance imaging (MRI) scans are especially useful since they are non-invasive and may produce finely detailed images of inside body structures using microwave pulses and magnetization. 3 primary MRI patterns are employed to diagnose brain tumors: T1-weighted, T2-weighted, and Fluid Attenuated Inversion Recovery (FLAIR). One major problem is accurately identifying tumor-infected regions utilizing brain MRI (Rundo et al. 2016). The utilization of artificial intelligence (AI) techniques in the medical domain has grown in significance, especially with the creation of intricate frameworks that can automatically analyze and understand vast amounts of medical data (Hamamoto et al. 2020; Shukur

and Mijwil 2023; Mijwil et al. 2022; Barragán-Montero et al. 2021; Kapoor et al. 2019; Aggarwal et al. 2022). This data includes genetics and electronic health records in addition to several medical imaging modalities including MRI (Zhao et al. 2021), CT-scan (Mijwil and Al-Zubaidi 2021), X-ray (Mijwil 2021), and ultrasound (Shen et al. 2021). Deep learning, a subset of ML, is one of these methods that has attracted a lot of interest. Deep learning algorithms are skilled at taking characteristics out of medical datasets, finding novel patterns, helping doctors diagnose patients, and supporting them in making well-informed judgments (Battineni et al. 2020; Jerez et al. 2010; Dhahri et al. 2019). Interestingly, these algorithms find their applications in the classification of medical images related to diseases like COVID-19, brain cancer, and breast cancer. Their predictive capabilities range from cardiovascular illness to diabetes and even Alzheimer's disease for early diagnosis and personalized treatment planning. The application of AI in the medical field may improve patient outcomes and further healthcare delivery.

Table 1 presents a compilation of recent investigations on brain tumor categorization, showcasing the diversity of ML techniques employed for accurate tumor identification from MRI images. The studies vary in their approach from traditional ML and optimized DL methods. Cheng et al. (Cheng et al. 2015) employed a Support Vector Machine (SVM) scheme to classify brain tumors, achieving an accuracy of 91.28%. This study demonstrated the effectiveness of SVM in distinguishing tumor types from non-tumor brain tissues in MRI scans. Zacharaki et al. (Zacharaki et al. 2009) combined SVM and k-nearest Neighbors (KNN) for brain tumor categorization, achieving an accuracy of 85%. Their approach involved the extraction of shape and intensity features from MRI images to differentiate between primary gliomas and metastatic brain tumors, as well as to grade gliomas. The study's classification method involved recursive feature elimination, making it a robust solution for classifying tumor grades. Wahlang et al. (Wahlang et al. 2022) employed hybrid CNN-DNN models integrated with SVM, reaching an accuracy of 88%. The study included gender and age information as higher-order attributes to boost categorization accuracy, with their deep learning-based architectures outperforming traditional approaches like AlexNet. Yazdan et al. (Yazdan et al. 2022) proposed a Multi-Scale CNN (MSCNN) to classify MRI images into glioma, meningioma, pituitary tumor, and non-tumor categories, achieving an accuracy of 91.2%. This model demonstrated robustness by addressing Rician noise and utilizing a fuzzy similarity-based denoising method for improved performance. Sandhiya and Raja (Sandhiya and Raja 2024) introduced a Particle Swarm Optimized Kernel Extreme Learning Machine (PSO-KELM), which achieved

**Table 1** Recent investigations for the classification of brain tumour

| Authors                             | References               | Model           | Accuracy |
|-------------------------------------|--------------------------|-----------------|----------|
| Cheng et al.                        | (Cheng et al. 2015)      | SVM             | 0.9128   |
| Zacharaki et al.                    | (Zacharaki et al. 2009)  | SVM and KNN     | 0.85     |
| Wahlang et al.                      | (Wahlang et al. 2022)    | CNN-DNN and SVM | 0.88     |
| Yazdan et al.                       | (Yazdan et al. 2022)     | Multi-Scale CNN | 0.912    |
| Sandhiya, B and Kanaga Suba Raja, S | (Sandhiya and Raja 2024) | PSO-KELM        | 0.96     |
| Remzan et al.                       | (Remzan et al. 2024)     | MLP             | 0.9771   |

a high accuracy of 96%. By incorporating radiomic features alongside DL models like Inception V3 and DenseNet201, they optimized the classification process for gliomas, meningiomas, and pituitary tumors. Remzan et al. (Remzan et al. 2024) adopted a Multi-Layer Perceptron (MLP) model, incorporating RadImageNet for pre-trained feature extraction, achieving a top accuracy of 97.71%. This study demonstrated the benefits of ensemble learning methods, particularly the stacking ensemble strategy, which combined multiple base learners like SVM, k-NN, and Extra Trees with a Logistic Regression meta-learner to enhance model performance.

This investigation explores a modern tactic to brain tumor detection by leveraging ML algorithms and advanced optimization tactics to tackle gaps in existing literature. Unlike previous studies that primarily rely on traditional ML classifiers without substantial optimization, this research focuses on enhancing prediction accuracy and reliability through the integration of two sophisticated optimization methods: Differential Squirrel Search Algorithm (DSSA) and Escaping Bird Search Optimization (EBSO). These optimizers were applied to two popular ML classifiers, the Decision Tree Classifier (DTC) and the Support Vector Classifier (SVC), to develop optimized models: DTDS (DTC optimized with DSSA), DTEB (DTC optimized with EBSO), SVDS (SVC optimized with DSSA), and SVEB (SVC optimized with EBSO). This study's novelty lies in its dual focus on optimization and interpretability. Decision Trees provide transparency in decision-making, while SVCs excel in handling high-dimensional and complex data. By optimizing these models, the study introduces a new paradigm for accurately identifying patterns in brain tumor data, surpassing the performance of conventional methods. The research stands apart by addressing the critical challenge of parameter tuning, a key factor often overlooked in previous studies. This innovative approach paves the way for improved diagnostic precision, enabling more personalized healthcare interventions and better outcomes for patients.

## 2 Study methods

### 2.1 Data collection

**Data** The dataset used in this research contains 3,762 observations, from (<https://www.kaggle.com/datasets/jakeshboha/ju/brain-tumor>), which have been divided into two phases: 70% for the training phase and 30% for the testing phase. This brain tumor feature dataset is a well-structured set of quantitative descriptors that were elaborated to capture the radiographic properties of brain tumor images. First-order statistical measures and second-order texture descriptors

are two major categories into which the features may be divided. These include the mean, variance, standard deviation, skewness, and kurtosis of the intensity distribution of pixels in images. It gives information concerning central tendency, dispersion, and shape of the intensity distribution that helps in the characterization of tumor regions in the images. Second-order features represent the texture and spatial patterns inside the images; they capture fine, intricate relationships that occur between neighboring pixels. Contrast, energy, angular second moment, entropy, homogeneity, dissimilarity, correlation, and coarseness, constitute some of these metrics. It gives an appropriate and more profound level in distinguishing the presence and absence of tumors and non-tumor regions that describe structural complexity with spatial heterogeneity. The data column "Class" represents the "0" indicating a brain without any tumor and the value "1" when the person had a tumor. Each entry of data corresponds to an image ID, allowing traceability and cross-referencing with the original images.

#### 2.1.1 Results visualization

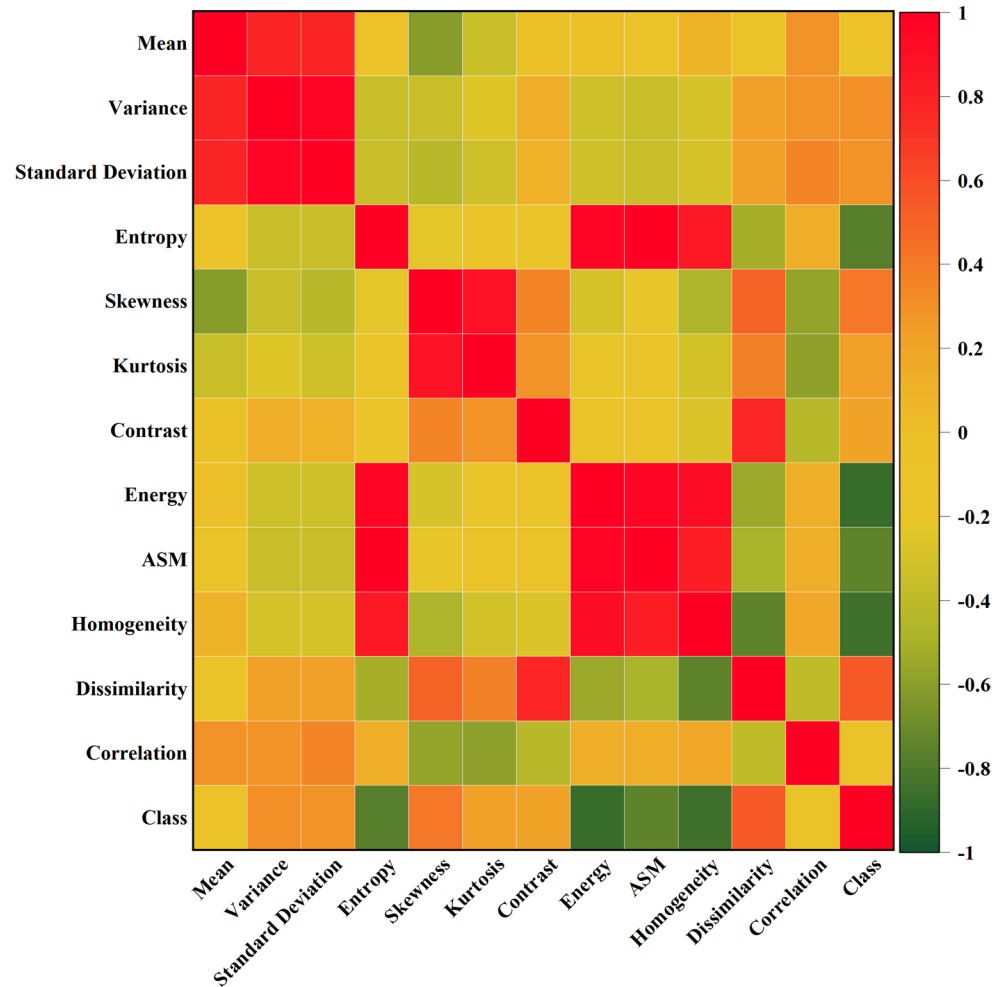
Color coding is used in Fig. 1 to show the impact of each parameter on the categorization results of people with and without brain tumors. It is evident that in this model, the hues green, red, and yellow represent  $-1$ ,  $+1$ , and neutral (zero) impacts, respectively. Results analysis indicates that parameters with orange and red colors positively affect classification. Interestingly, dissimilarity is the metric with the biggest positive impact, followed somewhat by skewness, variance, standard deviation, kurtosis, and contrast. Conversely, metrics like correlation and mean show no influence. Moreover, the entropy and ASM parameters show negative impacts ranging from  $-0.6$  to  $-0.8$ . Energy and uniformity show the strongest detrimental consequences among them. This graphic illustrates each parameter's different levels of impact on the classification results, offering important information about the characteristics used in the analysis's discriminative power.

### 2.2 Decision tree and classification (DTC)

With leaf nodes signifying specific class labels, branches showing test results, and inside nodes reflecting attribute testing, a DT is a hierarchical structure that resembles a flowchart. In a DT prediction, the attribute values of a data point must be evaluated by moving from the root to a leaf node, which produces the predicted class label (Sharma and Kumar 2016).

The following 3 factors are the DT's primary operating parameters:

**Fig. 1** The plot illustrates the correlation between the input and output



**Split information** The initial dataset (represented by D) included training samples and corresponding class labels.

**List of attributes** A collection of attributes that describe the characteristics of the information that makes up this parameter.

**Method of attribute selection** This option explains how to divide the DT into branches or divisions by choosing the best attribute. Usually, one must use an attribute selection metric, like the Gini index or information gain.

Here is a summary of the algorithm's functioning:

- Create a node first, and name it A.
- When the class in the current dataset is the same for all samples, Node A turns into a leaf node with that same class.
- Node A should be classified as a leaf node and given the most common class among the data samples if the attribute list is empty.
- Select a feature for data splitting that will maximize the quality of the subsets that are produced.
- Assign the selected property, node A, and make it the deciding factor.
- Take the chosen attribute off of the attribute list if it is discrete.
- Create data subsets based on the selected attribute's findings.
- Attach a leaf node to node A that is labeled with the majority class of the original dataset if any subsets are empty.

When working with non-empty subsets, create new nodes continually; this recursive process should be carried out until all data partitions are processed. The final DT structure is the outcome of all of this work. Figure 2 presents the design of DTC.

### 2.3 Support vector classification (SVC)

SVC is an ML technique (Vapnik 1998) that lowers risk by altering independent variables non-linearly using SVMs.

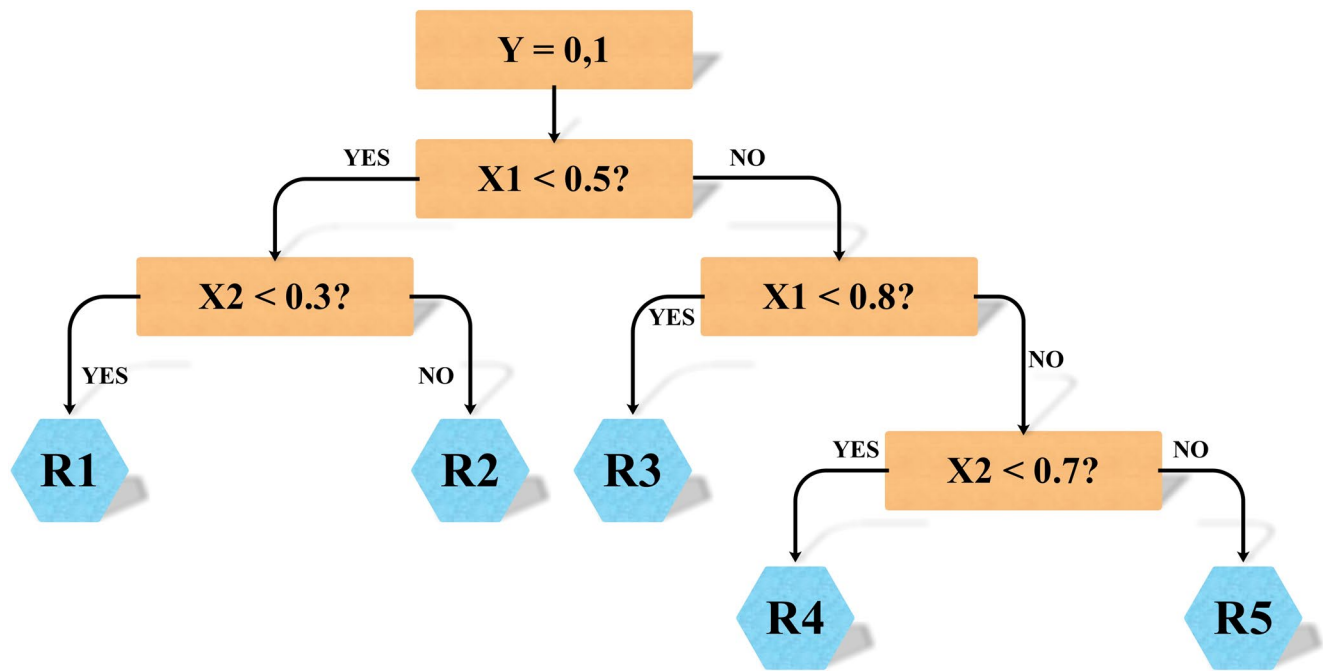


**Fig. 2** The structure of the DTC model

Minimizing classification errors and building the optimal hyperplane to split groups are the goals, along with maximizing margins, hyperplane distances, and training samples closest to each class (Maldonado et al. 2014). The fundamental model is then presented using equations (Chang and Lin 2011).

$$\min_{p, \epsilon} \frac{\|A_v\|^2}{2} + C_{svc} \sum_{i=1}^r \Delta_i \quad (1)$$

$$y_i(A^t \cdot \partial(x_i) + \vartheta) \geq 1 - \Delta_i \quad i = 1, \dots, r \quad (2)$$

$$\Delta_i \geq 0 \quad i = 1, \dots, r \quad (3)$$

Each observation is represented by an explanatory  $x_i$ , which is translated into a higher-dimensional space using a non-linear transformation function  $\partial(x_i)$ . The feature space represents this new domain, while  $P_v$  stands for the weight vector related to the independent variables. The bias term  $\epsilon$  and the regularization parameter  $C_{svc}$  are also introduced alongside. Slack variables show the distance between an observation ( $i$ ) and the class boundary or edge.  $\Delta_i$  is the symbol used to represent them.

Class labels applied to each sample ultimately indicate the classes that correlate with it, and Eq. (4) starts an optimization procedure to find the optimal hyperplane in this high-dimensional space that maximizes the margin. This is done in an attempt to reduce both the number of cases that are wrongly classified and the norm of the weight vector.

$$\mathcal{H}(x_i) = A_v^t \phi(x_i) + \epsilon \quad (4)$$

While the original model's scale was dependent on the complexity of the problem, the dual version of the model is based on sample counts. Therefore, as shown by Eq. (5) to (7), handling the dual model is preferable when dimensionality is large enough.

$$\max_h \sum_{i=1}^r h_i - \frac{1}{2} \sum_{i=1}^r h_i h_j y_i y_j \mathcal{K}(x_i, x_j) \quad (5)$$

$$\sum_{i=1}^r h_i y_i = 0 \quad (6)$$

$$0 \leq h_i \leq C_{svc} \quad i = 1, \dots, r \quad (7)$$

Pairs of data points are mapped to feature space by the function  $\mathcal{K}(x_i, x_j)$ . Many Kernel functions are available. However, they need to be symmetric, polynomial, radial basis, positive semi-definite, and sigmoidal. Previous studies show that the radial basis Kernel function, often known as Eq. (8), is helpful, particularly for classification applications (Aydogdu and Firat 2015). As a result, where  $\gamma$  serves as a hyperparameter that indicates the inverse range of effect of data points that have support vector labels (Géron 2022).

$$\mathcal{K}(x_i, x_j) = \partial(x_i)^R \partial(x_j) = \exp(-\gamma \|x_j - x_i\|^2) \quad (8)$$

Forecasts for more samples and estimates for the weights and bias factor are given in Eq. (9), which can be utilized to solve the model.

$$SVC: \quad y_i = \begin{cases} -1 & \text{if } A_v^t \partial (x_i) + b \leq 0 \\ 1 & \text{if } A_v^t \partial (x_i) + b > 0 \end{cases} \quad (9)$$

## 2.4 Escaping bird search for constrained optimization (EBSO)

Using artificial bird pairs as search agents to explore the design space, EBS is a population-based method. In each randomly chosen couple from the population, the less fit bird becomes the prey (escaping bird) while the more fit bird becomes the predator (Hedenström and Rosén 2001; Lentink et al. 2007). The position of a synthetic bird in the exploration area is matched by a design vector that changes as the bird travels. The bird's agility is determined by its body area, speed, and wing beat rate (Howland 1974). Equation (10) defines the maneuvering power of the  $i^{th}$  artificial bird in the current numerical simulation,  $MP_i$ .

$$MP_i = r_i \mathcal{V}_i^\omega \quad (10)$$

The moving power of the  $i^{th}$  bird is calculated by taking the velocity vector,  $\mathcal{V}_i$ , and subtracting it from the current position of the bird. The method for calculating the square root of the total squared deviation of  $\mathcal{V}_i$  is  $\|\mathcal{V}_i\| = \sqrt{\sum_j \mathcal{V}_{i,j}^2}$ , which provides the norm of  $\mathcal{V}_i$ . The parameter  $\omega$  is incorporated, randomly fluctuating between 0 and 2, to replicate the effect of wing beat-rate variability on maneuver power. The effect of the bird's body area (cost) in modeling is indicated by the normalized relation body factor  $r_i$ .

$$r_i = \frac{\mathcal{G}_{max} - \mathcal{G}_i}{\mathcal{G}_{max} - \mathcal{G}_{min} + \epsilon} \quad (11)$$

The aforementioned expression can be utilized to calculate the  $i^{th}$  agent's cost, denoted as  $\mathcal{G}_i$ . Here,  $\mathcal{G}_{max}$  and  $\mathcal{G}_{min}$  represent the current population's maximum and lowest costs, respectively. The tiny constant  $\epsilon$  is included to avoid division by zero. Additionally, the mobility of the Attacking Bird ( $\alpha$ ) and Escaping Bird ( $\epsilon$ ), acting as the predator and prey, respectively, is considered while determining the Escaping Rate (ER). The ER is found using Eq. (12).

$$ER = \frac{MP_\epsilon}{MP_\alpha + MP_\epsilon} \quad (12)$$

The maneuvering ability of the attacking bird and the escaping bird is indicated by the labels  $MP_\alpha$  and  $MP_\epsilon$ , respectively. When a predatory bird does a vertical maneuver, it selects a direction that is counter to the predator. To mimic this, artificial flight of the following kind was employed:

$$X_\epsilon = X_\epsilon + s \times ER \times (Op(X_\alpha) - X_\epsilon) \quad (13)$$

$X_\alpha$  and  $X_\epsilon$ , respectively, indicate the places of the attacking bird and the escaping bird. From 0 to 1, a uniformly distributed variable,  $s$ , is randomly generated. The ER is found using Eq. (12). Use the  $Op(\cdot)$  function, to find a given vector's opposite, which is defined as the vector's opponent in the design space (Shahrouzi and Salehi 2020).

$$Op(X_\alpha) = X_L + X_U - X_\alpha \quad (14)$$

The artificial prey will choose a lateral movement, which involves veering from its present route if the  $ER$  is modest. A randomly generated position vector serves as a switch for numerical replication:

$$X_\epsilon = X_L + \epsilon \tau (X_U - X_L) \quad (15)$$

The vectors  $X_L$  and  $X_U$  represent the lower and upper bounds of the design variables, respectively. The  $\tau$  symbol indicates the product of elements, and vector  $\epsilon$  contains random values reaching from 0 to 1. The marauder's main objective while in flight is to find its prey.

The present simulation has added another direction to this trip, though: it is in the direction of  $X_{Gbest}$ , which is the global average of all prey encounters to date. As a result, the marauder's movement is shown as:

$$X_\alpha^{Can} = X_\alpha + \epsilon_1 \times CR \times (X_\epsilon - X_\alpha) + \epsilon_2 \times (X_{Gbest} - X_\epsilon) \quad (16)$$

Separate random values,  $\epsilon_1$  and  $\epsilon_2$ , are generated between 0 and 1. The following is the representation of the Capturing Rate (CR):

$$CR = 1 - ER \quad (17)$$

A reciprocal relationship exists between the catching rate (CR) and escape rate (ER); when one rises, the other falls. Predator and prey impacts are considered in Eq. (12), particularly in cases of aerial problems. Using vector-sum position updating (Eq. (16)), the technique, EBS, departs from Particle Swarm Optimization (PSO). In contrast to PSO, which ignores inertial and cognitive factors, EBS only considers modifying the coefficient,  $CR$ . Additionally, the previous term in Eq. (16) does not scale proportionately. The algorithm proceeds by simulating synthetic predator-prey behaviors, as described below.

**First step** ascertain where each  $i^{th}$  bird is located inside the lower and higher bounds of the design variables. To generate a colony of  $M$  birds, randomly assign a position to each artificial bird. Applying the subsequent Eq. (18)

$$X_i = X_L + \epsilon \tau (X_U - X_L) \quad (18)$$

The calculation in Eq. (15) has determined the right-side limits. After that, each bird's cost function will be calculated, and its velocity will be set to 0. The third phase will identify the initial population of predators and prey birds.

**Second step** After normalizing the cost values obtained from Eq. (11), use Eq. (10) to get the maneuver power value for each bird.

**Third step** Keep running the main loop until the specified closure condition is satisfied.

- - For each  $i$  between 1 and  $M$ , a bird is randomly selected at position  $i$ th from the population, and it is paired with another bird..
- - Decide which bird is ideal for the AB and which bird is the EB.

Each pair (AB, EB) should do the following actions:

- - Determine the  $MP$ ,  $ER$ , and  $CR$  for the present pair using Eqs. (10) to (12) and (17).
- - Apply Eq. (16) to generate a possible solution for AB.
- - Determine the cost of the potential remedy,  $X_a^{Can}$ .
- - if  $X_a^{Candidate}$  is less expensive than  $X_a$ , Replace  $X_a$  with  $X_a^{Can}$ .
- - When the entire number of fitness function evaluations ( $NFE$ ) is reached, the end of the main loop, also called  $NFE_{max}$ , reaches a given maximum value.

Next, use either level –turning (Eq. (15)) or vertical maneuvering (Eq. (13)) to produce a workable EB solution. Alternate between these evasion techniques based on the given relationship:

$$X_e^{Can} = \begin{cases} X_L + r\tau (X_U - X_L) & \text{if } ER < 1/M \\ X_e + r \times ER \times (Opp(X_a) - X_e) & \text{otherwise} \end{cases} \quad (19)$$

Take the following steps on behalf of EB:

- - Examine the prospective solution's price,  $X_e^{Can}$ .
- - If  $X_e^{Can}$  is less expensive than  $X_e$ , replace  $X_e$  with  $X_e^{Can}$  (greedy collection).
- - Terminate the main loop directly when NFE approaches  $NFE_{max}$ .
- - Update  $X_{Gbest}$ .

Return to the third Step if  $NFE < NFE_{max}$  and  $i = M$  are true. If not, proceed to the 4th Step and close the loop.

**4th step** After terminating the main loop, declare the changed  $X_{Gbest}$  to be the best option. Just  $M$  and  $NFE_{max}$  are required for EBS to function.

## 2.5 Differential squirrel search algorithm (DSSA)

DSSA, a novel optimization strategy combining differential evolution (DE) and the squirrel search algorithm (SSA), was first presented by Manoj et al. (Jena et al. 2021). DSSA enhances exploration capabilities by fusing DE-inspired crossover operations with squirrel behaviors, which update their places by tracking others in trees. In their work, the various foraging methods DSSA uses are carefully and mathematically characterized.

### 2.5.1 Setting up the position and assessing fitness

Squirrels are first arranged at random across the exploration area. Their fitness levels are assessed based on where they are in the fitness function, which indicates the caliber of the food they locate. Sorting is used to determine which squirrel is the fittest in the hickory tree  $PS_{ht}$  and which 3 are in the acorn tree  $PS_{at}$  (1:3). The remaining squirrels,  $PS_{nt}^{P1}$ , are positioned normally inside the habitat and have not yet discovered suitable food.

### 2.5.2 Update on position

When there are no predators around, squirrels in the acorn tree use  $PS_{ht}$  to guide them to the best source. In the meantime, squirrels on the regular tree imitate the acorn or hickory tree's movements. When there is a chance of encountering predators, this stochastic foraging behavior occurs. Squirrel placements are updated using mathematical models, especially for those in the acorn tree, as follows:

$$PS_{at}^m = \begin{cases} PS_{at}^o + d_g \cdot G_c (PS_{ht}^o - PS_{at}^o - P_{avg}), & r_1 \geq P_{dp} \\ r_{position}, & \text{otherwise} \end{cases} \quad (20)$$

The mean of all squirrel's locations in the available population is denoted by  $P_{avg}$ , and  $r_{position}$  denoted as random position.

By including the crossover mechanism into the DE algorithm, the possibility of a local minimum is decreased, hence broadening the scope of the squirrel population under consideration. The current position and the matching new location were subjected to the crossover operation, which was derived from Eq. (21).

$$PS_{at,i,j}^{scr} = \begin{cases} PS_{at,i,j}^n, & \text{if } (r_j \leq Cr) \text{ or } j = j_s, \\ PS_{at,i,j}^o, & \text{if } (r_j > Cr) \text{ or } j \neq j_s, \end{cases} \quad j = 1, 2, 3, \dots, D \quad (21)$$

$i = 1, 2, 3, \dots, NP$

Where are the locations of the new and old squirrels,  $PS_{at,i,j}^n$  and  $PS_{at,i,j}^o$  in acorn or regular trees.

Here are the squirrel's locations following the crossover procedure:  $PS_{at,i,j}^{cr}$ .

The population is denoted by  $NP$ .

The crossover rate, or  $Cr$ , is set to 0.5.

$D$  is the problem's dimension.

A randomly produced index is denoted by  $j_s \in [1, D]$ , and the  $j$ -th evaluation of uniformly generated random numbers in the distinct range is denoted by  $r_j \in [1, D]$ .

Some of the squirrels in regular trees go to new locations by trailing the squirrels in acorn trees, as per Eq. (22).

$$PS_{nt}^n = \begin{cases} PS_{nt}^o + d_g \cdot G_c (PS_{nt}^o - PS_{nt}^n), & r_2 \geq P_{dp} \\ r \text{ position}, & \text{otherwise} \end{cases} \quad (22)$$

$r_2$  is consistent with an unchanging distribution over the range  $[0, 1]$ .

The modified positions of the squirrels that remain in conventional trees are as follows, and they correspond with the current ideal location:

$$PS_{nt}^n = \begin{cases} PS_{nt}^o + d_g \cdot G_c (PS_{nt}^o - PS_{nt}^n), & r_3 \geq P_{dp} \\ r \text{ position}, & \text{otherwise} \end{cases} \quad (23)$$

The following is how a crossover operation is applied to squirrels living in regular trees:

$$PS_{nt,i,j}^{cr} = \begin{cases} PS_{nt,i,j}^n, & \text{if } (r_j \leq Cr) \text{ or } j = j_s \\ PS_{nt,i,j}^o, & \text{if } (r_j > Cr) \text{ or } j \neq j_s \end{cases}, \quad j = 1, 2, 3, \dots, D \quad (24)$$

Allowing the squirrel in the hickory tree to adjust its location in relation to the pace of convergence may be accelerated by using Eq. (25) to calculate the average of the squirrels' locations within the tree.

$$PS_{ht}^n = PS_{ht}^o + d_g \cdot G_c (PS_{ht}^o - PS_{at}^{avg}) \quad (25)$$

The average of all the squirrel's places in the acorn trees is denoted by  $PS_{at}^{avg}$ .

The best open positions and their hybrid positions are then contrasted with the historical positions in the decision handle for occupying the population of the upcoming generation. Figure 3 presents the flowchart of the DSSA models.

## 2.6 Performance evaluation metrics

Performance evaluator metrics, like accuracy ( $\mathbb{A}$ ), precision ( $\mathbb{P}$ ), F1-score ( $\mathbb{F}$ ), and recall ( $\mathbb{R}$ ), are frequently used to

evaluate the efficacy of a model or algorithm. Accuracy is the percentage of correctly predicted observations relative to all predictions made; precision is the ratio of true positive predictions to all positive predictions; F1-score is the harmonic mean of precision and recall, offering a balance between the 2 metrics; and recall is the percentage of actual positives that the model correctly identified. These metrics are key to assessing the effectiveness of ML models and establishing their suitability for particular tasks. The following are the formulas for these metrics:

$$\mathbb{A} = \frac{\mathcal{T}p + \mathcal{T}n}{\mathcal{T}p + \mathcal{T}n + \mathcal{F}p + \mathcal{F}n} \quad (26)$$

$$\mathbb{P} = \frac{\mathcal{T}p}{\mathcal{T}p + \mathcal{F}p} \quad (27)$$

$$\mathbb{R} = \mathcal{T}pR = \frac{\mathcal{T}p}{P} = \frac{\mathcal{T}p}{\mathcal{T}p + \mathcal{F}n} \quad (28)$$

$$\mathbb{F} = \frac{2 \times \mathbb{R} \times \mathbb{P}}{\mathbb{R} + \mathbb{P}} \quad (29)$$

True Positives ( $\mathcal{T}p$ ) are the cases in which the model properly predicted a positive outcome in binary classification. Accurate forecasts of negative outcomes are denoted by True Negatives ( $\mathcal{T}n$ ). A model produces False Positives ( $\mathcal{F}p$ ) when it predicts a positive outcome wrongly and False Negatives ( $\mathcal{F}n$ ) when it fails to forecast a positive outcome.

## 3 Result

### 3.1 Convergence outcomes and hyperparameter

Figure 4 shows the iterative evolution of 4 optimal models (SVDS, SVEB, DTDS, and DTEB), each observed after more than 200 iterations. All models show poor accuracy levels at the beginning of the iterations. However, as the rounds go on, a pattern becomes apparent: starting about iteration 140 and continuing until iteration 200, every model converges to a constant accuracy. The model that performs the best among them is the one with the maximum accuracy. The DTEB model performs exceptionally well in this specific phase of convergence, with an accuracy score of 0.9691, which is very near to the ideal value of 1. This outstanding result suggests that, during the convergence process, the DTEB model has essentially reached its maximum accuracy. Closely trailing, with an accuracy of 0.9564, is the DTDS model, which takes second place. As impressive as it is, this accuracy is only 1.2% less than the top DTEB model. Although the DTDS model is not as good as the

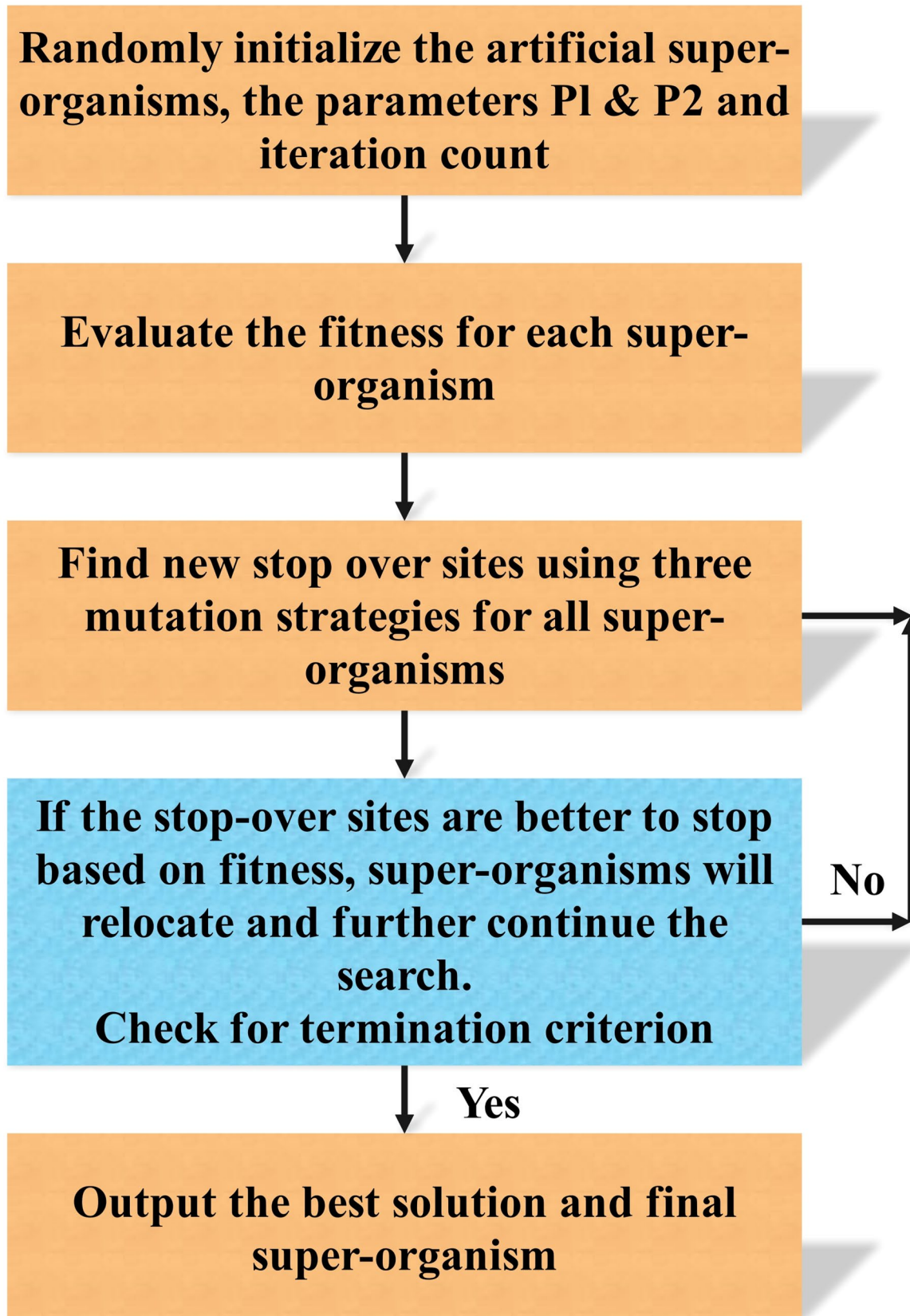


Fig. 3 The flowchart of the DSS Algorithm

**Fig. 4** 3D The convergence curve for the presented models

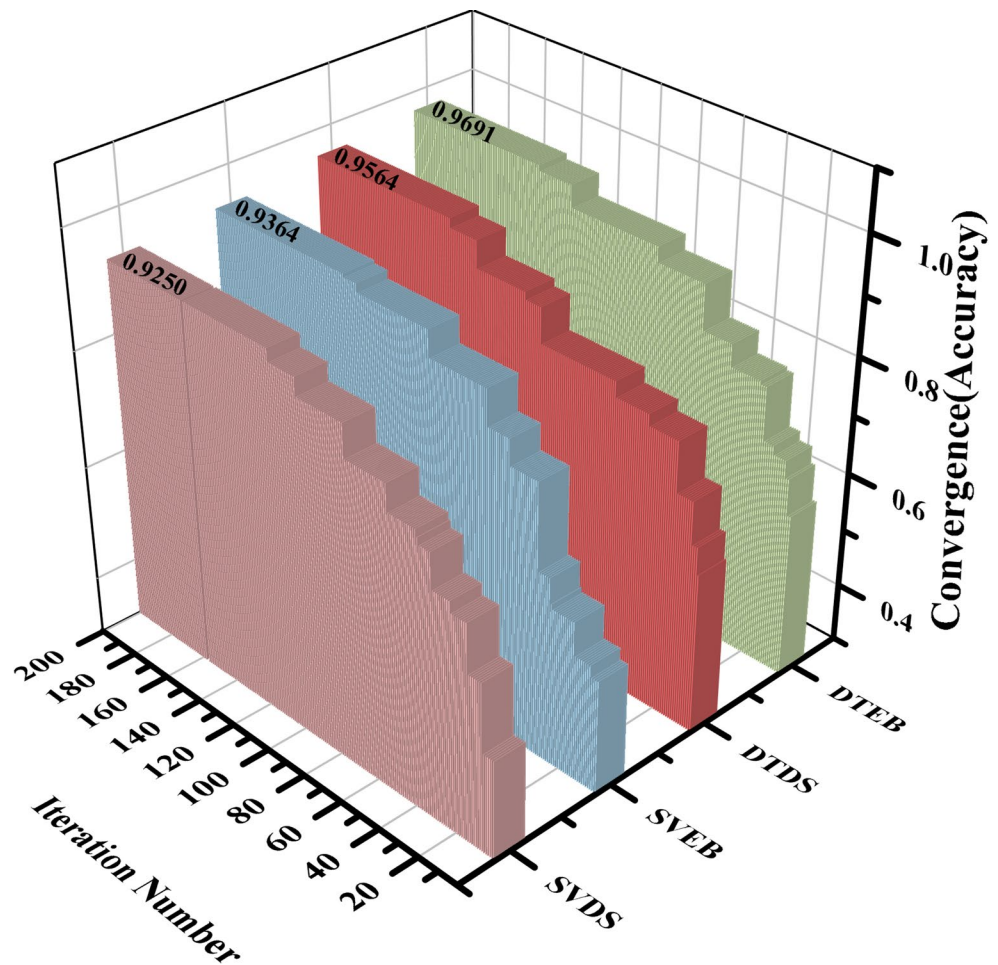


**Table 2** The result of hyperparameter

| Models | <i>Hyperparameter</i> |                   |                  |                |
|--------|-----------------------|-------------------|------------------|----------------|
|        | max_depth             | min_samples_split | min_samples_leaf | max_leaf_nodes |
| DTC    | -                     | 2                 | 1                | -              |
| DTDS   | 19                    | 0.019             | 0.0095           | 190            |
| DTEB   | 6                     | 0.00447           | 0.001091         | 168            |
| Models | <i>Hyperparameter</i> |                   |                  |                |
|        | C                     | gamma             |                  |                |
| SVC    | 1                     | scale             |                  |                |
| SVDS   | 2151                  | 0.49328           |                  |                |
| SVEB   | 8617                  | 0.13872           |                  |                |

top-ranked DTEB model, it still performs well despite this slight discrepancy.

Table 2 Summarizes the hyperparameter tuning results for the machine learning models used in the study, leveraging the random search method for optimization. The DTC and its optimized versions, DTDS and DTEB, are evaluated based on key hyperparameters: max\_depth, min\_samples\_split, min\_samples\_leaf, and max\_leaf\_nodes. The default DTC model uses minimal configuration, with values of 2 for min\_samples\_split and 1 for min\_samples\_leaf, leaving max\_depth and max\_leaf\_nodes unspecified. Using random

search, DTDS achieved optimal performance with a max\_depth of 19, min\_samples\_split of 0.019, min\_samples\_leaf of 0.0095, and max\_leaf\_nodes of 190. Similarly, DTEB attained its best configuration with a max\_depth of 6, min\_samples\_split of 0.00447, min\_samples\_leaf of 0.001091, and max\_leaf\_nodes of 168. For SVC models, the default SVC used standard parameters ( $C=1$  and  $\text{gamma}=\text{scale}$ ). Random search optimization significantly improved performance. SVDS achieved optimal results with  $C=2151$  and  $\text{gamma}=0.49328$ , while SVEB further refined these parameters to  $C=8617$  and  $\text{gamma}=0.13872$ . These results

emphasize the value of random search as a robust hyperparameter tuning method, enabling the efficient identification of high-performing configurations and improving the overall accuracy and reliability of the models.

The models in this study were developed using a system equipped with an Intel(R) Core(TM) i7-3770 K CPU @ 3.50 GHz processor, 16.0 GB of RAM, a 64-bit Windows 11 Pro operating system, and an NVIDIA GeForce GT 640 GPU, supported by 1 terabyte of storage. The programming was conducted in Python, utilizing scikit-learn for machine learning tasks and Pandas, NumPy, and Matplotlib for data analysis and visualization. The runtime performance of the models varied, with the DTEB model requiring 3186 s, DTSD taking 2500 s, DTC completing in 500 milliseconds, SVEB needing 18,000 s, SVDS taking 20,000 s, and the default SVC model running within 1–4 s. This setup enabled the efficient implementation, optimization, and evaluation of the models while highlighting the significant computational demands associated with hyperparameter optimization processes, especially for advanced models like SVDS and SVEB. These runtime details emphasize the resource requirements, which are critical for scaling or deploying the models in real-time medical applications.

### 3.2 Prediction outcomes

2 separate phases of model training and testing are usually used in the assessment of prediction outcomes:

- **Splitting Data into Training and Testing Sets (70/30 Split):** 30% of the available data is set aside for testing the model's performance in this phase, with the

remaining 70% going toward training it. The algorithm adjusts its parameters during training to identify patterns and links in the data, effectively classifying patients related to brain tumors and those without them. The prediction performance of the trained model is then assessed by utilizing the testing set that was set aside.

- **Utilizing All Available Data (100% Data):** During this stage, the whole dataset is used for testing and training the model. Since the entire dataset is used to train the model, it may be able to capture a more complete depiction of the underlying data distribution. Using the same dataset, the model's performance is assessed after training, allowing for a comprehensive evaluation of its predictive power.

The model's performance is assessed using important assessment measures in both cases, such as *accuracy*, *precision*, *recall*, and *F1 – score*. The model's generality and robustness in a variety of data configurations are revealed through a comparison of the outcomes from the 2 approaches.

The numerical results are recorded in Table 3 for a thorough summary of the model's predictive ability. These results are also graphically displayed in Fig. 5 for a schematic representation of the model's predictive power. It is possible to gain important insights into the model's predictive ability by carefully examining either of these resources. A close inspection of the test phase in Table 3 reveals significant performance differences between the individual SVC and DTC models. In particular, the DTC model performs admirably, with an accuracy of 0.941. This result supports the idea that models with greater values of singular accuracy typically perform better when fixed optimizers are used. A closer examination of the domain of DTS's optimized models clarifies this idea. Interestingly, the improved DTSD model outperforms its single counterpart with an accuracy of 0.959, which is a 1.9% improvement. Likewise, the refined DTEB model performs better than its single iteration, with an accuracy of 0.970, which is a noteworthy 3.1% improvement. When the 2 optimized DTC models are compared, it is clear that the DTEB model performs better than the single model counterpart, showing higher accuracy improvement.

### 3.3 Classification outcomes

To help with accurate diagnosis of patients, the main goal of this study is to differentiate between patients who have brain tumors and those who do not. Clinical decisions are informed by the real-time assessments of illness status made possible by this classification. To achieve this goal, essential metrics, including F1-score, precision, and recall, are used to assess how well different models perform. A review of

**Table 3** The outcome of the showcased developed models

| Section | Model | Metric values |           |        |          |
|---------|-------|---------------|-----------|--------|----------|
|         |       | Accuracy      | Precision | Recall | F1-score |
| Train   | SVC   | 0.907         | 0.910     | 0.907  | 0.905    |
|         | SVSD  | 0.927         | 0.930     | 0.928  | 0.927    |
|         | SVEB  | 0.938         | 0.940     | 0.939  | 0.938    |
|         | DTC   | 0.945         | 0.946     | 0.945  | 0.944    |
|         | DTSD  | 0.955         | 0.956     | 0.955  | 0.955    |
|         | DTEB  | 0.969         | 0.970     | 0.969  | 0.969    |
| Test    | SVC   | 0.894         | 0.904     | 0.894  | 0.894    |
|         | SVSD  | 0.927         | 0.930     | 0.928  | 0.927    |
|         | SVEB  | 0.938         | 0.940     | 0.939  | 0.938    |
|         | DTC   | 0.941         | 0.944     | 0.941  | 0.941    |
|         | DTSD  | 0.959         | 0.962     | 0.959  | 0.959    |
|         | DTEB  | 0.970         | 0.972     | 0.970  | 0.970    |
| All     | SVC   | 0.903         | 0.908     | 0.903  | 0.902    |
|         | SVSD  | 0.925         | 0.929     | 0.925  | 0.924    |
|         | SVEB  | 0.936         | 0.939     | 0.937  | 0.936    |
|         | DTC   | 0.943         | 0.945     | 0.943  | 0.943    |
|         | DTSD  | 0.956         | 0.958     | 0.956  | 0.956    |
|         | DTEB  | 0.969         | 0.970     | 0.969  | 0.969    |

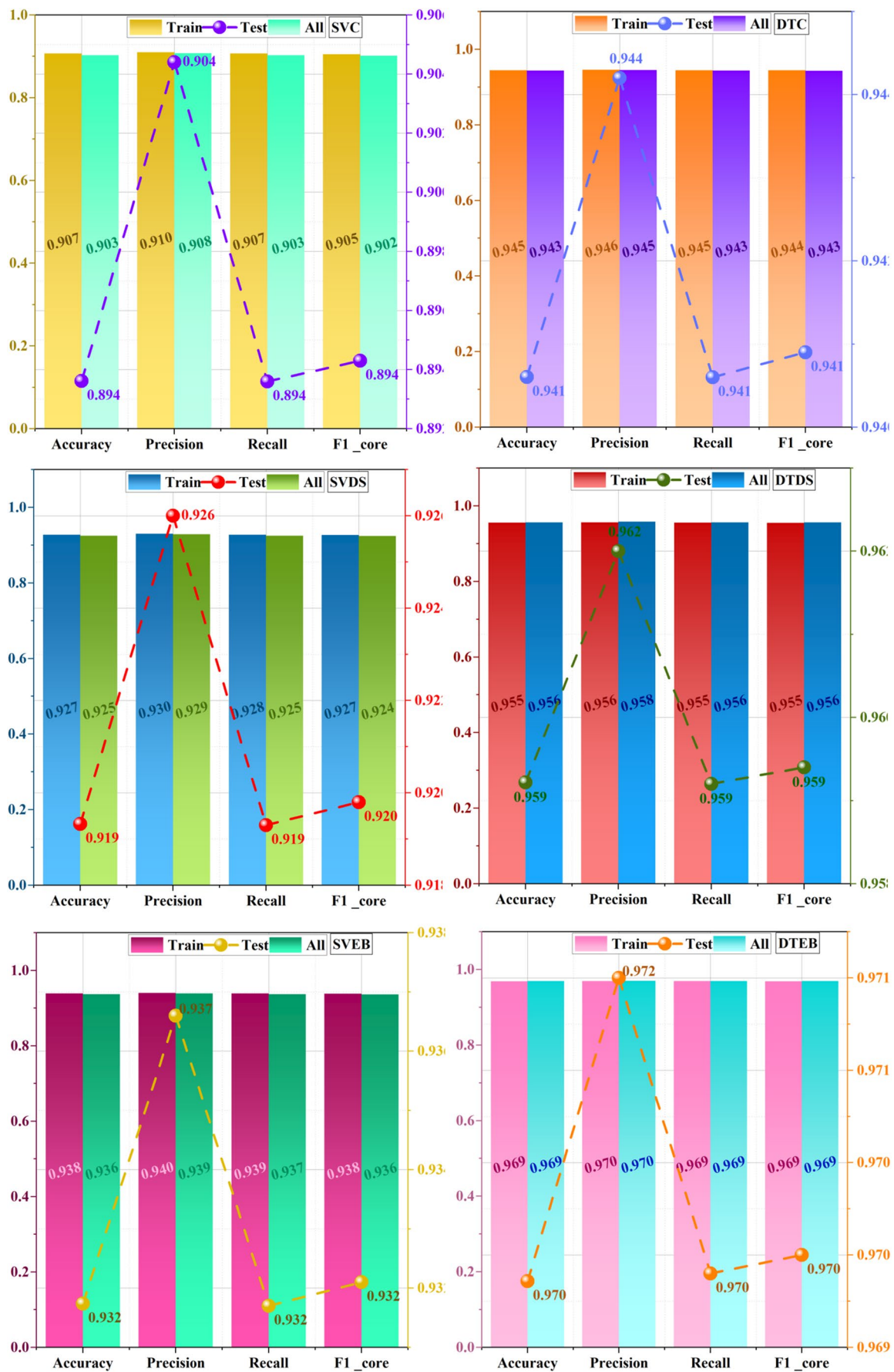


**Fig. 5** Column line symbol plot for the performance of the models across phases

Table 4, which shows the numerical results of these metrics for various scenarios, indicates that the DTC model and its optimized versions perform better than the SVC model and its improved versions. After more examination, it is clear that the DTEB model is the most effective of the improved DTC models. With values close to 1 for each of the 3 metrics in both conditions, the DTEB model performs better than any other. For example, the brain tumor group's F1-score is 0.965, while the non-tumor group's is 0.973. Similarly, for the non-tumor and brain tumor groups, the DTEB model scores 0.993 and 0.940 in terms of memory, respectively. This pattern is supported by precision results, where the DTEB model has scores of 0.991 for brain tumors and 0.953 for patients without tumors.

The dataset that is being examined consists of 1683 individuals who have been diagnosed with brain tumors and 2079 persons who have not. Figure 6 presents specific observations about how well different models anticipate the future. Users are likelier to trust a model that shows the greatest reliability when aligning with the measured data. However, users should be aware of the potential flaws in the model's predictions and proceed with caution. One prominent instance may be found in the DTEB model, which shows around 2064 out of 2079 cases classified as non-tumors (i.e., an error rate of 0.73%). Comparably, in the class of brain tumors, the DTEB model makes predictions for roughly 1582 out of 1683 cases (i.e., an error rate of 6.39%). Users must thoroughly assess the model's outputs and take into account the underlying uncertainties when making decisions, as the error rates shown above are highlighted.

The accuracy percentages of the different models are shown in a pie chart in Fig. 7, where 100% is the total number of measured patients. Each model's prediction of this value is carefully evaluated, and the model that obtains the highest percentage is considered the best performer. The DTEB model is the most dominant in the brain tumor cohort, accounting for 14.4% of the total. The DTDS model follows in second place, with a lower representation of 14.3%. In the reverse direction, the non-tumor group indicates that the best performance is from the DTEB model, reaching 14.9% of the total. In that same group, the DTDS model occupies the second position in performance, with 14.5% of the total, consistent with its result in the cohort with brain tumors. This study further solidifies the DTEB model as a robust predictive tool, since this investigation has further established its performance in cohorts with and without brain tumors. Besides that, the stable performance of the DTDS model shows that it can be considered a reliable alternative in clinical decision-making.

### 3.4 Analysis of attributes

SHAP sensitivity analysis is one of the robust tools for learning how specific input features affect model predictions. SHAP takes their ideas a bit further by quantifying the effect of each feature on the model output, where global and local model behaviors become explainable. This dual-level explanation will let the researchers understand how individual features impact the predictions at the dataset level while identifying feature importance for particular data points. Important principles underlying SHAP values concern consistency, coherence, and handling of missing data; hence, it allows for interpreting model predictions and finding hidden relationships in data reliably.

This study showed the contribution of specific features to the model's decision-making process with clinical implications using SHAP analysis. Figure 8 depicts the influence of input variables on the output of the model. For example, Entropy is the most influential feature with an impact magnitude of about 0.7, which indicates its critical role in the discrimination of tumor and non-tumor regions. Homogeneity followed, contributing significantly with an influence value of 0.2, indicating its relevance in capturing textural uniformity. Finally, with an impact of 0.1, kurtosis underscored its moderate but meaningful role. Importantly, the influence of these features was balanced across tumor and non-tumor classes, indicating their diagnostic value in identifying both categories. However, while SHAP analysis provided valuable insights, further elaboration on the clinical relevance of these features, such as why Entropy or Homogeneity correlates with tumor characteristics, would enhance the interpretability of results in a clinical context. This could bridge the gap between statistical analysis and practical diagnostic utility.

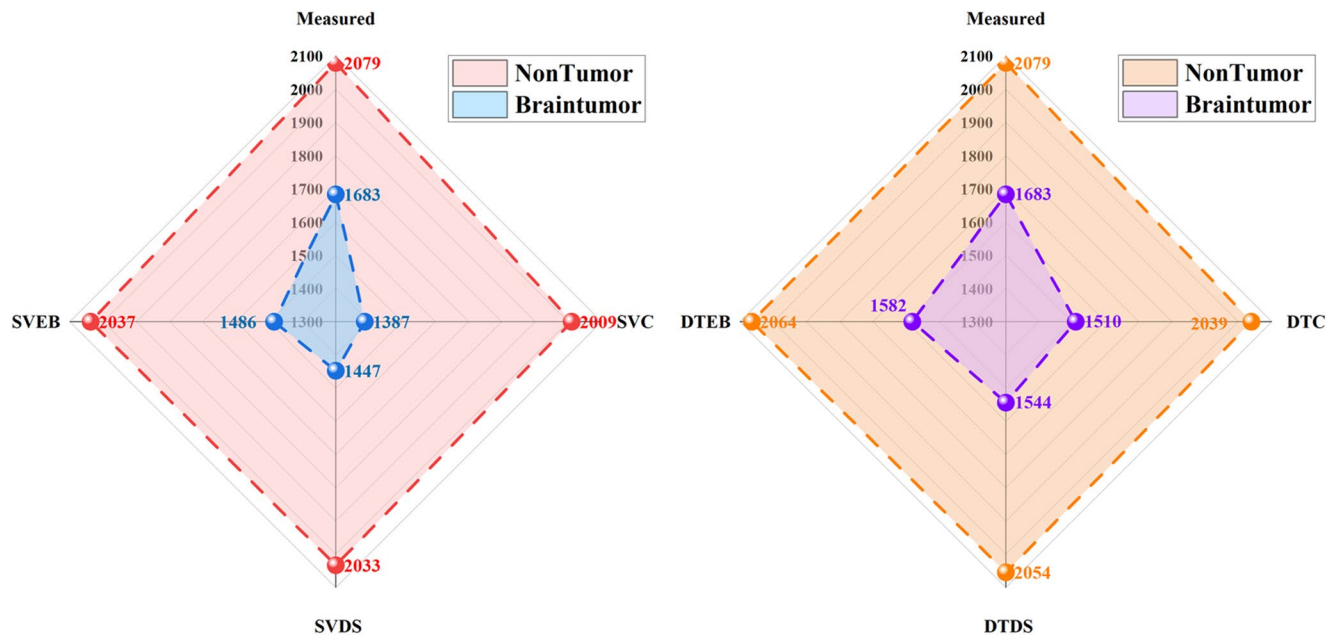
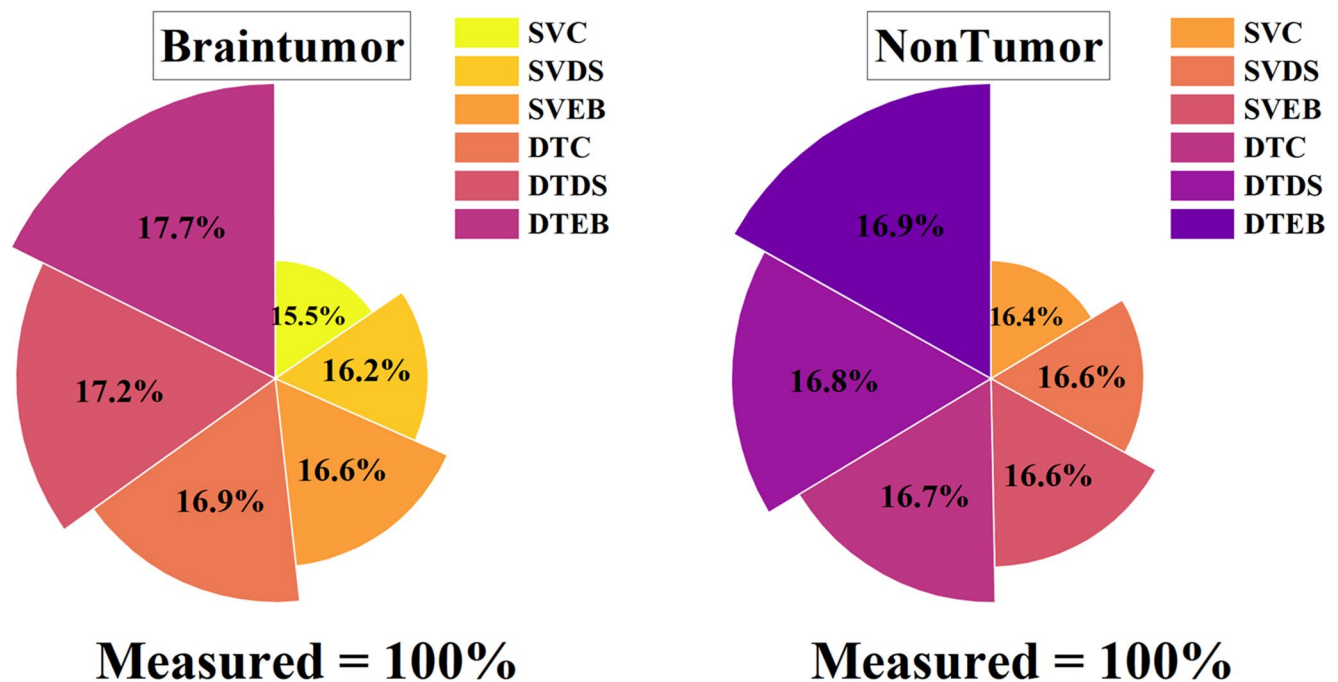
## 4 Discussion

### 4.1 Practical implication

Integrating advanced enhancement tactics, such as DSSA and EBSO, into machine learning models, enhances their prediction accuracy and reliability. These methods enable refined hyperparameter tuning, ensuring that the models achieve optimal configurations for classifying brain tumors. The optimized models, DTDS, DTEB, SVDS, and SVEB, not only improve classification performance but also enhance the ability to capture complex patterns in high-dimensional datasets. This has significant implications in practical medical diagnostics, where accurate and reliable classification of brain tumors can aid in early detection

**Table 4** Categorization of assessment criteria for the performance of the developed models

| Metric values | Condition  | Model |       |       |       |       |       |
|---------------|------------|-------|-------|-------|-------|-------|-------|
|               |            | SVC   | SVSD  | SVEB  | DTC   | DTSD  | DTEB  |
| Precision     | NonTumor   | 0.872 | 0.896 | 0.912 | 0.922 | 0.937 | 0.953 |
|               | Braintumor | 0.952 | 0.969 | 0.973 | 0.974 | 0.984 | 0.991 |
| Recall        | NonTumor   | 0.966 | 0.978 | 0.980 | 0.981 | 0.988 | 0.993 |
|               | Braintumor | 0.824 | 0.860 | 0.883 | 0.897 | 0.917 | 0.940 |
| F1-score      | NonTumor   | 0.917 | 0.935 | 0.945 | 0.950 | 0.962 | 0.973 |
|               | Braintumor | 0.883 | 0.911 | 0.926 | 0.934 | 0.950 | 0.965 |

**Fig. 6** Radar plot to represent the difference among the models**Fig. 7** Pie chart for the accuracy percentage of models

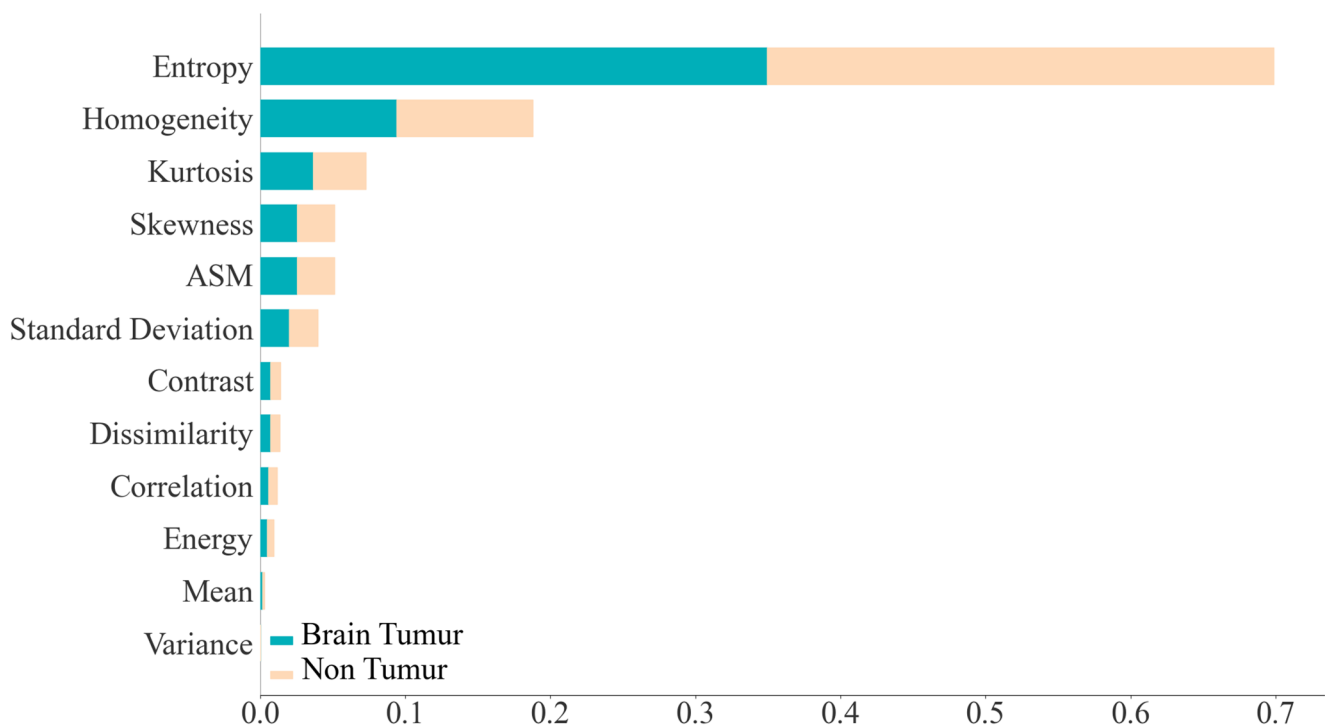


**Fig. 8** SHAP sensitivity analysis for the output of models

**Table 5** The comparison between this study and published studies

| Number | Article                       | Model   | Evaluator<br>Accuracy |
|--------|-------------------------------|---------|-----------------------|
| 1      | Sultan et al. [38]            | CNN-DNN | 0.9613                |
| 2      | Deepak, S and Ameer, P M [39] | CNN-SVM | 0.9582                |
| 3      | Alanazi et al. [40]           | TL-CNN  | 0.9575                |
| 4      | Present study                 | DTEB    | 0.969                 |

and treatment planning, ultimately leading to better patient outcomes.

While these methods of optimization undeniably yield some gain in accuracy, there is a lack of detailed discussion on computational costs and scalability. Practical implementation in real-world scenarios, more so in healthcare where resources are always constrained, requires this understanding. For example, the computational time of optimization increases with dataset and model complexity, which might be limiting for real-time system deployment. Apart from this, the scalability issue occurs in these models, particularly for larger data sets and varied clinical settings. Further studies must focus on weighing the gains induced by optimization with computational overhead to eventually present a practical tool in everyday diagnostics, besides being an effective one.

## 4.2 Comparison with published studies

Table 5 presents a comparison between the present study and previously published research on brain tumor categorization

utilizing machine learning frameworks. The models and their corresponding accuracy values are listed for each study. Sultan et al. (Sultan et al. 2019) utilized an incorporation of CNN and DNN models, achieving an accuracy of 0.9613. Deepak and Ameer (Deepak and Ameer 2021) employed a CNN-SVM framework, which yielded an accuracy of 0.9582. Alanazi et al. (Alanazi et al. 2022) implemented a Transfer Learning-based CNN (TL-CNN) model with an accuracy of 0.9575. In comparison, the present study utilized the DTEB model, achieving the highest accuracy of 0.969. The outcomes showcase that DTEB surpasses the tactics from the referenced studies, demonstrating superior performance regarding categorization accuracy. This gives an idea of the strength brought in by the advanced optimization tactics and hybrid machine learning models in improving precision and robustness for brain tumor diagnosis. Comparatively, this method holds great value in getting more precise diagnostic tools into the hands of medical professionals, a fact that's considered critical for better clinical patient outcomes.

## 5 Conclusion

This work aims to boost the ML models for classifying brain tumor patients, with more emphasis on the Support Vector Classifier and Decision Tree Classifier. To further improve these models in their performance, 2 optimization techniques

were added, namely DSSA and EBSO, after realizing that the performances had room for improvement. The resulting optimized models, SVEB, SVDS, DTEB, and DTDS, were tested on a dataset with twelve factors for segregation into groups related and not related to brain tumors. The performance of these can be described by the following 4 major metrics: accuracy, recall, precision, and F1-score. 70% of the data has been used for training, and the remaining 30% is used for testing. An additional check was done on the entire dataset to ensure a wide analysis. The results indicated that the SVC model constantly performed lower than the DTC model, with a difference in performance of 4.43%. Also, the DTEB model was the best performing of the optimized DTC models, with an accuracy of 0.969, much higher than the 0.956 accuracy of the DTSD model. The DTEB model performed excellently in classification findings, correctly categorizing 1582 cases of brain tumors and 2064 cases of non-tumors, demonstrating its efficacy in both categories. It is essential to note that DTEB demonstrated error rates of 0.73% and 6.39% in cases of brain tumors and non-tumors, respectively, underscoring the necessity for cautious interpretation of findings. Users of this model should proceed with caution and consider these constraints when interpreting results, acknowledging the study's limits, including the possibility of mistakes resulting from unexplained parameters and other factors. This work emphasizes the potential of ML models, especially optimized DTC models like DTEB, to help clinical decision-making processes and increase the accuracy of brain tumor categorization.

**Acknowledgements** We wish to state that no individuals or organizations require acknowledgment for their contributions to this investigation.

**Author contributions** TH performed Data collection also XY carried out simulation and analysis. EJ evaluate the first draft of the manuscript. editing and writing.

**Funding** This work was supported by the Key Scientific Research Foundation of Hunan Provincial Department of Education (Nos. 23A0575, 24A0598), in part by the Hunan Provincial Natural Science Foundation (Nos. 2024JJ7184, 2024JJ7187), in part by the Guiding Science and Technology Plan Project of Yongzhou City (No. 2024YZ011), in part by the Research Project of Hunan University of Science and Engineering (No. 24XKYZC03), in part by the Teaching Reform Research Project of Hunan University of Science and Engineering (No. XKYJ2024012), in part by the Hunan Provincial Social Science Review Committee Project (XSP25YBC575), in part by the construction program of applied characteristic discipline in Hunan University of Science and Engineering, and in part by the Aid program for Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province.

**Data availability** No datasets were generated or analysed during the current study.

## Declarations

**Ethics approval and consent to participate** The paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and adherence to ethical standards.

**Competing interests** The authors declare no competing interests.

## References

- Aggarwal K et al (2022) Has the future started? The current growth of artificial intelligence, machine learning, and deep learning. *Iraqi J Comput Sci Math* 3(1):115–123
- Alanazi MF et al (2022) Brain tumor/mass classification framework using magnetic-resonance-imaging-based isolated and developed transfer deep-learning model. *Sensors* 22(1):372
- Aydogdu M, Firat M (2015) Estimation of failure rate in water distribution network using fuzzy clustering and LS-SVM methods. *Water Resour Manag* 29:1575–1590
- Barragán-Montero A et al (2021) Artificial intelligence and machine learning for medical imaging: A technology review. *Phys Med* 83:242–256
- Battineni G, Sagaro GG, Chinatalapudi N, Amenta F (2020) Applications of machine learning predictive models in the chronic disease diagnosis. *J Pers Med* 10(2):21
- Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol* 2(3):1–27
- Cheng J et al (2015) Enhanced performance of brain tumor classification via tumor region augmentation and partition. *PLoS ONE* 10(10):e0140381
- de Robles P et al (2015) The worldwide incidence and prevalence of primary brain tumors: a systematic review and meta-analysis. *Neuro. Oncol* 17(6):776–783. <https://doi.org/10.1093/neuonc/nou283>
- Deepak S, Ameer PM (2021) Automated categorization of brain tumor from mri using Cnn features and Svm. *J Ambient Intell Humaniz Comput* 12(8):8357–8369
- Dhahri H, Al Maghayreh E, Mahmood A, Elkilani W, Nagi MF (2019) Automated breast cancer diagnosis based on machine learning algorithms. *J. Healthc. Eng* 2019
- Fernando T, Gammulle H, Denman S, Sridharan S, Fookes C (2021) Deep learning for medical anomaly detection—a survey. *ACM Comput Surv* 54(7):1–37
- Géron A (2022) Hands-on machine learning with Scikit-Learn, keras, and tensorflow. O'Reilly Media, Inc.
- Hamamoto R et al (2020) Application of artificial intelligence technology in oncology: towards the establishment of precision medicine. *Cancers (Basel)* 12(12):3532
- Hedenström A, Rosén M (2001) Predator versus prey: on aerial hunting and escape strategies in birds. *Behav Ecol* 12(2):150–156
- Howland HC (1974) Optimal strategies for predator avoidance: the relative importance of speed and manoeuvrability. *J Theor Biol* 47(2):333–350
- Jena B, Naik MK, Wunnava A, Panda R (2021) A differential squirrel search algorithm BT - Advances in intelligent computing and communication, pp. 143–152
- Jerez JM et al (2010) Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artif Intell Med* 50(2):105–115
- Jun W, Liyuan Z (2022) Brain tumor classification based on attention guided deep learning model. *Int J Comput Intell Syst* 15(1):35

- Kapoor R, Walters SP, Al-Aswad LA (2019) The current state of artificial intelligence in ophthalmology. *Surv Ophthalmol* 64(2):233–240
- Lakshmi MJ, Nagaraja Rao S (2022) Brain tumor magnetic resonance image classification: A deep learning approach. *Soft Comput* 26(13):6245–6253
- Lentink D et al (2007) How swifts control their glide performance with morphing wings. *Nature* 446(7139):1082–1085
- Lundervold AS, Lundervold A (2019) An overview of deep learning in medical imaging focusing on MRI. *Z Med Phys* 29(2):102–127
- Maldonado S, Pérez J, Weber R, Labbé M (2014) Feature selection for support vector machines via mixed integer linear programming. *Inf Sci (Ny)* 279:163–175
- Mijwil MM (2021) Implementation of machine learning techniques for the classification of lung x-ray images used to detect covid-19 in humans. *Iraqi J Sci*, pp. 2099–2109
- Mijwil MM, Al-Zubaidi EA (2021) Medical image classification for coronavirus disease (COVID-19) using convolutional neural networks. *Iraqi J Sci* 62(8):2740–2747
- Mijwil MM, Aggarwal K, Doshi R, Hiran KK, Sundaravadivazhagan B (2022) Deep learning techniques for COVID-19 detection based on chest X-ray and CT-scan images: a short review and future perspective. *Asian J Appl Sci*
- Rehman A, Naz S, Razzak MI, Akram F, Imran M (2020) A deep learning-based framework for automatic brain tumors classification using transfer learning. *Circuits Syst Signal Process* 39(2):757–775
- Remzan N, Tahiry K, Farchi A (2024) Advancing brain tumor classification accuracy through deep learning: Harnessing radimagenet pre-trained convolutional neural networks, ensemble learning, and machine learning classifiers on MRI brain images. *Multimed Tools Appl*, pp. 1–29
- Rundo L et al (2016) Semi-automatic brain lesion segmentation in Gamma Knife treatments using an unsupervised fuzzy c-means clustering technique, in *Advances in Neural Networks: Computational Intelligence for ICT*, pp. 15–26
- Sandhiya B, Raja SKS (2024) Deep learning and optimized learning machine for brain tumor classification. *Biomed Signal Process Control* 89:105778. <https://doi.org/10.1016/j.bspc.2023.105778>
- Shahrouzi M, Salehi A Design of large-scale structures by an enhanced metaheuristic utilizing opposition-based learning, in (2020) 4th Conference on Swarm Intelligence and Evolutionary Computation (CSIEC) 2020:27–31
- Sharma H, Kumar S (2016) A survey on decision tree algorithms of classification in data mining. *Int J Sci Res* 5(4):2094–2097
- Shen Y-T, Chen L, Yue W-W, Xu H-X (2021) Artificial intelligence in ultrasound. *Eur J Radiol* 139:109717
- Shukur BS, Mijwil MM (2023) Involving machine learning techniques in heart disease diagnosis: a performance analysis. *Int J Electr Comput Eng* 13(2):2177
- Sultan HH, Salem NM, Al-Atabany W (2019) Multi-Classification of brain tumor images using deep neural network. *IEEE Access* 7:69215–69225. <https://doi.org/10.1109/ACCESS.2019.2919122>
- Vapnik V (1998) *Statistical Learning Theory*. New York: John Wiley & Sons, Inc
- Wahlang I et al (2022) Brain magnetic resonance imaging classification using deep learning architectures with gender and age. *Sensors* 22(5):1766
- Yazdan SA, Ahmad R, Iqbal N, Rizwan A, Khan AN, Kim D-H (2022) An efficient multi-scale convolutional neural network based multi-class brain MRI classification for SaMD. *Tomography* 8(4):1905–1927
- Zacharaki EI et al (2009) Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme. *Magn Reson Med Off J Int Soc Magn Reson Med* 62(6):1609–1618
- Zhao X, Ang CKE, Acharya UR, Cheong KH (2021) Application of artificial intelligence techniques for the detection of alzheimer's disease using structural MRI images. *Biocybern Biomed Eng* 41(2):456–473

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.