Received: 8 May 2023 | Revised: 29 June 2023 | Accepted: 9 July 2023 | Published online: 20 July 2023

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2023, Vol. 00(00) 1–14
DOI: 10.47852/bonviewJCCE32021049

BON VIEW PUBLISHING

Trends of Optimization Algorithms from Supervised Learning Perspective

Rahul Paul[1] and Kedar Nath Das[1,*]

[1]*Department of Mathematics, National Institute of Technology Silchar, India*

Abstract: Machine learning (ML) is rapidly evolving, leading to numerous theoretical advancements and widespread applications across multiple fields. The goal of ML is to enable machines to carry out cognitive tasks by acquiring knowledge from past encounters and resolving intricate issues despite varying circumstances that deviate from previous instances. Supervised learning (SL) being one of the most popular type of ML has become an area of significant strategic importance due to its practical applications, data collection, and computing power's exponential growth. On the other hand, optimization is a crucial component of ML that has garnered significant attention from researchers. Numerous proposals have been made one after another for solving optimization problems or enhancing optimization techniques in the field of ML. A comprehensive review and application of optimization methods from the perspective of ML is crucial to guide the development of both optimization and ML research. This paper presents information specifically on the area of SL and a wide range of optimization methods, applied in conjunction to address various scientific issues. Additionally, this paper explores some of the challenges and open problems in optimizing SL models.

Keywords: machine learning, regression, classification, gradient descent, evolutionary optimization techniques

1. Introduction

Machine learning (ML) is a subset of artificial intelligence which focuses on creating algorithms and statistical models that allow computers to acquire knowledge from data without requiring explicit programming. These algorithms and models are used to make predictions, classify data, and identify patterns in large datasets. It makes use of statistical methods to allow computers to gain expertise and become better at a particular job (Alpaydin, 2020; Simeone, 2018).

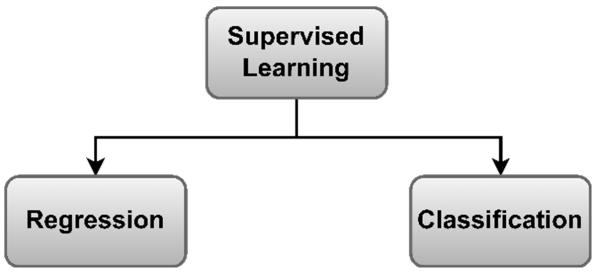
As one of the most prevalent types of ML, supervised learning (SL) has undergone rapid innovation and is used in numerous real-world applications. Over the years, researchers have been trying to maximize the solution quality in various real-life applications including the field of ML where the model performance is being optimized (Chong & Zak, 2013; Maier et al., 2019).

On the other hand, it is a general practice that the parameters involved in ML are being optimized with the help of suitable optimization techniques. These techniques are predominantly employed to minimize the error rate, which denotes the disparity between the predicted output and the actual output. Some common optimization techniques used in ML include gradient descent (GD) (Khasanov & Primqulov, 2021; Mustapha et al., 2020), stochastic gradient descent (SGD) (Bottou, 2010, 2012), and conjugate gradient methods (Hager & Zhang, 2006; Nazareth, 2009). These optimization techniques are also frequently used to solve various types of SL algorithms, such as linear and nonlinear regression, support vector machines (SVMs), and neural networks. GD is a widely used optimization algorithm in ML. It is used to minimize the error in a model by adjusting the parameters of the model. SGD is a variant of GD that uses random samples of the data to update the parameters of the model. For training models with a large number of parameters, the optimization approach known as the conjugate gradient is frequently employed in ML. It is particularly useful for large-scale optimization problems, as it is computationally efficient. In addition to these algorithms, there are also other important optimization techniques such as evolutionary algorithms (Vikhar, 2016; Yu & Gen, 2010) and Bayesian optimization (BO) (Snoek et al., 2012; Wu et al., 2019), which are used in ML for specific tasks such as anomaly detection, hyperparameter optimization, and antenna designing (Zhou et al., 2020). The selection of an optimization technique is reliant on both the problem at hand and the algorithm being used. Eventually, understanding the advantages of different optimization techniques and their limitations plays a crucial role in selecting the best problem-specific method that can provide optimal performance in ML.

It is observed from the literature that quite a good amount of research works on optimization algorithms used in ML perspectives were cited in the last few decades. However, the works are being presented in a scattered manner over the years. An attempt has been made in this paper to make a critical review specifically on SL method that could make a sense for the future researchers to proceed with the past knowledge. This paper presents an updated and systematic review of the optimization algorithms used in SL perspectives.

*Corresponding author: Kedar Nath Das, Department of Mathematics, National Institute of Technology Silchar, India. Email: kedarnath@math.nits.ac.in

Figure 1
Category of SL



This paper is structured into the subsequent sections. Section 2 provides a concise overview of the correlation between SL techniques and optimization algorithms. Section 3 discusses various problems related to optimization in SL and explains how frequently used optimization techniques are utilized to handle those problems. Finally, Section 4 concludes this paper by addressing the challenges and unresolved issues in the field of SL and optimization.

Figure 1 describes the categorization of SL into two main domains, and the detailed explanation is discussed in the subsequent section.

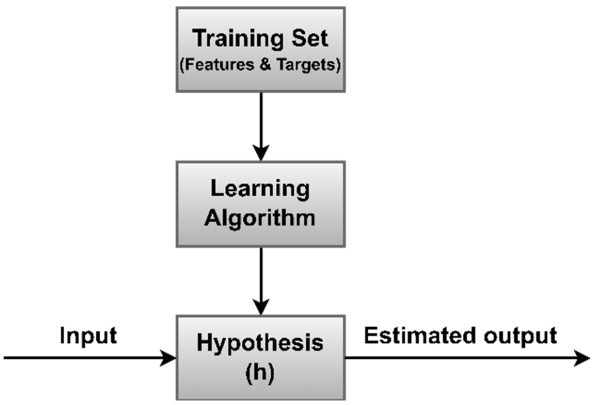
2. The Interplay of SL Techniques and Optimization Algorithms

The growing advantages of ML and optimization techniques are yet successfully applied in the field of medical sciences, social sciences, environmental sciences, industries, agricultural sciences, and engineering. The ML mechanism basically depends on various mathematical and statistical models. These models are useful in analyzing vast quantities of data, identifying patterns, and making predictions or decisions, enabling ML models to improve their performance over time.

Moreover, SL is a ML approach that involves training a model on data that has already been labeled, with the objective of enabling the model to predict outcomes for new, unseen data. In SL, we have inputs x (features) and y (targets), and we find a mapping $h : X \rightarrow Y$, $x \in X$ and $y \in Y$ (Nasteski, 2017; Ng, 2000). There are a wide variety of fields where SL is useful, including

(i) **Predictive modeling:** Predictive models can be constructed using SL techniques for various problems like stock prices (Panwar et al., 2021; Vijh et al., 2020), housing prices (Ho et al., 2021; Soltani et al., 2022), and weather forecasts (Paras et al., 2009; Zhao et al., 2021).
(ii) **Classification:** SL algorithms are widely used for classification tasks, such as spam filtering (Navaney et al., 2018) and sentiment analysis (Balahur & Turchi, 2014; Pannala et al., 2016).
(iii) **Regression analysis:** SL algorithms can also be used for regression analysis, which predicts continuous values based on input data, such as stock prices, housing prices, and energy consumption (Shin & Woo, 2022).
(iv) **Automated decision-making:** SL algorithms can be used to automate decision-making processes, such as loan approval (Faisal et al., 2021; Joshi et al., 2019), fraud detection (Patil et al., 2021; Shivagangadhar et al., 2015; Tax et al., 2021), and medical diagnosis (Kumar et al., 2021; Solanki et al., 2021; Suresh et al., 2022).

Figure 2
Process of SL



In the field of ML, SL is a critical area that is extensively studied and considered to be of utmost importance due to its significant role in numerous practical applications. Figure 2 shows the flow of SL.

Supervised ML encompasses various types of learning algorithms, including the following:

(i) **Linear regression:** Linear regression (Montgomery et al., 2021; Wu, 2022) is a statistical technique used in SL that aims to model the connection between one (or more) independent variables and a dependent variable and by finding a linear equation that best fits the data.

To minimize the sum of the squared errors, the model tries to fit a straight line to the data. It is one of the most widely used and simple models in ML.

The basic equation for simple linear regression is

$$y = h_{w,b}(x) = wx + b \tag{1}$$

where y is the dependent variable, x denotes the independent variable, w stands for slope of the line, and b represents y intercept.

For multiple linear regression, the equation becomes

$$y = h_{\vec{w},b}(\vec{x}) = \vec{w}.\vec{x} + b \tag{2}$$

where y denotes the dependent variable, $\vec{x} = [x_1, x_2, \ldots, x_n]$ denotes the feature vector of independent variables, and $\vec{w} = [w_1, w_2, \ldots, w_n]$, b are the regression coefficients or parameters of the model that show the effect of each predictor on the dependent variable, the operation here is the dot product of linear algebra.

In order to determine the best-fitting line or hyperplane, the model employs a cost function (Lubis et al., 2014) that measures the difference between the predicted values and the actual values of the dependent variable. Linear regression seeks to minimize this cost function.

The application of linear regression in SL is widespread. It is used in various fields like finance (Dixon et al., 2020); economics (Athey & Imbens, 2019; Storm et al., 2020); social sciences (Boelaert & Ollion, 2018; Hindman, 2015); and engineering (Khurana et al., 2016; Myers et al., 2012). Some examples of applications of linear regression include: predicting stock market prices and other marketing policies (Liu, 2022; Panwar et al., 2021; Vijh et al., 2020); forecasting sales (Kohli et al., 2021); medical diagnosis (Welsch & Kuh, 1977); and climate modeling (Hadley et al., 2006; Monteleoni et al., 2011).

(i) **Logistic regression:** It is a statistical modeling technique for predicting binary outcomes (usually 0 or 1). It is potent and extensively used in data science and for a variety of applications, like fraud detection (Itoo & Singh, 2021; Sahin & Duman, 2011), spam filtering (Alberto et al., 2015; Dedeturk & Akay, 2020), and disease diagnosis (Cui et al., 2021; Li et al., 2022; Xiao et al., 2021).

The sigmoid function (also called logistic function) forms the basis of the logistic regression model, which transforms the linear equation into a probability estimate.

The logistic regression model is

$$h_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w}.\vec{x}+b)}} \tag{3}$$

where $\vec{x}$, $\vec{w}$, and b are same as described in the context of equation (2) above. It takes features $\vec{x}$ and outputs a number between 0 and 1 by setting a suitable threshold value. The interpretation of logistic regression output is the probability that the class is 1 or positive.

The logistic regression model estimates the values of $\vec{w}$ and b by minimizing a loss function. Cross-entropy loss function (Ruby & Yendapalli, 2020; Sypherd et al., 2019) is a frequently employed loss function in logistic regression which is also known as the log loss. There are several numerical optimization techniques available, ranging from basic GD to more intricate second-order methods, which could be employed to ascertain the optimal parameter values (Saputro & Widyaningsih, 2017; Zou et al., 2019).

Moreover, to prevent overfitting and optimize the logistic regression model, it is possible to conduct variable selection by retaining only the most relevant subsets of x variables in the model, as suggested in Hastie et al. (2009). This can be achieved through heuristic techniques like forward selection or backward elimination, which involve adding or removing variables depending on the statistical importance of their estimated coefficients. Although interaction terms can be included to make the model more complex, doing so may increase the risk of training data overfitting.

Once trained, the logistic regression model has the capability to forecast fresh data by calculating the probability of a positive outcome given the values of the independent variables. If the probability goes over a specific threshold, generally 0.5, the model anticipates a positive outcome; otherwise, it predicts a negative outcome.

Over other classification algorithms, logistic regression has many advantages. It is simple to implement, computationally efficient, and can handle a large number of independent variables. Logistic regression enables researchers to determine the significance of each independent variable, thereby identifying the key factors that influence the outcome.

However, logistic regression has some limitations. It assumes that the independent variables are linearly related to the log odds of the positive outcome, and that the relationship is constant across all levels of the independent variables. Logistic regression also makes assumptions that the observations are independent of one another, and that the independent variables do not have multicollinearity. Finally, logistic regression is not appropriate for predicting outcomes with more than two categories, and an alternative for this is multinomial logistic regression (El-Habil, 2012; Hosmer Jr et al., 2013; Kwak & Clayton-Matthews, 2002).

(ii) **Decision Trees (DTs):** A DT is a popular ML model utilized for classification and regression analysis (Myles et al., 2004). In a classification problem, the goal is to predict a discrete class label for a given input. In a regression problem, the goal is to predict a continuous value for a given input. DTs can also be used for feature selection, where the most important features are selected based on their information gain.

The mechanism of building a DT involves the following steps: feature selection, node creation, recursion, and leaf node. For building DTs, there are various algorithms, like ID3 (Hssina et al., 2014; Singh & Gupta, 2014); C4.5 (Chauhan & Chauhan, 2013; Hssina et al., 2014; Singh & Gupta, 2014); CART (Aziza et al., 2019; Rutkowski et al., 2014; Singh & Gupta, 2014); and random forest (RF) (Breiman, 2001).

A technique for finding provably optimal DTs was suggested in Bertsimas and Dunn (2017), but its intricacy grows exponentially with the tree's depth. The paper (Günlük et al., 2021) offers a substitute formulation for optimal DTs that is specialized for categorical features, exploiting the combinatorial structure of this type of data to improve computational performance. Their approach is limited to binary classification and a fixed structure for the tree, which lowers the amount of computational work needed to solve the optimization problem. Both models, however, only consider one variable at each node in the DT to enhance computational performance. Data analysts often favor using single DT models due to their strong interpretability, but the accuracy of the model can be significantly enhanced by incorporating multiple DTs. The paper (Mišić, 2020) introduced a tree ensemble mixed integer optimization model as a recent solution to this issue.

DTs have some limitations such as overfitting and instability, which can be mitigated by pruning and ensemble methods. Pruning (Almuallim, 1996; de Marchi et al., 2023) involves removing nodes that do not add value to the model's accuracy, while ensemble methods combine multiple DTs to create a more robust model.

(iii) **Random Forest:** RF is a highly effective ML algorithm that finds widespread usage in classification (Cutler et al., 2007; Paul et al., 2018), regression (Li, 2013; Svetnik et al., 2003), feature selection (Menze et al., 2009; Nguyen et al., 2013; Sylvester et al., 2018), and other SL tasks such as predicting the heart disease risk based on medical data (Jabbar et al., 2016; Wang et al., 2023), predicting stock prices based on financial data (Khaidem et al., 2016; Sadorsky, 2021), and predicting customer churn based on marketing data (Kiguchi et al., 2022; Ullah et al., 2019). Recently, it is also used in remote sensing (Belgiu & Drăguţ, 2016), body motion detection (Kianoush et al., 2023), and genetics (Murgas et al., 2023). It is an ensemble learning method where multiple DTs are combined to develop a more accurate and robust model. RF is particularly useful when dealing with high-dimensional data, where the number of features is much larger than the number of samples.

In RF, optimization is significant, particularly in the construction of the individual DTs. Each DT is built by selecting the best split at each node based on some optimization criteria. The most commonly used optimization criteria in RF are Gini impurity and entropy, which measure the purity of the split based on the distribution of class labels in the data (Mohapatra et al., 2020).

Optimization is also used in RF for tuning of hyperparameters. Hyperparameters are parameters that are specified prior to training the model and have an impact on its performance. Examples of hyperparameters in RF include the maximum depth of each tree, the number of trees, and the number of features used for each split. Hyperparameter tuning entails determining the hyperparameters' optimal values to get the model's optimum performance on a validation set (Probst et al., 2019; Sun et al., 2020).

(iv) **Linear Discriminant Analysis (LDA):** LDA, a popular SL method in ML and pattern recognition, finds a linear combination of features that separates two or more data classes. Finding a decision boundary between classes that maximizes their separation is the primary objective of LDA. LDA achieves this by projecting the high-dimensional feature space onto a lower-dimensional subspace where the class separation is maximized.

The mathematical formulation of LDA involves finding the linear discriminant function $y(\mathbf{x})$ that maps a D-dimensional input vector $\mathbf{x}$ to a scalar value y. Finding the projection that maximizes the ratio of between-class variance to within-class variance is the aim.

The linear discriminant function for LDA is given by

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \tag{4}$$

where $\mathbf{x}$ is a vector of input features, $\mathbf{w}$ is a weight vector that defines the direction of the decision boundary, and b is a bias term that shifts the decision boundary. The weight vector and the bias term are chosen to maximize the separation between the classes. The mathematical operations are discussed in the paper (Balakrishnama & Ganapathiraju, 1998; Tharwat et al., 2017).

The role of optimization in LDA is critical as it helps to estimate the model parameters that maximize the separability of the classes. With optimization we find the optimal linear discriminant function that can separate the classes in the data with maximum margin. It involves finding the values of the parameters that minimize the within-class scatter and maximize the between-class scatter. This is done by finding the eigenvectors of the scatter matrices that correspond to the largest eigenvalues. The eigenvectors represent the directions along which the data are most separable.

While LDA and principal component analysis (PCA) both methods involve transforming the input features into a lower-dimensional space, they have different objectives and work in different ways.

PCA seeks to capture the most important patterns in the data by finding a set of orthogonal axes (principal components) that maximize the variance of the data when projected onto those axes. The principal components are computed by finding the eigenvectors of the covariance matrix of the data. On the other hand, LDA seeks to find a linear combination of the input features that can best separate the classes in the data.

The primary distinction between PCA and LDA is that PCA is an unsupervised method that does not take into account the class labels of the data, whereas LDA being a supervised method is more robust as it uses the class labels to find the best discriminant functions (Belhumeur et al., 1997).

However, LDA has some limitations; it assumes that the covariance matrix is the same for all classes, which may not be true in practice. It also assumes that the decision boundary is linear, so it may not work well for data that are not linearly separable. In such cases, nonlinear discriminant analysis algorithms like quadratic discriminant analysis or support vector machines may be more appropriate. The assumption of normal distribution of the data is also a limitation of LDA.

(v) **Support Vector Machines:** SVM (Cristianini & Shawe-Taylor, 2000) is a potent SL algorithm that has extensive usage in ML for classification (Awad & Khanna, 2015; Pal & Mather, 2005; Vijayarani & Dhayanand, 2015) and regression tasks (Brereton & Lloyd, 2010). It is particularly useful for tasks that involve data that cannot be easily separated by a simple linear boundary. SVMs can handle complex decision boundaries and can work with high-dimensional data with a relatively small sample size.

To find a hyperplane in the feature space that maximally separates the different classes of data is the fundamental tenet of SVM. The hyperplane is chosen such that it maximizes the margin between the different classes of data. Finding the hyperplane that maximizes this margin is the purpose of SVM.

However, in many cases, the data are not linearly separable, and SVM needs to find a nonlinear decision boundary that separates the classes of data. This is achieved by transforming the feature space into a higher-dimensional space using a kernel function. The kernel function maps the input space into a feature space where it is simpler to get a linear decision boundary. The most popular kernel functions include the linear, polynomial, radial basis function (RBF), and sigmoid functions (Hearst et al., 1998; Noble, 2006; Steinwart & Christmann, 2008).

SVM is closely linked to optimization, as it involves finding the hyperplane that maximizes the margin between the classes of data. This optimization problem is a quadratic programming problem, which involves minimizing a convex quadratic function subject to linear constraints. The problem of quadratic programming can be handled with a variety of optimization algorithms, such as the sequential minimal optimization algorithm or the GD algorithm (Scholkopf & Smola, 2018; Shawe-Taylor & Sun, 2011).

(vi) **k-Nearest Neighbors (k-NN):** k-NN is a simple yet powerful algorithm with strong applications in classification (Adeniyi et al., 2016; Guo et al., 2003), regression (Goyal et al., 2014; Kohli et al., 2021), and recommendation problems (Adeniyi et al., 2016; Singh et al., 2020a). It is an instance-based learning approach, where the algorithm learns by storing the entire training dataset and using it to make predictions for new instances (Guo et al., 2003, Zhang & Zhou, 2007).

The k-NN algorithm has a few important parameters, including the distance metric used to measure the distance between the data points, the value of "k," and how the algorithm handles ties (e.g., when there is an equal number of neighbors with different class labels). These parameters can be tuned to achieve better performance on different datasets.

k-NN has a number of advantages as a ML algorithm, including its simplicity, interpretability, and ability to handle nonlinear decision boundaries. It also does not require any training time, since it simply stores the entire training dataset, making it a good choice for small- to medium-sized datasets. However, k-NN can be computationally expensive for larger datasets, and its performance can degrade if the dataset contains irrelevant or noisy features.

In addition to ML, k-NN can also be used in optimization problems. For example, k-NN can be used in the field of logistics to optimize delivery routes (Mohammed et al., 2017). Given a set of delivery locations, k-NN can be used to find the "k" nearest neighbors for each location and then use that information to group the delivery locations into clusters that can be efficiently serviced by a single delivery vehicle.

(vii) **Naïve Bayes:** It is a prevalent probabilistic ML algorithm for classification tasks (Farid et al., 2014), such as text classification (Kim et al., 2006), and is popularly used as a classifier for making predictions in healthcare industries (Pattekari & Parveen, 2012; Vijayarani & Dhayanand, 2015). It is based on Bayes' theorem, which is a fundamental principle of probability theory. The algorithm is considered "Naïve" because it makes the assumption that the features used in the classification are independent of each other, which may not always be true in practice (Webb et al., 2010).

In terms of optimization, Naïve Bayes does not require an explicit optimization step during training (Zhang, 2004).

However, some optimization techniques can be used to improve its performance. For example, *Smoothing techniques*, to avoid zero probabilities (Cherian & Bindu, 2017; Kaur & Oberai, 2014). *Feature selection,* to enhance the accuracy of the algorithm (Chen et al., 2009, Dey Sarkar et al., 2014). *Hyperparameter tuning*, hyperparameters can have a significant impact on how well Naïve Bayes performs, so achieving the optimal values needs careful tuning (Sasongko et al., 2019). Optimization techniques such as grid search or randomized search can be used to systematically search for the optimal values of these hyperparameters.

(viii) **Neural networks:** These are ML models that draw inspiration from the structure and function of the human brain. They are composed of interconnected nodes, or neurons, which process and transmit information through a series of weighted connections (Goodfellow et al., 2016).

The capability of neural networks to learn and advance via training is one of its main features. During training, the network is presented with a set of labeled examples, called a training set, and adjusts the weights of the connections between neurons to minimize the difference between the predicted output and the true output. The importance of optimization in neural networks is fundamental, as it is the process by which the network learns to make accurate predictions from input data. Optimization's objective is to adjust the network's weights and biases in order to minimize a chosen loss function. There are various optimization algorithms that can be used to train neural networks, but one of the most commonly used is SGD. In SGD, the weights and biases of the network are updated iteratively based on the gradients of the loss function with respect to these parameters. The gradients are estimated using a small subset, or batch, of the training data, which makes the computation more efficient (Kalimeris et al., 2019).

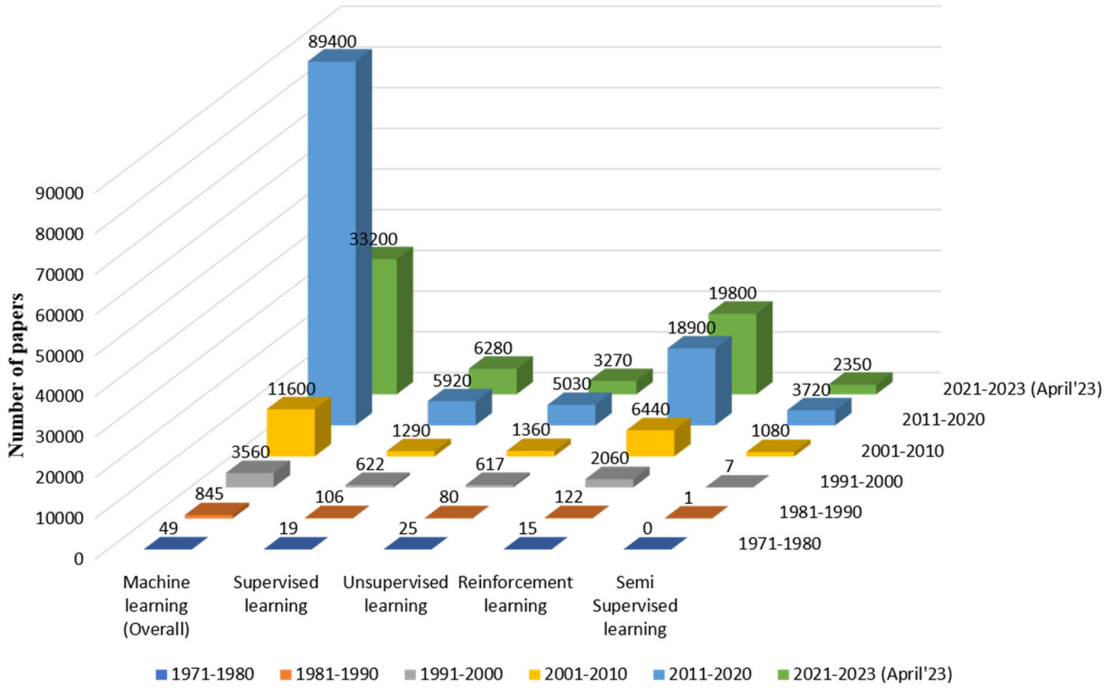
The key challenge in optimization for neural networks is to avoid overfitting (Liu et al., 2008; Piotrowski & Napiorkowski, 2013), where the network becomes too complex and starts to memorize the training data rather than learning general patterns. This can be mitigated using regularization techniques such as L_1 or L_2 regularization, dropout, or early stopping. Another important aspect of optimization is the choice of hyperparameters, like the number of epochs, batch size, and learning rate. The hyperparameter configuration or optimization can be achieved by using genetic algorithm (Aszemi & Dominic, 2019; Johnson et al., 2020) or Gaussian process (Dernoncourt & Lee, 2016). These hyperparameters can have a substantial effect on network performance and must be meticulously tuned for optimal results.

One common type of neural network is the feedforward neural network (Bebis & Georgiopoulos, 1994), where the data flow in one direction from input to output. Another type is the recurrent neural network (Grossberg, 2013; Salehinejad et al., 2017), which can process sequential data by allowing information to loop back through the network. Convolutional neural networks (CNNs) (Gu et al., 2018) are another type that are commonly used for image and signal processing tasks (Chaudhary et al., 2019; Ratnasingam, 2019).

A variety of tasks have been successfully completed using neural networks, including image recognition and natural language processing (Wang et al., 2020; Wang & Gang, 2018), speech recognition (Abdel-Hamid et al., 2012; Abdel-Hamid et al., 2013), and predictive modeling (Singh et al., 2020b). They have achieved state-of-the-art performance in many of these areas and have become a key tool in the field of ML.

These are some of the most popularly employed algorithms in supervised ML. The kind of problem at hand and the characteristics of the data will determine which algorithm should be performed. The approximate publication frequency of major categories ML approaches over the last 5 decades is briefly picturized in Figure 3.

Figure 3
Frequency analysis of published papers in ML and its variants at Google Scholar database



3. Optimization Challenges in SL

Optimization challenges are present throughout the field of ML. Nearly all ML algorithms can be represented as a problem of optimization, where finding the extremum of a given objective function is the primary aim. The creation of models and the development of appropriate objective functions are crucial initial steps in the application of ML methods. Once the objective function is established, appropriate numerical or analytical or evolutionary optimization techniques are typically employed to address the problem of optimization.

The objective in SL is to determine the optimal mapping function $h_{\vec{w},b}(\vec{x})$ that minimizes the loss function for the training samples

$$\min_{\vec{w},b} J(\vec{w},b) \tag{5}$$

$$J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m} L\left(h_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) + \frac{\lambda}{2m}\sum_{j=1}^{n} {w_j}^2 \tag{6}$$

where $\vec{x}^{(i)}$ is the feature vector of i^{th} training examples and $y^{(i)}$ is the corresponding target, L is the loss function, J is the regularized cost function, $\lambda > 0$ is the regularization parameter, m denotes the number of training examples, n is the number of features, and $\vec{w}, b$ are the model parameters.

Typically, the loss function for linear regression is

$$L\left(h_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \frac{1}{2}\left(h_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)^2 \tag{7}$$

and the loss function for logistic regression is

$$\begin{aligned} &L\left(h_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = \\ &-y^{(i)} log\left(h_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - \left(1 - y^{(i)}\right) log\left(1 - h_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) \end{aligned} \tag{8}$$

The objective is to determine the optimal $\vec{w}, b$ by minimizing $J(\vec{w}, b)$. To accomplish this, we will use a search algorithm that begins with an initial estimate for $\vec{w}, b$ and repeatedly adjusts it to decrease $J(\vec{w}, b)$ until we reach a $\vec{w}, b$ value that minimizes $J(\vec{w}, b)$. There are several algorithms that can be used in SL to find the value of $\vec{w}, b$ that minimizes the cost function; some of the most commonly used algorithms include GD, SGD, mini-batch gradient descent, ordinary least squares (OLS), conjugate gradient, Broyden–Fletcher–Goldfarb–Shanno (BFGS), limited-memory BFGS (L-BFGS), and Newton's method. The choice of algorithm depends on the specific problem being solved and the characteristics of the dataset.

Numerous distinct loss functions are used in SL in various types of models and tasks; some of the most common ones include least squares estimate (LSE), mean squared error, mean absolute error, binary cross entropy, categorical cross entropy, hinge loss, Kullback–Leibler divergence, cosine similarity loss, Wasserstein loss, Poisson loss, focal loss, contrast loss, and information gain.

The LSE is the most frequently employed loss function for linear regression. In this method, the goal of fitting a regression model is to minimize the residual sum of squares between the predicted outputs and the labels (Gambella et al., 2021). There are several algorithms that can be used to compute the LSE for a linear regression model, such as OLS (Craven & Islam, 2011), GD, SGD, ridge regression (Hoerl & Kennard, 1970), Lasso regression (Ranstam & Cook, 2018; Roth, 2004), and elastic net (Zou & Hastie, 2005). The choice of algorithm for computing the LSE relies on the particular needs of the problem at hand, like the dataset's size, the complexity of the model, and the computational resources available. The LSE is recognized for having the smallest variance out of all linear unbiased estimates, and it can be solved through a closed-form solution. Nonetheless, this option may not always be the best for fitting because it can result in a model with poor predictive accuracy due to high variance and may involve numerous non-zero regression coefficients, which makes it less interpretable. Shrinkage methods and linear dimension reduction (Gambella et al., 2021) offer other possibilities instead of the LSE. Forward or backward elimination are additional often employed techniques for variable selection to prevent overfitting (Hastie et al., 2009).

Noise can interfere with how accurately statistical learning techniques work during the process of collecting input data. To account for noise in the features of linear regression problems, the paper (Bertsimas & Copenhaver, 2018) introduced a model that considers this factor. The study also explores the connection between regularization and resilience to noise.

Creating linear regression models of superior quality necessitates several desirable characteristics that are often conflicting and cannot be executed simultaneously. The fitting process for linear regression models proposed in paper (Bertsimas & King, 2016) takes into account sparsity, selective sparsity, robustness to noisy data, stability against outliers, modeler expertise, statistical significance, and low global multicollinearity using mixed integer quadratic programming. In addition, the paper (Bertsimas & Shioda, 2007) explored mixed integer programming models for both regression and classification, and the regression problem involves assigning data points to clusters with identical regression coefficients.

Linear regression models have been extensively utilized in SL to predict a numerical outcome since the early days of statistics. Linear regression models are still relevant due to their simplicity, wide range of applications, and ease of interpretability. Specifically, in its most basic form, interpretability in ML refers to the ability to explain the contribution of the inputs to the output in a way that is understandable to humans, as discussed in Doshi-Velez and Kim (2017). The distribution's mode is a crucial data summary and is often estimated using a non-parametric kernel density estimator. The paper (Yao & Li, 2014) introduces a novel tool for analysis of data, known as modal linear regression, to investigate datasets with a large number of variables. To estimate the regression coefficients, an expectation-maximization algorithm was proposed. Asymptotic properties were offered for the proposed estimator without assuming the error density is symmetric. Experiments with both simulated and real data reveal that the proposed modal regression produces narrower predictive intervals compared to mean linear regression, median linear regression, and method of moments estimators.

Linear regression models can be expanded by incorporating nonlinear terms to capture more intricate relationships between regressors and predictors. This leads to the development of nonlinear regression models such as polynomial regression, local regression, regression splines, exponential regression, step functions, and smoothing splines. Some literature sources that discuss this topic include Hastie et al. (2009); Gambella et al. (2021); and James et al. (2013). When there is a nonlinear relationship between the independent and dependent variables, nonlinear regression models are employed. There are several algorithms that can be used to estimate the parameters of a nonlinear regression model, such as gradient-based optimization, Levenberg–Marquardt algorithm (Moré, 2006), nonlinear least squares (Dennis et al., 1981; Teunissen, 1990), Bayesian methods, neural networks, and support vector regression (Kavitha et al., 2016; Smola & Schölkopf, 2004).

Moreover, the quintessence of the recent papers on the interplay of SL and optimization techniques is also presented in Table 1.

Table 1
Gist of the recent papers on the association of optimization techniques and SL algorithms

Authors	Year	ML technique(s)	Optimization method(s) and statistical/probabilistic tool(s)	Key contribution/application
"V. Hoste, I. Hendrickx, W. Daeleman, A. van den Bosch" (Hoste et al., 2002)	2002	Feature weighting metrics (gain ratio weighting, information gain weighting, chi-squared weighting, shared variance weighting and log-likelihood weighting), memory-based learning (MBL) algorithms, DT	Algorithm parameter optimization	Word sense disambiguation (WSD) for Dutch and English language
"Jasper Snoek, Hugo Larochelle, Ryan P. Adams" (Snoek et al., 2012)	2012	Structured SVMs, CNN	BO, Gaussian process, Markov Chain Monte Carlo (MCMC), gradient-based search, covariance	Identified effective methods for Bayesian optimization of ML algorithms for hyperparameter selection, development of new algorithm such as GP EI MCMC, GP EI per second, N GP EI MCMC for managing varying time schedules and conducting simultaneous experiments
"Jakub Konecny, H. Brendan McMahan, Daniel Ramage" (Konečný et al., 2016)	2016	L2-regularized logistic regression, SVM	Federated optimization, GD, SGD, stochastic variance reduced gradient (SVRG), distributed approximate newton (DANE), expected improvement (EI)	Developed a new distributed optimization algorithm *Federated SVRG (FSVRG)* for the non-IID, unbalanced, and massively distributed data
"Aaron Klein, Stefan Falkner, Simon Bartels, Philipp Hennig, Frank Hutter" (Klein et al., 2017)	2017	SVMs, deep neural networks, information gain, CNN, residual neural network (RNN)	MCMC, Gaussian process, EI, entropy search (ES), multi-task opportunistic Bayesian optimization (MTOB), hyperband, random search, multi-task Bayesian optimization (MTOB)	Introduced *FAst Bayesian Optimization on LArge data Sets (FABOLAS)* to accelerate hyperparameter optimization
"Mohammad Noor Injadat, Fadi Salo, Ali Bou Nassif, Aleksander Essex, Abdallah Shami" (Injadat et al., 2018)	2018	SVM with Gaussian kernel (SVM-RBF), RF, and k-NN algorithms	Bayesian optimization technique	Presented how the Bayesian optimization approach is used to improve the effectiveness of a technique for detecting anomalies
"Pavas Navaney, Gaurav Dubey, Ajay Rana" (Navaney et al., 2018)	2018	Naïve Bayes algorithm, SVM algorithm, maximum entropy algorithm	Conditional probability, probability rule for independent events	The research analyzed the recognition of spam and non-spam (ham) messages through the use of various supervised ML algorithms. According to the findings, the SVM approach was the most precise in differentiating between ham and spam messages, trailed by the Naïve Bayes technique and lastly the maximum entropy method
"Weimin Zhou, Hua Li, Mark A. Anastasio" (Zhou et al., 2019)	2019	CNN, single-layer neural networks (SLNNs), ANN	MCMC, covariance matrices, Gaussian, Laplacian, mixed Poisson-Gaussian, receiver operating characteristic (ROC), MSE	Research was conducted to examine SL approaches for estimating the test statistics of the ideal observer and hotelling observer
"Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, Si-Hao Deng" (Wu et al., 2019)	2019	RF, CNN, recurrent neural network (RNN), multi-grained cascade forest	BO, Gaussian process, grid search, random search	An algorithm for tuning hyperparameters of ML models using Bayesian optimization was introduced

(*Continued*)

Table 1
(*Continued*)

Authors	Year	ML technique(s)	Optimization method(s) and statistical/probabilistic tool(s)	Key contribution/application
"Gyanendra Singh, Mahesh Pal, Yogendra Yadav, Tushar Singla" (Singh et al., 2020a)	2020	DNN, back-propagation neural network	Root mean squared error (RMSE), correlation coefficient, mean absolute error (MAE), Gaussian distribution, Adam	Used DNN model for prediction of road accidents
"Deliang Sun, Haijia Wen, Danzhou Wang, Jiahui Xu" (Sun et al., 2020)	2020	RF, recursive feature elimination	Bayes algorithm, random sampling, Gaussian process	A RF model was created for mapping the likelihood of landslides occurring, and the hyperparameters of the model were optimized using the Bayes algorithm
"Marco C. Campi, Simone Garatti" (Campi & Garatti, 2020)	2020	SVM, support vector regression (SVR), support vector data description (SVDD)	Convex optimization, joint probability distribution, data-driven optimization, stochastic optimization, distributionally robust optimization (DRO), Gaussian distribution, empirical distribution	Created a novel concept regarding scenario optimization with relaxation, which can be utilized as a tool for designing and implementing solutions to ML problems
"Lars Hertel, Julian Collado, Peter Sadowski, Jordan Ott, Pierre Baldi" (Hertel et al., 2020)	2020	DNN	GPyOpt algorithm, random search, grid search, successive halving, population-based training, local search	Developed a robust hyperparameter optimization library for ML models: Sherpa (code)
"Bilge Kagan Dedeturk, Bahriye Akay" (Dedeturk & Akay, 2020)	2020	Logistic regression	Artificial bee colony (ABC)	Proposed a novel spam filtering approach
"Yogendra Singh Solanki, Prasun Chakrabarti, Michal Jasinski, Zbigniew Leonowicz, Vadim Bolshev, Alexander Vinogradov, Elzbieta Jasinska, Radomir Gono, Mohammad Nami" (Solanki et al., 2021)	2021	Wrapper-based feature selection, SVM, J48 (C4.5 DT algorithm), multilayer-perceptron (MLP) (a feed-forward ANN), Naive Bayes, KNN, RF	PSO, genetic search, Greedy stepwise, RMSE, Matthew's correlation coefficient (MCC), Kappa statistics, MAE, relative absolute error (RAE), root relative squared error	A system that combines supervised ML methods for breast cancer prognosis has been created. This system uses feature selection techniques and methods for dealing with imbalanced data
"Xinchun Cui, Ruyi Xiao, Xiaoli Liu, Hong Qiao, Xiangwei Zheng, Yiquan Zhang, Jianzong Du" (Cui et al., 2021)	2021	Adaptive LASSO logistic regression	PSO	Suggested a new approach for detecting Alzheimer's disease (AD)
"Mohsen Yoosefzadeh-Najafabadi, Dan Tulpan, Milad Eskandari" (Yoosefzadeh-Najafabadi et al., 2021)	2021	MLP, radial basis function (RBF), RF, bagging strategy (ensemble method), PCA	Coefficient of determination (R^2), MAE, RMSE, genetic algorithm (GA), nearest neighbor analysis (NNA), best linear unbiased prediction (BLUP), Pearson coefficient of correlations	An application that uses ML and genetic optimization algorithms has been suggested to model and improve the yield of soybeans by focusing on its component traits (data)
"Qingzhi Zhao, Yang Liu, Wanqiang Yao, Yibin Yao" (Zhao et al., 2021)	2021	SVM, SVR, radial basis function (RBF) kernel, neural network	Average RMSE, relative RMSE, Lagrange multiplier, R^2	The SVM algorithm was used to develop a model called HRF (hourly rainfall forecast)
"Majid Emami Javanmard, S.F. Ghaderi" (Javanmard & Ghaderi, 2022)	2022	ANN, autoregressive model (AR), autoregressive integrated moving average model (ARIMA), seasonal autoregressive integrated moving average model (SARIMA), seasonal autoregressive integrated moving average model with	PSO, grey wolf optimization (GWO), stepwise regression algorithm, mean absolute percentage error (MAPE), MAE, RMSE, normalized root mean squared error (NRMSE), RAE	Research has been done to use a combination of ML algorithms and an optimization model to predict greenhouse gas emissions using energy market data, creating a hybrid model

(*Continued*)

Table 1
(*Continued* **)**

Authors	Year	ML technique(s)	Optimization method(s) and statistical/probabilistic tool(s)	Key contribution/application
		exogenous factors (SARIMAX), RF, SVR, KNN, long short-term memory model (LSTM)		
"Santosh Jhansi Kattamuri, Ravi Kiran Varma Penmatsa, Sujata Chakravarty, Venkata Sai Pavan Madabathula" (Kattamuri et al., 2023)	2023	Wrapper-based feature selection, KNN, nearest centroid (NC), RF, Gaussian Naive Bayes (GNB), SVM, DT	Ant colony optimization (ACO), cuckoo search optimization (CSO), GWO	Created a novel SOMLAP dataset (swarm optimization and ML applied to portable executable malware detection) with the aim of investigating how cyber threat intelligence could potentially be enhanced by improving the accuracy of malware detection

4. Conclusion

The use of mathematical programming is a crucial component of many ML models, as the process of training these models involves solving large-scale optimization problems. Both the traditional and evolutionary optimization techniques are being widely employed to establish many milestones in the area of ML. However, there are still challenges to handle the problems involving nonlinearity of the models, uncertainty in the data, and the existence of larger scale in the problem at hand. This paper mainly reviews the optimization techniques and challenges in the field of SL. A concise overview of the notable contributions in the aligned field of research is being reported in a tabular form, at the end. It can be concluded that among the research applications based on the SL mechanism, the neural networks and SVM have gained much attention of researchers and can be considered as the widely used algorithms. Over the last few decades, researchers apply both traditional (BO, Markov Chain Monte Carlo optimization, GD, etc.) and nature-inspired (random search, ABC, PSO, GA, DE, GWO, etc.) approaches to solve the SL problems. The study on the application of such approaches ensures that there was a vast implementation of nature-inspired approaches as compared to the traditional techniques. Thus, possibly evolutionary techniques might have a better capability in handling complex problems.

As a future scope, some efficient and robust optimization technique along with unsupervised and reinforcement learning can be synergized in SL to be applied in various real-time scenarios like enhancing the quality of spam filtering, fraud detection, and developing an improved version of K-Means clustering optimizer to fine-tune the parameters in various fields like using thermal imaging, IoT-based sensors to predict breast cancer, etc.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

References

Abdel-Hamid, O., Deng, L., & Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. *Interspeech*, *2013*, 3366–3370.

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G. (2012). Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing,* 4277–4280.

Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics*, *12*(1), 90–108.

Alberto, T. C., Lochter, J. V., & Almeida, T. A. (2015). Tubespam: Comment spam filtering on YouTube. In *2015 IEEE 14th International Conference on Machine Learning and Applications,* 138–143.

Almuallim, H. (1996). An efficient algorithm for optimal pruning of decision trees. *Artificial Intelligence*, *83*(2), 347–362.

Alpaydin, E. (2020). *Introduction to machine learning*. USA: MIT Press.

Aszemi, N. M., & Dominic, P. D. D. (2019). Hyperparameter optimization in convolutional neural network using genetic algorithms. *International Journal of Advanced Computer Science and Applications*, *10*(6), 269–278.

Athey, S., & Imbens, G. W. (2019). Machine learning methods that economists should know about. *Annual Review of Economics*, *11*, 685–725.

Awad, M., & Khanna, R. (2015). *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*. Germany: Springer Nature.

Aziza, E. Z., El Amine, L. M., Mohamed, M., & Abdelhafid, B. (2019). Decision tree CART algorithm for diabetic retinopathy classification. In *2019 6th International Conference on Image and Signal Processing and their Applications,* 1–5.

Balahur, A., & Turchi, M. (2014). Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *Computer Speech & Language*, *28*(1), 56–75.

Balakrishnama, S., & Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, *18*(1998), 1–8.

Bebis, G., & Georgiopoulos, M. (1994). Feed-forward neural networks. *IEEE Potentials*, *13*(4), 27–31.

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 24–31.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific

linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720.

Bertsimas, D., & Copenhaver, M. S. (2018). Characterization of the equivalence of robustification and regularization in linear and matrix regression. *European Journal of Operational Research*, *270*(3), 931–942.

Bertsimas, D., & Dunn, J. (2017). Optimal classification trees. *Machine Learning*, *106*, 1039–1082.

Bertsimas, D., & King, A. (2016). OR forum—An algorithmic approach to linear regression. *Operations Research*, *64*(1), 2–16.

Bertsimas, D., & Shioda, R. (2007). Classification and regression via integer optimization. *Operations Research*, *55*(2), 252–271.

Boelaert, J., & Ollion, É. (2018). The great regression: Machine learning, econometrics, and the future of quantitative social sciences. *Revue Française de Sociologie*, *59*, 475–506.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of 19th International Conference on Computational Statistics Paris France,* 177–186.

Bottou, L. (2012). Stochastic gradient descent tricks. In G. Montavon., G. Orr & K. R. Müller (Eds.), *Neural networks: Tricks of the trade* (pp. 421–436). Springer.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Brereton, R. G., & Lloyd, G. R. (2010). Support vector machines for classification and regression. *Analyst*, *135*(2), 230–267.

Campi, M. C., & Garatti, S. (2020). Scenario optimization with relaxation: A new tool for design and application to machine learning problems. In *2020 59th IEEE Conference on Decision and Control,* 2463–2468.

Chaudhary, S., Taran, S., Bajaj, V., & Sengur, A. (2019). Convolutional neural network based approach towards motor imagery tasks EEG signals classification. *IEEE Sensors Journal*, *19*(12), 4494–4500.

Chauhan, H., & Chauhan, A. (2013). Implementation of decision tree algorithm c4. 5. *International Journal of Scientific and Research Publications, 3*(10), 1–3.

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, *36*(3), 5432–5435.

Cherian, V., & Bindu, M. S. (2017). Heart disease prediction using Naive Bayes algorithm and Laplace Smoothing technique. *International Journal of Computer Science Trends and Technology*, *5*(2), 68–73.

Chong, E. K., & Zak, S. H. (2013). *An introduction to optimization*. USA: John Wiley & Sons.

Craven, B. D., & Islam, S. M. (2011). Ordinary least-squares regression. *The SAGE Dictionary of Quantitative Management Research*, *2011*, 224–228.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other Kernel-based learning methods*. UK: Cambridge University Press.

Cui, X., Xiao, R., Liu, X., Qiao, H., Zheng, X., Zhang, Y., & Du, J. (2021). Adaptive LASSO logistic regression based on particle swarm optimization for Alzheimer's disease early diagnosis. *Chemometrics and Intelligent Laboratory Systems*, *215*, 104316.

Cutler, D. R., Edwards Jr., T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, *88*(11), 2783–2792.

de Marchi, D., Welch, M., & Kosorok, M. (2023). A robust hypothesis test for tree ensemble pruning. *arXiv Preprint: 2301.10115*.

Dedeturk, B. K., & Akay, B. (2020). Spam filtering using a logistic regression model trained by an artificial bee colony algorithm. *Applied Soft Computing*, *91*, 106229.

Dennis Jr., J. E., Gay, D. M., & Walsh, R. E. (1981). An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software*, *7*(3), 348–368.

Dernoncourt, F., & Lee, J. Y. (2016). Optimizing neural network hyperparameters with Gaussian processes for dialog act classification. In *2016 IEEE Spoken Language Technology Workshop,* 406–413.

Dey Sarkar, S., Goswami, S., Agarwal, A., & Aktar, J. (2014). A novel feature selection technique for text classification using Naive Bayes. *International Scholarly Research Notices*, *2014*, 1–10.

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in finance*. Germany: Springer International Publishing.

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

El-Habil, A. M. (2012). An application on multinomial logistic regression model. *Pakistan Journal of Statistics and Operation Research*, *8*(2), 271–291.

Faisal, M. F., Saqlain, M. N. U., Bhuiyan, M. A. S., Miraz, M. H., & Patwary, M. J. (2021). Credit approval system using machine learning: Challenges and future directions. In *2021 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications,* 76–82.

Farid, D. M., Zhang, L., Rahman, C. M., Hossain, M. A., & Strachan, R. (2014). Hybrid decision tree and Naïve Bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, *41*(4), 1937–1946.

Gambella, C., Ghaddar, B., & Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, *290*(3), 807–828.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. USA: MIT Press.

Goyal, R., Chandra, P., & Singh, Y. (2014). Suitability of KNN regression in the development of interaction based software fault prediction models. *IERI Procedia*, *6*, 15–21.

Grossberg, S. (2013). Recurrent neural networks. *Scholarpedia*, *8*(2), 1888.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., . . . , & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377.

Günlük, O., Kalagnanam, J., Li, M., Menickelly, M., & Scheinberg, K. (2021). Optimal decision trees for categorical data via integer programming. *Journal of Global Optimization*, *81*, 233–260.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. In *On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003*, 986–996.

Hadley, S. W., Erickson III, D. J., Hernandez, J. L., Broniak, C. T., & Blasing, T. J. (2006). Responses of energy use to climate change: A climate modeling study. *Geophysical Research Letters*, *33*(17).

Hager, W. W., & Zhang, H. (2006). A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, *2*(1), 35–58.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Germany: Springer.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and Their Applications*, *13*(4), 18–28.

Hertel, L., Collado, J., Sadowski, P., Ott, J., & Baldi, P. (2020). Sherpa: Robust hyperparameter optimization for machine learning. *SoftwareX*, *12*, 100591.

Hindman, M. (2015). Building better models: Prediction, replication, and machine learning in the social sciences. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 48–62.

Ho, W. K., Tang, B. S., & Wong, S. W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48–70.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression*. USA: John Wiley & Sons.

Hoste, V., Hendrickx, I., Daelemans, W., & van den Bosch, A. (2002). Parameter optimization for machine-learning of word sense disambiguation. *Natural Language Engineering*, *8*(4), 311–325.

Hssina, B., Merbouha, A., Ezzikouri, H., & Erritali, M. (2014). A comparative study of decision tree ID3 and C4. 5. *International Journal of Advanced Computer Science and Applications*, *4*(2), 13–19.

Injadat, M., Salo, F., Nassif, A. B., Essex, A., & Shami, A. (2018). Bayesian optimization with machine learning algorithms towards anomaly detection. In *2018 IEEE Global Communications Conference,* 1–6.

Itoo, F., & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, *13*, 1503–1511.

Jabbar, M. A., Deekshatulu, B. L., & Chandra, P. (2016). Intelligent heart disease prediction system using random forest and evolutionary approach. *Journal of Network and Innovative Computing*, *4*(2016), 175–184.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Germany: Springer.

Javanmard, M. E., & Ghaderi, S. F. (2022). A hybrid model with applying machine learning algorithms and optimization model to forecast greenhouse gas emissions with energy market data. *Sustainable Cities and Society*, *82*, 103886.

Johnson, F., Valderrama, A., Valle, C., Crawford, B., Soto, R., & Nanculef, R. (2020). Automating configuration of convolutional neural network hyperparameters using genetic algorithm. *IEEE Access*, *8*, 156139–156152.

Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., & Ghosh, J. (2019). Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv Preprint: 1907.09615*.

Kalimeris, D., Kaplun, G., Nakkiran, P., Edelman, B., Yang, T., Barak, B., & Zhang, H. (2019). SGD on neural networks learns functions of increasing complexity. *Advances in Neural Information Processing Systems*, *32*.

Kattamuri, S. J., Penmatsa, R. K. V., Chakravarty, S., & Madabathula, V. S. P. (2023). Swarm optimization and machine learning applied to PE Malware detection towards cyber threat intelligence. *Electronics*, *12*(2), 342.

Kaur, G., & Oberai, E. N. (2014). A review article on Naive Bayes classifier with various smoothing techniques. *International Journal of Computer Science and Mobile Computing*, *3*(10), 864–868.

Kavitha, S., Varuna, S., & Ramya, R. (2016). A comparative analysis on linear regression and support vector regression. In *2016 Online International Conference on Green Engineering and Technologies,* 1–5.

Khaidem, L., Saha, S., & Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *arXiv Preprint: 1605.00003*.

Khasanov, D., & Primqulov, O. (2021). Gradient descent in machine learning. In *2021 International Conference on Information Science and Communications Technologies,* 1–3.

Khurana, U., Turaga, D., Samulowitz, H., & Parthasrathy, S. (2016). Cognito: Automated feature engineering for supervised learning. In *2016 IEEE 16th International Conference on Data Mining Workshops,* 1304–1307.

Kianoush, S., Savazzi, S., Rampa, V., Costa, L., & Tolochenko, D. (2023). A random forest approach to body motion detection: Multi-sensory fusion and edge processing. *IEEE Sensors Journal*, *23*(4), 3801–3814.

Kiguchi, M., Saeed, W., & Medi, I. (2022). Churn prediction in digital game-based learning using data mining techniques: Logistic regression, decision tree, and random forest. *Applied Soft Computing*, *118*, 108491.

Kim, S. B., Han, K. S., Rim, H. C., & Myaeng, S. H. (2006). Some effective techniques for Naive Bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, *18*(11), 1457–1466.

Klein, A., Falkner, S., Bartels, S., Hennig, P., & Hutter, F. (2017). Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 54,* 528–536.

Kohli, S., Godwin, G. T., & Urolagin, S. (2021). Sales prediction using linear and KNN regression. *In Advances in Machine Learning and Computational Intelligence, 2019,* 321–329.

Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv Preprint: 1610.02527*.

Kumar, N., Narayan Das, N., Gupta, D., Gupta, K., & Bindra, J. (2021). Efficient automated disease diagnosis using machine learning models. *Journal of Healthcare Engineering*, *2021*.

Kwak, C., & Clayton-Matthews, A. (2002). Multinomial logistic regression. *Nursing Research*, *51*(6), 404–410.

Li, X. (2013). Using "random forest" for classification and regression. *Chinese Journal of Applied Entomology*, *50*(4), 1190–1197.

Li, Y., Lu, F., & Yin, Y. (2022). Applying logistic LASSO regression for the diagnosis of atypical Crohn's disease. *Scientific Reports*, *12*(1), 11340.

Liu, G. (2022). Research on prediction and analysis of real estate market based on the multiple linear regression model. *Scientific Programming*, *2022*, 1–8.

Liu, Y., Starzyk, J. A., & Zhu, Z. (2008). Optimized approximation algorithm in neural networks without overfitting. *IEEE Transactions on Neural Networks*, *19*(6), 983–995.

Lubis, F. F., Rosmansyah, Y., & Supangkat, S. H. (2014). Gradient descent and normal equations on cost function minimization for online predictive using linear regression with multiple variables. In *2014 International Conference on ICT for Smart Society, 2014,* 202–205.

Maier, H. R., Razavi, S., Kapelan, Z., Matott, L. S., Kasprzyk, J., & Tolson, B. A. (2019). Introductory overview: Optimization using evolutionary algorithms and other metaheuristics. *Environmental Modelling & Software*, *114*, 195–213.

Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., & Hamprecht, F. A. (2009). A comparison of random forest and its Gini importance with standard

chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics*, *10*, 1–16.

Mišić, V. V. (2020). Optimization of tree ensembles. *Operations Research*, *68*(5), 1605–1624.

Mohammed, M. A., Abd Ghani, M. K., Hamed, R. I., Mostafa, S. A., Ibrahim, D. A., Jameel, H. K., & Alallah, A. H. (2017). Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution. *Journal of Computational Science*, *21*, 232–240.

Mohapatra, N., Shreya, K., & Chinmay, A. (2020). Optimization of the random forest algorithm. In *Advances in Data Science and Management, 2019,* 201–208.

Monteleoni, C., Schmidt, G. A., Saroha, S., & Asplund, E. (2011). Tracking climate models. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *4*(4), 372–392.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. USA: John Wiley & Sons.

Moré, J. J. (2006). The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis: Proceedings of the Biennial Conference Held at Dundee,* 105–116.

Murgas, L., Pollastri, G., Riquelme, E., Saez, M., & Martin, A. J. (2023). Random forest approach for the identification of relationships between epigenetic marks and its application to robust assignment of chromatin states. *bioRxiv*, 2023–01.

Mustapha, A., Mohamed, L., & Ali, K. (2020). An overview of gradient descent algorithm optimization in machine learning: Application in the ophthalmology field. In *Smart Applications and Data Analysis: Third International Conference, 3,* 349–359.

Myers, R. H., Montgomery, D. C., Vining, G. G., & Robinson, T. J. (2012). *Generalized linear models: With applications in engineering and the sciences*. USA: John Wiley & Sons.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *18*(6), 275–285.

Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons. B*, *4*, 51–62.

Navaney, P., Dubey, G., & Rana, A. (2018). SMS spam filtering using supervised machine learning algorithms. In *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence),* 43–48.

Nazareth, J. L. (2009). Conjugate gradient method. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(3), 348–353.

Ng, A. (2000). CS229 Lecture notes. *CS229 Lecture Notes*, *1*(1), 1–3.

Nguyen, C., Wang, Y., & Nguyen, H. N. (2013). Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. *Journal of Biomedical Science and Engineering*, *6*(5), 1–10.

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, *24*(12), 1565–1567.

Pal, M., & Mather, P. M. (2005). Support vector machines for classification in remote sensing. *International Journal of Remote Sensing*, *26*(5), 1007–1011.

Pannala, N. U., Nawarathna, C. P., Jayakody, J. T. K., Rupasinghe, L., & Krishnadeva, K. (2016). Supervised learning based approach to aspect based sentiment analysis. In *2016 IEEE International Conference on Computer and Information Technology,* 662–666.

Panwar, B., Dhuriya, G., Johri, P., Yadav, S. S., & Gaur, N. (2021). Stock market prediction using linear regression and SVM. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering,* 629–631.

Paras, S. M., Kumar, A., & Chandra, M. (2009). A feature based neural network model for weather forecasting. *International Journal of Computational Intelligence*, *4*(3), 209–216.

Patil, N. S., Kamanavalli, S., Hiregoudar, S., Jadhav, S., Kanakraddi, S., & Hiremath, N. D. (2021). Vehicle insurance fraud detection system using robotic process automation and machine learning. In *2021 International Conference on Intelligent Technologies*, 1–5.

Pattekari, S. A., & Parveen, A. (2012). Prediction system for heart disease using Naïve Bayes. *International Journal of Advanced Computer and Mathematical Sciences*, *3*(3), 290–294.

Paul, A., Mukherjee, D. P., Das, P., Gangopadhyay, A., Chintha, A. R., & Kundu, S. (2018). Improved random forest for classification. *IEEE Transactions on Image Processing*, *27*(8), 4012–4024.

Piotrowski, A. P., & Napiorkowski, J. J. (2013). A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. *Journal of Hydrology*, *476*, 97–111.

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3), e1301.

Ranstam, J., & Cook, J. A. (2018). LASSO regression. *Journal of British Surgery*, *105*(10), 1348–1348.

Ratnasingam, S. (2019). Deep camera: A fully convolutional neural network for image signal processing. In P*roceedings of the IEEE/ CVF International Conference on Computer Vision Workshops.*

Roth, V. (2004). The generalized LASSO. *IEEE Transactions on Neural Networks*, *15*(1), 16–28.

Ruby, U., & Yendapalli, V. (2020). Binary cross entropy with deep learning technique for image classification. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(4), 5393–5397.

Rutkowski, L., Jaworski, M., Pietruczuk, L., & Duda, P. (2014). The CART decision tree for mining data streams. *Information Sciences*, *266*, 1–15.

Sadorsky, P. (2021). A random forests approach to predicting clean energy stock prices. *Journal of Risk and Financial Management*, *14*(2), 48.

Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by ANN and logistic regression. In *2011 International Symposium on Innovations in Intelligent Systems and Applications,* 315–319.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv Preprint: 1801.01078*.

Saputro, D. R. S., & Widyaningsih, P. (2017). Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method for the parameter estimation on geographically weighted ordinal logistic regression model (GWOLR). In *AIP Conference Proceedings, 1868*(1), 040009.

Sasongko, T. B., Arifin, O., & Al Fatta, H. (2019). Optimization of hyper parameter bandwidth on Naïve Bayes kernel density estimation for the breast cancer classification. In *2019 International Conference on Information and Communications Technology,* 226–231.

Scholkopf, B., & Smola, A. J. (2018). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. USA: MIT Press.

Shawe-Taylor, J., & Sun, S. (2011). A review of optimization methodologies in support vector machines. *Neurocomputing*, *74*(17), 3609–3618.

Shin, S. Y., & Woo, H. G. (2022). Energy consumption forecasting in Korea using machine learning algorithms. *Energies*, *15*(13), 4880.

Shivagangadhar, K., Sagar, H., Sathyan, S., & Vanipriya, C. H. (2015). Fraud detection in online reviews using machine learning techniques. *International Journal of Computational Engineering Research*, *5*(5), 52–56.

Simeone, O. (2018). A brief introduction to machine learning for engineers. *Foundations and Trends in Signal Processing*, *12*(3–4), 200–431.

Singh, G., Pal, M., Yadav, Y., & Singla, T. (2020a). Deep neural network-based predictive modeling of road accidents. *Neural Computing and Applications*, *32*, 12417–12426.

Singh, R. H., Maurya, S., Tripathi, T., Narula, T., & Srivastav, G. (2020b). Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology*, *9*(5), 556–559.

Singh, S., & Gupta, P. (2014). Comparative study ID3, cart and C4. 5 decision tree algorithm: A survey. *International Journal of Advanced Information Science and Technology*, *27*(27), 97–103.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*, 199–222.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *25*.

Solanki, Y. S., Chakrabarti, P., Jasinski, M., Leonowicz, Z., Bolshev, V., Vinogradov, A., . . . , & Nami, M. (2021). A hybrid supervised machine learning classifier system for breast cancer prognosis using feature selection and data imbalance handling approaches. *Electronics*, *10*(6), 699.

Soltani, A., Heydari, M., Aghaei, F., & Pettit, C. J. (2022). Housing price prediction incorporating spatio-temporal dependency into machine learning algorithms. *Cities*, *131*, 103941.

Steinwart, I., & Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Storm, H., Baylis, K., & Heckelei, T. (2020). Machine learning in agricultural and applied economics. *European Review of Agricultural Economics*, *47*(3), 849–892.

Sun, D., Wen, H., Wang, D., & Xu, J. (2020). A random forest model of landslide susceptibility mapping based on hyperparameter optimization using Bayes algorithm. *Geomorphology*, *362*, 107201.

Suresh, T., Assegie, T. A., Rajkumar, S., & Kumar, N. K. (2022). A hybrid approach to medical decision-making: Diagnosis of heart disease with machine-learning model. *International Journal of Electrical and Computer Engineering*, *12*(2), 1831–1838.

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003). Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, *43*(6), 1947–1958.

Sylvester, E. V., Bentzen, P., Bradbury, I. R., Clément, M., Pearce, J., Horne, J., & Beiko, R. G. (2018). Applications of random forest feature selection for fine-scale genetic population assignment. *Evolutionary Applications*, *11*(2), 153–165.

Sypherd, T., Diaz, M., Sankar, L., & Kairouz, P. (2019, July). A tunable loss function for binary classification. In *2019 IEEE International Symposium on Information Theory,* 2479–2483.

Tax, N., de Vries, K. J., de Jong, M., Dosoula, N., van den Akker, B., Smith, J., . . . , & Bernardi, L. (2021). Machine learning for fraud detection in e-Commerce: A research agenda. In *Deployable Machine Learning for Security Defense: Second International Workshop, 2,* 30–54.

Teunissen, P. (1990). Nonlinear least squares.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, *30*(2), 169–190.

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A churn prediction model using random forest: Analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE Access*, *7*, 60134–60149.

Vijayarani, S., & Dhayanand, S. (2015). Liver disease prediction using SVM and Naïve Bayes algorithms. *International Journal of Science, Engineering and Technology Research*, *4*(4), 816–820.

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, *167*, 599–606.

Vikhar, P. A. (2016). Evolutionary algorithms: A critical review and its future prospects. In *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication,* 261–265.

Wang, H., Li, Y., Khan, S. A., & Luo, Y. (2020). Prediction of breast cancer distant recurrence using natural language processing and knowledge-guided convolutional neural network. *Artificial Intelligence in Medicine*, *110*, 101977.

Wang, J., Rao, C., Goh, M., & Xiao, X. (2023). Risk assessment of coronary heart disease based on cloud-random forest. *Artificial Intelligence Review*, *56*(1), 203–232.

Wang, W., & Gang, J. (2018). Application of convolutional neural network in natural language processing. In *2018 International Conference on Information Systems and Computer Aided Education,* 64–70.

Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. *Encyclopedia of Machine Learning*, *15*, 713–714.

Welsch, R. E., & Kuh, E. (1977). *Linear regression diagnostics*. National Bureau of Economic Research.

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26–40.

Wu, Y. (2022, April). Linear regression in machine learning. In *International Conference on Statistics, Applied Mathematics, and Computing Science, 12163*, 1253–1264.

Xiao, R., Cui, X., Qiao, H., Zheng, X., Zhang, Y., Zhang, C., & Liu, X. (2021). Early diagnosis model of Alzheimer's disease based on sparse logistic regression with the generalized elastic net. *Biomedical Signal Processing and Control*, *66*, 102362.

Yao, W., & Li, L. (2014). A new regression model: Modal linear regression. *Scandinavian Journal of Statistics*, *41*(3), 656–671.

Yoosefzadeh-Najafabadi, M., Tulpan, D., & Eskandari, M. (2021). Application of machine learning and genetic optimization algorithms for modeling and optimizing soybean yield using its component traits. *Plos One*, *16*(4), e0250665.

Yu, X., & Gen, M. (2010). *Introduction to evolutionary algorithms*. Germany: Springer Science & Business Media.

Zhang, H. (2004). The optimality of Naive Bayes. *Aa*, *1*(2), 3.

Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048.

Zhao, Q., Liu, Y., Yao, W., & Yao, Y. (2021). Hourly rainfall forecast model using supervised learning algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–9.

Zhou, J., Yang, Z., Si, Y., Kang, L., Li, H., Wang, M., & Zhang, Z. (2020). A trust-region parallel Bayesian optimization method

for simulation-driven antenna design. *IEEE Transactions on Antennas and Propagation*, *69*(7), 3966–3981.

Zhou, W., Li, H., & Anastasio, M.A. (2019). Approximating the ideal observer and hotelling observer for binary signal detection tasks by use of supervised learning methods. *IEEE Transactions on Medical Imaging*, *38*, 2456–2468.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.

Zou, X., Hu, Y., Tian, Z., & Shen, K. (2019). Logistic regression model optimization and case analysis. In *2019 IEEE 7th International Conference on Computer Science and Network Technology,* 135–139.

How to Cite: Paul, R. & Das, K. N. (2023). Trends of Optimization Algorithms from Supervised Learning Perspective. *Journal of Computational and Cognitive Engineering* https://doi.org/10.47852/bonviewJCCE32021049