INFOGAIN PUBLICATION

International Journal of Advanced Engineering, Management and Science (IJAEMS)
Peer-Reviewed Journal
ISSN: 2454-1311 | Vol-10, Issue-3; Mar, 2024
Journal Home Page: https://ijaems.com/
DOI: https://dx.doi.org/10.22161/ijaems.103.4

An Overview of Supervised Machine Learning Paradigms and their Classifiers

Njideka Nkemdilim Mbeledogu, Roseline Uzoamaka Paul, Daniel Ugoh and Kaodilichukwu Chidi Mbeledogu

Received: 30 Jan 2024; Received in revised form: 14 Feb 2024; Accepted: 22 Mar 2024; Available online: 30 Mar 2024

Abstract**—** *Artificial Intelligence (AI) is the theory and development of computer systems capable of performing complex tasks that historically requires human intelligence such as recognizing speech, making decisions and identifying patterns. These tasks cannot be accomplished without the ability of the systems to learn. Machine learning is the ability of machines to learn from their past experiences. Just like humans, when machines learn under supervision, it is termed supervised learning. In this work, an in-depth knowledge on machine learning was expounded. Relevant literatures were reviewed with the aim of presenting the different types of supervised machine learning paradigms, their categories and classifiers.*

Keywords**— *Artificial intelligence, Machine learning, supervised learning paradigms***

I. INTRODUCTION

For intelligent system to perform complex tasks that historically requires human intelligence such as recognizing speech, making decisions and identifying patterns (Staff, 2023), it requires the ability to learn from past experiences. Learning is a process that leads to change and it is an attribute that is possessed by humans. It occurs as a result of experience and increases the potential for improved performance and future learning (Ambrose *et al*., 2010). As the intelligence demonstrated by machines are said to be artificial, their learning ability is referred to as machine learning (ML). ML is a type of Artificial Intelligence (AI) focused on building computer systems that learn from data. It has applications in all types of sectors including manufacturing, retail, cyber-security, real-time chatbot agents, humanities disciplines, Agriculture, Social media, healthcare and life sciences, Email, Image processing, travel and hospitality, financial services and energy, feedstock and utilities (Bansal *et al*., 2019).

In the light of its applications, it is undoubtedly more valuable than other branches of AI because for a system to be intelligent, it must possess the ability to learn in order to improve the performance of their AI software applications over time and as well as possess the ability to adapt to changes. This in turn fuels the advancements in AI and progressively blurs the boundaries between machine intelligence and human intellect (Tucci, 2023).

II. MACHINE LEARNING

ML are computational techniques (scientific algorithms and statistical models) that enable computers to learn from data without being explicitly programmed. If programming is automation, then ML is automating the process of automation. It provides machines with the ability to learn independently (Ghahremani-Nahr *et al*., 2021) and makes programming scalable.

According to NetApp (2023), ML is made up of three parts. They are:

a) Computational Algorithm: A formal procedure describing an ordered sequence of operations to be performed a finite number of times (Falade, 2021). This is at the core of considering determinations.
b) Variables and features that make up the decisions.
c) Knowledge Base: The known facts which the system trains to learn from.

In a typical simple model of machine learning (Fig. 1), the environment supplies the information to the learning element which uses the information to make improvements in the knowledge base in order for the performance element to perform its task accurately. The kind of information

supplied to the machine by the environment is usually imperfect, with the result that the learning element does not know in advance how to fill in missing details or ignore details that are unimportant. The machine therefore, operates by guessing and then receives feedback from the performance element. The feedback mechanism enables the machine to evaluate its hypotheses and revise them if necessary.

Two different kinds of information processing are involved in machine learning. They are the inductive and deductive information processing. General pattern and rules are determined from raw data and experience in the inductive information processing and it is used in similarity-based learning where as in deductive, general rules are used to determine the specific facts and is used in proof of a theorem where deductions are made from known axioms to other existing axioms (Haykin, 1994).

In comparison with the traditional programming, ML uses data and output to run on the computer to generate a program which can then be used in traditional programming while traditional programming uses data and program on the computer to produce output (Brownlee, 2020).

(a) Traditional Programming (b) Machine Learning

Fig. 1: Typical simple model of machine learning

Machine Learning Classifiers

The technique for determining which class a dependent belongs to base on one or more independent variables is termed as Classification. The type of machine learning algorithm that assigns a label to a data input is known as Classifier.

Supervised Machine Learning Paradigm and their Classifiers

As the name implies, it is when a machine learns under supervision. This is the learning paradigm for acquiring the input-output relationship information of a system based on a given set of paired input-output training samples. The model is provided with a correct answer (output) for every input pattern (Samarasinghe, 2006) and as such referred to as "learning with a teacher" (Jain, 1996), that is, available data comprises feature vectors together with the target values. The learner (computer program) is provided with two sets of data, training set and test set. The training set has labelled dataset examples (solution to each problem dataset) which the learner can use to identify unlabeled examples in the test set with the highest possible accuracy as depicted in Fig. 2. The data is analyzed in order to tune the parameters of the model that were not in the training set to predict the target value for the new set of data (test data).

The major tasks of supervised learning paradigms are:

i. Classification: Labeled data and classifiers are used to produce predictions about data input classifications. The function is discrete and it is a categorical type.
ii. Regression: The function is continuous. The target variable is numeric.
iii. Forecasting (Probability Estimation): The function is a probability.
iv. The supervised learning paradigm classifiers are Decision trees, Naïve Bayes, Regression, Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Discriminant Analysis, Ensemble Methods and Neural Networks.

Decision Trees

This is a statistical classifier used for both classification and regression problems. It incorporates nominal and numerical values that are expressed as a recursive partition of the instance space. Decision tree is a graphical representation of a well-defined decision problem (Fig. 3). It consists of nodes that are concerned with decision making and arcs which connects the nodes (decision rules). The decision tree forms the rooted (directed) tree that has basically three types of nodes: the root nodes, the internal nodes and the terminal nodes. The root node originates from the tree and in turn is called the parent node. It has no incoming edges and zero or more outgoing edges. Every other nodes have one incoming node and are called child node. A node with outgoing edges is termed an internal node. It is also referred to as the test node. It represents the features of the dataset. Each internal node has exactly one incoming edge, two or more outgoing edges and splits the instance space into two or more sub-spaces based on the discrete function of the input attribute values (attribute test condition) to separate records that have different characteristics. This latter process is called Splitting.

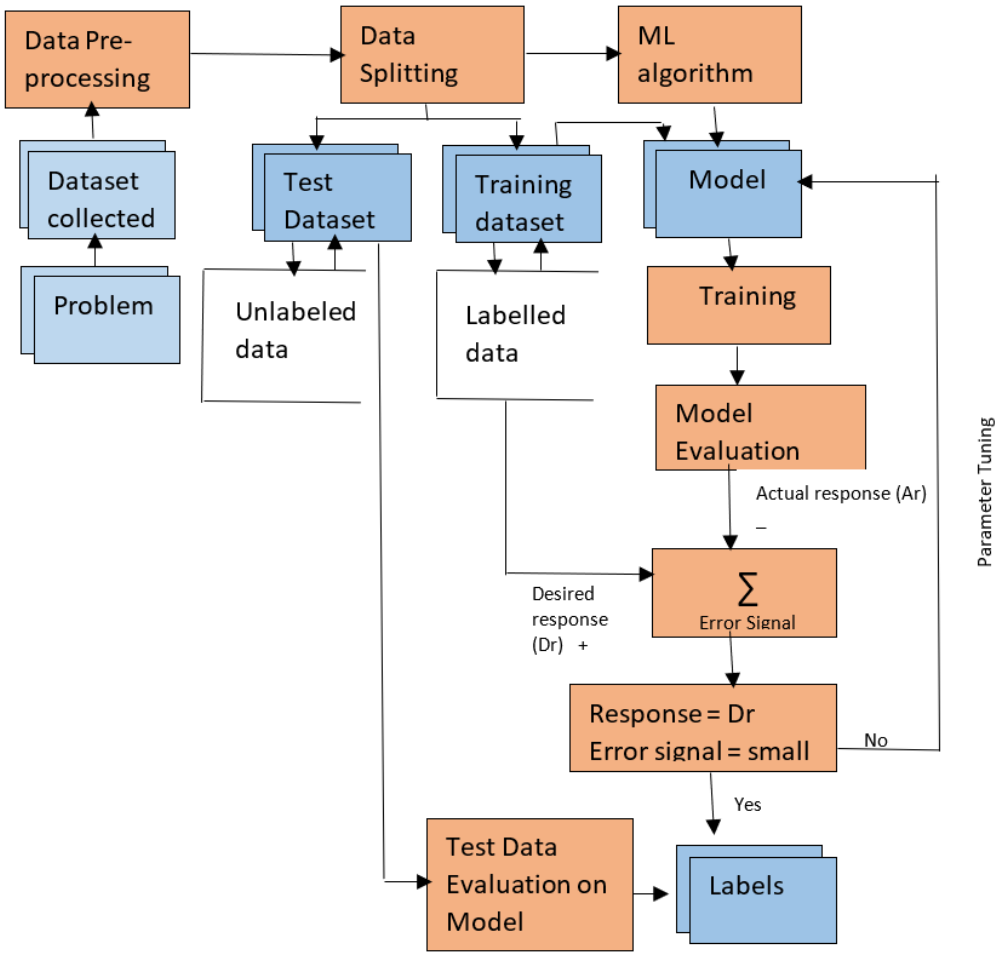

Fig. 2: Data Flow Diagram of Supervised Learning Paradigm

This is the process of dividing a node into two or more nodes and decision branches off into variables. For numeric attributes, the range is considered as the partition criteria where the decision tree can be geometrically interpreted as a collection of hyperplanes, each orthogonal to one of the axes. For classification problem, the entropy, Gini index and information gain (IG) are the splitting metrics used while for regression, residual sum of squares is applied. All other nodes apart from the root and internal nodes are termed as the leaves/terminal/decision nodes. Each of the leaf has exactly one incoming edge and no outgoing edges because it represents the outcome. The leaf node is assigned to the class label describing the most appropriate target value. Instances are classified by navigating them from the root down through the arcs to the leaf (Figure 4). Pruning in decision tree classifier is the opposite of splitting. It is the process of going through and reducing the tree to only the most important nodes or outcomes.

Decision Tree Pseudocode:

1. Start the decision tree with a root node, P that contains the complete dataset.
2. Using the Attribute Selection Measure (ASM), determine the best attribute in the dataset P to split it.
3. Divide P into subsets containing possible values for the best attributes.
4. Generate a tree node that contains the best attribute.
5. Make new decision trees recursively by using the subsets of the dataset P created in Step 3. Continue the process until a point is reached that the nodes cannot be further classified.

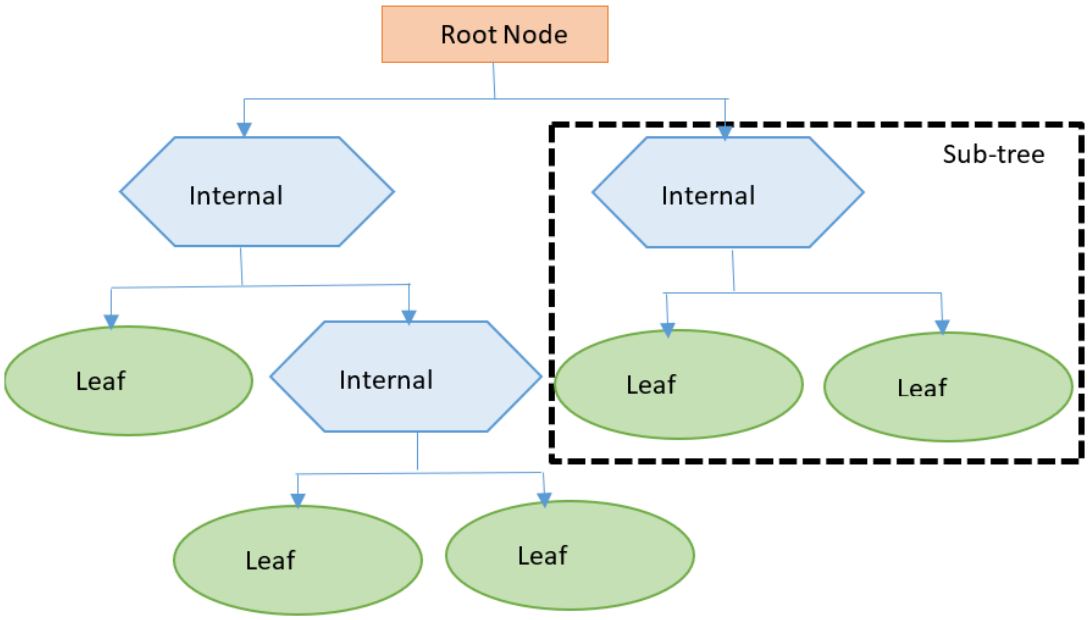

Fig. 3: Decision tree showing the root, internal and leaf nodes

Naive Bayes

This is a probabilistic classifier and a generative learning algorithm that is based on Bayes' theorem. It is used for text classification task. Given the data and some prior knowledge, the theorem is based on the probability of a hypothesis. The classifier assumes that all features in the input data are conditionally independent of each other, given the class label (note: this assumption is not true for all real world cases) thereby, permitting the algorithm to make predictions quickly. The dataset is divided into two: the feature matrix and the response vector. The feature matrix contains all the vector of the dataset in which each vector consist of the value of the dependent features. The response vector contains the value of class variable (prediction) for each row of the feature matrix.

Assumptions of Naive Bayes

i. Feature independence: The features of the data are conditionally independent of each other, given the class label.
ii. Continuous features are normally distributed: If a feature is continuous then it is assumed to be normally distributed within each class.
iii. Discrete features have multinomial distributions: If a feature is discrete then it is assumed to have a multinomial distribution within each class.
iv. Features are equally important: All features are assumed to contribute equally to the prediction of the class label.
v. No missing data: The data should not contain any missing values.

For the mathematical analysis from Bayes theorem, if A and B are events and P(B) ≠ 0, to find the probability of event A:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \ldots(1)$$

where Event B is an evidence (true), P(A) is the priori of A, P(B) is the marginal probability, $P(A|B)$ is the posteriori probability of B and $P(B|A)$ is the Likelihood probability that a hypothesis will come true based on the evidence.

Applying Bayes theorem:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad \ldots(2)$$

y is the class variable and X is the dependent feature vector (of size n), where

$$X = x_1, x_2, x_3, \ldots, x_n \quad \ldots(3)$$

Putting the naïve assumption into the Bayes' theorem (independence among the features), we split the evidence into independent parts.

If A and B are independent, then:

$$P(A,B) = P(A)P(B) \quad \ldots(4)$$

Hence,

$$P(y|x_1, x_2, x_3, \ldots, x_n) = \frac{P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)}{P(x_1)P(x_2)\ldots P(x_n)} \quad \ldots(5)$$

which can be expressed as:

$$P(y|x_1, x_2, x_3, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1)P(x_2)\ldots P(x_n)} \quad \ldots(6)$$

As the denominator remains constant for any given input, we remove $P(y|x_1, x_2, x_3, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$

In order to create the classifier model, we find the probability of the given set of inputs for all possible values of the class variable y, and with maximum probability.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y) \quad ..(7)$$

Regression

The goal of this statistical classifier is to plot the best-fit line or curve between the data (Kurama, 2023). A continuous outcome (y) is predicted based on the value of the predictor variables (x). Linear regression is the most common regression model due to ease (Fig. 4). It finds the linear relationship between the dependent variables (continuous) and one or more independent variables (continuous or discrete).

Steps in determining the best-fit line:

1. Considering the linear problem $y = mx + c$ where y is the dependent data, x is the independent data within the dataset, m is the coefficient (contribution of the input value in determining the best fit line) and c is the bias or intercept (deviations added to the line equation for the predictions made).
2. Adjust the line by varying m and c.
3. Randomly determine values initially for m and c and plot the line.
4. If the line does not fit best, adjust m and c using gradient descent algorithm or least square method.

$$y = mx + c \quad \dots(8)$$

y = the dependent variable and it is plotted along the y-axis

x = the independent variable and plotted along the x-axis

m = Slope of the line

c = the intercept (the value of y when x = 0)

Line of regression = Best fit line for a model

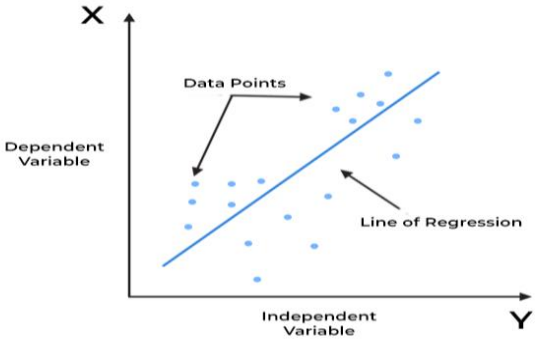

Fig. 4: Linear Regression Model showing the Best Fit Line

Logistic Regression

This does binary classification tasks by predicting the probability of an outcome, event, or observation. Based on the independent variables, it predicts the probability of an event occurring by fitting the data to a logistic function (Fig. 5). The coefficients of the independent variables in the logistic function are optimized by maximizing the likelihood function. A decision boundary is determined such that the cost function is minimal using Gradient Descent. The model delivers a binary or dichotomous outcome limited to two possible outcomes: yes/no, 0/1, or true/false. This is mathematically defined as:

$$y = \frac{e^{(b_0 + b_1 X)}}{1 + e^{(b_0 + b_1 X)}} \quad \dots(9)$$

where x = input value, y = predicted output, b_0 = bias or intercept term and b_1 = coefficient for input (x)

Logistic regression is similar to linear regression where the input values are combined linearly to predict an output value using weights or coefficient values but differs in the output value model. Logistic regression returns a binary value (0 or 1) as output rather than a numeric value as with the linear regression.

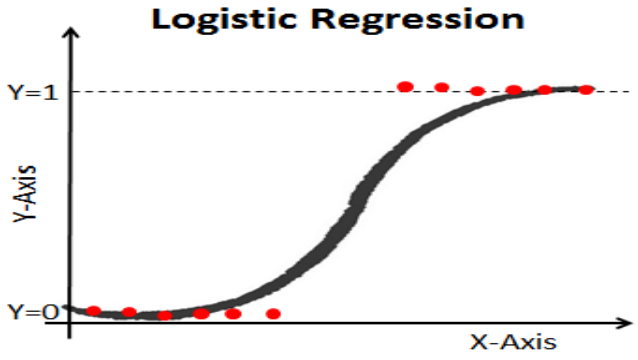

Fig. 5: Logistic Regression with predicted y between 0 and 1

Support Vector Machine (SVM)

This is used for classification (pattern recognition) and regression (function approximation) problems. It is based on statistical learning theory that can transform the input data into an N-dimensional (where N is the number of features that is high) by the use of kernel function to clearly create a linear model in the feature space. The kernel functions used in SVM include linear, polynomial, radial basis function and sigmoid function.

It constructs an optimal hyperplane (decision boundary) in a multidimensional space that separates cases of different class labels by using the objects (samples) on the edges of the margin (support vectors) to separate objects rather than using the differences in class means. It is based on the separation mechanism of the algorithm to obtain a hyperplane by supporting (defining) using the vectors (data points) nearest to the margin that it was called the Support Vector Machine.

Sahu and Sharma (2023) noted that SVM uses the Hinge Loss function to maximize the margin distance between the observations of the classes (training) as in Equ. 10.

$$l(y) = \max(0,1 + \max_{y \neq t} w_y x - w_t x) \qquad \ldots(10)$$

where w is the model parameter, x is the input variable and t is the target variable.

SVM can efficiently be used in high dimensional space where the number of spaces is higher than the number of samples, though it can result to poor outcome. The fame of SVM rests on two key properties: it finds solutions to classification tasks that have generalization and it solves non-linear problems using the kernel trick, thus, referred to as kernel machine. It uses Gaussian

distribution, thereby, making the induction paradigm for parameter estimation the maximum likelihood method which is then reduced to the minimization of sum-of-errors-square cost function.

K-Nearest Neighbour (K-NN)

This is a non-parametric instance base learning classifier that uses proximity (distance) to make predictions about the grouping of individual data. Due to the fact that it is unlikely for an object to exactly match another, the classifier finds a group of *k* objects in the training set that are closest to the test object by measuring the distance between the data (similarity measure) and assigns a label based on the predominance of a particular class in their neighbor (Steinbach and Tan, 2009). K-NN is a lazy learning technique because it delays until the query occurs to generalize beyond the training data.

K-NN Pseudocode

1. Determine parameter k = number of nearest neighbor.
2. Calculate the distance between the query-instance and all the training examples.
3. Sort the distance and determine the nearest neighbour based on the k-t minimum distance.
4. Gather the category Y of the nearest neighbor.
5. Use simple majority of the category of the nearest neighbor as the prediction value of the query instance.

Linear Discriminant Analysis (LDA)

This is also known as normal discriminant analysis (NDA) or discriminant function analysis (DFA). This technique aids in optimizing machine learning models in data science. It has generative model frame work because the data distribution for each class is modeled and uses Bayes theorem to classify new data points by calculating the probability of whether an input data set will belong to a particular output. Also, this is used to solve multi-class classification problems by separating multiple classes with multiple features through data dimensionality reduction.

Assumptions of LDA

1. Every feature such as variable, dimension, or attribute in the dataset has Gaussian distribution.
2. Each feature holds the same variance and has varying values around the mean with the same amount on average.
3. Each feature is assumed to be sampled randomly.
4. Lack of multicollinearity in independent features and there is an increment in correlations between independent features and the power of prediction decreases.

In reducing the features from higher dimension space to lower dimensional space, the following steps should be considered:

1. Compute the separate ability amid the various classes. This is to determine the between-class variance of the different classes (the distance between the mean of the different classes).
2. Compute the distance among the mean and the sample of each class (within class variance).
3. Determine the lower dimensional space that maximizes the between class variance and minimizes the within class variance.

Ensemble Methods

This classifier encapsulates multiple learning algorithms for better predictive results. It aims to mitigate errors or biases that may exist in individual models by leveraging the collective intelligence of the ensemble (Singh, 2023). The outputs of many models are combined thereby utilizing the strengths of these models to improve accuracy and handle uncertainties in data in its learning system. The various ensemble techniques are Max Voting, Averaging, Weighted Average, Stacking, Blending, Bagging and Boosting.

Artificial Neural Network (ANN)

It is designed to mimic the function and structure of the human brain. ANN is an intricate network of interconnected nodes or neurons that collaborates to tackle complicated tasks. The main characteristics of ANN is the ability to learn in classification task. It learns by example and through experience. In high dimensionality data, learning is needful in modeling non-linear relationships or recognizing not well established relationship amongst the input variables. The learning process is achieved by adjusting the weights of the interconnections according to the applied learning algorithm. The basic attributes of ANNs can be classified into Architectural attributes and Neuro-dynamic attributes (Kartalopoulos, 1996). The architectural attributes define the number and topology of neurons and interconnectivity while the neuro-dynamic attributes define the functionality of the ANN. Based on this, ANN is also referred to as Deep Learning (DL) when it has more than three layers (the depth of the layers are considered) to handle complex non-linear tasks. The Feed forward neural network comprises of the single layer (Hopfield net architecture) and Multiple layer perceptron (MLP) uses back propagation learning (Levenberg Marquardt) and Radial basis neural network are supervised learning.

Feed Forward Neural Networks (FFNN): This is a layered neural network in which an input layer of source nodes projects on to an input layer of neurons but not vice versa.

a. **Single-layer Feed Forward Network:** This is the simplest kind of neural network that is flat and consists of a single layer of output nodes (Fig. 6). It is also called single perceptron. The inputs are fed directly to the outputs through a series of weights. The sum of the products of the weights and the inputs are calculated in each node, and if the value is above some threshold (typically 0), the neuron fires and takes the activated value (typically 1); otherwise it takes the deactivated value (-1). Single perceptron is only capable of learning linearly separable patterns.

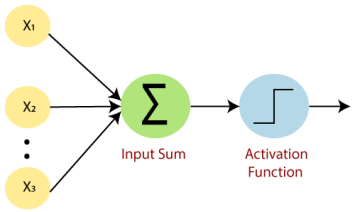

Fig. 6: A Single layer Feed Forward Network

The mapping of single unit perceptron is expressed as:

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \quad \ldots(11)$$

where w_i are the individual weights, x_i are the inputs and b is the bias

b. **Multilayer Feed Forward Network (MLP):** This distinguishes itself by the presence of one or more hidden layers called hidden neurons between the input units and the output units (Fig. 7). This aids the network in dealing with more complex non-linear problems. MLP is structured in a feed forward topology whereby each unit gets its input from the previous one (back propagation).

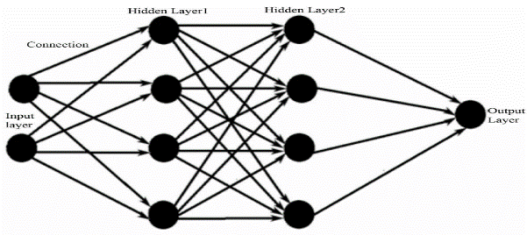

Fig. 7: Multiple Layer Perceptron

The mapping of the inputs to the outputs using an MLP neural network can be expressed as:

$$y_k = f\left(\sum_{j=1}^{m} w_{kj}^{(2)} \left(\sum_{i=1}^{n} w_{ji}^{(1)} + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \quad \ldots(12)$$

Where $w_{ji}^{(1)}$ and $w_{kj}^{(2)}$ indicate the weights in the first and second layers respectively, going from input i to hidden unit j (hidden layer 1), m is the number of the hidden units, y_k is the output unit, $w_{j0}^{(1)}$ and $w_{k0}^{(2)}$ are the biases for the hidden units j and k respectively. For simplicity, the biases have been omitted from the diagram.

c. **Radial Basis Neural Network (RBNN):** This is also called Radial Basis Feed Forward (RBF) network. It is a two layer feed forward type network in which the input is transformed by the basis function at the hidden layer (Fig. 8). At the output layer, linear combinations of the hidden layer node responses are added to form the output. The name RBF comes from the fact that the Basis function in the hidden layer nodes are radially symmetric, that is, the neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron.

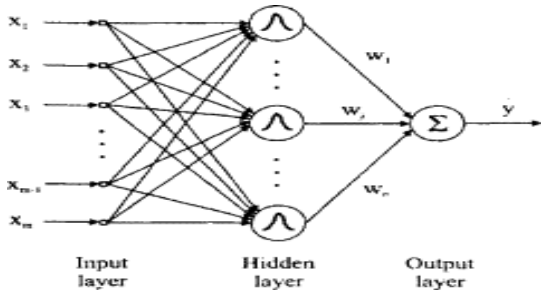

Fig. 8: Radial Basis Neural Network

Mathematically, it can be expressed as:

$$y(x) = \sum_{i=1}^{N} w_i \emptyset(\|x - c_i\|) \quad \ldots(13)$$

where x is the input vector, N is the number of neurons in the hidden layer, w_i are weights of the connections from the hidden layer to the output layer, c_i are the centers of the radial basis functions, $\|x - c_i\|$ is the Euclidean distance between the input vector and the center of the radial basis function and $\emptyset$ is the radial basis function usually chosen to be a Gaussian Function.

III. CONCLUSION

As the present world revolts round AI for its benefits, machine learning has been of immense importance to the building body of such intelligent systems to improve their performances. Learning under supervision to predict the output of a system when given new inputs has been more accurate and of ease when the decision boundary is not overstrained. The overview of supervised machine learning paradigms gave a detailed insight to the various statistical and scientific classifiers used in building functions that map new data onto the expected output values in tasks that requires either or both classification and regression issues.

REFERENCES

[1] Ambrose, S.A., Bridges, M.N., Dipietro, M, Lovett, M.C. and Norman, M.K. (2010). How Learning Works: Seven Research-Based Principles for Smart Teaching, Jossey-Bass A Wiley Imprint Publisher, San Francisco, pp. 1-301

[2] Bansal, R., Singh, J. and Kaur, R. (2019). Machine Learning and its Applications: A Review, Journal of Applied Science and Computations, Vol. VI Issue VI, pp. 1392-1398

[3] Brownlee, J. (2020). Basic Concepts in Machine Learning. Retrieved from https://machinelearningmastery.com/basic-concepts-in-machine-learning/

[4] Falade, K.I. (2021). Introduction to Computational Algorithm, Numerical and Computational Research Laboratory, pp.1-50

[5] Ghahremani-Nahr, J., Hamed, N. and Sadeghi, M.E. (2021). Artificial Intelligence and Machine Learning for Real-World Problem (A Survey), International Journal of Innovation in Engineering 1 (3), pp. 38-47

[6] Haykin, S. (1998). Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, Inc. USA, pp. 1-696

[7] Jain, A.K. (1996). Artificial Neural Networks: A tutorial. Pp. 1-14. Retrieved from www.cogsci.ucsd.edu/ajyu/Teaching/cogs202_sp12/Readings/jain_ann96.pdf

[8] Kartalopoulous, S.V. (1996). Understanding Neural Networks and Fuzzy Loic: Basic Concepts and Applications, IEEE press, NY, pp. 1-232

[9] Kurama, V. (2023). Regression in Machine Learning: What it is and Examples of Different Models. Retrieved from https://builtin.com/data-science/regression=machine-learning

[10] NetApp (2023). What is Machine Learning? Retrieved from https://www.netapp.com/artificial-intelligence/what-is-machine-learning/

[11] Sahu, C.K. and Sharma, M. (2023). Hinge Loss in support Vector Machine. Retrieved from https://www.niser.ac.in/~smishra/teach/cs460/23cs460/lectures/lecII.pdf

[12] Samarasinghe, S. (2006). Neural Networks for Applied Sciences and Engineering from Fundamentals to Complex Pattern Recognition, Auerbach Publications, Taylor and Francis Company , New York, pp. 1-570

[13] Singh, A. (2023). A Comprehensive Guide to Ensemble Learning (with Python Codes). Retrieved from https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/

[14] Staff, C. (2023). What is Artificial Intelligence? Definition, Uses and Types. Retrieved from https://www.coursera.org/articles/what-is-artificial-intelligence

[15] Steinbach, M. and Tan, P. (2009). KNN: K- Nearest Neighbors, Chapter 8, Taylor and Francis, pp. 151-159

[16] Tucci, L. (2023). What is Machine Learning and How does it work? In-depth guide. Retrieved from https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML