Decision Analytics Journal 3 (2022) 100071

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning

Malti Bansal [*], Apoorva Goyal, Apoorva Choudhary

Department of Electronics and Communication Engineering, Delhi Technological University (DTU), Delhi 110042, India

ARTICLE INFO

Keywords:
Machine Learning
K-Nearest Neighbor
Genetic Algorithm
Support Vector Machine
Decision Tree
Long Short-Term Memory

ABSTRACT

Machine learning (ML) is a new-age thriving technology, which facilitates computers to read and interpret from the previously present data automatically. It makes use of multiple algorithms to build models, mathematical in nature, and then makes predictions for the new data using the past data and knowledge. Lately, it has been adopted for text detection, hate speech detection, recommender system, face detection, and more. In this paper, majorly all the aspects concerning five machine learning algorithms namely-K-Nearest Neighbor (KNN), Genetic Algorithm (GA), Support Vector Machine (SVM), Decision Tree (DT) , and Long Short Term Memory (LSTM) network have been discussed in great detail which is a prerequisite for venturing into the field of ML. This paper throws light on various new results and conclusions related to these algorithms via research and review of recently published papers that carried out quantitative and qualitative research on real-time problems, mainly predictive analytics in multidisciplinary fields. This paper also talks about the circumstantial origin of these algorithms, which although has been rarely talked about in previous publications, is a preeminent point of discussion for ML enthusiasts and amateurs, both. To explain and understand the accuracy, robustness, and reliability of the algorithms, they were exhaustively reviewed and researched in all aspects qualitatively and quantitatively, wherein the LSTM network and SVM algorithm have projected a superior behavior over the rest. The paper answers all relevant questions that may arise during the study of these algorithms ranging from their origin, to their definition, methodologies of execution, real-time applications attached with sufficient novel evidence, followed by the advantages and major trade-offs; lastly an elaborate comparison of their performances on quantitative and qualitative grounds has been presented. To conclude, the paper also highlights the future scope of ML algorithms and artificial intelligence in the coming times and their roles in automation and holistic development, not just in technology-related aspects but also, the humanitarian aspects, finally followed by reliable and relevant conclusions derived from this exhaustive research.

Contents

[*] Corresponding author.
E-mail address: maltibansal@gmail.com (M. Bansal).

https://doi.org/10.1016/j.dajour.2022.100071
Received 25 November 2021; Received in revised form 20 May 2022; Accepted 23 May 2022
Available online 27 May 2022

1. Introduction

1.1. Machine learning

Machine Learning, an amalgamation of statistical concepts and scientific knowledge of computers, this term was coined by Arthur Samuel back in 1959, and today, it is considered to be a subset or a subpart of Artificial Intelligence (AI), associated to algorithms that permit processors or computers to automatically process and classify new data based on old data and information. Without comprehensive programming, computers are able to predict and decide themselves, since they use mathematical models that are constructed by these machine learning algorithms with the help of training data (which is the existing sample data set). For a given complex problem statement, if some kind of prediction is to be made, it is not mandatory to design and write the code for the entire problem, instead just by serving the algorithm with the available information, a mathematical model, or logic may be built by the machine, so as to predict the outcome [1]. Going further, machine learning is broadly segregated into three classes: Unsupervised, Supervised and Reinforcement learning (see Fig. 1).

1.1.1. Supervised learning

In this category, the machine is provided with labeled sample data intended for training it, based on which it would later be predicting outputs. Following this process, the machine is tested for correct and

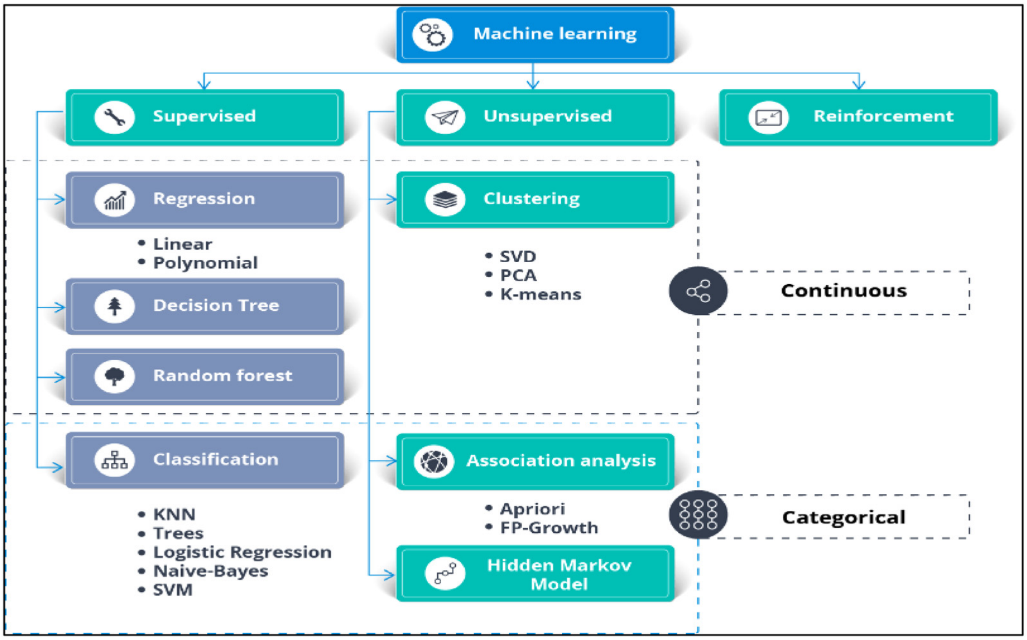


Fig. 1. Classification of Machine Learning Algorithms [2].

exact outputs with some random data. The theory of supervised type of learning is centered on the word 'supervision', where it aims at mapping the data associated with the input to that associated with the output. This method undoubtedly needs a substantial amount of human application to construct the model, but eventually leads to faster performance of an otherwise tedious task. Supervised machine learning is a widely adopted category of Machine learning. This is further classified into Regression algorithms and Classification algorithms [1].

1.1.2. Unsupervised learning

Unsupervised learning enables the machine to learn without any supervision. In unsupervised learning, an unsegregated and unlabeled data set is provided to the machine, and the algorithm is supposed to perform on the data without any supervision. This theory aims at regrouping the input data elements exhibiting like patterns. It is not possible to predict any outcomes in this theory, and the machine attempts to present important understandings based on the enormous amount of data. This is again further divided into Clustering and Association [1].

1.1.3. Reinforcement learning

This theory exists as a feedback-based mechanism, where the learner is rewarded for each correct move, and penalized for the incorrect action. With these prompts, the learners, can mend the system and increase its performance. In this type of learning, the person basically intermingles with the environment and tries to discover more about it [1]. As mentioned earlier, there are two categories under supervised learning, regression, and classification. The algorithms belonging to the regression sub-category are useful when the input variable is related to the output variable in some way, and it is required to predict variables of continuous nature, like stocks, or some population trends. Whereas, classification algorithms are handy when the outcome is of categorical nature, like 'Circle or Triangle, true or False, Right or left, Yes or No', etc.

2. K-nearest neighbor algorithm (K-NN)

K-nearest-neighbor (K-NN) being one of the most essential and effective algorithms for data segregation is capable of becoming the primary choice for implementation especially when the given data is quite ambiguous. This algorithm was invented back in 1951 by Evelyn Fix and Joseph Hodges for discriminant examination when it was relatively challenging to decide the probabilistic densities by parametric estimation [3]. Further in the year 1967, a couple of characteristics belonging to this algorithm were calculated, for example where 'k' = 1 and 'n' tends to infinity then the K-NN classification fallacy or error is limited above by two times the error rate of Bayes [4]. Post establishment of such particular characteristics and properties, research and experimentation followed over long periods to count novel rejection approaches [5], improvements for Bayes error rate [6], procedures relying solely on distance [7], [8], methods for soft computing [9] and other approaches. The K-NN algorithm is positioned under the supervised type learning technique and is considered one of the easiest-to-use algorithms in Machine Learning. Although it is suited for classifying as well as regressing both, it is predominantly utilized for classifying objects. It is an extremely handy algorithm, used to assign any missing value and to re-sample the data [10]. For a given data set, this algorithm predicts the connection among the unseen data and the already existing data and based on that prediction, it imputes the new data to a prevailing category that best matches with it. Therefore, fresh data can be certainly classified by the K-NN algorithm. It sorts the new data point or figure based on arrangements of its neighbors. K-NN can also be referred to as the lazy learner algorithm, as the data set is only stored initially, but the learning process of the training data set does not take place until there is a demand for classification or prediction of the new data set. It is also non-parametric in nature, i.e., in K-NN there does not exist any predetermined method or form of the relationship between the input and output [11].

In Fig. 2, there are two cases, either benign tumor, or malignant tumor. A separate data point has been assigned to be specified as either benign or malignant. In this case, K-NN algorithm can easily help the analyzers in the classification procedure of the new different point from the data set on the basis of the likeness or similarity index of the point with both the existing cases. K-NN can be used when, the data set taken is labeled, and noise free.

2.1. Working of KNN algorithm

The 'K' here in K-NN refers to the count of neighbors of the new data point. Deciding a suitable value for K is the foremost process in

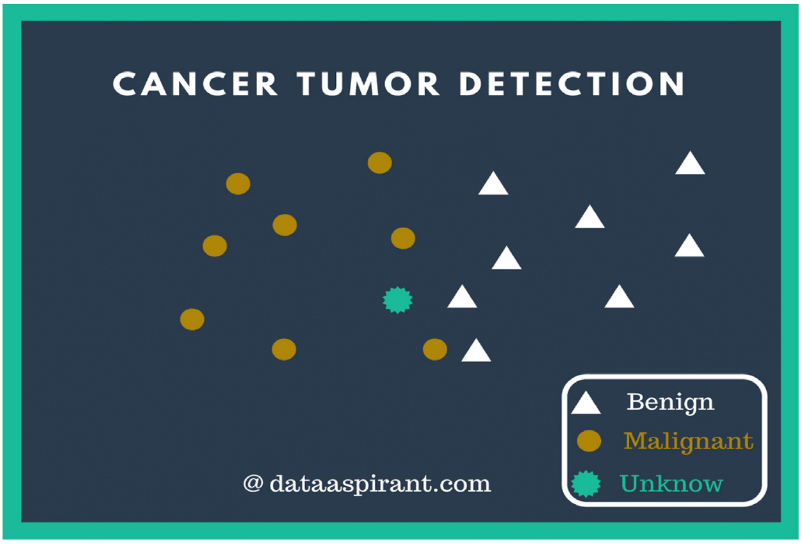


Fig. 2. Plotting of classes and the new data point [12].

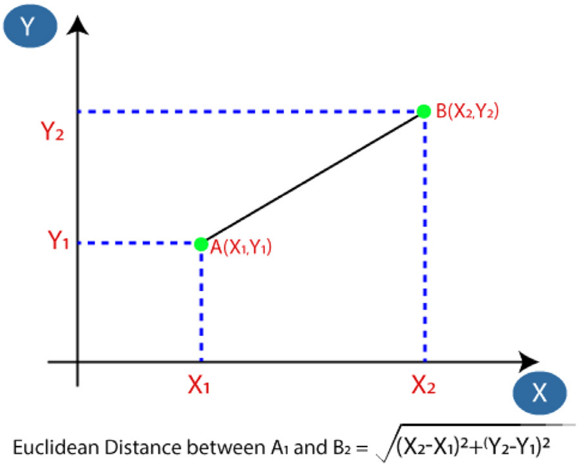

Y
X
Y_2
Y_1
X_1
X_2
$B(X_2,Y_2)$
$A(X_1,Y_1)$
Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Fig. 3. Calculation of Euclidean Distance b/w two points [13].

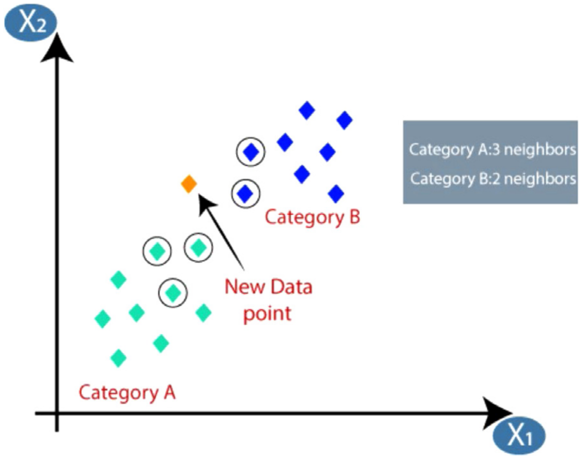


Fig. 4. Classification of new data point based on neighbors [10].

this algorithm. For better accuracy, it is imperative that one chooses the accurate value of K, and this process is called parameter tuning. A very low value of K like 1 or 2 can lead to noisy results, whereas, a very high value can create confusion at times, depending on the data set [10]. There is no fixed value for K, however, one of the standard values that K often assumes is '5' i.e., for the majority voting process, the 5 neighbors closest to the new data point are considered. To avoid mistakes and confusion among two classes of data sets, generally, an odd value of K is suitable. Another formula-based calculation for K can be done through this formula:

$$K = \sqrt{n} \tag{1}$$

And, **n** is the overall count of data points.

Followed by that, the Euclidean type distance of the prevailing points in the data set from the new data point is calculated. In order to do so, it is imperative, that the data set be plotted in a graphical manner. Euclidean distance is calculated as shown in Fig. 3.

Upon calculating the values of the Euclidean distances of all the points from the new data point, one should observe the category to which the majority of the nearest neighbors belong (say, at K = 5), and hence after careful computation impute that class to the data point, assigned for classification. Like in Fig. 4, it can be concluded that the point, goes to class A, since it has 3 (majority) nearest neighbors from that category [10].

2.2. Comparison of logistic regression, Naive Bayes and KNN machine learning algorithms for credit card fraud detection — recent application

2.2.1. Background of the recent work

Credit cards are a widely adopted method for payments these days due to the unstoppable advancement of internet technology. Having said that, banking scams are also way more commonly heard these days than before, which has indelibly affected many segments of the population, be it individuals or institutions. With every advanced security feature, the tricksters have a new way to approach the victims. One of the loopholes that exist in credit card information is the skewness of the data, causing ineffective forecasting of incoming frauds. This research work presented by Fayaz Itoo et al. (2020) [14] makes use of three symmetries of databases for the purpose of the study and apart from that, an under-sampling method is selected on a random basis for datasets that are skewed. The experimental research carried out by Fayaz Itoo et al. (2020) [14] has incorporated three algorithms, k-nearest neighbor, Naive Bayes, and logistic regression. The evaluation metrics that were considered for judgment by them are accuracy, specificity, sensitivity, F-measure, the area under the curve, and precision.

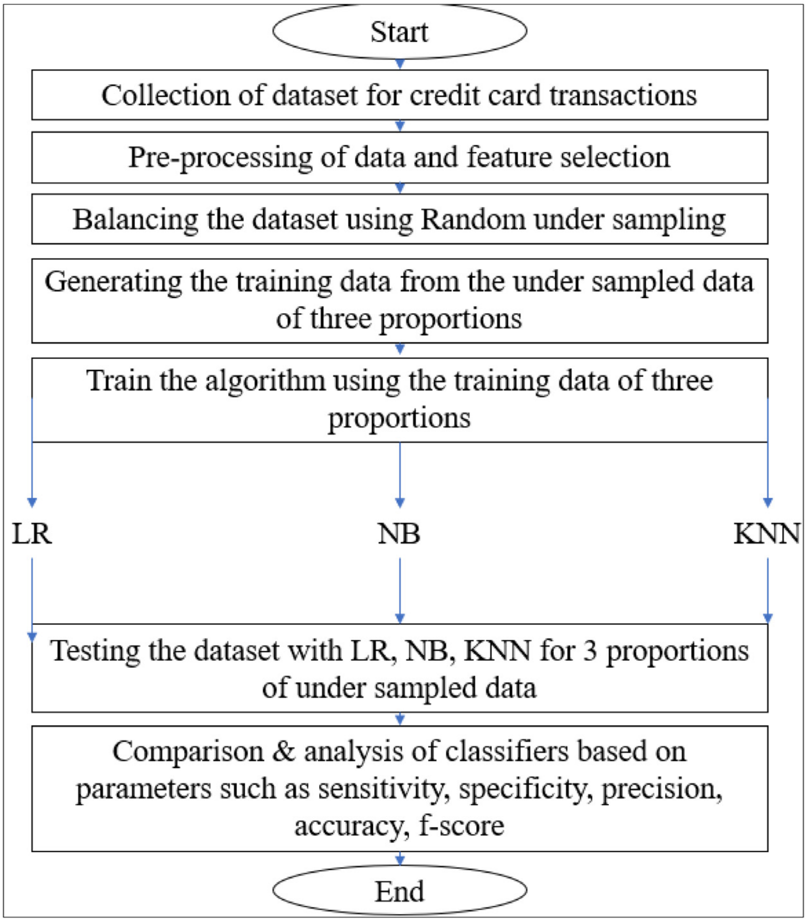

Start
Collection of dataset for credit card transactions
Pre-processing of data and feature selection
Balancing the dataset using Random under sampling
Generating the training data from the under sampled data of three proportions
Train the algorithm using the training data of three proportions
LR
NB
KNN
Testing the dataset with LR, NB, KNN for 3 proportions of under sampled data
Comparison & analysis of classifiers based on parameters such as sensitivity, specificity, precision, accuracy, f-score
End

Fig. 5(a). (a) Flow diagram of research work [14].

Table 1
Division of dataset by ratio 50:50 [14].

Data division	Training data	Resampling method RUS
Fraud	492	344
Non-fraud	284,315	344
Total	284,807	688

Table 2
Division of dataset by ratio 34:66 [14].

Data division	Training data	Resampling method RUS
Fraud	492	341
Non-fraud	284,315	692
Total	284,807	1033

Table 3
Division of dataset by ratio 25:75 [14].

Data division	Training data	Resampling method RUS
Fraud	492	353
Non-fraud	284,315	1024
Total	284,807	1377

Table 4
Testing dataset preparation [14].

Data proportion	Fraud	Non-Fraud	Total
50:50	35	261	296
34:66	137	306	443
25:75	141	450	591

The final result of the research has proved that logistic regression has given more reliable results compared to the other two algorithms used.

2.2.2. Description and results

The flow of the approach followed for this research work has been shown in Fig. 5(a). Splitting the dataset into two segments, the ratios that have been used for the same are 50:50, 34:66, and 25:75, wherein the distribution is from fraud to non-fraud data. The division can be seen in Fig. 5(b). Furthermore, Tables 1, 2, 3 explain more about the dataset division (see Table 4).

From Tables 5, 6, 7; it is observed that the logistic regression algorithm has rendered the most accurate and reliable results as compared to the Naïve Bayes, and K-Nearest Neighbor algorithms. The K-NN algorithm as can be seen from the above figures has shown the poorest performance of all the algorithms, this is because of the small sample training data set, as there is high similarity between the fraud and non-fraud data and hence the algorithm was not efficiently able to classify between the two. [14]

2.3. Advantages of KNN algorithm

■ K-NN algorithm is an easy-to-apply algorithm to problems.

■ K-NN algorithm is tolerant and resistant to the noise prevailing in the data set used for training.

■ It is fast and easy to interpret and effective even if the data set is large enough [10].

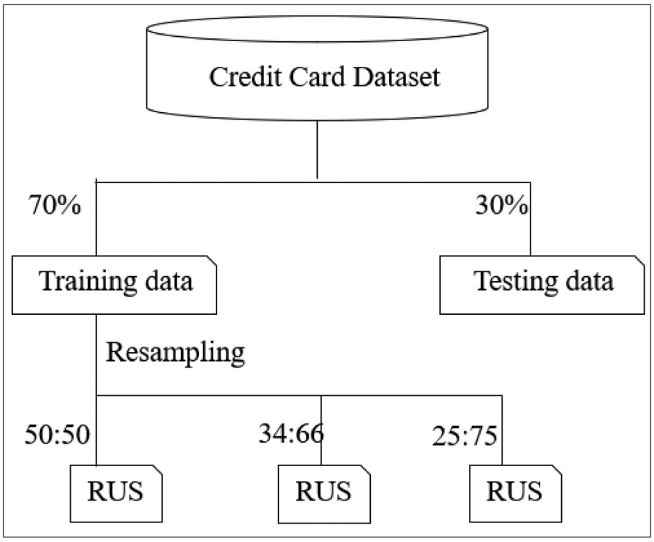


Fig. 5(b). Division of dataset to be used [14].

Table 5
Results obtained after testing from ratio 50:50 [14].

Techniques	Sensitivity	Specificity	Accuracy	Precision	F-measure	AUC
Logistic regression	0.878	0.949	0.912	0.951	0.913	0.914
Naïve Bayes	0.757	0.964	0.854	0.959	0.846	0.860
K-nearest neighbor	0.687	0.669	0.679	0.701	0.694	0.678

Table 6
Results obtained after testing from ratio 34:66 [14].

Techniques	Sensitivity	Specificity	Accuracy	Precision	F-measure	AUC
Logistic regression	0.777	1.0	0.923	1.0	0.875	0.888
Naïve Bayes	0.718	1.0	0.902	1.0	0.836	0.859
K-nearest neighbor	0.477	0.789	0.681	0.544	0.508	0.633

Table 7
Results obtained after testing from ratio 25:75 [14].

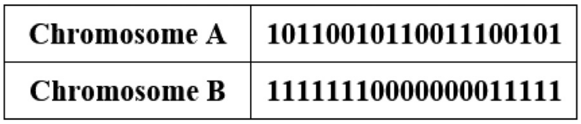
Techniques	Sensitivity	Specificity	Accuracy	Precision	F-measure	AUC
Logistic regression	0.839	0.997	0.959	0.991	0.909	0.918
Naïve Bayes	0.664	0.995	0.915	0.979	0.789	0.829
K-nearest neighbor	0.405	0.861	0.751	0.483	0.441	0.633

2.4. Disadvantages of KNN algorithm

- Deciding a suitable value for K is a complexity, as it drastically changes the results sometimes.
- Since, the requirement is to calculate the Euclidean type distance between each and every data point belonging to the dataset used for training; it leads to high cost of computation [15].

3. Genetic algorithm (GA)

During the 1950s, an English mathematician named Alan Turing presented a machine that was intended to replicate evolutionary theories or principles [16]. The computerized simulations concerning evolution were initiated in 1954 by Nils Aall Barricelli, who was making use of the machines and computers available in Princeton at the Institute for Advanced Study [17,18]. However, his publication was not well recognized by the audience. Later in 1957, a quantitative genetics specialist Alex Fraser from Australia worked and disseminated a set of papers related to simulations of selecting artificially the organisms [19]. Followed by this, evolution-related computer simulations offered by various biologists came into wide existence in the 1960s, and the techniques were published in texts by Fraser and Burnell [20] and Crosby [21], and all the fundamental aspects of Genetic algorithms were covered. Furthermore, a set of papers published by Hans-Joachim Bremermann consisted of a wide variety of solutions to problems undergoing selection, mutation, and recombination related to optimization mainly. Aspects belonging to modern Genetic algorithms were also covered by Bremermann in his research work [22]. Artificial Evolution was not a very well-known technique until Ingo Rechenberg and Hans-paul Schwefel presented their research in the 1960s and 1970s where Rechenberg and his group had competently provided solutions to intricate engineering situations via principles of genetics and evolution [23–27]. An alternate programming method was proposed by Lawrence J. Fogel for evolutionary problems, mainly for generating Artificial Intelligence. Initially, the concept of evolutionary programming made use of finite state machines for environmental predictions, and selection and variation techniques were used for predictive designing. Finally, in the early 1970s, John Holland was the one who could publicize Genetic algorithms through his book Adaptation in Natural and Artificial Systems wherein his work commenced with the analysis of cellular automata, carried out by him and his students personally. The studies and research regarding Genetic algorithms were mainly theoretical until the mid-1980s when in Pittsburgh, Pennsylvania the very first International Conference on Genetic algorithms was conducted. Based on the important biological concepts of natural selection, and heredity, Genetic Algorithms were founded; they are predominantly, search



Chromosome A	10110010110011100101
Chromosome B	11111110000000011111

Fig. 6. Binary encoding [30].

optimization algorithms. These can be called a clever manipulation of random search supported with past information to guide the search towards the domain of improvised performance concerning the space for solutions. Genetic algorithms are predominantly useful for the generation of supreme quality solutions for problems related to search and optimization [28,29]. Genetic Algorithms are a subset of a relatively much larger domain of computation known as Evolutionary Computation. Genetic algorithm is mainly a probability-based optimization algorithm. Similar to genetics from biology, here, the multiple solutions attained undergo mutation and recombination, which eventually gives rise to fresh offspring, followed by the repetition of this process for several generations [30]. Every offspring is assigned a fitness measure, determined by the assessment of its objective function and at the end, the fitter ones have a higher probability of reproducing a more fit generation. This technique guarantees the establishment of fit entities, or solutions in the successive generations, till the final generation is attained.

The creation of offspring follows the principle given below.

1. The characters or entities strive for resources and then breed.
2. The characters with higher fitness scores breed to reproduce offspring.
3. The best genes from the parental chromosomes are carried forward through successive generations.
4. Hence, as each new generation is achieved, it gets better and more suitable for the presiding environment.

3.1. Search space

The entire population is confined to a specific region called the search space. Every entity present here carries a key or solution for a given problem statement. Every chromosome is encoded like a finite length vector of constituents. Upon the selection and creation of the primitive generation, the genetic algorithm leads to the evolution of the group, using the process of selection, crossover, and mutation.

3.2. Selection

In this process, basically the chromosomes with higher fitness scores, are sought, and permitted to produce successive generations, in order to pass on the better and more competitive genes.

3.3. Encoding

The format which a chromosome acquires includes data relevant to the outcome or solution that it represents. One of the most common ways to encode is using a binary string shown in Fig. 6. Each and every chromosome can be encoded through this format. Every bit present in the string contains some fragment of the output solution [30].

3.4. Crossover

In crossing over, two parental chromosomes are selected through the selection process and an arbitrary point for the crossover of genes. Once the crossover takes place, it results in fresh offspring (see Fig. 7).

Table 8
Details of database [31].

Database	Number of classes	Images per class	Size of image
Yale	15	11	243 * 320
ORL	40	10	112 * 92
UMIST	20	24	Variable

Table 9
Parameters for genetic algorithm [31].

Parameters	Values
Population Size	50
Number of generations	100
Crossing probability	0.5
Mutation probability	0.1

3.5. Mutation

In order to prevent untimely convergence of population, random genes are introduced into the newly produced offspring to encourage the variety in the population (see Fig. 8).

3.6. Face recognition based on genetic algorithm optimization — recent application

3.6.1. Background of the recent work

Mourad Moussa et al. (2018) [31], performed research work on face recognition approach based on the well-known principal component analysis and discrete cosine transform methods. To develop a stable, and reliable face recognition operation, it is essential to first look after the feature selection, which is responsible for canceling the unnecessary noise, superfluous data, and various other irrelevant features. However, the development of Genetic algorithm, which is relatively a newer algorithm for feature selection can be brought to use to look after this matter. In order to apply genetic algorithm for the purpose of solving an issue, it demands the coding of effective solutions in measurable bit chains to include the chromosomes hailing from particular points. The end goal is to derive the genetic operators to be used and to develop decent discrimination among the chromosomes. Mourad Moussa et al. (2018) [31], designed a face recognition system by employing Genetic algorithm along with a combination of Discrete Cosine Transform-Principal Component Analysis (DCT-PCA) to be deployed for dimensionality reduction and feature selection to be used on a set of face images of humans. The results presented by Mourad Moussa et al. (2018) [31], demonstrate the efficiency of this method relative to other previous works.

3.6.2. Description and results

Mourad Moussa et al. (2018) [31], made use of three standard schemes named University of Manchester Institute of Science and Technology (UMIST), Olivetti Research Laboratory (ORL), and Yale as described in Table 8 given below, along with the representative faces (Fig. 9) that were employed for testing. The databases on a random basis were used either for training or as testing sets and all expedient arrangements were used for the purpose of research. Average results and observations have been attached. The minimum Euclidean Distance was the parameter for judgment and classification. For this experimental research, MATLAB R2015a was used and the genes to be taken were locked to 30. Various other variables that were considered for research have been shown in Table 9.

The results obtained by Mourad Moussa et al. (2018) [31], are presented in the Table 10 and the approach followed by them has helped in the achievement of this face identification system to reach a recognition rate of 99%. It is evident that this novel approach has led to the betterment of 8% compared to other previous works. Therefore, this Genetic algorithm-based approach has successfully helped in making this face recognition system more efficient and robust, and in the suitable selection of required coefficients (see Fig. 10).

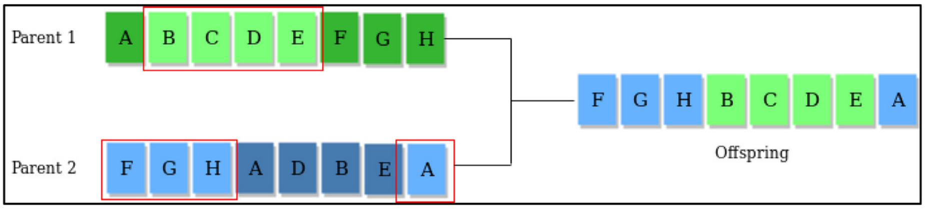


Fig. 7. Crossover operation.

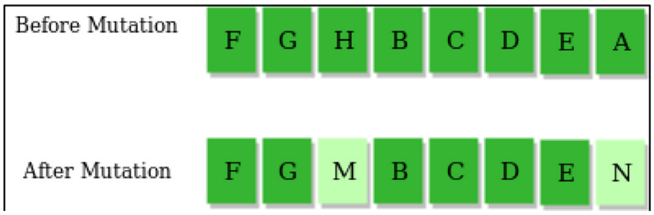


Fig. 8. Mutation operation.

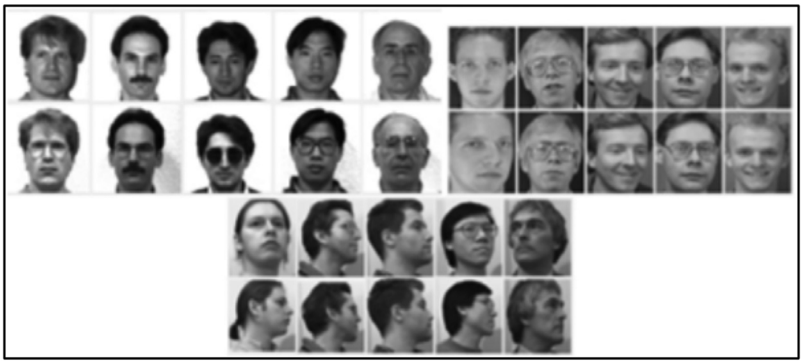


Fig. 9. Sample data [31].

Table 10
Results of the research work (Compared with previous results) [31].

Database	Number of classes	Number of train cases	Number of test cases	Recognition rate of previous works	Recognition rate of our works
ORL	40	3	7	92.5%	92.62%
ORL	20	6	4	97.5%	98.45%
UMIST	20	24	Variable	91.66%	99.4%
YALE	15	5	6	93.33%	96.5%
YALE	15	4	7	95.23%	95.5%

3.7. Advantages of genetic algorithm

- Genetic Algorithm provides a highly robust performance against local maxima or minima.
- They render improvisation over enormous-sized space state.
- They do not weaken upon varying inputs, and noise, as compared to the conventional artificial intelligence systems.
- Genetic algorithms upgrade the distinctive and incessant functions, both.
- This algorithm does not require imitative data or information.
- Far more extensive and optimal in nature as compared to the basic methods [32].

3.8. Drawbacks of genetic algorithm

- One of the potential drawbacks of the genetic algorithm is that it could quite often lead to untimely convergence of the population, due to homogenization of genes. This restrains any type of valuable investigation.
- Although this algorithm does not need as much information regarding the problem statement, it is quite a challenge to design an objective function and achieve the operations.
- Genetic Algorithm is highly time-consuming to apply [33].

4. Support vector machine (SVM) algorithm

Support Vector Machines (SVM) were created by Alexey ya. Chervonenkis and Vladimir N. Vapnik in the year of 1963 [34]. Ever since the discovery of Support Vector Machines, this technique has been widely adopted for use in image, hypertext, and text segregation and classification problems. These algorithms are quite advanced and can be utilized for handwritten text, as well as for protein sorting in biological laboratories. They are used in many other domains, like self-driving cars, chatbots, face recognition, etc. [35] Among one of the highly prevalent supervised type learning algorithms, the Support Vector Machine algorithm is intended for regression and classification problems. SVM algorithm aims at forming the finest suitable decision limit or boundary, known as hyperplane, which separates n-dimensional space

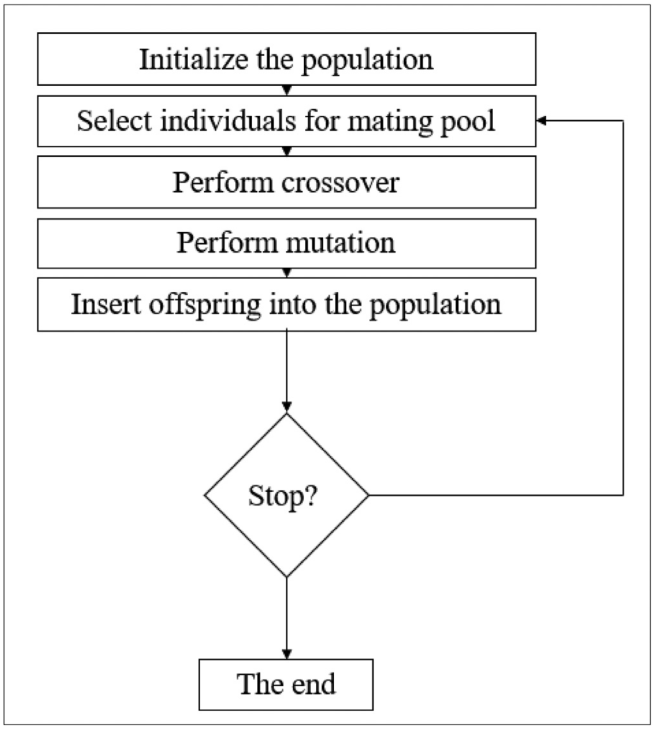

Initialize the population
Select individuals for mating pool
Perform crossover
Perform mutation
Insert offspring into the population
Stop?
The end

Fig. 10. Entire flow of Genetic Algorithm [30].

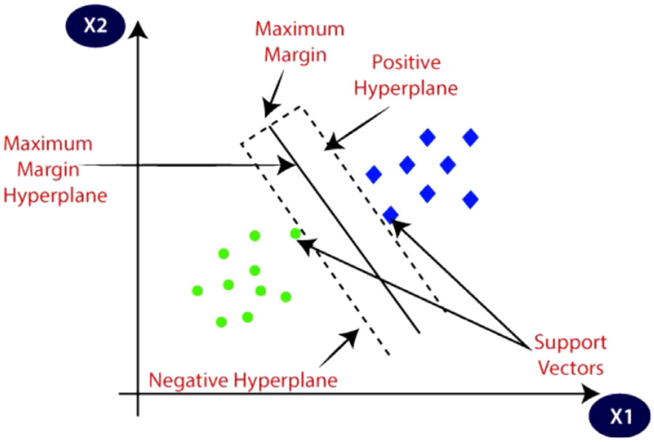


Fig. 11. Illustration of a decision boundary in SVM [36].

into various classes, making it easy to place a different point in the apt category. In SVM algorithm, extreme vector points called Support Vectors are chosen which help in creating an appropriate hyperplane. SVM algorithm finds its applications in detection of faces, classification of some images, categorization of texts, etc. An example is illustrated in Fig. 11 wherein two contrasting categories are classified using SVM algorithm. For instance, a person comes across an image of a strange cat having some similar features as that of a dog. Therefore, in order to create a model that precisely identifies whether it is a dog or a cat, the SVM algorithm is quite beneficial [36].

4.1. Types of SVM

4.1.1. Linear type SVM

Linear type SVM algorithm is useful in cases wherever the data needs to be segregated in a linear fashion, which implies that the data set can be divided into two classes separated by a single straight line.

4.1.2. Non-linear type SVM

Non-Linear type SVM algorithm is useful in cases wherever the data is to be segregated in a non-linear fashion, which implies that the data set cannot be divided into classes using a straight line.

4.2. Hyperplane and support vectors in the SVM algorithm

4.2.1. Hyperplane

Hyperplane of SVM algorithm is defined as the best possible decision boundary out of various possible decision boundaries that accurately classifies the classes in n-dimensional space. Features of the data set determine the dimensions of the hyperplane i.e. if a data set has 2 features it implies a one-dimensional hyperplane whereas 3 features imply a two-dimensional hyperplane. A hyper-plane having maximum margins, which means the distance between two data points is maximum, is preferred.

4.2.2. Support vectors

Support vectors are the nearest data pointers affecting the position of the hyperplane. It is because of their supporting nature towards the hyperplane, that they are termed as supporting vectors.

4.3. Working of SVM algorithm

4.3.1. Linear SVM

SVM algorithm working model can be illustrated using an example. Consider a data set having two different objects (red and yellow), and the two features, say X1 and X2 is allotted. A classifying algorithm that can accurately segregate the pair of coordinates (X1, X2) in either red or in yellow is desirable. Since it is a 2-dimensional space and having two features, so it is rather easy to separate out the two categories simply by means of a straight line but there are many such straight lines possible. Here comes the role of this algorithm that finds the most apt decision line out of all the possible lines or decision boundaries; this most apt decision border is called hyperplane. SVM algorithm locates the support vectors i.e., closest points from the decision boundary within both the classes. SVM algorithm maximizes the distance between hyperplane and vectors to return an optimal solution.

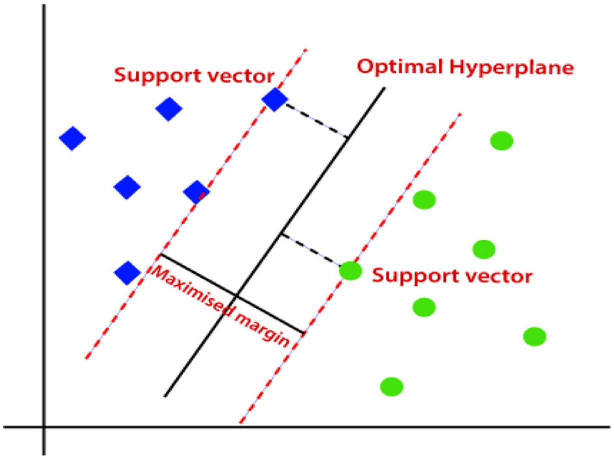


Fig. 12. Implementation of SVM [36].

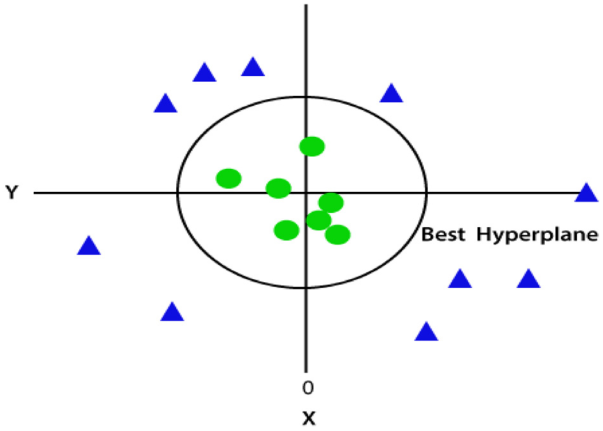


Fig. 13. 3-D Arrangement [36].

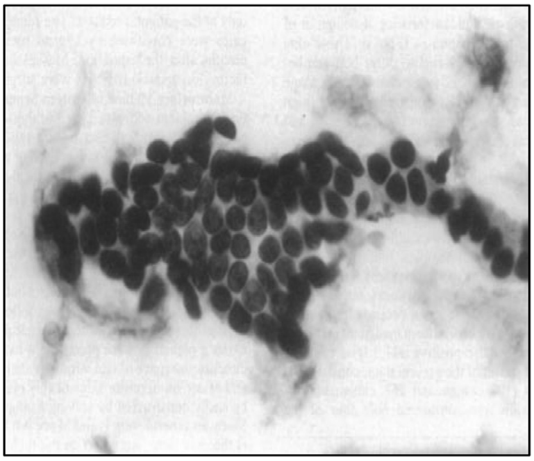


Fig. 14. Sample image of breast mass used for extracting other features [37].

4.3.2. Non-linear SVM

Consider having data that is arranged in a non-linearly fashion. Here one cannot just simply draw a single straight line. Therefore, in order to separate these data points, an additional dimension is needed. For linearly arranged data, just two dimensions were utilized i.e. x and y, but for a non-linearly arranged data, an additional third dimension is required i.e. z which can be simply calculated using the geometric formula ($\mathbf{z} = \mathbf{x}^2 + \mathbf{y}^2$). Since the work involves a 3-dimensional space, hence it resembles a plane that is parallel to the x-axis. On conversion of it into a 2-dimensional space with z = 1, a circle of radius 1 unit is obtained as shown in Fig. 13 (see Fig. 12).

4.4. Use of SVM algorithm for breast cancer detection — recent application

4.4.1. Background of the recent work

Jenni A. M. Sidey-Gibbons et al. (2019) [37] carried out elaborate research on particular machine learning algorithms that can be used for cancer prediction, especially breast cancer. Machine learning algorithms are highly efficient and can be very helpful in medical science for the early detection or forecasting of numerous fatal diseases. In their experimental research predictive designs were made using different algorithms for cancer detection based on matter sampled out of breast mass. The algorithms used for their research work included single-layered artificial neural networks, support vector machine algorithm with a radial basis function kernel, and General Linear Model (GLM) regression. Approximately 456 samples of breast masses were used for evaluation and 227 samples for validation. Before testing the algorithms and models in the validation set for detecting the disease, they were trained using the evaluation samples. In order to compare the performances of the respective algorithmic models, the evaluation metrics that were used by Jenni A. M. Sidey-Gibbons et al. (2019) [37], were, sensitivity, specificity, and accuracy. Post the research performed by them, it was found that the SVM algorithm provided maximum area under the curve and accuracy, when compared to the other two algorithms.

4.4.2. Description and results

The dataset that was employed for their research was the Wisconsin Breast cancer dataset, which is freely accessible via the University of California Irvine Machine Learning Repository. The characteristics of the breast masses that were mainly focused on in this research are the cell nuclei, which underwent the Fine-needle aspiration (FNA), a routine oncological diagnostic procedure. A sample figure of the breast mass is displayed in Fig. 14.

The number of samples used for this research work is in fact the number of instances, which are assigned a unique Identity (ID), and apart from that, various other characteristics are attached to it. The class column displayed in the resulting figure is the diagnosis which is either a benign or malignant tumor, which in turn depends on the Fine-needle aspiration (FNA) if it was cancerous or not. As can be seen in this Table 11, 241 samples turned out to be malignant, whereas 458 of them were benign. Benign samples have a class of two, while malignant, a class of four.

There are nine attributes to test in this experiment shown in Table 11, wherein every trait is judged on a scale of 1 to 10. The closer

Table 11
Attributes of the data set that was used for the experiment [37].

Instance No.	Sample ID	Thickness	Cell shape	Cell size	Adhesion	Epithelial size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
1	1000025	5	1	1	1	2	1	3	1	1	2
2	1002945	5	4	4	5	7	10	3	2	1	2
3	1015425	3	1	1	1	2	2	3	1	1	2
.											
.											
699	897471	4	8	8	5	4	5	10	4	1	4

Table 12
Evaluated performance metrics of the algorithms on cancer dataset [37].

			Actual outcomes		Sensitivity	Specificity	Accuracy
			Benign (0)	Malignant (1)			
Predicted outcomes	GLM	Benign (0) Malignant (1)	148 42	10 67	0.99	0.87	0.95
	SVM	Benign (0) Malignant (1)	1146 4	5 72	0.97	0.94	0.96
	ANN	Benign (0) Malignant (1)	148 2	11 66	0.99	0.86	0.94

the value is to 10, the trait is more malignant in nature, and the closer the value gets to 1, the trait is more benign in nature.

Although all algorithms have a greatly varied style of working, they have rendered a decent level of accuracy, sensitivity, and specificity in their performance and the SVM algorithm has represented the best performance of all with an accuracy of 0.96 shown in Table 12.

4.5. *Advantages of SVM algorithm*

- SVM algorithm is best suited where a clear division is present between the classes.
- SVM shows greater efficiency when it comes to higher-dimensional spaces.
- SVM also works effectively in situations where the number of dimensional spaces exceeds the amount of sample values.
- SVM is relatively efficient in terms of memory, which is a quite desirable feature [36].

4.6. *Disadvantages of SVM algorithm*

- SVM algorithm cannot work suitably for huge-sized data sets.
- SVM fails to perform efficiently in cases where the data set contains huge amount of noise i.e. overlapping target classes, which is practically very often.
- SVM under performs in cases where the numeric value of features of each data point is higher than that of training data samples.
- SVM algorithm has no possible probabilistic explanation for the classification it so carried out [38].

5. Decision tree (DT) algorithm

Decision Tree (DT) algorithm, belonging to the supervised learning class of algorithms is mostly preferred for solving classification problems but either way, it may be used in classifying as well as in regressing cases. It consists of inner nodes representing the structures of the branches, dataset, representing the verdict given by the algorithm, and each leaf node representing an outcome. There are two nodes, first is the decision node, that is used to make a decision and has various branches; and second is the leaf node, which is the output of decision nodes and has no further branches. It owes its name to that of a tree because of its similarity in shape. The root node is a starting point that further expands to various branches making it a tree-like structure. Decision tree simply forks the tree into sub-trees on the basis of answer to question i.e. whether a yes or a no [39].

5.1. *Classification of decision trees*

5.1.1. *Decision tree consisting of a categorical variable*

The Decision Tree having a categorical variable as the target.

Example: - Problem statement having the target variable as "whether on tossing a coin head will appear or not" (see Fig. 15).

5.1.2. *Decision tree consisting of a constant variable*

The Decision Tree having a constant variable as the target.

Example: - Whether a person can repay a loan or not. In case the banks do not have income details, which is a significant variable in this case, then a decision tree could be built for predicting the monthly revenue of a person on the basis of various factors like assets, living standard, occupation, etc. Here the values being predicted are for variables continuous in nature.

5.2. *Decision tree terminologies*

- **Root Node** The initial part of the Decision Tree from where the entire data set starts getting divided further, into various possible sets that are homogeneous in nature.
- **Leaf Node:** The final outcome node beyond which no further segregation of trees is possible.
- **Splitting:** It involves the process of division of the main node further, upon the provided constraints into sub-nodes.
- **Sub Tree:** Splitting up a hierarchy results into a sub tree or branch.
- **Pruning:** This involves the elimination of superfluous branches of the Decision Tree in order to get optimal results. It actually minimizes the tree size without hampering its accuracy. It is of two types, cost complexity, and error reduction pruning.
- **Child and Parent node:** It is the base node also called the parental node whereas the remaining nodes are simply called child nodes [40].

5.3. *Attribute selection measures*

Attribute selection measure (ASM) involves the collection of the optimum attribute concerning the source node as well as for the sub-nodes. The two major practices for ASM are:

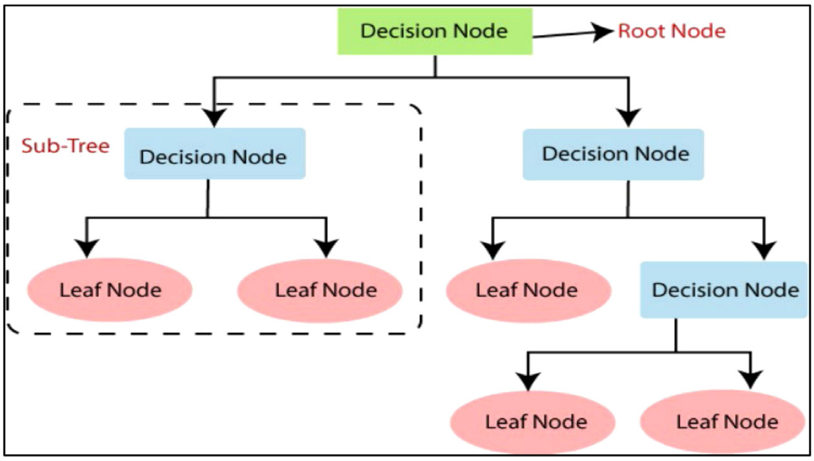


Fig. 15. Schematic diagram of a decision tree [40].

5.3.1. *Information gain*

Information gain as the name suggests calculates the amount of information that is provided by a feature regarding the class. The node is split and the tree is constructed on the basis of information gain values. The DT algorithm maximizes the information gain function besides splitting the node/attribute possessing the greatest amount of information, first. Information gain is mathematically denoted as:

$$Information\ Gain = Entropy\,(s) - [(Weighted\,Avg.) * Entropy(Every\ Feature)] \tag{2}$$

Entropy Entropy measures the amount of arbitrariness present in the dataset. It is represented as follows:

$$Entropy\,(s) = -P\,(yes)\,log_2 P\,(yes) - P(no)\;log_2 P(no) \tag{3}$$

and, **S** denotes the 'samples number', P(yes) denotes the 'Chance of getting yes' and **P(no)** denotes the 'chance of obtaining no' (see Fig. 16).

5.3.2. *Gini index*

Gini index measures the impurity or purity used during the creation of a decision tree algorithm. Small Gini index attributes are preferred by the decision tree algorithm over the attributes possessing larger Gini index, while taking the decision.

The calculation of Gini index can be performed using the expression given as follows:

$$\text{Gini index} = 1 - \sum_j p_j^2 \tag{4}$$

5.4. *Steps for making a decision tree*

- The root node, say X, that contains the entire data set is considered the starting point of the tree.
- By using ASM look for the best matching characteristic from the data set.
- Split X into subsections comprising values having the finest possible qualities.
- Develop the decision tree nodes only using the idyllic attribute.
- Repetitively keep developing unique decision tree nodes using the available subsets of the data set created in '3'.
- Keep continuing this process until it reaches a point where it is impossible to further get sub-nodes.
- Such ultimate end resultant node is known as a leaf node [41].

5.5. *Liver disease prediction using different decision tree techniques-recent application*

5.5.1. *Background of the recent work*

Liver-related diseases are one of the most fatal diseases that can take a toll on human lives. The discovery of any technology that could predict such diseases in their early stages could be very helpful for saving human lives. Nazmun Nahar et al. (2018) [43], performed research in this field by testing and comparing different types of decision tree algorithms to help in the prediction of liver disease in their early stages. Decision tree algorithms are employed in numerous disciplines, notably in the domain of medical science. Nazmun Nahar et al. (2018) [43], adopted the dataset consisting of properties such as direct bilirubin, total bilirubin, gender, age factor, total proteins, etc. The decision tree techniques that were tested under this experimental research were Logistic Model Tree (LMT), Java48 (J48), Random tree, Decision Stump, Random Forest, Hoeffding Tree, and Reduced Error Pruning Tree (REPTree). Their experimental study has showed that Decision Stump rendered the most reliable and precise results.

5.5.2. *Description and results*

The primary goal of this research was to discover whether a patient is affected by liver-related diseases or not, using various decision tree algorithms. The different techniques were tested and compared on the basis of numerous evaluation metrics such as accuracy, mean absolute error, Kappa Statistics, Run time, precision, recall, etc. Nazmun Nahar et al. (2018) [43], used Weka, which is a robust data mining tool to test the precision of various algorithms by applying them on various datasets (see Fig. 17).

Table 13 is the dataset that was used by Nazmun Nahar et al. (2018) [43], for the purpose of their experimental research on the different types of decision tree algorithms for early diagnosis of liver diseases (see Fig. 18, Table 14).

From the above plots, it can be seen that the comparison of performance conducted between the various algorithms on the liver dataset is pictured. It can be derived from the observations that the Decision stump has outperformed the rest of the techniques by gaining an accuracy of 70.67%. Other superior decision trees such as Classification and Regression Trees (CART) could also be used in the future for research and study related to liver disease diagnosis. From the above recent work, it is evident that decision tree algorithms can be extremely handy for the early diagnosis of diseases by providing accurate and reliable results [43].

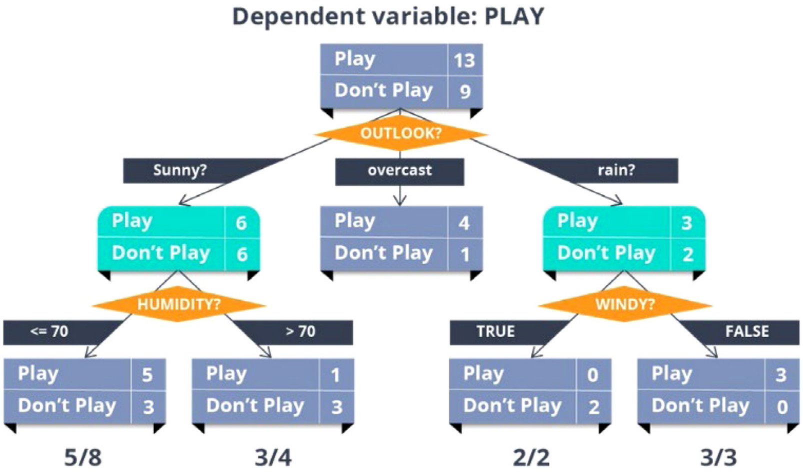


Fig. 16. Illustration of decision tree [42].

Table 13
Dataset used for testing of Decision tree algorithm [43].

No.	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	A/G ratio	Class
1	65	Female	0.7	0.1	187.0	16.0	18.0	6.8	3.3	0.9	Yes
2	62	Male	10.9	5.5	699.0	64.0	100.0	7.5	3.2	0.74	Yes
3	62	Male	7.3	4.1	490.0	60.0	68.0	7.0	3.3	0.89	Yes
4	58	Male	1	0.4	182.0	14.0	20.0	6.8	3.4	1.0	Yes
5	72	Male	3.9	2	195.0	27.0	59.0	7.3	2.4	0.4	Yes
6	46	Male	1.8	0.7	208.0	19.0	14.0	7.6	4.4	1.3	Yes
7	26	Female	0.9	0.2	154.0	16.0	12.0	7.0	3.5	1.0	Yes
8	29	Female	0.9	0.3	202.0	14.0	11.0	6.7	3.6	1.1	Yes
9	17	Male	0.9	0.3	202.0	22.0	19.0	7.4	4.1	1.2	No
10	55	Male	0.7	0.2	290.0	53.0	58.0	6.8	3.4	1.0	Yes
11	57	Male	0.6	0.1	210.0	51.0	59.0	5.9	2.7	0.8	Yes
12	72	Male	2.7	1.3	260.0	31.0	56.0	7.4	3.0	0.6	Yes
13	64	Male	0.9	0.3	310.0	61.0	58.0	7.0	3.4	0.9	No
14	74	Female	1.1	0.4	214.0	22.0	30.0	8.1	4.1	1.0	Yes
15	61	Male	0.7	0.2	145.0	53.0	41.0	5.8	2.7	0.87	Yes
16	25	Male	0.6	0.1	183.0	91.0	53.0	5.5	2.3	0.7	No
17	38	Male	1.8	0.8	342.0	168.0	441.0	7.6	4.4	1.3	Yes
18	33	Male	1.6	0.5	165.0	15.0	23.0	7.3	3.5	0.92	No
19	40	Female	0.9	0.3	293.0	232.0	245.0	6.8	3.1	0.8	Yes
20	40	Female	0.9	0.3	293.0	232.0	245.0	6.8	3.1	0.8	Yes
21	51	Male	2.2	1	610.0	17.0	28.0	7.3	2.6	0.55	Yes
22	51	Male	2.9	1.3	482.0	22.0	34.0	7.0	2.4	0.5	Yes
23	62	Male	6.8	3	542.0	116.0	66.0	6.4	3.1	0.9	Yes
24	40	Male	1.9	1	231.0	16.0	55.0	4.3	1.6	0.6	Yes
25	63	Male	0.9	0.2	194.0	52.0	45.0	6.0	3.9	1.85	No
26	34	Male	4.1	2	289.0	875.0	731.0	5.0	2.7	1.1	Yes
27	34	Male	4.1	2	289.0	875.0	731.0	5.0	2.7	1.1	No
28	34	Male	6.2	3	240.0	168.0	850.0	7.2	4.0	1.2	Yes
29	20	Male	1.1	0.5	128.0	20.0	30.0	3.9	1.9	0.95	No
30	84	Female	0.7	0.2	188.0	13.0	21.0	6.0	3.2	1.1	No
31	57	Male	4	1.9	190.0	45.0	111.0	5.2	1.5	0.4	Yes
32	52	Male	0.9	0.2	156.0	35.0	44.0	4.9	2.9	1.4	Yes
33	57	Male	1	0.3	187.0	19.0	23.0	5.2	2.9	1.2	No
34	38	Female	2.6	1.2	410.0	59.0	57.0	5.6	3.0	0.8	No
35	38	Female	2.6	1.2	410.0	59.0	57.0	5.6	3.0	0.8	No
36	30	Male	1.3	0.4	482.0	102.0	80.0	6.9	3.3	0.9	Yes

Table 14
Evaluated performance metrics of decision tree algorithms on liver dataset [43].

Techniques	Tree size	ACC (%)	MAE	PRE	REC	FME	Kappa statistics	Time
J48	65	65.49	0.3678	0.657	0.651	0.654	0.158	0.11
LMT	1	69.47	0.4116	0.695	0.632	0.628	0.065	0.88
Random Forest		69.30	0.3464	0.693	0.667	0.674	0.186	0.5
Random Tree	267	66.55	0.3382	0.666	0.662	0.663	0.183	0.01
REPTree	27	66.13	0.3800	0.691	0.630	0.629	0.067	0.03
Decision Stump	Single level	70.67	0.4392	0.707	0.499	0.585	0.379	0.01
Hoeffding Tree	1	69.75	0.4091	0.700	0.634	0.619	0.0501	0.12

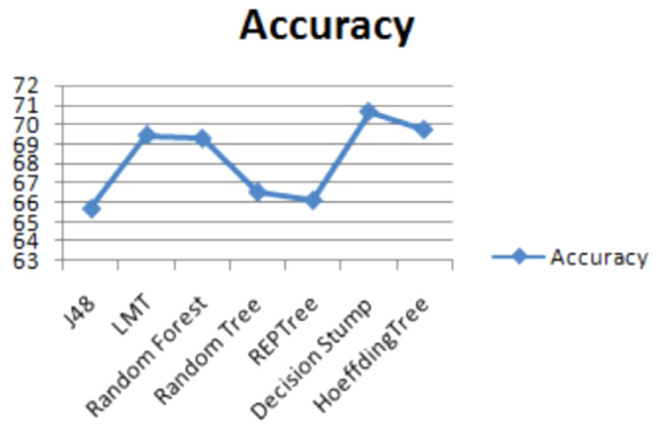


Fig. 17. Accuracy plot of algorithms on liver Dataset [43].

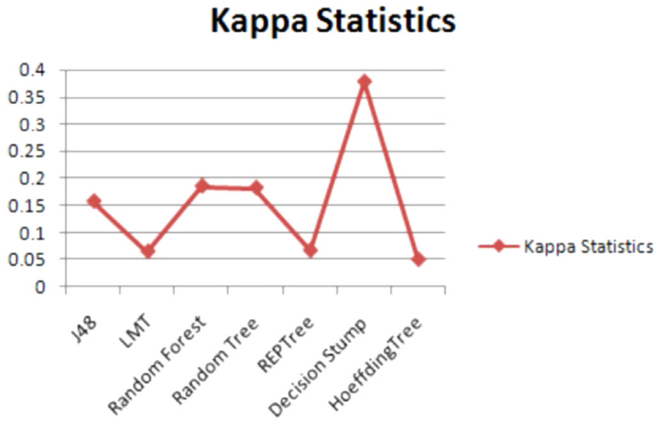


Fig. 18. Kappa statistics plot of algorithms on liver Dataset [43].

5.6. Advantages of decision tree algorithm

- **Possesses very low complexity** – This algorithm is quite simple to comprehend, and is not dependent on any specialized knowledge related to statistics for its interpretation.
- **Useful in Data exploration -** It **c**an also be used in the stages of data exploration as decision tree algorithm proves to be one of the quickest algorithms in creating or identifying new features.
- **Less cleaning of data required -** It comparatively requires fewer data cleaning steps and is not influenced by values and data that is missing.
- **No data type constraint -** It can flexibly handle numerical as well as variables having categorical nature.
- **Non-Parametric Method -** Decision tree uses a non-parametric method, which implies making no assumptions regarding the spatial distribution.

5.7. Drawbacks of decision tree algorithm

- **Overfitting -** Overfitting is among the major practical problem affecting the decision tree model. However, by pruning and setting model parameter constraints the problems of overfitting can be ebbed.
- **Not suitable for variables continuous in nature** - The decision tree loses some of the valuable information while categorizing variables in various categories [44].

6. Long Short-Term Memory (LSTM) algorithm

Due to backpropagation with real-time recurrent learning or time, the error-incorporated signals running rearward in time are likely to disappear or blow up; the temporal shifts of the error incorporated signal to a great extent relies on the weight sizes. In case of blowing up, the weights are quite likely to start oscillating and in case of disappearance, either the time consumed to learn bridging longer time lags is out of bounds, or in the worst case it does not work [45]. As a remedy, the Long Short-Term Memory (LSTM) algorithm, a novel type of recurrent neural network came into existence in 1991, developed by Sepp Hochreiter and Jurgen Schmidhuber to outperform the existing systems and overcome the error backpropagation issues discussed above. The primary version of this Long short-term memory algorithm only consisted of cells, input, and output gates [46]. This algorithm is capable of bridging time breaks in excess of steps even when the sequences being used for input are incompressible or noisy in nature while preventing losses of short time break abilities [47]. Long Short-Term Memory, designed by Hochreiter & Schmidhuber is a special case recurrent neural network (RNN) that is well equipped to handle long term dependencies by default. In LSTM algorithm, the input of a current step is the output of the previous step, thereby solving the issues of long-term dependencies of RNN where the RNN give precise predictions on recent information but are incapable of predicting data stored in long term memory. However, with the increasing gap length, the efficiency of RNN decreases. Some of the major applications of LSTM are captioning images, generation of handwriting chatbots answering questions, and various others [48] (see Fig. 19).

6.1. Structure of LSTM

LSTM structure has been depicted below, consisting of four neural networks and various memory blocks known as the cells. The gates perform memory manipulations on the data stored in cells. The gates are of three types.

6.1.1. Forget gate

The information that is not needed anymore is removed from the cell using the forget gate. The input at a specific time i.e. x_t and the output of the previous cell h_{t-1} are multiplied by the weighted matrices and further addition of bias. To get a binary output the resultant is passes through an activation function. The information in the cell state is then retained if the output is ‘1’ and erased if the output is ‘0’.

6.1.2. Input gate

It performs the function of adding vital information in a cell state. The information is processed through a sigmoid function and the values to be retained are filtered. Next step involves vector creation using the function tanh which gives an output ranging from −1 to + 1, containing all possible values from h_{t-1} and x_t. Lastly, vector values and sigmoid function filtered results are multiplied to derive useful results.

6.1.3. Output gate

It declares the output based on data stored in the current cell state. Firstly, a vector is created with the help of the function tanh for cell values. Next step involves regulation of the information using sigmoid function and filtering of the values that are to be retained. Lastly, the product of the regulated values and vector values is sent as an output, which acts as an input for next cell [48].

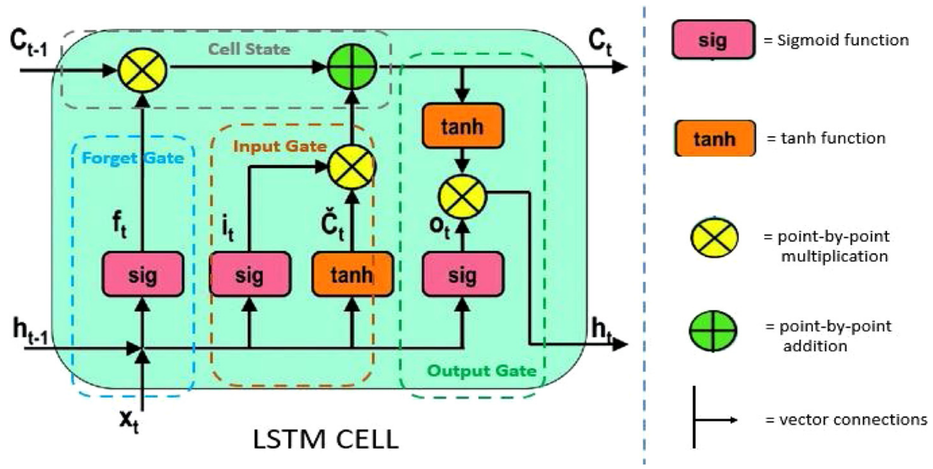


Fig. 19. Structure of LSTM [49].

6.2. LSTM working

The foremost stage requires a decision regarding the removal of unnecessary information from the cell state. Such decisions are resolved by the 'forget gate layer', which is one of the sigmoid layers [50]. While decision-making, x_t and h_{t-1} are considered and the outcomes for all the numbers belonging to the cell C_{t-1}, could be any number ranging from 0 to 1. In case, the output is '1', it indicates that the information has to be saved, where as a '0' indicates that the information needs to be discarded.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{5}$$

Followed by that, one now has to plan as to what information needs to be stored in the cells. This process constitutes two parts. Firstly, the gate layer used for input, that is also a sigmoid coat layer, resolves the values to be updated. Secondly, a new character's vector i.e., t is generated by a tanh layer, intended for the purpose of addition, in this state.

$$i_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_t) \tag{6}$$

$$\tilde{c}_t = \tan h\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \tag{7}$$

After all the planning and decision-making has been done, the execution takes place in this step. Cell state C_{t-1}, gets updated to C_t. Now in order to forget the information as planned earlier, f_t is multiplied with the previous state, followed by the addition of ($\mathbf{i_t}$ * $\widetilde{C}_t$). This value thus acquired is the new character's value, which is scaled by a limit in accordance with the decision that was made to update the cell state value.

$$C_t = f_t\, \tilde{c}^*_{t-1} + i_t * C_t \tag{8}$$

In conclusion, now it is important to plan the output, which is decided on the basis of the cell state, however, it is going to be a filtered output. So, first, a sigmoid layer chooses the portion from the cell state that needs to be presented as the output. Following that, the state of this cell is passed along tanh (in order to confine the outcomes from −1 to 1) then later it can be increased by multiplying it along with the outcome of sigmoid gate layer, to obtain the exact output as decided.

$$o_t = \sigma(W_o\left[h_{t-1}, x_t\right] + b_o) \tag{9}$$

$$h_t = o_t * \tan h\,(C_t) \tag{10}$$

6.3. LSTM- RNN model for prediction of electric load requirement - Recent application

6.3.1. Background of the recent work

Since the evolution of smart powerhouses, the estimation, and prediction of required electricity load is a matter of extreme importance, mainly because the companies and associations concerned with power and electricity, can do sounder planning and scheduling of the loads and cut down the unnecessary power generation. Salah Bouktif et al. (2018) [51], performed experimental research on the usage of LSTM algorithm model for electric load forecasting using feature selection and genetic algorithm. They aimed to create an LSTM based model to design predictive models for load planning and scheduling. Many non-linear and linear algorithms underwent training so that the most appropriate one could be made the base, adopting the most suitable parameters and lastly making use of the genetic algorithm to determine the optimal and suitable time lags and the layers to be used by the LSTM network. The France metropolitan's electricity expenditure data was used for research and analysis. They proved via their experimental research that the LSTM model rendered highly accurate results when compared to machine learning models optimized with hyper-parameter tuning. Their results proved that the LSTM network by employing restricted time-lagged characteristics, gained all traits of the complex time series while showing diminished root mean square error and mean absolute error for a large metropolitan space regarding the prediction and forecasting.

6.3.2. Description and results

Salah Bouktif et al. (2018) [51], developed a model which uses wrapper and heterogeneous system, reasonable time lag and layers for LSTM model, and finally, genetic algorithm enabling them to have a check on overfitting and achieve more precise and reliable prediction. They gathered large datasets for a metropolitan space covering a time span of about 9 years at a 30 min definition using which they trained a system comprising LSTM-RNN to predict the average electricity load requirements. The evaluation metrics that they used for analysis were coefficient of variation, the root mean squared error, and the mean absolute error.

$$CV\,(RMSE)\% = \sqrt{\frac{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\bar{y}}} \tag{11}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{12}$$

$$MAE = \sqrt{\frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{n}} \tag{13}$$

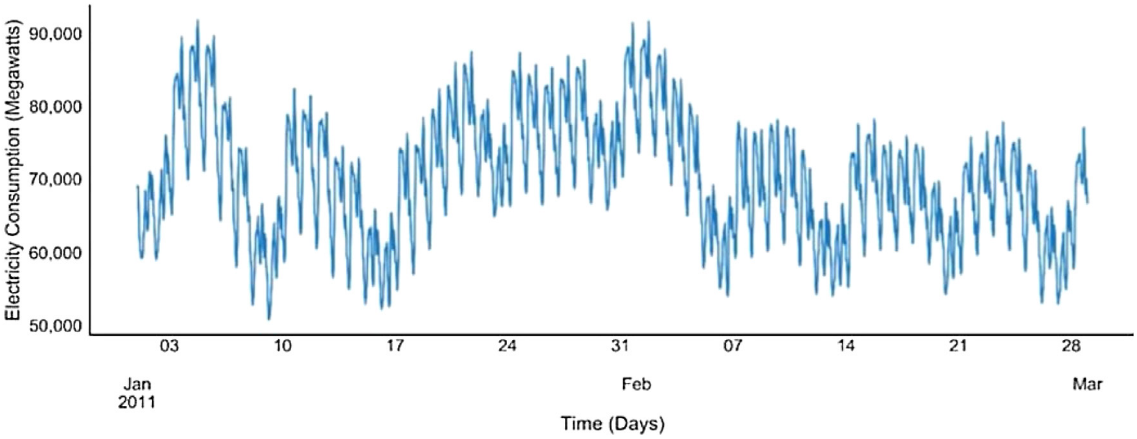


Fig. 20. Electricity load (France Metropolitan) vs. time (January–February 2011) [51].

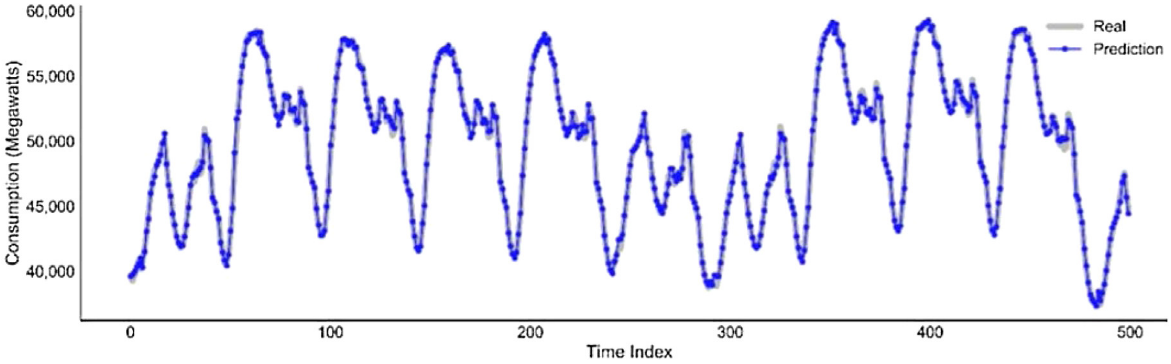


Fig. 21. Actual vs. predicted forecast by the LSTM model [51].

Table 15
Performance metrics of other machine learning models on test set [51].

Model	RMSE	CV (RMSE)	MAE
Linear Regression	847.62	1.55	630.76
Ridge	877.35	1.60	655.70
K-Nearest Neighbor	1655.70	3.02	1239.35
Random Forest	539.08	0.98	370.09
Gradient Boosting	1021.55	1.86	746.24
Neural Network	2741.91	5.01	2180.89
Extra Trees	466.88	0.85	322.04

Table 16
Performance metrics of LSTM model on test set [51].

Metrics	LSTM metrics 30 lags	LSTM metrics optimal time lags	Extra tree model metrics	Error reduction (%)
RMSE	353.38	341.40	428.01	20.3
CV (RMSE)	0.643	0.622	0.78	20.3
MAE	263.14	249.53	292.49	14.9

As part of their experimental research, the designed LSTM-RNN system was compared with the machine learning standard, and the designed model rendered the best results amongst various non-linear and linear models. Given below are the relevant plots and results related to the LSTM algorithm.

The plot depicted above shows the electricity consumption in Mega Watt ranging from January to February in 2011.

Tables 15, 16 show the performance metrics of all the machine learning models including the LSTM model on the test set, seeing which it can be derived that the LSTM model is superior to other models in this case, as the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) are comparatively quite small in case of LSTM. Furthermore, the plot shown in Fig. 20, represents the difference between the real and predicted loads by the LSTM model. The research performed by Salah Bouktif et al. (2018) [51], shows that the LSTM model has provided highly accurate results for the forecasting of the electric load.

6.4. Advantages of LSTM algorithm

- Huge time lags can be bridged by constant error back propagation within the memory cell itself
- LSTMs prove to be robust when it comes to vanishing gradients.
- LSTMs are capable of handling long-term sequential dependencies.
- LSTMs have no parameter fine-tuning
- LSTMs have memory for a longer duration of time.
- LSTMs show high accuracy in terms of prediction.

6.5. Drawbacks of LSTM algorithm

- LSTMs fail to completely solve vanishing gradient problems as the cell has become more complex.
- LSTMs are resource and time-intensive when it comes to training i.e., they require very high memory bandwidth. Therefore, inefficient in terms of hardware.
- With the increasing demand for data mining, there is a quest for models having longer storage time.
- Random weight initialization affects LSTMs making them similar to a feed-forward neural net.
- Problems of fitting which fail to get corrected even by dropout algorithm.

Table 17
Details of the dataset used [52].

No.	Feature title	Variable data type	Feature categorization
1	GP	Continuous	0–4
2	GPA	Continuous	0–4
3	Hometown	Categorical	1: City close from campus 0: City far from campus
4	Type of school	Categorical	1: Public School 0: Private School
5	Major	Categorical	1: Computer/Informatics 2: Science Major 3: Others
6	Parents' Job	Categorical	1: Civil Servant 2: Employee 3: Entrepreneur 4: Farmer/Fisherman 5: Others
7	Active	Categorical	1: Active 0: Others

Table 18
Accuracy before testing [52].

Algorithm	Result	Accuracy
KNN	K = 3	94.50%
SVM	Value C = 1	95.09%
Decision Tree	Cp = 0.6689113	95.65%

Table 19
Confusion matrix for KNN [52].

Prediction	Reference	
	Active	Non-active
Active	309	14
Non-Active	7	52

Table 20
Confusion matrix for SVM [52].

Prediction	Reference	
	Active	Non-active
Active	311	13
Non-Active	5	53

Table 21
Confusion matrix for Decision Tree [52].

Prediction	Reference	
	Active	Non-active
Active	308	18
Non-Active	4	48

7. Quantitative comparison of the algorithms

7.1. *Case study presented for comparison of K-NN, SVM and decision tree algorithm*

For the purpose of quantitative analysis, the research paper titled, "Comparative Study of KNN, SVM and Decision Tree Algorithm for Student's Performance Prediction" written by Slamet Wiyono et al. published in 2020, has been reviewed, wherein a real-time dataset, consisting of 6 different variables was adopted by them for carrying out the exhaustive research [52]. Their research work was a continuation of the various works published in the past for student performance prediction based on a couple of other ML algorithms. The platform used for their analysis was R studio, and they provided reliable and credible results after rigorously carrying out several processes like data collection, pre-processing of the data, building of robust models after training, validating and testing the models on different algorithms, and finally comparing and evaluating them, on a quantitative note [53]. The six variables involved in the dataset, were the grade point, grade point average, hometown, majors, type of school, parents' work, and student performance as shown in Table 17. They resorted to pre-processing of data in order to eliminate certain glitches related to missing data point values, or different attributes, for instance, to make the models further robust and reliable. One of the most common ways to conduct pre-processing is to split the given dataset into training and testing data, out of which the former is used to train and build the model, whereas the latter is required to test the model on grounds of accuracy in prediction. Slamet Wiyono et al. has used a similar approach to compute and execute this research work, and finally compared the three models based on different performance parameters for predicting the students' performance [53–56].

As discussed above, before processing the data, it was split into training and testing data in a ratio of 75:25, wherein it was decided that 1148 samples would be used, with 6 predictors, and 2 classes, supported by cross-validation of 10 folds, repeated thrice. The model built by them has been displayed in Table 18, after which the model was tested using the testing data, and the resulting confusion matrices have been displayed in Tables 19, 20, 21 for KNN, SVM, and decision tree algorithm respectively [57,58]. Before testing these models Table 18, it was seen that the KNN algorithm renders the best performance when k = 3, and has an accuracy of 94.5%, whereas the SVM algorithm provides an accuracy of 95.09% with C = 1, and finally the decision tree algorithm shows an accuracy of 95.65% with cp = 0.6689113. Out of these observations, it was concluded that the Decision tree should be the best model for student performance prediction, however after testing the models, it turned out that the SVM model is indeed the aptest for prediction here. As it can be observed from the pictures displayed, SVM was able to predict 311 active and 53 non-active students, whereas KNN predicted 309 active and 52 non-active students, and lastly, the decision tree algorithm could predict 308 active and 48 non-active students. Therefore, without testing, the Decision tree was the best model; however, post-testing it was concluded that the SVM model can achieve higher predictive accuracy out of the three discussed models, Table 22, Fig. 21. Even though the confusion matrices of the algorithms may show similar predictive accuracy percentages for two or more algorithms, when it was deeply analyzed and calculated, it was corroborated that the SVM model should be considered the best possible method out of the three models for student performance prediction here where it is accurate to a scale of 95% followed by the KNN model with an observed accuracy of 94.5%, and lastly by the decision tree model possessing an accuracy of 93% [59]. Furthermore, upon calculation of other parameters like the specificity, precision,

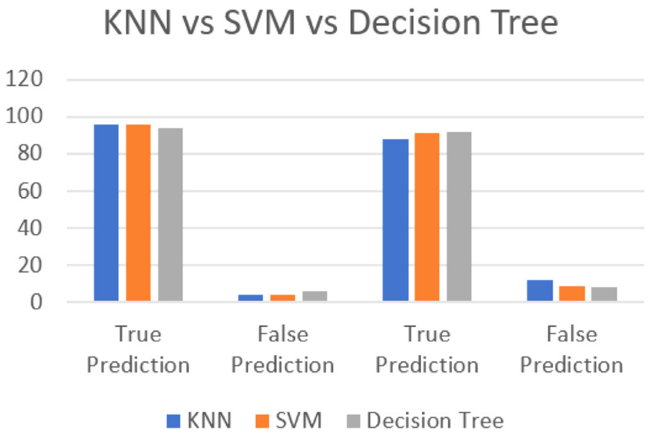


Fig. 22. Comparison of testing accuracy [52].

Table 22
Comparison of confusion matrices [52].

Prediction		KNN	SVM	Decision Tree
Active	True	96%	96%	94%
	False	4%	4%	6%
Non-Active	True	88%	91%	92%
	False	12%	9%	8%

Table 23
Quantitative comparison of performance of various ML algorithms.

Parameter	KNN	SVM	Decision Tree
Accuracy	94.5%	95%	93%
Sensitivity	95%	96.34%	94.75%
Specificity	98.32%	97.8%	97.3%
Precision	97.09%	98.5%	97.76%

and sensitivity, on a collective note, it is safe to conclude that SVM algorithm renders the best performance compared to the other two algorithms (see Fig. 22).

Table 23 presents the summarized quantitative comparison of the results of this case study based on the research paper written by Slamet Wiyono et al. which includes three algorithms KNN, SVM and Decision tree algorithms; used for the prediction of students' performance.

7.2. Case study presented for comparison of genetic algorithm and LSTM algorithm

For a quantitative comparison and analysis of the remaining two algorithms i.e., the Genetic algorithm and the Long Short-Term memory algorithm, the paper titled "A Method Based on GA, CNN and LSTM for Daily Tourist Flow Prediction at Scenic Spots" written by Wenxing Lu et al. [60], published in the year 2020, has been reviewed and given below is the case study from this paper which helps to compare the algorithms better. This paper authored by Wenxing Lu et al. aimed at creating a model to predict the tourist outpour so that the common picturesque and tourist spots can be smoothly maintained and operated. Nonetheless, since any model cannot solely make accurate predictions because of the widely fluctuating data, the authors of this paper worked on a model which uses Convolution Neural Networks (CNN), along with a Deep Learning algorithm i.e., Long Short-Term Memory algorithm and later optimized by the Genetic Algorithm, to predict the day-to-day crowd of a place called Huangshan in China. As part of their research implementation, they primarily formed continuous feature maps from various sorts of data types like those of meteorological, network search, etc. Followed by this is the vector extraction by the CNN, and after this successful extraction, the derived vectors are fed to the LSTM network for prediction of the time-series data. Conventionally, the dataset being used undergoes preprocessing and normalization prior to the prediction stage. The model designed is quantitatively compared in terms of its performance without Genetic algorithm optimization and with genetic algorithm optimization using the common performance parameters, like the mean absolute error (MAE), mean absolute percentage error (MAPE), index of agreement (IA), and personal correlation coefficient. Post the fair comparison between the LSTM-CNN, GA-LSTM-CNN, LSTM and the GA model carried out by the authors, the model incorporating the Genetic algorithm (GA-LSTM-CNN) has outperformed the one without the Genetic algorithm (LSTM-CNN) by approximately 8.22%. However, if the algorithms are solely and individually considered for performance comparison, the LSTM algorithm has rendered a much better performance as compared to that of Genetic algorithm, shown in the tables below.

Table 24 describes the different types of data and their respective features that were gathered and considered by Wenxing Lu et al. to consolidate this reliable and wholesome piece of research.

Table 25 shows a comparison of the performances of the different models that were used to predict the tourist inflow on a day-to-day basis in the city named Huangshan located in China. As it can be seen from the table, if the algorithms are considered for individual performance, LSTM has outperformed Genetic algorithm by approximately 5%. However, one parameter is not wholly sufficient to conclude about the performance of a certain algorithm, therefore, further parameters were calculated by the authors as given below.

Table 26 has the results tabulated for the Pearson correlation coefficient (r), another performance metric which signifies results contrary to the prior Table 25 showing MAPE, wherein LSTM had outperformed GA, however in Table 26, GA has outperformed LSTM by a small value.

Table 27 has the results tabulated for the parameter Index of Agreement (IA), wherein the performance of LSTM has clearly exceeded that of the Genetic algorithm. Therefore, on the basis of the above results and calculation of three performance metrics, MAPE, r, and IA, it would be safe to conclude that LSTM is superior in performance to the Genetic Algorithm for such predictive analytics.

8. Future scope

Due to the revolutionary characteristics of machine learning, its scope is expanding day in and day out. The automobile industry is one such example portraying the excellent innovations using machine learning. Leading automotive brands, like Tesla, Toyota, Mercedes Benz, Google, Nissan, etc have put in huge sums of money in this domain to come up with innovative programs employing machine learning and other artificial intelligence. The well-known Self-driving car brought out Tesla, is built using Internet of Things (IoT) sensors, Machine Learning, high-definition cameras, etc. which would require human input just to feed the desired destination into the system, and all the

Table 24
Characteristics of the dataset that was considered for research.

Impact factors	Characteristics
Tourist flow related historical data	The number of tourists yesterday The number of tourists the day before yesterday The number of tourists 365 days ago The number of tourists same day last week The number of tickets
Time factors	Monday to Sunday Holiday or working day
Meteorological factors	Weather Wind Speed Average Temperature Average Humidity
Baidu search index	Huangshan Scenic Spot Huangshan Travel Guide Huangshan Hong Village Huangshan Travel Map Huangshan Day Tour

Table 25
Performance comparison based on the parameter MAPE.

Test	GA-LSTM-CNN	LSTM-CNN	LSTM	GA
1	20.73	22.90	24.92	29.81
2	20.50	22.29	23.96	29.81
3	20.86	22.56	26.64	29.80
4	20.79	22.64	24.54	29.81
5	20.96	22.74	24.56	29.81
Average	20.77	22.63	24.92	29.81

Table 26
Performance comparison based on the Pearson correlation coefficient (r).

Test	GA-LSTM-CNN	LSTM-CNN	LSTM	GA
1	0.911	0.908	0.847	0.887
2	0.912	0.900	0.842	0.887
3	0.912	0.908	0.847	0.887
4	0.916	0.901	0.846	0.887
5	0.912	0.905	0.837	0.885
Average	0.913	0.904	0.844	0.887

Table 27
Performance comparison based on the Index of Agreement (IA).

Test	GA-LSTM-CNN	LSTM-CNN	LSTM	GA
1	0.923	0.929	0.915	0.901
2	0.922	0.919	0.911	0.901
3	0.919	0.912	0.920	0.906
4	0.921	0.910	0.917	0.906
5	0.904	0.913	0.918	0.906
Average	0.919	0.917	0.916	0.904

rest of the work would be done by the machine, i.e., choosing a good route, free of traffic, and safely driving the passenger to his destination. Robotics is another field that has been constantly under talks among scientists, researchers, and even the common people. Machine learning and artificial intelligence has made it possible to come up with inventions like the first programmable robot named Unimate back in 1954, followed by the creation of the first artificially intelligent robot named Sophia. There is a bright scope for research in this field, and in future robots created using ML and AI can be expected and other revolutionary technologies, which would be able to perform functions akin to those of humans, in all fields, including medicine [61]. ML is still yet to be explored beyond limits, and one of the fields that would greatly help in exploring ML is Quantum Computing. It constitutes the mechanical occurrence of quantum like superposition and entanglement [62] (see Table 28).

A detailed description of the novel applications of the five machine learning algorithms has been done in the paper. Below, a table including the novel applications, the authors of the respective papers and their findings to summarize the descriptions written above, is presented (see Table 29).

9. Conclusions

This paper presents a comparative study of the KNN, Genetic, SVM, Decision Tree, and LSTM machine learning algorithms, along with some of their recent novel applications which have a great scope for research in the future times [63–70]. Algorithms and related concepts have been explained in great detail, commencing from their origin till their most novel applications. This paper throws light on a lot of crucial aspects like when and under what circumstances did the algorithms originate, and how are they useful in today's scenario for real-time predictive and other applications. Visions concerning the methodologies to implement these algorithms have also been discussed in detail and their results and performances in pristine and novel research work have been discussed. A detailed comparison of the various machine learning algorithms on qualitative and quantitative basis has been done; and has also been summarized in the form of tables. After conducting an exhaustive review and research in this domain, we have been able to arrive at important conclusions like LSTM network and the SVM algorithm have rendered one of the best results when it comes to predictive analytics in real-time applications related to multidisciplinary spheres like medicine, bank frauds, face detection, student performance prediction, electricity usage prediction, etc. The LSTM network is deep learning networking with feedback and has an added advantage of retaining required information, which allows it to show highly accurate results in terms of prediction. Finally, the future scope highlights, the expected demand and popularity of machine learning and artificial intelligence in the future, which is anticipated to either support humans in multiple fields or completely replace them and bring in automation at a large scale and pace with the help of more advanced and rigorous research [71–75].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 28
Qualitative comparison of various ML algorithms.

Algorithm → Parameter ↓	KNN	GA	SVM	DT	LSTM
Type	Supervised Classification algorithm	Supervised algorithm	Supervised, Classification algorithm	Supervised algorithm	Unsupervised algorithm
Advantages	Easy to apply. Tolerant and resistant to the noise. Fast and easy to interpret. Effective for large datasets.	Highly robust against local maxima or minima. Can improve enormous-sized space state. Resistant to varying inputs and noise. Does not require imitative data. Elaborative and ideal in nature.	Efficient when there is a clear margin of separation between classes. Good for high-dimensional spaces. Useful when no. of dimensions exceed the no. of samples Efficient in terms of memory.	Simple and easy to comprehend. Beneficial during exploration of data. Less cleaning and segregation of data is required It is flexible in terms of data type Uses Non-Parametric Method and does not assume in spatial distribution.	Time lags are bridged by constant error backpropagation within the memory cell itself Robust for vanishing gradients and do not require parameter fine-tuning Can handle long-term sequential dependencies and save memory for longer durations Accurate prediction
Disadvantages	Deciding a suitable value for K is a challenge. Calculating the Euclidean distance between all the points leads to high cost of computation.	Can lead to untimely convergence of the population Tough to design objective function and achieve the operations. GA is time consuming to apply.	Inefficient for huge data set. Does not work with noisy data set. Unsuitable when no. of features exceed no. of training data samples.	Overfitting is a major drawback in the decision tree model, can be solved by pruning. Unsuitable for continuous variables.	Due to high complexity of the cell, the vanishing gradient problems are unsolved. Resource and time-intensive during training. Requires high memory B.W, hence inefficient in terms of hardware.

Table 29
Novel applications of various ML algorithms.

Algorithm → Parameter ↓	KNN	GA	SVM	DT	LSTM
Novel Application	Comparison of LR, NB & KNN ML algorithms for credit card fraud detection.	Face Recognition Based on Genetic Algorithm Optimization	Use of General Linear Model, Artificial Neural Networks & Support Vector Machine algorithm for breast cancer detection	Liver disease prediction using different decision tree techniques	LSTM-RNN model for prediction of electric load requirement
Year of publishing	2020	2018	2019	2018	2018
Author (s)	Fayaz Itoo, Meenakshi, Satwinder Singh	Mourad Moussa, Maha Hamila, Ali Douik	Sidey-Gibbons, J. Sidey-Gibbons, C.	Nazmun Nahar, Ferdous Ara	Salah Bouktif, Ali Fiaz, Ali Ouni, Mohamed Adel Serhani
Major Findings	K-NN showed the least accuracy, due to small training set.	This novel approach increased the accuracy by 8% than previous works related to this field.	Out of the three algorithms that were compared, SVM represented the most accurate results.	Decision stump outperformed the rest of the techniques by gaining an accuracy of 70.67%.	LSTM model is superior to other models in this case, as the RMSE and the MAE are comparatively quite small in case of LSTM.

References

[1] https://www.javatpoint.com/machine-learning.
[2] https://images.app.goo.gl/eLBR6gBjRGnSyJ7S9.
[3] E. Fix, J.L. Hodges, Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties, Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
[4] M. Bansal, H. Singh, The genre of applications requiring the use of IoT in Day-to-Day Life, Int. J. Innov. Adv. Comput. Sci. (IJIACS) 6 (11) (2017) 147–152.
[5] M.E. Hellman, The nearest neighbor classification rule with a reject option, IEEE Trans. Syst. Man Cybern. 3 (1970) 179–185.
[6] K. Fukunaga, L. Hostetler, K-nearest-neighbor bayes-risk estimation, IEEE Trans. Inform. Theory 21 (3) (1975) 285–293.
[7] S.A. Dudani, The distance-weighted k-nearest-neighbor rule, IEEE Trans. Syst. Man Cybern. SMC-6 (1976) 325–327.
[8] M. Bansal, M. Nanda, M.N. Husain, Security and privacy aspects for internet of things (IoT), in: 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 199–204, http://dx.doi.org/10.1109/ICICT50816.2021.9358665.
[9] M. Bansal, V. Sirpal, Fog computing-based internet of things and its applications in healthcare, J. Phys.: Conf. Ser 916 (2021) 012041, pp. 1-9.
[10] https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.
[11] https://www.analyticsvidhya.com/blog/2021/04/simple-understanding-and-implementation-ofknn-algorithm/.
[12] https://images.app.goo.gl/Lx61DdipcQyXZ2287.
[13] https://images.app.goo.gl/WQbK8Ak4KaFzQs6r9.
[14] F. Itoo, S. Meenakshi Singh, Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection, Int. J. Inf. Technol. 13 (2021) 1503–1511, http://dx.doi.org/10.1007/s41870-020-00430.
[15] K. Taunk, S. De, S. Verma, A. Swetapadma, A brief review of nearest neighbor algorithm for learning and classification, in: 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 1255–1260, http://dx.doi.org/10.1109/ICCS45141.2019.9065747.
[16] M. Turing, I.—Computing machinery and intelligence, Mind LIX (236) (1950) 433–460, http://dx.doi.org/10.1093/mind/LIX.236.433.
[17] N.A. Barricelli, Numerical testing of evolution theories, Acta Biotheor. 16 (1962) 69–98, http://dx.doi.org/10.1007/BF01556771.
[18] M. Bansal, S. Gupta, S. Mathur, Comparison of ECC and RSA algorithm with DNA encoding for IoT security, in: 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1340–1343, http://dx.doi.org/10.1109/ICICT50816.2021.9358591.
[19] M. Bansal, S. Garg, Internet of things (IoT) based assistive devices, in: 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1006–1009, http://dx.doi.org/10.1109/ICICT50816.2021.9358662.

[20] A. Bernstein, H. Rubin, Artificial evolution of problem-solvers, Amer. Behav. Sci. 8 (9) (1965) 19–23, http://dx.doi.org/10.1177/000276426500800907.

[21] AS. Fraser, Simulation of genetic systems by automatic digital computers I. Introduction, Aust. J. Biol. Sci. 10 (1957) 484–491, http://dx.doi.org/10.1071/BI9570484.

[22] Alex Fraser, Donald Burnell, Computer Models in Genetics [by] Alex Fraser [and] Donald Burnell, McGraw-Hill, New York, 1970.

[23] Jack L. Crosby, Computer Simulation in Genetics, John Wiley & Sons, London, ISBN: 978-0-471-18880-3, 1973.

[24] David B. Fogel (Ed.), Evolutionary Computation: The Fossil Record, IEEE Press, New York, ISBN: 978-0-7803-3481-6, 1998.

[25] M. Bansal, P. Priya, Performance comparison of MQTT and CoAP protocols in different simulation environments, in: G. Ranganathan, J. Chen, Á. Rocha (Eds.), Inventive Communication and Computational Technologies, in: Lecture Notes in Networks and Systems, vol. 145, Springer, Singapore, 2021, pp. 549–560, http://dx.doi.org/10.1007/978-981-15-7345-3_47.

[26] M. Bansal, P. Priya, Machine learning perspective in VLSI computer aided design at different abstraction levels, in: S. Shakya, et al. (Eds.), Mobile Computing and Sustainable Informatics, in: Lecture Notes on Data Engineering and Communications Technologies, vol. 68, Springer, Singapore, 2021, pp. 95–112, http://dx.doi.org/10.1007/978-981-16-1866-6_6.

[27] N.A. Barricelli, Numerical testing of evolution theories, Acta Biotheor. 16 (1963) 99–126, http://dx.doi.org/10.1007/BF01556602.

[28] Ingo Rechenberg, Evolutionsstrategie, Holzmann-Froboog, Stuttgart, ISBN: 978-3-7728-0373-4, 1973.

[29] Hans-Paul Schwefel, Numerische Optimierung Von Computer-Modellen (Ph.D. thesis), 1974.

[30] https://www.geeksforgeeks.org/encoding-methods-in-genetic-algorithm/.

[31] Mourad Moussa, Maha Hamila, Ali Douik, A novel face recognition approach based on genetic algorithm optimization, Stud. Inform. Control (ISSN: 1220-1766) 27 (1) (2018) 127–134, http://dx.doi.org/10.24846/v27i1y201813.

[32] Hans-Paul Schwefel, Numerische Optimierung Von Computor-Modellen Mittels Der Evolutionsstrategie : Mit Einer Vergleichenden Einführung in Die Hill-Climbing- Und Zufallsstrategie, Birkhäuser, Basel; Stuttgart, ISBN: 978-3-7643-0876-6, 1977.

[33] Hans-Paul Schwefel, Numerical optimization of computer models, in: (Translation of 1977 Numerische Optimierung Von Computor-Modellen Mittels Der Evolutionsstrategie, Wiley, Chichester; New York, ISBN: 978-0-471-09988-8, 1981.

[34] https://www.geeksforgeeks.org/simple-genetic-algorithm-sga/.

[35] . Lambora, K. Gupta, K. Chopra, Genetic algorithm- A literature review, in: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 380–384.

[36] https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

[37] J. Sidey-Gibbons, C. Sidey-Gibbons, Machine learning in medicine: a practical introduction, BMC Med. Res. Methodol. 19 (2019) 64, http://dx.doi.org/10.1186/s12874-019-0681-4.

[38] M. Bansal, S. Malik, M. Kumar, N. Meena, Arduino based smart walking cane for visually impaired people, in: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 462–465, http://dx.doi.org/10.1109/ICISC47916.2020.9171209.

[39] M. Bansal, . Prince, R. Yadav, P.K. Ujjwal, Palmistry using machine learning and opencv, in: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 536–539, http://dx.doi.org/10.1109/ICISC47916.2020.9171158.

[40] https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[41] https://www.analyticsvidhya.com/blog/2021/02/machine-learning-101-decision-tree-algorithm-for-classification/.

[42] https://images.app.goo.gl/Ugwp564wYFUqjPsd8.

[43] Nazmun Nahar, Ferdous Ara, Liver disease prediction by using different decision tree techniques, Int. J. Data Min. Knowl. Manage. Process 8 (2018) 01–09, http://dx.doi.org/10.5121/ijdkp.2018.8201.

[44] M. Bansal, A. Goyal, A. Choudhary, Industrial internet of things (iIoT): A vivid perspective, in: V. Suma, J.IZ. Chen, Z. Baig, H. Wang (Eds.), Inventive Systems and Control, in: Lecture Notes in Networks and Systems, vol. 204, Springer, Singapore, 2021, pp. 939–949, http://dx.doi.org/10.1007/978-981-16-1395-1_68.

[45] M. Bansal, N. Oberoi, M. Sameer, IoT In online banking, J. Ubiquit. Comput. Commun. Technol. (UCCT) 2 (4) (2020) 219–222.

[46] Sasan Karamizadeh, Shahidan Abdullah, Mehran Asl, Jafar Shayan, Mohammad Rajabi, Advantage and drawback of support vector machine functionality, in: I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings, 2014, http://dx.doi.org/10.1109/I4CT.2014.6914146.

[47] Ms M.P. Phalak, D.R. Bhandari, . Sharma, Analysis of deci- sion tree-A survey, Int. J. Eng. Res. Technol. (IJERT) 03 (03) (2014).

[48] M. Bansal, V. Sirpal, M.K. Choudhary, Advancing e- government using internet of things, in: S. Shakya, et al. (Eds.), Mobile Computing and Sustainable Informatics, in: Lecture Notes on Data Engineering and Communications Technologies, vol. 68, Springer, Singapore, 2021, pp. 123–137, http://dx.doi.org/10.1007/978-981-16-1866-6_8.

[49] https://images.app.goo.gl/QiZ2cYj6MfrZqCbGA.

[50] http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[51] S. Bouktif, A. Fiaz, A. Ouni, M. Serhani, Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches †, Energies 11 (1636) (2018).

[52] Slamet Wiyono, Taufiq Abidin, Comparative study of machine learning knn, svm, and decision tree algorithm to predict student's performance, Int. J. Res. - Granthaalayah 7 (2019) 190–196, http://dx.doi.org/10.29121/granthaalayah.v7.i1.2019.1048.

[53] M. Bansal, T. Chopra, S. Biswas, Organ simulation and healthcare services: An application of IoT, in: 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 205–208, http://dx.doi.org/10.1109/ICICT50816.2021.9358677.

[54] https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[55] https://www.analyticsvidhya.com/blog/2021/02/machine-learning-101-decision-tree-algorithm-for-classification/.

[56] https://images.app.goo.gl/Ugwp564wYFUqjPsd8.

[57] Nazmun Nahar, Ferdous Ara, Liver disease prediction by using different decision tree techniques, Int. J. Data Min. Knowl. Manage. Process 8 (2018) 01–09, http://dx.doi.org/10.5121/ijdkp.2018.8201.

[58] https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[59] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, Lstm: A search space odyssey, IEEE Trans. Neural Netw. Learn. Syst. 28 (10) (2017) 2222–2232, http://dx.doi.org/10.1109/TNNLS.2016.2582924.

[60] M. Bansal, . Priya, Application layer protocols for internet of healthcare things (IoHT), in: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 369–376, http://dx.doi.org/10.1109/ICISC47916.2020.9171092.

[61] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, in: 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), Vol. 2, 1999, pp. 850–855, http://dx.doi.org/10.1049/cp:19991218.

[62] https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.

[63] B. Lantz, Machine Learning with R, Packt Publishing Ltd, Birmingham, 2013.

[64] M. Ciolacu, A. Tehrani, R. Beer, H. Popp, Education 4.0fostering student's performance with machine learning methods, in: 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME), 2017, pp. 438–443.

[65] A. Khasanah, A. Harwati, A comparative study to predict students performance using educational data mining techniques,, IOP Conf. Ser.: Mater. Sci. Eng. 215 (2) (2017) 1–7.

[66] A. Vihavainen, Predicting students' performance in an introductory programming course using data from students' own programming process, in: 2013 IEEE 13th International Conference on Advanced Learning Technologies, 2013, pp. 498–499.

[67] M. Quadri, N. Kalyankar, Drop out feature of student data for academic performance using decision tree techniques,, Glob. J. Comput. Sci. Technol. (2010).

[68] T. Devasia, T. Vinushree, V. Hegde, Prediction of students performance using educational data mining, in: 2016 International Conference on Data Mining and Advanced Computing (Sapience), 2016, pp. 91–95.

[69] M. Bansal, N. Adarsh, N. Kumar, M. Meena, 24×7 Smart IoT based integrated home security system, in: 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 477–481, http://dx.doi.org/10.1109/ICISC47916.2020.9171051.

[70] B. Albreiki, N. Zaki, H. Alashwal, A systematic literature review of student' performance prediction using machine learning techniques, Educ. Sci. 11 (9) (2021) 552, http://dx.doi.org/10.3390/educsci11090552.

[71] W. Lu, H. Rui, C. Liang, L. Jiang, S. Zhao, K. Li, A method based on GA-CNN-LSTM for daily tourist flow prediction at scenic spots, Entropy 22 (2020) 261, http://dx.doi.org/10.3390/e22030261.

[72] Abhinav Jain, et al., Overview and importance of data quality for machine learning tasks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020.

[73] Archana Singh, Rakesh Kumar, Heart disease prediction using machine learning algorithms, in: 2020 International Conference on Electrical and Electronics Engineering (ICE3), IEEE, 2020.

[74] https://intellipaat.com/blog/future-scope-of-machine-learning.

[75] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.