DOI: 10.15849/IJASCA.231130.19

Int. J. Advance Soft Compu. Appl, Vol. 15, No. 3, November 2023
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Pattern Prediction on Uncertain Big Datasets using Combined Light GBM and LSTM Model

G Divya Zion, *B. K. Tripathy

School of Computer Science and Engineering, VIT, Vellore-632014, TN, India
e-mail: gdivya.zion2016@vitstudent.ac.in

School of Computer Science Engineering and Information Systems, VIT, Vellore-632014, TN, India
e-mail: tripathybk@vit.ac.in
*Corresponding author

Abstract

Mining frequent patterns from voluminous datasets termed under 'Big data' and having inherent uncertainties poses a significant challenge. Minor changes carried out on the databases like; addition, deletion or modification of items should not lead to scanning the whole database. Besides, a number of algorithms proposed to handle these issues are effective, but their basis of mathematics and way of installation are complex. Keeping the above points in mind, we have proposed an approach, which innovatively combines the models Light Gradient Boosting Machine (LightGBM) and Long Short-Term Memory (LSTM) serially to improve the prediction accuracy. Here, the LightGBM brings its tree-based learning algorithms optimized for speed and performance, while LSTM contributes its advanced sequence modeling capabilities, effectively resolving the vanishing gradient dilemma that often plagues recurrent networks. Our approach is applied to the healthcare sector in general and particularly in the early detection of Breast Cancer from a dataset obtained from Kaggle, yielding outstanding results as are evident from the scores; precision rates of 0.92 for predicted negatives and 0.93 for predicted positives, recall rates of 0.96 for negatives and 0.88 for positives, alongside F1-scores of 0.94 and 0.90, respectively. With a comprehensive accuracy of 0.93 across 188 samples, our model demonstrates a remarkable potential for early medical diagnosis, outperforming existing single-model solutions. The robustness of our approach is further validated by the consistency of performance across various metrics, highlighting its suitability for deployment in high-stakes domains where predictive accuracy is paramount.

Keywords: ***Artificial Intelligence, Big Data, Data Mining, Data Science, Machine Learning, Soft Computing.***

Received 7 August 2022; Accepted 22 November 2023

1 Introduction

Frequent pattern (FP) mining is applied on datasets to come up with patterns in the form of substructures, itemsets or subsequences and appear more often than the others. The application areas of FP mining include image analysis, biometric recognition, medical diagnosis, bioinformatics, remote sensing, GIS, document classification, handwritten text analysis and many more. Models opted for pattern recognition can be categorized in to different categories such as Statistical Model, Structural Model, Template Matching Model, Neural Network Based Model, Fuzzy Based Model and Hybrid model.

Now days, datasets are mostly categorized under the label of Big data, which are primarily having the characteristics mentioned in the Fig. 1

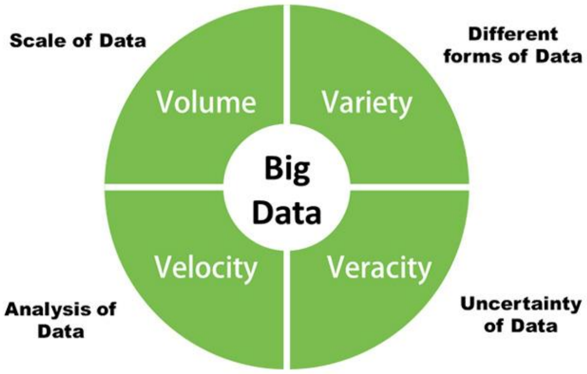


Fig. 1 Big Data Characteristics

Generating patterns and their analysis leads to many interesting properties like associations and correlations. Several data mining tasks; like the classification, clustering and finding associations rules can be carried out as a result of the patterns generated. Therefore, FP mining has attracted many researchers in the field of Data mining. Uncertainty has become an unavoidable part of the modern-day datasets. So, an investigator has to bear with it and work on it. Also, these datasets can contain only records or Graph-based records or ordered records. The storage of records can be in a Data Matrix with fixed set of numeric attributes having several dimensions or attributes in a multidimensional space. It may be a normal Data matrix or in a sparse Data matrix as is the requirement. All these types of data fall into the big data category along with inherent uncertainty in them. So, any Pattern mining or FP mining should be capable of handling these datasets.

Several of the FP mining algorithms have been extended to deal with uncertainty. Some such algorithms are candidate generate-and-test algorithms, hyper-structure algorithms and pattern growth-based algorithms. However, it has been observed that the experimental behavior of different classes of algorithms does not retain their efficiencies when extended to the uncertain ir respective uncertain cases as compared to the deterministic cases. As has been established in the deterministic case, the FP-growth algorithm which is very efficient in the crisp case loses its efficiency over the extensions of the candidate generate-and-test and the hyper-structure based algorithms are much more effective. So, the necessity of developing algorithms from the basics remains a challenge instead of extensions of crisp case algorithms to accommodate the uncertainty. In this article, we move a step ahead by combining two of the algorithms serially to take care of FP mining efficiently.

The first model we take is the LightGBM, which is a gradient-boosting framework based on decision trees to increase the efficiency of the model and reduces memory usage. Light GBM leads to accurate gain estimation than uniformly random sampling, with target sampling rate when the value of information gain has a large range. The Exclusive Feature Bundling (EFB) Technique for Light GBM takes care of reducing high-dimensional data and is considered as a process of reducing the number of features. Sparse feature has features that are mutually exclusive with no zero values. The exclusive feature is one of the bundled features with high performance gradient boosting method. This uses decision tree algorithms used for ranking classification, prediction and many other machine learning tasks [23], [24]. The other model we use is the Long-short term memory (LSTM) network, which is a prominent recurrent neural network invented by Schmidt Huber [21]. LSTM has three gated units mainly as forget gate, input and output gates which can control memory of past states and is capable of improving the prediction accuracy if applied after the LightGBM.

Cancer ranks as a leading cause of death and an important barrier to increasing life expectancy in every country of the world. According to estimates from the World Health Organization (WHO) in 2019 cancer is the first or second leading cause of death before the age of 70 years in 112 of 183 countries and ranks third or fourth in a further 23 countries. Rising prominence of Cancer as a leading cause of death partly reflects marked declines in mortality rates of stroke and coronary heart disease, relative to cancer, in many countries. Estimates of cancer incidence and mortality produced by the International Agency for Research on Cancer, worldwide, an estimated 19.3 million new cancer cases (18.1 million excluding nonmelanoma skin cancer) and almost 10.0 million cancer deaths (9.9 million excluding nonmelanoma skin cancers) occurred in 2020. Female breast cancer has surpassed lung cancer as the most commonly diagnosed cancer, with an estimated 2.3 million new cases. As an application and illustration of our approach we study the prediction of breast cancer. We use a dataset obtained from Kaggle and see that the results yielded are outstanding.

The rest of the paper is organized as follows. Section 2 gives an overview of the related works. Section 3 presents the working of the Machine Learning Algorithms. Section 4 shows the experimental results and evaluation of the performance of the algorithm. Finally, we provide concluding remarks in section 5.

2 Related Work

FPs play a vital role in the field of mining the associations, correlations and relationships among the data [1]. The advancement of technology results in the change in volume of various fields of as social networking data, banking data, biological data, marketing data, financial data and in medical data. This leads to new era called Big Data [2]. Big Data has all the three powerful Vs as velocity, variety, volumes of data that capture, manage, and process within a tolerable elapsed time. Data is processed by considering decision making and knowledge discovery. Big Data is processed by a programming model named Map

Reduce model. This Map Reduce Model consists of both the master node and multiple worker nodes [3] with two key roles: “map” and “reduce”. Big Data Mining and analytics discovers sets of different commodities. Few commodities are text categorization [4], text mining [5], social network [6], [43], and frequent sub graph [7]. Mining frequent patterns can be completed in 3 ways; Candidate Generation Methods, Without Candidate Generation Methods and Parallel Methods. The Candidate Generation Method scans the dataset numerous times in order to generate best candidates used as frequent patterns of dataset. Apriori algorithm, proposed by Agrawal and Srikantin1995 [8],[44,63] reduces the complexity and increases the efficiency of finding frequent patterns. Partitioning technique [8], pattern-based algorithms and incremental Apriori based algorithms are used in finding candidates using its generation methods [9]. Map reducing implementation using Apriori algorithm mines frequent patterns present in large datasets. Apriori based algorithms produces large number of candidate’s sets. As an alternate approach FP-growth tree-based bottom-up approach is used as transaction reduction technique [10], [45,62]. Several parallel techniques have been proposed in recent years, to solve the complexity problems in finding frequent patterns [11] [12]. To find frequent pattern occurrences based on time-series which is applied on satellite weather monitoring, parallel method has given good results [13]. Fuzzy based frequent pattern mining has been a new approach which is discussed [14], [46].

Though there were significant changes in finding frequent patterns, working with the varying database has been a challenge. Especially, it need not scan again the whole database whenever having need of adding a new element or deleting/modifying an element. Besides, a number of algorithms are effective, but their basis of mathematics and way of installation are complex. In addition, it is the limit of computer memory. Hence, storing the data mining context is based on time constraint. Finally, ability of dividing data into several parts for parallel processing is also concerned. At present, according to various forecasting systems and the tools that replenishes the predicting sales of items in their store; in order to make the assortment well the forecasting models can be divided into three categories: time series model, machine learning model, and deep learning model [15]. Time-series models such as exponential smoothing method, autoregressive moving average model, and autoregressive conditional heteroscedasticity model arch [16]. Machine learning models in data mining methods are random forest and support vector machine model [17], [47]. Deep learning (DL) is a modern tool for automatic feature extraction and prediction [18] It has strong adaptability and self-learning ability and does not need to show specific network relationships and mathematical models [19], [20], [21]. However, the traditional model is prone to the problem of over fitting and time-series dependence of data, and RNN recurrent neural network can solve the problem. However, RNN has some problems such as gradient explosion and it cannot converge to the optimal solution [22], [56]. Deep learning is applied on predicting stock price forecasting [23],[57,61] when a comparison is done among the models like and machine learning and neural, deep learning has more certainty, thorough explanation ability, and vigorous learning ability to adjust to new problems [24], [58-63].

3 Problem Formulations or Methodology

LightGBM is evolved from Microsoft which is a free redistribution algorithm and has gradient boosting framework. Decision tree is used to increase the model performance and reduces the usage of memory. (GOSS) Gradient-based One Side Sampling and

(EFB) Exclusive Feature Bundling, both algorithms are used by LightGBM. Both these algorithms use histogram-based algorithm [29] which process the training data faster, and memory consumption is reduced in all GBDT (Gradient Boosting Decision Tree) frameworks. These techniques allows the model work perfectly when compared with that of other GBDT frameworks. Leaf-wise strategy is used in LightGBM which finds a leaf with the largest gain of variance to do the splitting. LightGBM can be differentiated from other models by the way the gain of variation is calculated. If the instances have larger gradients, then we have information gain which is larger than a predefined threshold and deletes the inputs with small gradients to bring accuracy on information gain estimation. This process gives procures more gain estimation than uniformly remaining random sampling methods. LightGBM is an algorithm that combines a machine learning algorithms decision tree model with ensemble learning called boosting [30]. The combination of learning ensemble algorithms is called a random forest. The Gradient Boosting Decision Tree (GDBT) uses LightGBM where bagging and boosting are 2 parameters used by LightGBM, and other algorithms such as XGBoost.

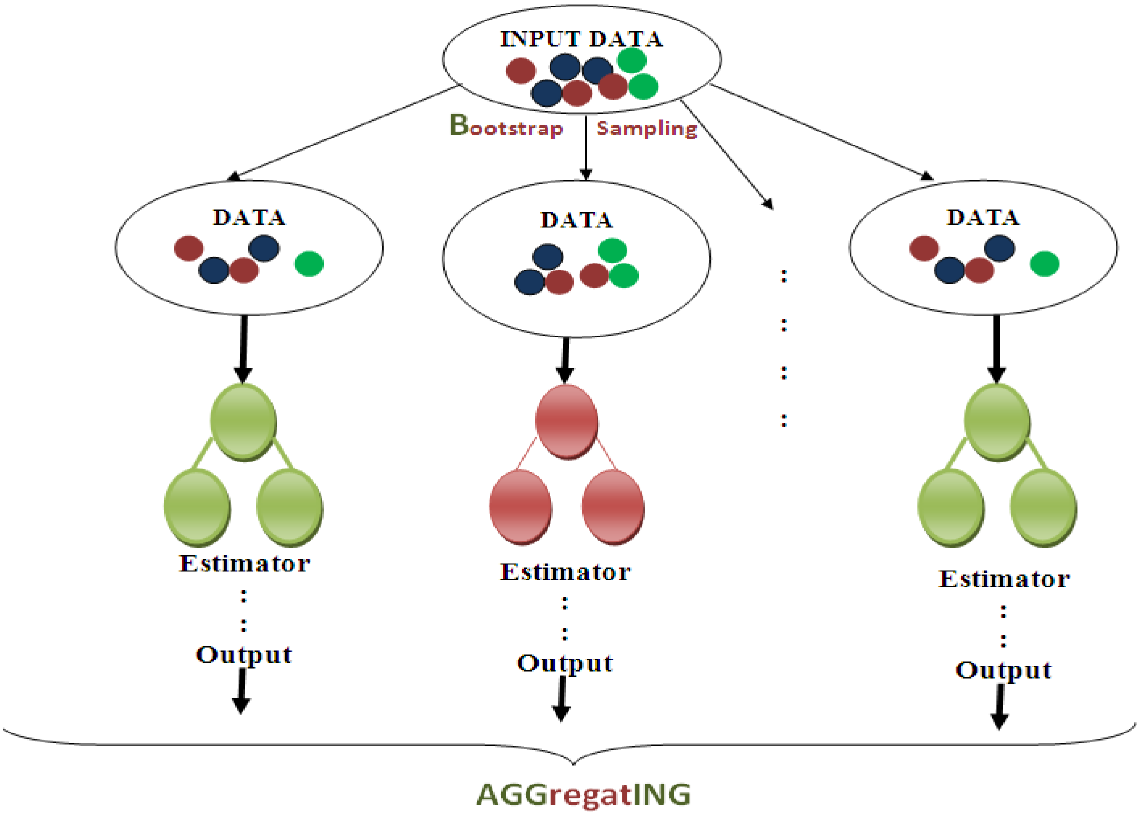


Fig.2: Working of Bagging

Bagging is constructed in parallel by-passing data from the input data to multiple predictors by bootstrap sampling. This is used in random forest /decision trees in parallel and comprehensively judging the results of all predictors. A diagram of bagging (where the decision tree is the predictor) is shown in Fig.2. Boosting: This algorithm calculates the loss from the output obtained by one predictor and uses it to construct the next predictor. Doing so increases the chances of getting a stronger predictor, which may outperform bagging. The more predictors connected in series, the more the next predictors will over fit the data, shown in Fig.3.

3.1 Working of LightGBM Algorithm:

LightGBM uses histogram-based splitting and is efficient as rate of change function is used in GOSS. Fig.4, explains the splitting of data done using histogram splitting and this makes the algorithm perform faster. Below is a dataset with salaries of a few employees as shown in Fig.4. Here, males are considered with a value 10 and 11 is considered for females. Exclusive Feature Bundling reduces the properties such that the model can be trained on the dataset easily.

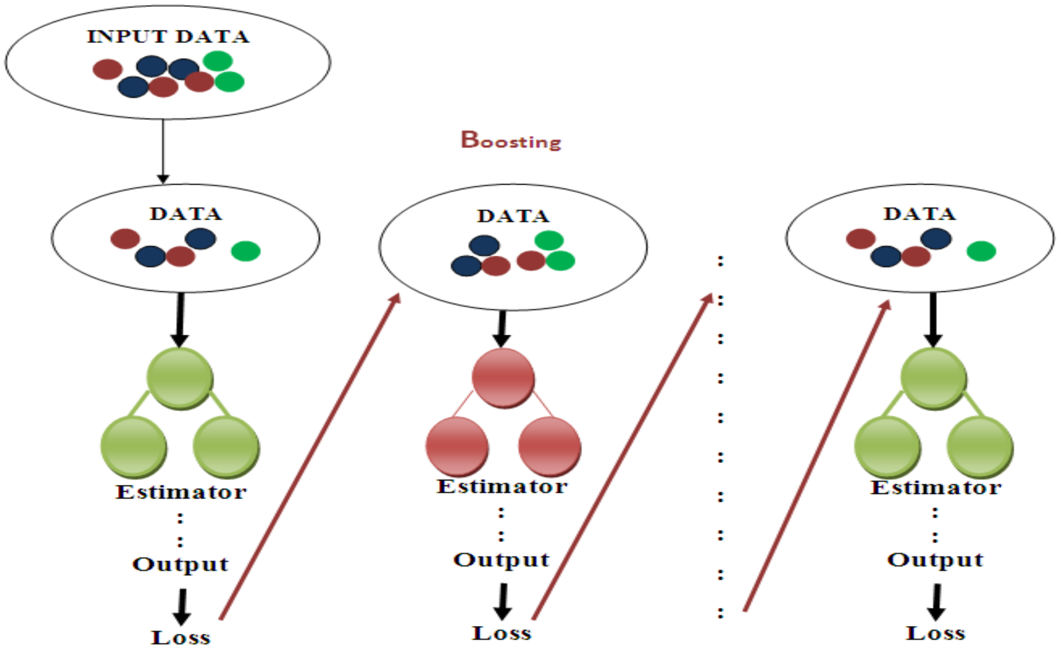
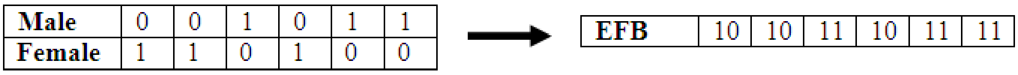


Fig.3: Working of Boosting

Male	0	0	1	0	1	1
Female	1	1	0	1	0	0

Fig.4: Employees' Salaries



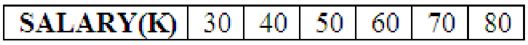
Male	0	0	1	0	1	1
Female	1	1	0	1	0	0

EFB	10	10	11	10	11	11

Fig.5: Dividing the dataset into bins

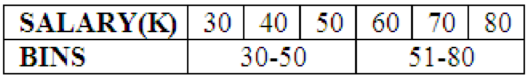
To perform a maximum split for a Decision tree, the values present in the dataset are considered. Suppose, there are millions of data rows, to do the best split lot of time is consumed. To minimize the time, LightGBM algorithm divides the dataset into multiple bins. The above example shows the dataset is divided into two bins. On the contrary of evaluating all the data values for maximum splitting, this is done by considering the bins such that the maximum split is done in less time as shown in Fig.5. Exclusive Feature Bundling (EFB) technique makes the LightGBM faster by decreasing the features that are small in number or amount. Exclusive Feature bundling (EFB) [31] this combines two features and adds offset to every feature that are available in feature bundles. The combination of categorical dataset regarding the attributes male and female shown in Fig.6

which have two columns for the gender of a person. Rather than having both the columns, EFB makes use of only one column which has the same information as shown in Fig.7. The larger the value the more error is shown in gradient [32]. All the inputs with large gradients that are considered as errors are considered to conduct random sampling on the inputs with small gradients. This model selects data points for the next iteration which has a high gradient value so that error can be reduced.



SALARY(K)	30	40	50	60	70	80

Fig. 6: Dataset with columns Male and Female



SALARY(K)	30	40	50	60	70	80
BINS	30-50			51-80		

Fig. 7: Minimizing to 2columns

3.1.1 Procedure

Datasets: Breast Cancer Dataset has been considered. This dataset is based on sequence of observations collected at some time intervals, which is Time series data. Data Preparation is done on missing values using any algorithm, which is also called as Data Cleaning. All the missing values are replaced based on the rule, where the number of missing observations is less than 5% of the total of instances for that feature then those values are not considered/ deleted. The model is trained so that predictions can be performed on the test set, where the main parameters considered are the number of leaves, the minimum amount of data in a leaf and the total number of threads as shown in Fig.8.

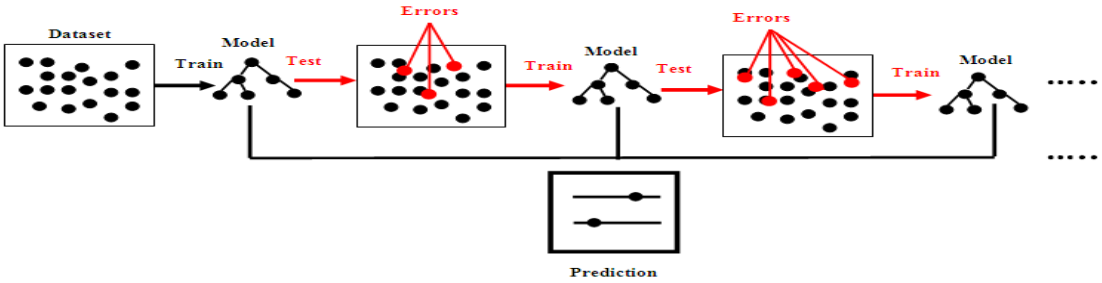


Fig.8: Working of LightGBM algorithm

Stratified random sampling method is considered to divide the dataset processed into smaller groups called as strata which are formed based on similar attributes or characteristics. The number of strata for the entire training performed has to be defining a forehand. The stratified random sampling method, can consider higher number of subsets when training a model, consequently this makes the model to perform better. Evaluation: The metrics that are used for evaluation are Absolute Error, Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error (RMSE). Root Mean Squared Error is considered to find the differences between values that are predicted by LightGBM. The differences are known as residuals. RMSE tracks the size of the errors in predictions and measures the performance of model. Therefore, RMSE is a measure that can be used to compare the accuracy in forecasting errors of different models on a particular dataset and not between datasets [34], [35]. The square root of the average of squared differences between prediction and actual observation is known as RMSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Where, n = number of observations, yj= predicted values and yˆj = actual values. RMSE always produces a positive number, if RMSE = 0 it conveys that model is best for dataset which is considered for the problem. So, Decision trees with less time span are constructed using LightGBM to make predictions such that error rate can be improved.

3.1.2 Drawbacks of LightGBM

The dataset is divided into the model similar to tree, where leaves grow in a level-wise. Leaves are chosen with maximum delta loss. Here the size of the leaf is fixed, therefore this algorithm has lower loss when compared with that of level-wise algorithm [33]. As the leaves in the tree increases the complicatedness of the model gets increased and this results to overfitting in small datasets. Leaf-Wise Tree Growth representation is shown in Fig.9;

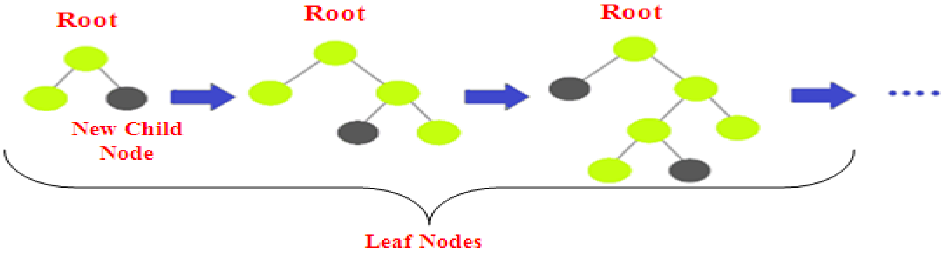


Fig.9: Leaf-wise Tree Growth

3.2 LSTM

Deep Neural Networks (DNN) are deep learning models which are extremely useful in data classification [40]. Out of the several DNN models existing recurrent Neural Networks (RNN) is capable of classifying temporal data. Long-short term memory was proposed by Schmidt Huber in 1997 [40], it is a special type of RNN model [36], which comes under the DNN models. There are several applications of this model in real life situations [37], [38], [39] This algorithm performs data processing, predicting, and data classification based on time-series data. LSTM disjoints the cell state which is denoted by c and output layer is denoted by h, it only does incremental updates to c, so that memories in c are constant. Therefore, the gradient flows through c are kept and hard to be deleted. This gradient issue can be overcome by adding three gated units: forget gate, input and output gates, which results in controlling the memory of past states. LSTM behaves similar to advanced Recurrent Neural Network. The RNN is used for persistent memory, where the previous information can be remembered and can be considered for processing the current input. The issue with RNN is it cannot have long term dependencies as the gradient gets vanished as shown in Fig.10. LSTMs are considered to avoid long-term dependency problems.

LSTM contains a cell state with 3 gates [41], [42] shown in Fig.11. The cell state allows the data to flow through the units without changing it. In each cell state there is a unit which has an input, output and a forget gate which allows to add or remove the information. The first gate decides whether the data coming from the previous timestamp is to be remembered or is irrelevant and can be deleted or forgotten. The second gate, tries to learn new information from the input to this cell.

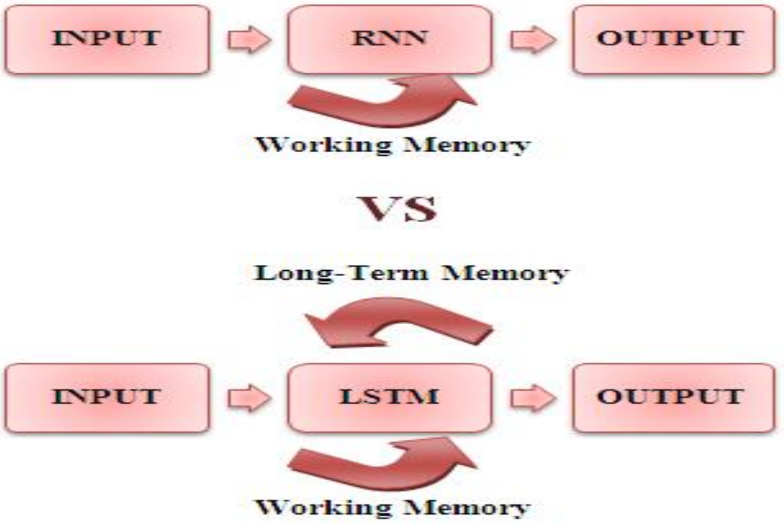


Fig. 10: Comparative of RNN and an LSTM

In the third gate/part, the updated information is passed from the current timestamp to the next timestamp. The hidden state which is present in third gate is known as short term memory and the cell state is known as long term memory. In [42] it says the LSTM architecture has a memory cell that has its state over time, and nonlinear gating units, to adjust the information flow in and out of the cell.

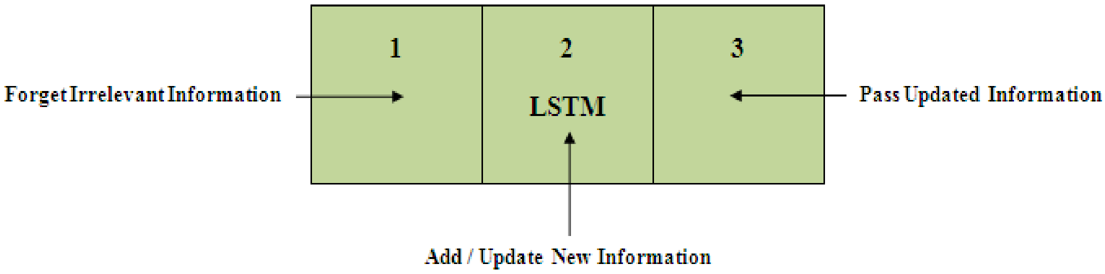


Fig.11: LSTM

3.2.1 Working of LSTM

Step 1: Both previous hidden state and new input data is given to forget gate in order to decide what bits of cell state (long term memory of the network) are necessary. Based on the decision, both the previous hidden state and the new input data are given to a neural network [43][44]. The neural network gives a vector such that each element lies in between the interval [0, 1] which should be justified by the sigmoid activation). As shown in Fig.12, The forget gate is trained with this neural network, so when outputs are close to 0 it conveys the input is meaningless and when output is closer to 1 conveys relevant. These values which are outputted have been point wise multiplied with the previous cell state. Point wise multiplication describes that components of the cell state which have been meaningless by the forget gate network and will be multiplied with a number that is close to 0. Therefore, the forget gate takes decision which pieces of the long-term memory should now be forgotten (have less weight) along with previous hidden state and the new data point which are in the sequence that needs to considered.

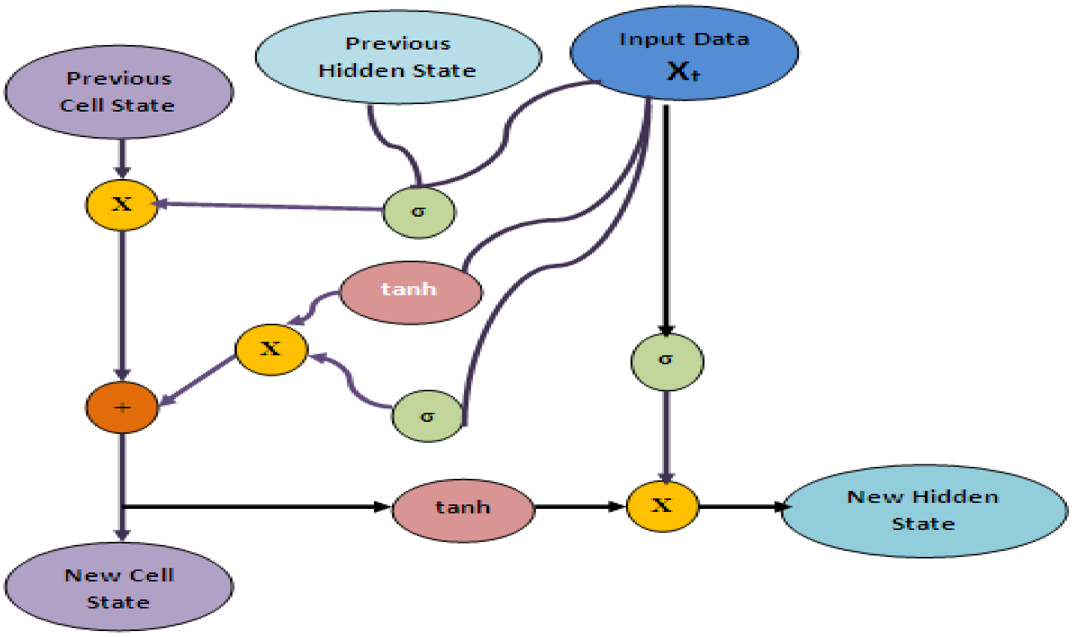


Fig. 12: working of Forget Gate

Step 2: The step performs adding of new information to the networks long-term memory (cell state), by adding the data from previous hidden state along with the new input data. The inputs considered are the same inputs to the forget gate. As shown in Fig.13, the network memory which is new is a neural network activated based on tanh which connects the previous hidden state and new input data to bring out a vector updated with new memory. The new vector basically has information from the new input data which has the context from the previous hidden state. Vector gives information on how to update each component of the long-term memory in network when given with new data.

Step 3: Output gate, which is the final step finalizes the new hidden state. To consider new hidden state the 3 states are considered; newly updated cell state, the previous hidden state along with the new input data. A filter is created for output gate, the inputs are the same with sigmoid [0, 1] as activation [45] [46]. As shown in Fig.14, this filter is applied to the cell state which is newly updated. This ensures that only necessary information is given as output to the new hidden state. The tanh is passed to the cell state with values into interval [-1, 1], before applying filter.

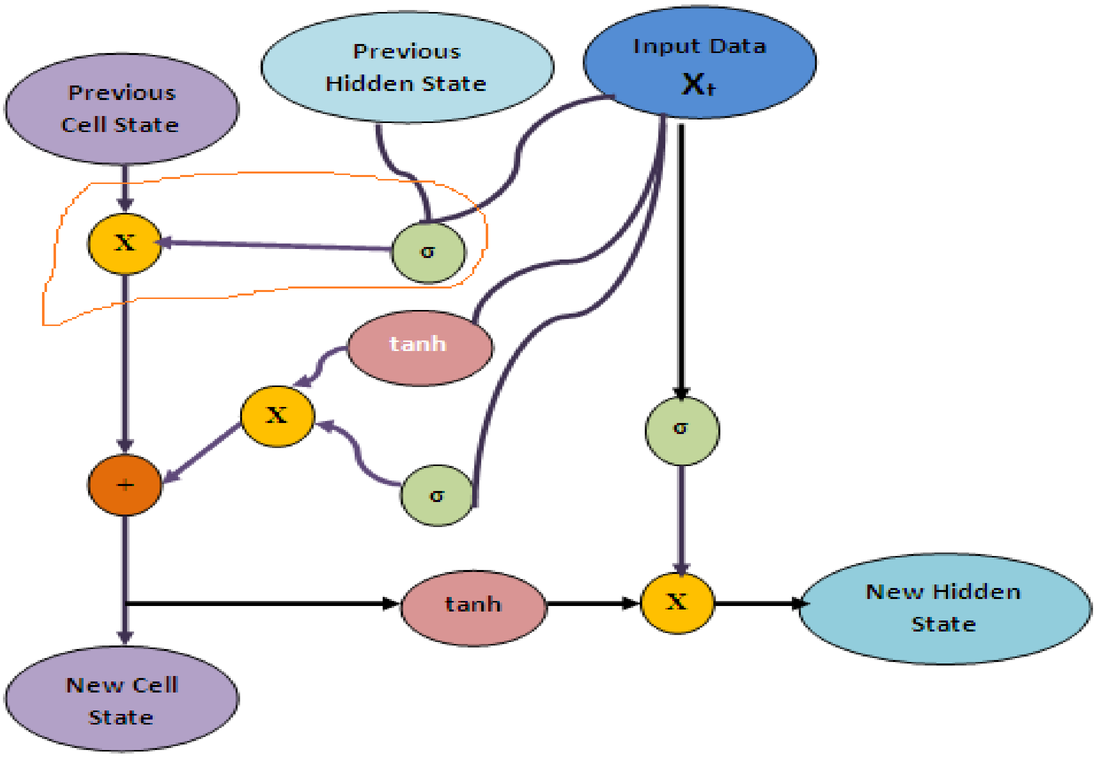


Fig.13: Input Gate.

3.2.2 Sequence of steps

1) tanh function is applied to the current cell state which lies in [-1,1]to obtain point wise.
2) The previous hidden state and current input data is passed through the activated neural network which is sigmoid in order to get the filter vector.
3) Filter vector is applied on the cell state in point wise multiplication manner.
4) The new hidden state is considered as the output state.

3.2.3 Drawbacks of LSTM

1) LSTMs overcome the issue of expired gradients but this model fails in removing them completely.
2) LSTM model considers lots of resources and consumes much time to get trained when working with real-time applications. In other terms, it needs high memory-bandwidth.[47][48].
3) In future, there can be a better model which remembers past information for a longer time than LSTM.

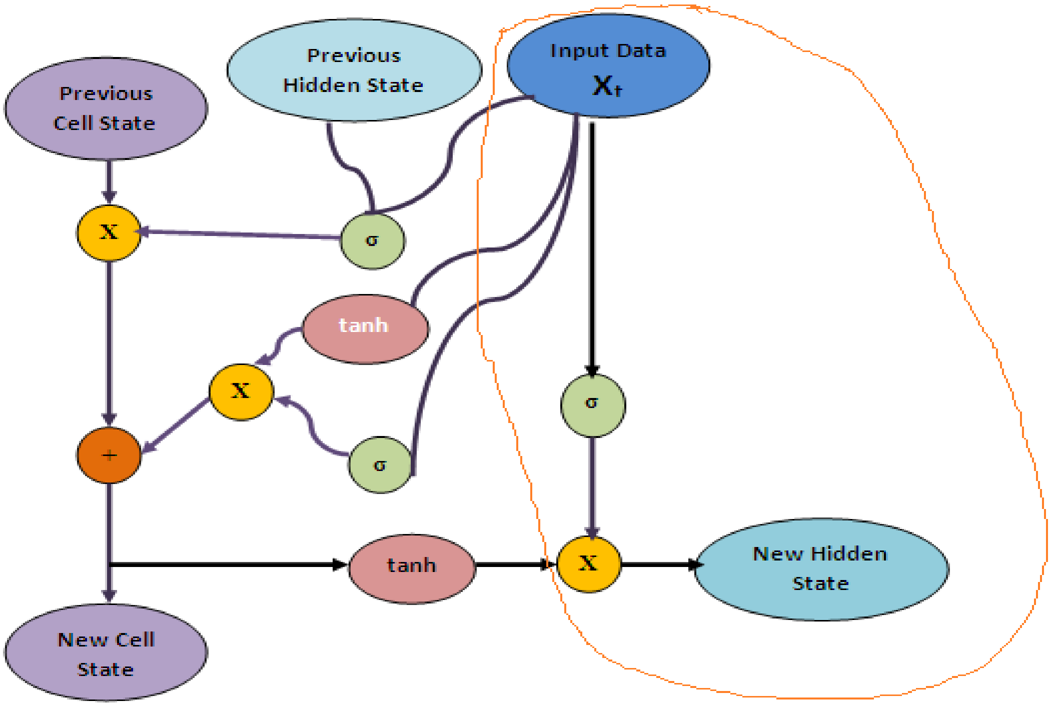


Fig.14: Output Gate

4 The Proposed Hybrid Method - Combined LightGBM and LSTM on Breast Cancer Dataset

The architecture for the hybrid model can be divided into four steps:
Dataset has been considered from public available source Kaggle, which has 32 attributes, 1205 instances with below list of Attributes or column names that are considered:
1) ID number
2) Diagnosis (1 represents malignant, 0 represents benign (not harmful in effect))
Below features are computed for each cell nucleus:
a) Radius (it’s the mean of distances from centre that points on to the perimeter)
b) texture or consistency
c) Perimeter or boundary
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of increasing/decreasing the shape)
h) concave distance
i) Symmetry
j) statistical index of complexity detail in a pattern (fractal dimension)
Class distribution: 848 benign, 357 malignant
Step 1: Handle Missing data values:
Data Dropping: This method is used to remove features or attributes with missing values from the dataset. Drop observations with missing values, Drop columns with missing values.
Mean/Median Imputation: This method is used to replace missing values of a given attribute with the mean ad median of the attribute which has non-missing values.

Step 2: Data preprocessing is one of the step in training a model which is applied on the original breast cancer data set in order to check for outliers, discrete feature processing is performed shown in Fig.18. Normalization technique is applied on the predicted sequences so that the predicted values lie between 0-1. Using the below equation:

$$S = \frac{S_i - S_{\min}}{S_{\max} - S_{\min}} \qquad (1)$$

where,

S_i = data which follows a standard,

S_{min} = minimum value in the entire data

S_{max} = maximum value in the entire data

Step 3: The processed prediction data from step-1 is given to LightGBM model. LightGBM model has 0.1 learning rate and the maximum depth is considered 10 along with no. of leaves as 40 figure-19.

Step 4: The LSTM model has two built in hidden layers. The first layer has 256 neurons and the second has 512 neurons. A Dropout layer is added in order to make the model generalized by putting the ratio to 0.3, the epoch to 256, and the Batch Size to 13.

Step 5: Based on the prediction results of the two models, ωi is the weight coefficients between models are compared using below equation and obtained result is shown in Table.2.

$$d_t = \omega_1 d^1{}_t + \omega_2 d^2{}_t, t = 1,2,3,\ldots n$$

$$\omega_1 = \frac{\delta_2}{\delta_1 + \delta_2}$$

$$\omega_2 = \frac{\delta_1}{\delta_1 + \delta_2} \qquad 2)$$

where,

$d^1{}_t$, $d^2{}_t$ = values that are predicted using LightGBM and LSTM models

ω_i = weight coefficient

δ1, δ2 = errors between models.

5 Results

The Breast cancer dataset is considered, the Breast Cancer is seen common among many women around the world. Millions of people have been affected in 2022 year and in the previous years. Breast cancer initially is been noticed when the breast size grows out of control and they lead to tumours which are felt as lumps in the breast area. The goal is to detect the tumour and classify them into harmful/malignant (cancerous) or benign/not harmful (non-cancerous). So, analysis has been performed on classifying these tumours using machine learning algorithms.

5.1 Dataset

The dataset consists of 32 attributes, 1205 instances, and 32 attributes listed below;
List of Attributes or column names that are considered:

1) ID number
2) Diagnosis (1 represents malignant, 0 represents benign (not harmful in effect))
Below features are computed for each cell nucleus:
a) Radius (it's the mean of distances from centre that points on to the perimeter)
b) texture or consistency
c) Perimeter or boundary
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of increasing/decreasing the shape)
h) concave distance
i) Symmetry
j) statistical index of complexity detail in a pattern (fractal dimension)
Class distribution: 848 benign, 357 malignant

5.2 Procedure for execution

5.2.1 Importing necessary modules and libraries

5.2.2 Load the data, Data Processing and Analysis:

All the necessary datasets which are in .csv files are loaded as shown in Fig.18. Processing and Analysis is applied on dataset such that the training data is compared along with test data to see accuracy score. Training data set is the biggest (in -size) subgroup of the actual dataset, which is considered to train the machine learning model. The test dataset is another subgroup or part of original data, which acts as independent when compared with the training dataset. The accuracy for the considered set is

Training set score: 1.0000
Test set score: 0.9298

5.2.3 Denoising

To remove the volatile data denoising is done, but some information from the original time series may be lost. Wavelet Denoising: Usually performed with electric signals, to remove the unwanted noise from a time series. Wavelet denoising calculates coefficients called the "wavelet coefficients". These coefficients decide what parts of information needs to be present (signal) and which ones to removed (noise). Mean Absolute Deviation value checks the randomness in the data and accordingly decide the minimum threshold for the wavelet coefficients in the time series. We filter out the low coefficients from the wavelets and reconstruct the breast cancer data set data from the remaining coefficients and this concludes in removing noise from the original data.

5.2.4 Data Exploration

Exploring all the datasets or all the .csv files, operations like merge, compare and sort can be applied on different attributes and different datasets.

5.2.5 Training a Model

The performance of Light GBM is shown using Confusion matrix, this summarizes the variations of a machine learning model on test dataset. A model is measured based on its accuracy using train/test data set. 80% or 70% of data is considered for training and 20% or 30% of data is considered to test the entire data available. Based on the percentage of consideration, to train and test a model both train and test data set is considered. Training a model creates a new model and testing a model leads to test the accuracy of the model.

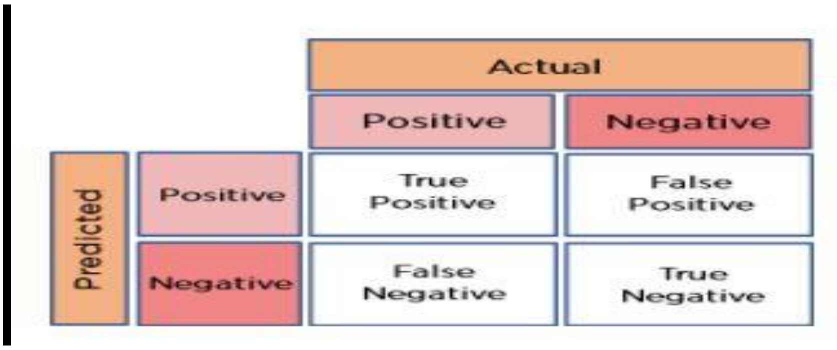


Fig.15: Representation of confusion matrix

The different possible outcomes of the prediction and the classification problems are visualized using a confusion Matrix. The classifier gives both actual values and the predicted values and is represented as shown below Fig.15.

Confusion Matrix:

$$\begin{bmatrix}104 & 4\\ 8 & 55\end{bmatrix}$$

True Positives (TP) = 104
True Negatives (TN) = 55
False Positives (FP) = 4
False Negatives (FN) = 8

A *true positive* explains the true/exact prediction of positive chances of a model and a *true negative* is an outcome of the model that predicts correctly the *negative* class. A *false positive* represents the outcome of the model that predicts incorrectly the positive class whereas the *false negative* gives the outcome of the model that predicts incorrectly the negative class.

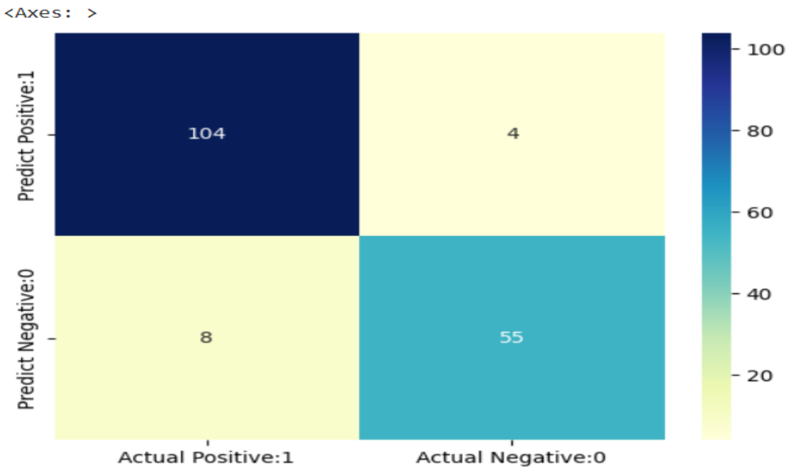
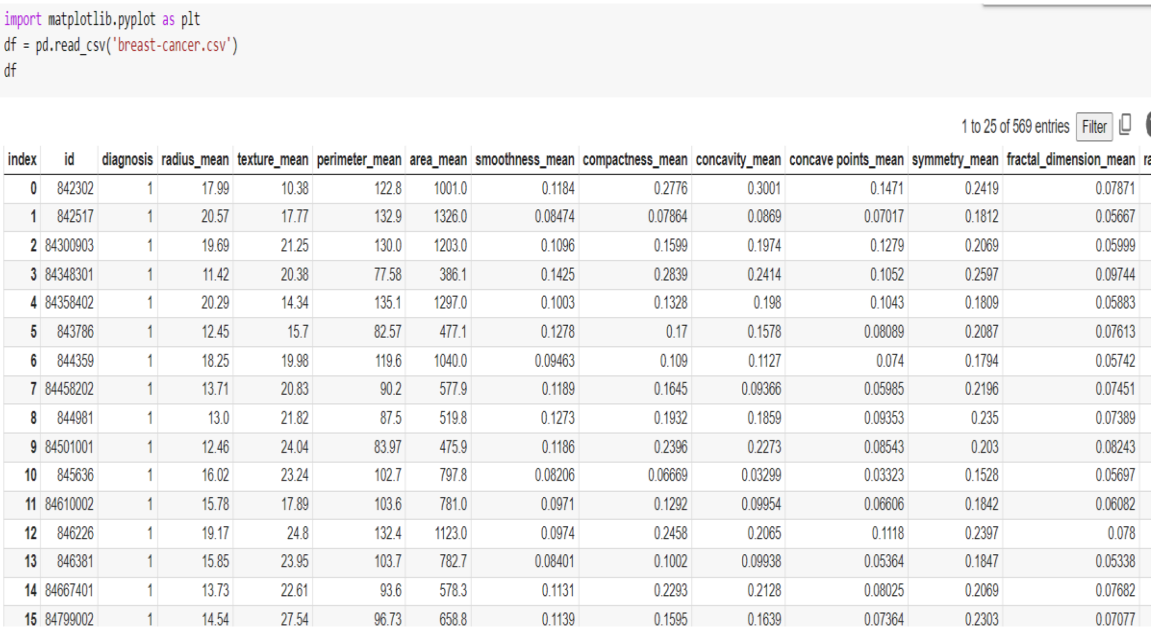


Fig.16: Confusion Matrix resulted using Light GBM

Fig.16 explains the model performance is evaluated (confusion matrix) using Light GBM, this summarizes different variations of a machine learning model when applied on test dataset where the parameters used are shown below; both training and test datasets are considered and predictions are done based on sorting the column names and describing the dataset with describe( ) displays the basic statistical details like percentile, mean, standard deviation of a data frame.Light GBM Classification report of a model is shown in Table.1.

	precision	recall	f1-score	support
0 (predicted negative)	0.93	0.96	0.95	108
1 (predicted positive)	0.93	0.87	0.9	63
accuracy			0.93	171
macro average	0.93	0.92	0.92	171
weighted average	0.93	0.93	0.93	171

Table.1: Classification Report using Light GBM



```
import matplotlib.pyplot as plt
df = pd.read_csv('breast-cancer.csv')
df
```

1 to 25 of 569 entries

index	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean	ra
0	842302	1	17.99	10.38	122.8	1001.0	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	
1	842517	1	20.57	17.77	132.9	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	
2	84300903	1	19.69	21.25	130.0	1203.0	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	
3	84348301	1	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	
4	84358402	1	20.29	14.34	135.1	1297.0	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	
5	843786	1	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	
6	844359	1	18.25	19.98	119.6	1040.0	0.09463	0.109	0.1127	0.074	0.1794	0.05742	
7	84458202	1	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	
8	844981	1	13.0	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	
9	84501001	1	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	
10	845636	1	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	
11	84610002	1	15.78	17.89	103.6	781.0	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	
12	846226	1	19.17	24.8	132.4	1123.0	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	
13	846381	1	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	
14	84667401	1	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	
15	84799002	1	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	

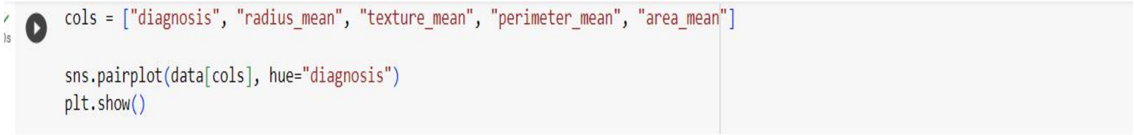
Fig.17: Loading and previewing the dataset using LSTM

Overall, the evaluation of the model using LSTM results with the following accuracy: 0.9357 and loss: 0.1527

Classification Report of LSTM:

	precision	recall	f1-score	support
0 (predicted negative)	0.92	0.96	0.94	115
1 (predicted positive)	0.93	0.88	0.9	73
accuracy			0.93	188
macro average	0.93	0.92	0.92	188
weighted average	0.93	0.93	0.93	188

Table.2: Classification reports of both algorithms/models



```
cols = ["diagnosis", "radius_mean", "texture_mean", "perimeter_mean", "area_mean"]

sns.pairplot(data[cols], hue="diagnosis")
plt.show()
```

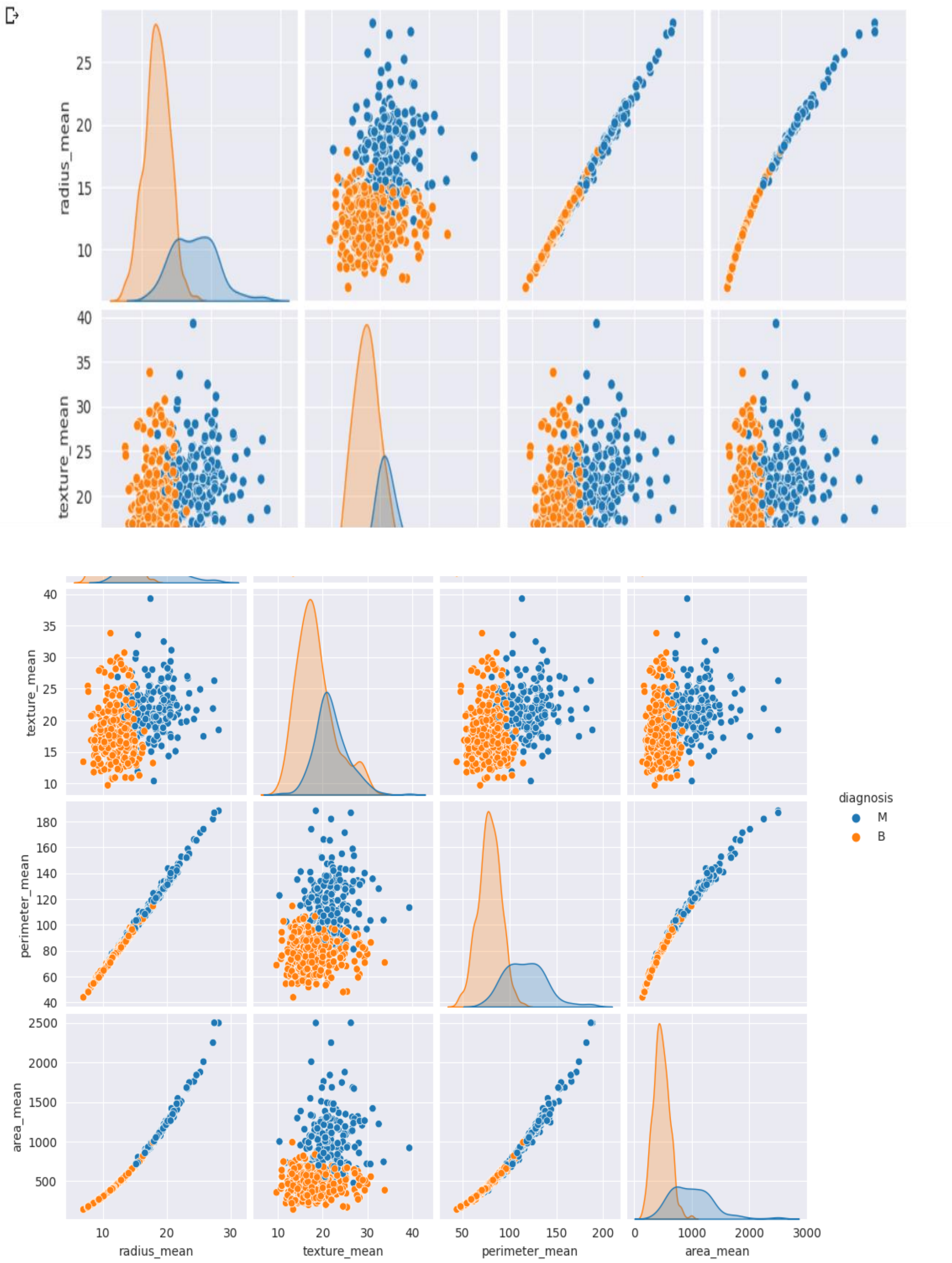


Fig. 18: Data Visualization performed on the column names (radius_mean, texture_mean, perimeter_mean)

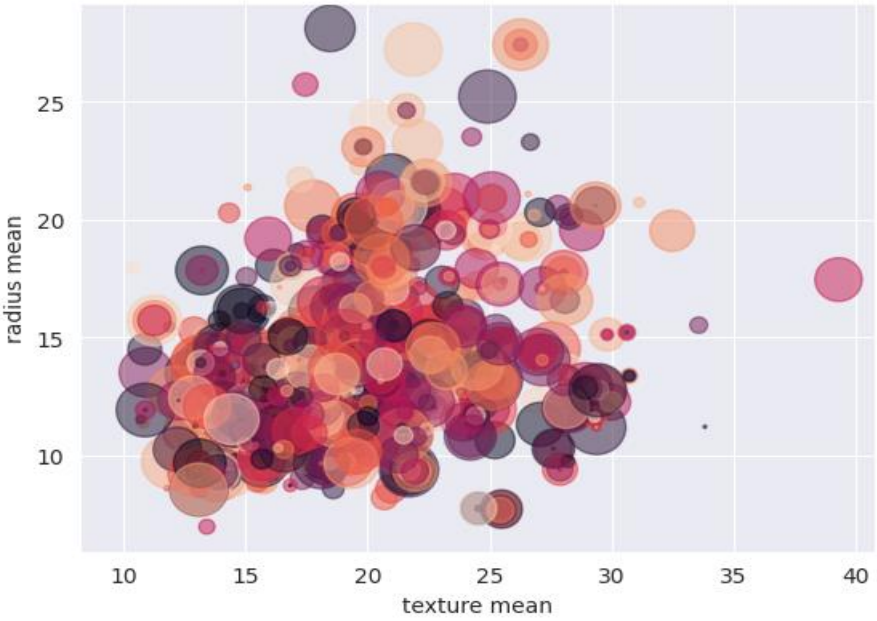


Fig.19: Data Visualization performed on the dataset having texture_mean and radius_mean on x and y axis

6 Conclusions

A serial combination of the models, LightGBM and LSTM is used by us to categorize the elements from the breast cancer dataset of size 1205 available in Kaggle to classify the images into two categories of malignant and benign with excellent results. This approach first puts together the detailed analysis ability of LightGBM and the pattern-finding skill of LSTM. The prediction through this approach of serial combination is more precise and shows an accuracy of 93% as measured through the F1-score. The precision and recall scores are also high with values of 93% and 92% respectively.

References

[1] R. V. Prakash, Govardhan, and S. S. V. N. Sarma, (2011) “Mining frequent item sets from large data sets using genetic algorithms,” IJCA Special Issue on “Artificial Intelligence Techniques–Novel Approaches &Practical Applications” AIT.

[2] S. Madden, (2012) “From Databases to Big Data,” IEEE Internet Computing, 16(3): 4–6, May–June.

[3] M.J. Zaki, (1999) “Parallel and distributed association mining: a survey,” IEEE Concurrency, 7(4): 14–25, Oct.–Dec.

[4] M. Yuan, Y. X. Ouyang, and Z. Xiong, (2013), “A text categorization method using extended vector space model by frequent term sets,” Journal of Information Science and Engineering, 99–114.

[5] H.D.Kim, D.H.Park, and Y.L.C.X.Zhai, (2012) “Enriching text representation with frequent pattern mining for probabilistic modeling”ASIST2012, October26–31, Baltimore, MD, USA.

[6] P. Nancy and R. G. Ramani, (2012), “Frequent pattern mining in social network data (face book application data)” European Journal of Scientific Research, ISSN 1450–216X, Vol.79No.4(2012), pp.531–540, Euro Journals Publishing.

[7] Y. Wang and J. Ramon, (2012) “An efficiently computable support measure for frequent sub graph pattern mining,”–ECML/PKDD, 362–377.

[8] Prasad, K.S.N., and Ramakrishna, S., (2011), “Frequent pattern mining and current state of the art,” International Journal of Computer Applications (0975 – 8887), July - Vol.26, No.7.

[9] A. K. Koundinya, N. K. Srinath, K. A. K. Sharma, K. Kumar, M. N.Madhu, and K. U. Shanbag, (2012), “Map/Reduce design and implementation of Apriori algorithm for handling voluminous data-sets”, Advanced Computing: An International Journal (ACIJ), November -Vol.3, No.6.

[10] S. N. Patro,S. Mishra, P.Khuntia, and C. Bhagabati, (2012), “Construction of FP tree using huffman coding” IJCSI International Journal of Computer Science Issues, Vol.9, Issue3, No.2.

[11] A.Appice,M. Ceci, and A. T. D. Malerba, (2011), “A parallel, distributed algorithm for relational frequent pattern discovery from very large datasets” Intelligent DataAnalysis15, pp.69–88.

[12] C. Xiaoyun, H. Yanshan, C. Pengfei, M. Shengfa, S. Weiguo, and Y.Min, (2009), “HPFP-Miner: a novel parallel frequent item set mining algorithm” Natural Computation, ICNC -09, Fifth International Conference, Aug-2009.

[13] A. Niimi, T. Yamaguchi, and O. Konishi, (2010), “Parallel computing method of extraction of frequent occurrence pattern of sea surface temperature from satellite data” The Fifteenth International Symposium on Artificial Life Robotics 2010, AROB 15th'10, Japan, February 4–6,.

[14] D. Picado-Muiño, I. C. León, and C. Borgelt, (2012), “Fuzzy frequent pattern mining in spike trains”, IDA2012:289–300

[15] A. Ebenstein, M. Fan, M. Greenston, G. He, and M. Zhou,(2013), “New evidence on the impact of sustained exposure to air pollution on life expectancy from China’s huai river policy,” Proceedings of the National Academy of Sciences of the United States of America, vol. 110, pp. 12936–12941.

[16] J. L. Elman, (1990), “Finding structure in time,” Cognitive Science, vol. 14, no. 2, pp. 179–211.

[17] V. Ferrtti and G. Montibeller, (2016), “Key challenges and meta choices in designing and applying multi-criteria spatial decision support systems, Decis,” Support Syst, vol. 84, pp. 41–52.

[18] S. Bhattacharyya, V. Snasel, A. E. Hassanian, S. Saha and B. K. Tripathy, (2020), Deep Learning Research with Engineering Applications, De Gruyter Publications, ISBN: 3110670909, 9783110670905.

DOI: 10.1515/9783110670905

[19] Adate, A., and Tripathy, B. K., (2018), Deep learning techniques for image processing, In S. Bhattacharyya, H. Bhaumik, A. Mukherjee & S. De (Eds.), Machine Learning for Big Data Analysis, Berlin, Boston: De Gruyter, pp. 69–90. DOI: 10.1515/9783110551433-00357

[20] Ankita Bose and B. K. Tripathy, (2020), Deep Learning for Audio Signal Classification, (Ed: S. Bhattacharyya, A. E. Hassanian, S. Saha and B. K. Tripathy,

Deep Learning Research and Applications), De Gruyter Publications, pp. 105-136. DOI: 10.1515/9783110670905-00660.

[21] Satin Jain, Udit Singhania, B.K. Tripathy, Emad Abouel Nasr, Mohamed K. Aboudaif and Ali K. Kamrani, (2021), Deep Learning based Transfer Learning for Classification of Skin Cancer, Sensors (Basel), Dec 6- 1(23):8142.
doi: 10.3390/s21238142,

[22] N. H. A. Rahman, S. H. Lee, and M. T. Latif, (2015), "Artificial neural networks and fuzzy time series forecasting: an application to air quality," Quality & Quantity, vol. 49, no. 6, pp. 1–15.

[23] M. S¸aylı and E. Yılmaz, (2017) , "Anti-periodic solutions for state dependent impulsive recurrent neural networks with time varying and continuously distributed delays," Annals of Operations Research, vol. 258, no. 1, pp. 159–185.

[24] J. Schwartz, (1993), "Particulate air pollution and chronic respiratory disease" Environmental Research, vol. 62, no. 1, pp. 7–13.

[25] Sepp, H., and Schmidhuber,J. (1997) "Long short-term memory." Neural computation 9.8, pp: 1735-1780.

[26] Taylor, S. J., and Letham, B. (2018), Fore casting at scale. The American Statistician, 72(1), 37-45

[27] D. Wang, Y. Zhang, and Y. Zhao, (2017), "Light gbm: An effective mirna classification method in breast cancer patients," in Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics, ser. ICCBB 2017. New York, NY, USA: ACM, pp. 7–11. http://doi.acm.org/10.1145/3155077.3155079

[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, (2017), "Light gbm: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., pp. 3146–3154.

[29] K. Alsabti, S. Ranka, and V. Singh, (1998), "Clouds: A decision tree classifier for large datasets," in Proceedings of the 4th Knowledge Discovery and Data Mining Conference, pp. 2–8.

[30] D. Wang, Y. Zhang, and Y. Zhao, (2017), "Lightgbm: An effective mirna classification method in breast cancer patients," in Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics, ser. ICCBB 2017. New York, NY, USA: ACM, pp. 7–11.

[31] X. Sun, M. Liu, and Z. Sima, (2018) "A novel cryptocurrency price trend forecasting model based on lightgbm," Finance Research Letters.

[32] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, , (2018), "Study on a prediction of p2p network loan default based on the machine learning lightgbm and xgboost algorithms according to different high dimensional data cleaning," Electronic Commerce Research and Applications, vol. 31, pp. 24 – 39.

[33] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, (2017), "Lightgbm: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., pp. 3146–3154.

[34] P. Harrington, (2012), Machine Learning in Action. Greenwich, CT, USA: Manning Publications Co.,

[35] R. J. Hyndman and A. B. Koehler, (2006), “Another look at measures of forecast accuracy,” International Journal of Forecasting, pp. 679–688.

[36] K. S. Ser Gill, Sharmila Banu, K. And B. K. Tripathy (2018): An Improved Differential Neural Computer Model Using Multiplicative LSTM. In: Zelinka I., Senkerik R., Panda G., Lekshmi Kanthan P. (eds) Soft Computing Systems, ICSCS 2018, Communications in Computer and Information Science, vol 837, Springer, Singapore. https://doi.org/10.1007/978-981-13-1936-5_31

[37] A. Adate and B. K. Tripathy, (2019), S-LSTM-GAN: Shared Recurrent Neural Networks with Adversarial Training, In: Kulkarni A., Satapathy S., Kang T., Kashan A. (Eds) Proceedings of the 2nd International Conference on Data Engineering and Communication Technology. Advances in Intelligent Systems and Computing, Vol. 828, Springer, Singapore, pp 107-115.

[38] Kumar V and B. K. Tripathy (2020): Detecting Toxicity with Bidirectional Gated Recurrent Unit Networks. In: Bhateja, V., Satapathy, S., Zhang, YD., Aradhya, V. (eds) Intelligent Computing and Communication, ICICC 2019, Advances in Intelligent Systems and Computing, vol 1034, Springer, Singapore. https://doi.org/10.1007/978-981-15-1084-7_57

[39] K. Baktha and B. K. Tripathy, (2017), Investigation of recurrent neural networks in the field of sentiment analysis, 2017 International Conference on Communication and Signal Processing (ICCSP), 2017, pp. 2047-2050, doi: 10.1109/ICCSP.2017.8286763.

[40] Hochreiter, Sepp, and JrgenSchmidhuber, (1997),”Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.

[41] S. Hochreiter and J. Schmidhuber, (1995) “Long short-term memory,” Dept. Fakultät für Informatik, Tech. Univ. Munich, Munich, Germany, Tech. Rep. FKI-207-95, Aug. 1995.

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.3117

[42] S. Hochreiter and J. Schmidhuber, (1997) “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[43] Wang, Yu, and Xiaoxi Zhu., (2016),” A Supervised Adaptive Learning-based Fuzzy Controller for a non-linear vehicle system using Neural Network Identification.” American Control Conference (ACC), Boston.

[44] Wang, Yu., (2014),” Design of Triple-Level Multiple Models Fuzzy Logic Controller for Adaptive Speed Control with Unknown External Disturbances.” IFAC Proceedings Volumes 47.3 (2014): 6326-6331.

[45] Han, Zhuo, and Kumpati S. Narendra, (2012),” New concepts in adaptive control using multiple models.” IEEE Transactions on Automatic Control 57.1, pp: 78-89.

[46] Narendra, Kumpati S., Yu Wang, and Wei Chen., (2015),” The Rationale for Second Level Adaptation.” arXiv preprint arXiv:1510.04989

[47] Narendra, Kumpati S., Yu Wang, and Wei Chen., (2014),” Stability, robustness, and performance issues in second level adaptation”, American Control Conference. IEEE.

[48] Narendra, Kumpati S., Yu Wang, and Wei Chen., (2015),” Extension of second level adaptation using multiple models to SISO systems” American Control Conference (ACC), IEEE- 2015

[49] Taylor, S. J., and Letham, B. (2018). Fore casting at scale. The American Statistician, 72(1), 37-45.

[50] Application of Face book's Prophet Algorithm for Successful Sales Forecasting Based on Real-world Data. International Journal of Computer Science and Information Technology-2020, 12(2), 23–36. doi:10.5121/ijcsit.2020.12203

[51] Liu, X., Zheng, L., Zhang,W., Zhou,J., Cao,S., Yu,S., (2022), An Evolutive Frequent Pattern Tree-based Incremental Knowledge Discovery Algorithms, ACM Transactions on Management Information Systems,Volume 13, Issue 3, Article No: 30pp 1–20,https://doi.org/10.1145/3495213, Published:04 February 2022.

[52] Komarasay, D., Gupta, M., Murugesan, M., Hermina, J., Gokuldhev, ,(2022), Mining Frequent Pattern Mining in Big Data Using Integrated Frequent Pattern (IFP) Growth Algorithm, September 2022, DOI:10.1007/978-981-19-2225-1_38, Proceedings of the 6th International Conference on Advance Computing and Intelligent Engineering (pp.423-431)

[53] Reihaneh H. Hariri, Erik M. Fredericks, Kate M. Bowers, (2019), Uncertainty in big data analytics: survey, opportunities, and challenges, Journal of Big Data volume 6, Article number: 44

[54] Davashi, R., (2021)UP-tree & UP-Mine: A fast method based on upper bound for frequent pattern mining from uncertain data, Engineering Applications of Artificial Intelligence, Volume 106, November 2021, 104477

[55] Chen,J., Li,P., Fang,W., Zhou,N.,Yin,Y., Zheng,H., Xu,H., and Wang, R., (2021), Fuzzy Frequent Pattern Mining Algorithm Based on Weighted Sliding Window and Type-2 Fuzzy Sets over Medical Data Stream, Wireless Communications and Mobile Computing, Volume 2021, Article ID 6662254, https://doi.org/10.1155/2021/6662254

[56] Rage,U.K., Palla,L., Dao,M., (2021), Discovering Periodic-Frequent Patterns in Very Large Uncertain Temporal Databases, September 2021, Conference: ICONIP 2021, National Institute of Information and Communications Technology.

[57] Rahman,M,M., Ahmed,C,F., Leung,C,K., (2019), Mining weighted frequent sequences in uncertain databases, Information Sciences, Volume 479, April 2019, Pages 76-100

[58] Kumar,S., Mohbey,K,K., (2022), A review on big data based parallel and distributed approaches of pattern mining, Journal of King Saud University - Computer and Information Sciences Volume 34, Issue 5, pp 1639–1662, https://doi.org/10.1016/j.jksuci.2019.09.006

[59] Ding,S., Li,Z., Zhang,K., and Mao,F.,Jiménez,F., (2022), A Comparative Study of Frequent Pattern Mining with Trajectory Data, Oct- 22(19): 7608, doi: 10.3390/s22197608

[60] Alibasa,M,J., Calvo,R,A., Yacef,K., (2022), Predicting Mood from Digital Footprints Using Frequent Sequential Context Patterns Features, June-2022, Pages: 2061-2075

[61] Al-Hamadi,M., Ghillan,M.,Saeed,F., (2018), An Enhanced Accelerator Frequent Pattern Growth for Association Rules Mining, June-2022, Pages: 193-205

[62] Karmelia,M.E., Widjaja,M., and Hansun,S., (2022), Candlestick Pattern Classification Using Feedforward Neural Network, July-2022, Pages: 80-95

[63] Aoulalay.A., Mhouti,A,E., Massar,M., Fahim,M., Borji,Y.E., (2023), Classification of Moroccan decorative patterns based on computer vision approaches using complex datasets, March-2023, Pages: 31-48.

Notes on contributors

G. Divya Zion, Research Associate at VIT-Vellore. She is working as Associate Professor in the Department of Artificial Intelligence & Data Science having 11 years of teaching experience. She has completed her internship at Indian Space Research Organization., completed her Masters on computer Science & Engineering in 2012.

Dr. B.K. Tripathy, a triple gold medalist, is a Professor (HAG) at VIT, Vellore. He has supervised 52 research degrees and is a Fellow of IETE and IEI besides being a Senior member of IEEE, ACM, IRSS, ACEEE and CSI. He is a distinguished researcher in the fields of Mathematics and Computer science leading to publication of over 750 articles in reputed journals, international conference proceedings, Edited volumes and books/monographs. His name is included in the list of top 2% scientists in the world as per the data generated by Stanford University, USA. His areas of interest include rough sets, Fuzzy sets, social networks, Machine Learning, Soft Computing, Granular computing, MCDM, Neighbourhood systems, SIoT, Big Data Analytics, Deep Neural Networks, Blockchain Technology, Soft Sets. And Neutrosophic sets.