ISSN: 2584-0495 Vol. 2, Issue 8, pp. 1140-1146

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

Comparative Analysis of Machine Learning Models for Disease Detection using Blood Cell Images

Vikram Kakade, Amol P. Bhagat, M. S. Ali and Sachin Deshmukh

Cite as: Kakade, V., Bhagat, A. P., Ali, M. S., & Deshmukh, S. (2024). Comparative Analysis of Machine Learning Models for Disease Detection using Blood Cell Images. International Journal of Microsystems and Iot, 2(8), 1140-1146. https://doi.org/10.5281/zenodo.13729385

Published online: 20 August 2024

DOI: https://doi.org/10.5281/zenodo.13729385

International Journal of Microsystems and IoT
ISSN: 2584-0495; Vol. 2, Issue 8, pp. 1140-1146; DOI: https://doi.org/10.5281/zenodo.13729385

Comparative Analysis of Machine Learning Models for Disease Detection using Blood Cell Images

Vikram Kakade[1], Amol P. Bhagat[2], M. S. Ali[1] and Sachin Deshmukh[3]

[1]Research Center Electronics and Telecommunication Engineering, Prof Ram Meghe College of Engg. & Mgmt., Badnera, Amravati, India
[2]Research Center Information Technology, Prof Ram Meghe College of Engg. & Mgmt., Badnera, Amravati, India
[3]Research Center Computer Science and Engineering, Prof Ram Meghe College of Engg. & Mgmt., Badnera, Amravati, India.

ABSTRACT
This paper presents machine learning algorithms for blood test parameters estimation. The existing deep learning approaches for blood test parameters estimation are presented in this paper. The proposed machine learning algorithms for estimation of blood test parameters using blood cells images are described. The proposed deep learning algorithm for estimation of blood test parameters using blood cells images are described. The results and evaluations of the proposed model are presented. The machine learning algorithms decision trees, random forest, LightGBM, and SVM are proposed for blood test parameters estimation. While deep learning algorithm CNN is proposed for blood test parameters estimation in this research work.

KEYWORDS
Blood cells, decision trees, deep learning, machine learning, random forest, LightGBM, SVM

1. INTRODUCTION

An In recent times, the world has been gripped by the relentless surge of a new strain of coronavirus, causing a global pandemic that has wreaked havoc on a scale of unprecedented proportions. A powerful classification model is created [Mary and Raj, 2023] that successfully generates the predictive values of infected patients with doubtful symptoms and epidemiological history to overcome this. [Garcia et al., 2022] have put forth a multiple linear regression model that aims to estimate the Carotid-to-femoral pulse wave velocity (cf-PWV) using a single, non-invasive peripheral pulse wave, such as blood pressure or photoplethysmography. An ensemble approach is presented in [Yadav and Mittal, 2022] for health risk prediction based on the steepest descent algorithm and decision trees.

[Jalaleddini et al., 2020] set out with the objective of detecting and eliminating subpar quality beats from the Transcranial Doppler (TCD) signal before diving into a more in-depth analysis. They selected particular features to accomplish this, such as the Euclidean distance and cross-correlation between individual and average beat waveforms, the fraction of high-frequency power to total beat power, beat duration, and the variation of the diastolic part of the beat waveform. On the other hand, [Hossain et al., 2020] aimed to create a system capable of non-invasively measuring blood glucose levels. Given that blood glucose is a crucial health indicator, their system could potentially benefit a large number of people who need to frequently monitor their blood glucose levels.

Convolutional Neural Networks (CNN), a cornerstone of Deep Learning (DL), have recently garnered attention for their prowess in pattern recognition, computer vision, and as a technique for Artificial Intelligence (AI). The framework of neural networks is the primary vehicle for implementing machine learning. Within the sphere of deep learning, a neural network is a layered structure. Convolutional layers are distinguished by the data's journey through an array of filters, facilitating efficient pattern recognition. In contrast to conventional machine learning algorithms that extract features before learning, a convolutional neural network incorporates the image directly into the learning process. The classification procedure can be hindered by the similarities among the differences in perspective relative to other classes and their own.

[Al-Mualemi and Lu, 2020] set their sights on creating and applying a deep learning (DL) oriented technique with the ability to forecast occurrences of septic shock and severe sepsis, and to assess its influence on healthcare procedures and patient results. [Shinde and Rajeswari, 2021] offer an exploratory examination of current cuff-less BP estimation methods, discussing their strengths and potential areas for enhancement. An elaborated review of existent data mining techniques used against the foretelling of diabetic illness is presented in [Shabtari et al., 2021]. It also gives succeeding path towards acuteness estimation of diabetic illness using data mining tool RapidMiner.

[Mollura et al., 2020] have crafted a short-term prediction algorithm aimed at pinpointing patients suffering from septic shock within a group of 100 septic patients, obtained from the MIMIC-III clinical and waveform database. [Banerjee et al., 2022] have proposed the blueprint for a quality checker,

designed to assess the quality of the signal and capable of being directly implemented on edge devices like a smartwatch. This algorithm has been put to the test on PPG data gathered from wearable, ICU, and medical-grade devices. Despite significant advancements, there is still a pressing need for a swift, sturdy, and easily accessible method for BP sensing in the burgeoning point-of-care market. To address this issue, [Byfield et al., 2022] introduce a BP measurement unit, built based on two photoplethysmography (PPG) sensors, from which the pulse wave velocity (PWV) of blood flow can be estimated.

2. MACHINE LEARNING ALGORITHMS FOR BLOOD TEST PARAMETER ESTIMATION

In this scholarly endeavor, an array of Machine Learning (ML) methodologies - Decision Trees (DT), Random Forests (RF), LightGBM, and Support Vector Machines (SVM) - are employed for the computation of blood test parameters. Decision trees are like the flowcharts of the algorithm world. Imagine starting at the top with a question, and based on the answer, follow branches until a decision is reached. This tool serves as a valuable asset for executing classification and regression operations in the realm of ML. Decision trees, a widely utilized ML algorithm, are adept at performing both classification and regression tasks. They operate by posing a sequence of queries related to the data's attributes. Every node in the tree symbolizes a decision contingent on a particular feature, while the branches signify the potential results. The process continues until a decision or prediction is made at a leaf node. It's a powerful and interpretable way to make sense of complex data.

The decision trees algorithm uses measures like entropy to evaluate the disorder or uncertainty in a set of data. It aims to find the features that, when used as questions, reduce this uncertainty the most. This reduction in uncertainty is called information gain. Another criterion is the Gini index, which measures the impurity of a set. The algorithm looks for features that minimize the Gini index when creating splits. In predicting numerical values instead of classes, the algorithm uses mean squared error (MSE) to evaluate how well a split predicts the target variable.

The decision tree grows by repeatedly splitting the data based on selected attributes until it reaches a stop condition, such as a fixed maximum depth or a minimum number of samples in a leaf node. It handles both categorical and continuous features. For categorical features, it's like asking yes/ no questions. For continuous features, it looks for the best threshold to split the data. Decision trees can easily memorize the training data (overfit), so techniques like pruning or setting minimum sample requirements for splitting help prevent this. Pruning involves removing branches that don't contribute much to the model's performance.

The decision tree algorithm looks at all the features in data and chooses the one that best splits the data into distinct groups. The chosen feature becomes a node, and the data is split into subsets based on its values. Each subset is now a branch. The algorithm goes deeper, repeating the process for each subset. It keeps asking the best questions at each node until it either reaches a decision or decides its gone deep enough. The end nodes, or leaves, contain the final decision or prediction. During training, the algorithm learns the best questions and splits by analyzing the features and outcomes in training data. Sometimes, to prevent overfitting, branches that don't contribute much are pruned.

To utilize the decision trees algorithm, one can assemble a dataset encompassing blood test parameters (such as levels of cholesterol, glucose, and so forth) along with other pertinent attributes like age, gender, and potentially lifestyle factors. The data undergoes a cleaning process, where missing values are addressed and categorical variables are encoded if necessary. Decision trees are adept at managing a blend of numerical and categorical features. The dataset is partitioned into training and testing subsets. The training subset instructs the algorithm, while the testing subset assesses its efficacy..

Training data is feed into the decision tree algorithm. It will learn how different parameters relate to each other and how they can be used to estimate the blood test results. Depending on the particular library or tool employed, there may be parameters that can be adjusted, like the tree's utmost reach in depth or the smallest quantity of samples required for a node division. This helps prevent overfitting. Model is validated using the testing set. This gives an idea of how well it generalizes to new, unseen data. One of the cool things about decision trees is their interpretability. The path a prediction takes can be easily traced, making it clear which parameters are crucial in the estimation. If tree is too complex, it can be prune to avoid overfitting and improve generalization. This approach allows the algorithm to learn the relationships between different factors and blood test parameters, making it a valuable tool in healthcare analytics. The proposed decision trees algorithm is shown below:

Algorithm: Proposed Decision Trees Algorithm for Blood Test Parameters Estimation

1. Formulate Data df = pd.read_csv('dataset.csv')
2. Divide the Data X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
3. Build the Decision Tree Model dtree = DecisionTreeRegressor()
4. Use dtree.fit (X_train, y_train) to train the model
5. Generate Predictions predictions = dtree.predict(X_test)
6. Evaluate the Model

A Random Forest is a collaborative learning method that brings together a multitude of decision trees. Every single tree

is independently trained on a data subcategory and formulates its own predictions. For every tree in the forest, a random data subset is utilized for training, a process known as bootstrapping, which guarantees diversity among the trees. When constructing each tree, a random feature subset is considered at every split, adding an extra layer of randomness and aiding in the prevention of overfitting. For classification tasks, the forest's predictions are typically determined by a majority vote, while for regression tasks, the predictions are averaged.

The efficacy of Random Forests is rooted in their capacity to mitigate overfitting and enhance generalization by harnessing the power of numerous trees. They exhibit robustness, adeptly manage missing data, and offer feature importance, aiding in discerning which features play a pivotal role in predictions. Random Forests furnish a feature importance score that can be graphically represented to comprehend which blood test parameters hold the most sway in making predictions. The proposed Random Forest algorithm is depicted below:

Algorithm: Proposed Random Forest Algorithm for Blood Test Parameters Estimation

1. Import dataset df = pd.read_csv('dataset.csv')
2. Initiate data preprocessing, manage any absent values, and perform encoding on categorical variables as required
3. Partition the dataset into training and testing subsets using the following command: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
4. Construct the Random Forest Model using the following command:rf_model = Random Forest Regressor (n_estimators = 100, random_state = 42) Train the model rf_model.fit (X_train, y_train)
5. After the model has been trained, employ it to generate predictions on the test data using the following command: rf_predictions = rf_model.predict(X_test)
6. Assess the model's performance using appropriate metrics

Light GBM, alternatively referred to as Light Gradient Boosting Machine, is a gradient enhancement framework that utilizes tree-based learning algorithms. It's crafted for the distributed and efficient instruction of extensive datasets. Mirroring other gradient boosting platforms, Light GBM progressively builds an array of decision trees, where each subsequent tree aims to correct the errors made by its predecessor. Light GBM is fine-tuned for rapidity and effectiveness. It utilizes a learning method based on histograms, which categorizes continuous feature values into distinct bins, thereby diminishing memory consumption and accelerating the training procedure.

Unlike traditional tree algorithms that grow trees layer by layer, Light GBM nurtures tree's leaf by leaf. It opts for the leaf with the maximum delta loss to foster growth, thereby increasing the likelihood of identifying the optimal split. It adeptly handles categorical variables, thereby obviating the necessity for one-hot encoding. This proves to be a substantial benefit when working with datasets comprising a combination of numerical and categorical features. Light GBM is compatible with parallel and distributed training, making it ideal for large datasets and scalable across numerous machines. It integrates regularization techniques, namely L1 and L2 regularization, as a means to mitigate the issue of overfitting. Light GBM also supports GPU acceleration, further boosting its speed. The proposed LightGBM algorithm is depicted below:

Algorithm: Proposed LightGBM Algorithm for Blood Test Parameters Estimation

1. Import dataset df = pd.read_csv('dataset.csv')
2. Initiate the data pre-processing, addressing any missing values and implementing encoding for categorical variables as required
3. Partition the dataset into training and testing subsets X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_state = 42)
4. Create a Light GBM dataset train_data = lgb.Dataset(X_train, label=y_train) test_data = lgb.Dataset(X_test, label=y_test, reference=train_data)
5. Establish the parameters and initiate the model's training process.
6. Adjust the number of boosting rounds
7. Make predictions
8. Evaluate the model

SVMs are a flexible set of supervised machine learning models, proficient in performing both classification and regression tasks. The fundamental objective of an SVM is to discover a hyperplane that optimally partitions the data into distinct classes. In the context of classification, this hyperplane is designed to maximize the distance, or margin, between classes. This margin is defined as the distance from the hyperplane to the nearest data point from each class. The goal of SVMs is to identify the hyperplane that amplifies this margin. Support vectors are the data points that reside closest to the decision boundary (hyperplane) and play a pivotal role in determining the optimal hyperplane. SVMs have the ability to manage non-linear relationships by employing a kernel trick. Kernels transform the input features into a higher-dimensional space, enabling the identification of a hyperplane in this space that corresponds to a non-linear demarcation line in the initial attribute space.

In SVMs, the regularization parameter C governs the equilibrium between maintaining a seamless demarcation line and precise categorization of the training instances. A diminutive C results in a broader margin, albeit potentially

misclassifying some points, while an enlarged C strives for a constricted margin with fewer misclassifications. SVMs can handle a diverse range of kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid. The choice of kernel is contingent upon the attributes of the data and the problem under consideration. In certain scenarios, the data may not be linearly separable. Soft margin SVMs incorporate a slack variable that permits some misclassifications, thereby achieving a balance between maximizing the margin and minimizing misclassification. The proposed SVM algorithm is depicted below:

Algorithm: Proposed SVM Algorithm for Blood Test Parameters Estimation

1. Import dataset df = pd.read_csv('dataset.csv')
2. Pre-process the data, by addressing any gaps in values and transforming categorical variables as required
3. Partition the dataset into training and testing subsets using the following command: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
4. Given the sensitivity of SVMs to the magnitude of the features, it's advisable to implement scaling as a best practice.
5. Build and Train the SVM Model
6. Make Predictions
7. Evaluate the Model

The next section presents the deep learning approach proposed for blood test parameters estimation.

3. DEEP LEARNING ALGORITHM FOR BLOOD TEST PARAMETER ESTIMATION

In Convolutional Neural Networks (CNNs) represent a unique class of deep neural networks, purposefully engineered to handle structured grid data like images. They have demonstrated remarkable efficacy in a multitude of computer vision endeavors, encompassing image categorization, object identification, and image synthesis. CNNs utilize convolutional layers to independently and adaptively identify spatial feature hierarchies from the input data. The process of convolution slides a filter, or kernel, over the input to identify patterns ranging from simple edges and textures to more intricate features.

Pooling layers are employed to reduce the spatial scale of the input volume, thereby reducing the computational intricacy. Max pooling, a prevalent technique, selects the maximum value within a region, thereby retaining the most crucial information. Non-linear activation mechanisms, exemplified by ReLU (Rectified Linear Unit), are implemented to the output of convolutional and pooling layers. ReLU infuses non-linearity and aids the network in deciphering complex patterns. After the layers that filter and downsample the input, CNNs usually add one or more layers that connect every neuron for complex thinking. These layers establish a connection between every neuron in one layer to every neuron in the subsequent layer.

Prior to reaching the fully connected layers, the data undergoes a transformation into a vector. This step is crucial for transitioning from the spatial hierarchy gleaned by the convolutional layers to the flat input necessitated by the fully connected layers. CNNs are trained employing the backpropagation algorithm. The model is exposed to labeled training data, and the weights are iteratively tweaked to minimize the discrepancy between the predicted and actual outputs.

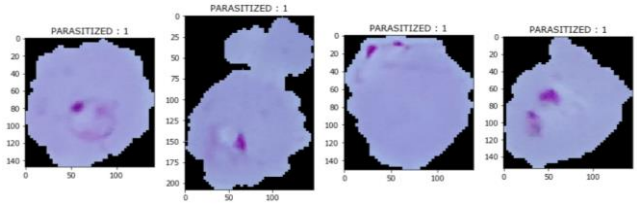


Fig. 1 Parasitized Blood Cells Images

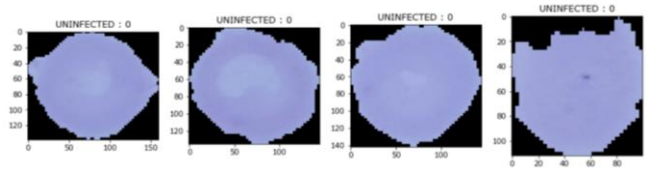


Fig. 2 Parasitized Blood Cells Images

Transfer learning entails the application of a pre-trained CNN on an extensive dataset, subsequently fine-tuning it for a particular task. This method becomes especially advantageous when faced with limited data. In the analysis of proposed deep learning approach, the used set of the parasitized and uninfected blood cells samples are shown in figure 1 and figure 2 respectively. The proposed SVM algorithm is shown below:

Algorithm: Proposed CNN Algorithm for Blood Test Parameters Estimation

1. Import dataset df = pd.read_csv('dataset.csv')
2. Initiate the data pre-processing, addressing any missing values and implementing encoding for categorical variables as required.
3. Partition the dataset into training and testing subsets using the following command: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42).
4. Normalize pixel values to be between 0 and 1
5. One-hot encode the labels.

6. **Construct the CNN Architecture**: This involves defining the structure of the Convolutional Neural Network (CNN) model.
7. **Assemble the Model**: This step involves setting up the model with appropriate configurations.
8. **Educate the Model:** Here, the model learns from the training data.
9. **Assess the Model's Performance:** Finally, the model's performance is evaluated using the test set

Every machine learning strategy outlined in the preceding segment, along with the deep learning methodology delineated here, undergo a thorough assessment in the subsequent section.

4. RESULTS AND EVALUATION

The proposed machine learning and deep learning approaches presented in previous sections are implemented in Anaconda Python 3.9 distribution in Jupyter Notebook. The python libraries pandas, sklearn, joblib, matplotlib, seaborn, cv2, os, keras, and tensorflow are utilized for implementing the proposed algorithms for blood test parameters estimation. The dataset comprising blood cells images is collected. The blood cell images dataset comprises 13780 parasitized and 13780 uninfected images, hence total 27560 blood cells images. The parasitized images are shown in figure 3 and uninfected images are shown in figure 4.

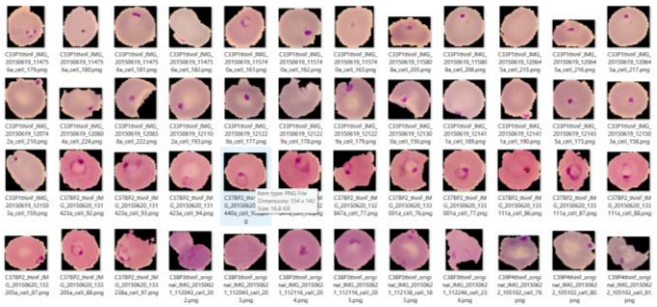


Fig. 3 Parasitized Blood Cells Images

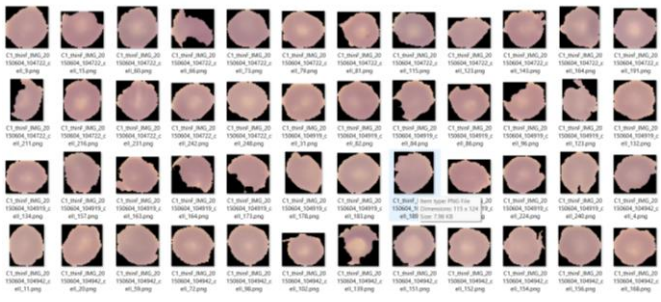


Fig. 4 Uninfected Blood Cells Images

The proposed approaches are implemented in Jupyter Notebook which can be started as shown in figure 3.8. In every machine learning methodology, the dataset is partitioned into training and test subsets for evaluation. A substantial 80% of the images are harnessed as training data, while the remaining 20% serve as test data for validation purposes. Rewrite creatively without plagiarism "The comparison of all the models on the basis of precision, recall and F1 score is shown in table 1. It can be observed from the comparison table all the machine learning algorithms has the precision of ~ 0.89 while the CNN algorithm performs better compare to the machine learning algorithms. The CNN has the precision of 0.94.

Table. 1 Comparison of ML and DL Algorithms for Blood Test Parameters Estimation

Computational Procedure	Accuracy	Recollection	F1-Score
D-Tree	0.895	0.90	0.90
R-Forest	0.895	0.90	0.90
LightGBM	0.895	0.895	0.90
Support Vector Machines	0.89	0.89	0.89
CNN	0.94	0.91	0.92

5. CONCLUSION

The machine learning algorithms decision trees, random forest, LightGBM, and SVM are proposed for blood test parameters estimation. While deep learning algorithm CNN is proposed for blood test parameters estimation in this research work. The proposed algorithms are implemented in Python and evaluated using Jupyter Notebook. These proposed algorithms are evaluated using various performance metrics. The assessment of model performance is conducted utilizing several measures such as the confusion matrix, accuracy, precision, recall, F1 score, and the AUC (area under the receiver operating characteristic curve) value, All the machine learning algorithms has the precision of ~ 0.89 while the CNN algorithm performs better compare to the machine learning algorithms. The CNN has the precision of 0.94.

REFERENCES

1. Mary, L.William and Raj, S. Albert A. (2023). RBTM: Prediction of COVID-19 from Random Blood Test with Machine Learning Algorithms. 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART).
2. Garcia, J.M.V., Bahloul, M.A. and Laleg-Kirati,T,M., (2022). A Multiple Linear Regression Model for Carotid-to-Femoral Pulse Wave Velocity Estimation Based on Schrodinger Spectrum Characterization. 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).
3. Yadav, A., Mittal,A.K. (2021). An Ensemble Machine Learning Based Approach for Health Risk Prediction. 2021 4th International Conference on Recent Trends in Computer Science and Technology (ICRTCST).
4. Jalaleddini, K., Canac, N., et. Al. (2020). Objective Assessment of Beat Quality in Transcranial Doppler Measurement of Blood Flow Velocity in Cerebral Arteries. IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, 67(3), 883-892.
5. Hossain, S. et.al.(2020). Estimation of Blood Glucose from PPG Signal Using Convolutional Neural Network. 2019 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON).

6. Al-Mualemi, B.Y. and Lu L. (2020). A Deep Learning Based Sepsis Estimation Scheme. IEEE Access, 9(1), 5442 – 5452.

7. Shabtari, M.M. et.al. (2021). Analyzing PIMA Indian Diabetes Dataset through Data Mining Tool RapidMiner. 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE).

8. Mollura, M. et.al.(2020). Prediction of Septic Shock Onset in ICU by Instantaneous Monitoring of Vital Signs. 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).

9. Banerjee,T.(2022). Design of a Realtime Photoplethysmogram Signal Quality Checker for Wearables and Edge Computing. 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).

10. Byfield et al., (2022). Towards Robust Blood Pressure Estimation From Pulse Wave Velocity Measured by Photoplethysmography Sensors," IEEE Sensors Journal.22(3), 2475 – 2483.

11. Shinde, S. and RajaRajeswari, P. (2021). CuffLess BP Stratification based on Bio-Signals processing using Machine Learning: An Investigative Study. 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC).

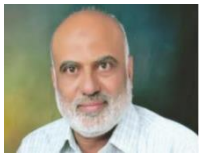
AUTHORS

Vikram Kakade received her B.E. degree in 2012 and M.E. degree in Electronics and Telecommunication in 2016 from PRMITR, Badnera, Amravati, Maharashtra, India. He served as Assistant Professor in PRMCEAM, Badnera from Jan 2013 to Nov 2021. He is working as Salesforce Developer in Springer Nature Technology and Publishing Solutions since Dec 2021. He is currently pursuing PhD at the Research Center Electronics and Telecommunication Engineering, Prof Ram Meghe College of Engg. & Mgmt., Badnera, Amravati, India. His areas of interest are image analysis, machine learning, artificial intelligence, and salesforce.

E-mail: vikramkakade.160@gmail.com

Amol P. Bhagat completed his B.E. (Information Technology) from Government College of Engineering, Amravati in the year 2005; with Master's degree in Computer Science and Engineering from Walchand College of Engineering, Sangli in the year 2009 and Ph.D. in Information Technology under the guidance of Dr. Mohammad Atique in the year 2016 and is a research supervisor since 2019. (Notification No. 22/2019 Dated: 14/03/2019). He is with Prof Ram Meghe College of Engineering & Management, Badnera- Amravati since the year 2010. He has to his credit: 24 Patents filed, 6 granted patents, 94+ Research papers published in National and International Journal and 5 books and 10 book chapters published. He is representing various Committees like Member of Board of Studies IT, Government Polytechnic Information Technology under the guidance of Dr. Mohammad Atique in the year 2016 and is a research supervisor since 2019. (Notification No. 22/2019 Dated: 14/03/2019). He is with Prof Ram Meghe College of Engineering & Management, Badnera- Amravati since the year 2010. He has to his credit: 24 Patents filed, 6 granted patents, 94+ Research papers published in National and International Journal and 5 books and 10 book chapters published. He is representing various Committees like Member of Board of Studies IT, Government Polytechnic Amravati, Technical Programme Committee Member in various International Conferences Organized by IEEE, Elsevier, Springer, etc. THE (Times Higher Education) World University Rankings, Advisory Board Member of BI and IIC, Editorial Board Member of International Journals, Session Chair in International Conferences, Nodal Officer for ARIIA. He served as Resource Person in more than 100 programmes. He has a vast experience in various domains like Medical Image Processing, Signal Processing, Soft Computing, Machine Learning, Deep Learning, Big Data Analytics, Data Science.

Corresponding Author E-mail: amol.bhagat84@gmail.com



M. S. Ali completed B. E. (Electronics & Power) from GCOE, Amravati in 1981, M. Tech (Power Electronics), from Indian Institute of Technology (IITB), Powai – Mumbai in 1984, Ph. D (Electronics Engineering) from Sant Gadge Baba Amravati University (SGBAU), Amravati, in 2006. He worked as Junior Engineer in Rashtriya Chemical Fertilizers (RCF), Mumbai from 1984 to 1986. He served as Professor in College of Engineering, Badnera – Amravati from 1986 – 2005, Head of Department (CSE) PRMITR, Badnera from 2005 – 2009, Principal PRMCEAM, Badnera from 2009 to 2024. He served as SGBAU Academic Council Member - SGBAU, Faculty of Sci. & Tech. Chairman - SGBAU Board of Studies (Electrical Engg.), Fellow – Institution of Electronics and Telecommunication Engineers Fellow – Institution of Engineers (India) Life Member - Indian Society for Technical Education. He has 35+ years of experience, 25 Ph.D. completed under his guidance, 4 copyrights, 6 granted patents, and 45 research papers published. His areas of expertise are neural networks, image processing, signal processing, soft computing.

E-mail: softalis@gmail.com

Sachin Deshmukh received her B.E. degree in 2013 from Sipna COET, Amravati and M.E. degree in CSE in 2016 from PRMCEAM, Badnera, Amravati, Maharashtra, India. He served as Assistant Professor in PRMCEAM, Badnera from Jun 2014 - Oct 2021. He is working as Software Engineer in Metacube Software Pvt. Ltd. since Nov 2021. He is currently pursuing PhD at the Research Center Computer Science and Engineering, Prof Ram Meghe College

of Engg. & Mgmt., Badnera, Amravati, India. His areas of interest are image analysis, machine learning, artificial intelligence, and salesforce.

E-mail: sachindeshmukh24@gmail.com