4th Indian International Conference on Industrial Engineering and Operations Management
Hyderabad, Telangana, India, November 07-09, 2024
Publisher: IEOM Society International, USA DOI: 10.46254/IN04.20240109
Published: November 07, 2024

Automated Email Spam Classification with TensorFlow in Python

T C Swetha Priya
Research Scholar, Department of Computer Science and Engineering,
Jawaharlal Nehru Technological University,
Hyderabad, Telangana, India
tcswetha3552@gmail.com

Dr. R Sridevi
Professor, Faculty of Computer Science and Engineering,
Jawaharlal Nehru Technological University,
Hyderabad, Telangana, India
sridevirangu@jntuh.ac.in

Dr. K S Sadasiva Rao
Professor, Department of Computer Science and Engineering,
Sri Indu College of Engineering and Technology,
Hyderabad, Telangana, India
kssadasivarao@gmail.com

Abstract

Email Spam Detection is a big issue for most email users and service providers. Email plays a major role in our online communication, like any other medium out there but email is also used to carry spam emails which can be treated as the biggest threat (spam). To build our model we use a dataset of emails with labels (spam or ham) in CSV file format. These methods include batch normalization (BN), dropout, and data augmentation, which let the model generalize better than unsupervised individual transfer learning. While training the model, we update its parameters using gradient descent-based algorithms and evaluate its performance based on accuracy, precision (true positive/true positives + false positives), recall (true positive/ true positives + false negatives), and F1-score. The results confirm that the designed model is reliable and thus very effective in capturing spam emails, providing high precision and recall.

Keywords
TensorFlow, Spam, Ham, Feature Extraction, preprocessing

1. Introduction
Scams and cyber-attacks are very common today, making it crucial to detect spam emails. Emails can be classified as spam or as normal mails by machine learning algorithms, especially deep learning models which have shown great promise in this aspect (Toma et al.2021). There is an open-source machine learning framework known as TensorFlow that can be used to build and train such models. Therefore, we can exploit the power of TensorFlow to create a strong spam detection system that assesses various email attributes including sender details, subject lines, body content, and attachments.

In this project, Natural language processing (NLP) techniques will be used to extract important characteristics from email data and train a TensorFlow model on these qualities so that the model learns patterns between them along with spam/legitimate labels. Our methodology involves using different feature extraction techniques like bag-of-words, TF-IDF, and word embeddings (e.g., Word2Vec, GloVe) to convert emails into numerical vectors (Nandhini and Jeen 2020). These vectors are passed through a neural network architecture like Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN). This will help in finding out if the mail is real or fake.

This paper aims at achieving high level of accuracy and precision in spam detection, using TensorFlow's efficient computation and expandable architecture. By doing so, we will develop a solution that can be implemented on email clients as well as web applications to protect users from spam thereby enhancing their safety on the internet (Farombi et al. 2024). We will have valuable insights about spam detection and machine learning by examining what TensorFlow can do using NLP techniques hence creating new ways to better this area in future.

The discussion in this chapter is on a machine learning classifier used in past research and projects (Navaney et al. 2018). It presents a search, reading, analysis, summarization, and evaluation of reading materials with respect to the project.

It is for this reason that research on spam detection using machine learning has been done. However, due to the evolution of spam and various developed technologies, the proposed methods are not dependable (Govindan et al. 2023). Natural language processing is one of the lesser-known fields in machine learning, and it is reflected here in the comparative less amount of work present.

2. Methodology

Spam detection and filtering are accomplished through various methods, hence appropriate methodologies must be selected based on the application's functionality.

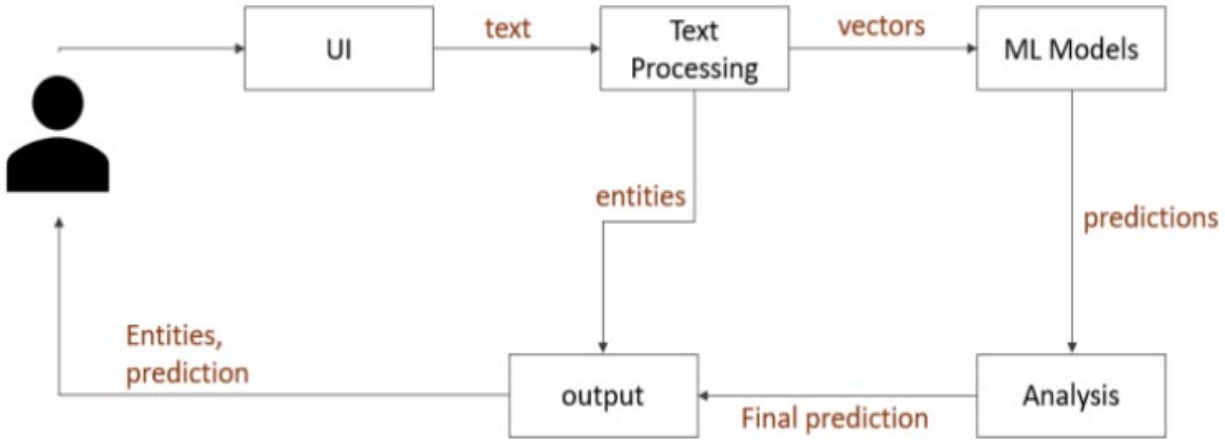

Figure 1. System Architecture

Figure 1 shows the system architecture of proposed system. The architecture has three main modules: User Interface, Text Processing, and ML Models. This process involves in several stages like collection of data from email servers or datasets. Then the collected data is pre-processed using some techniques like Tokenization, Stemming/Lemmatization and stop-word removal (Sethi et al. 2017).Then the pre-processed data is stored in database or file system like CSV. Next, the data is loaded into Python using libraries like Numpy and pandas. The data is split into two sets training and testing, mainly 80% for training and 20% for testing. A tensor-flow model is developed using a training dataset which gets trained and evaluated using testing dataset (Iqbal et al. 2022). That trained model can be deployed in a production environment, such as email server or web application, and also to classify the new emails whether they are spam or ham emails. For evolving spam patterns, we need to collect the new email data and the model is continuously retrained and updated. The web application is also developed to receive email inputs from users and the deployed model is used to classify the emails and display the results to users

(Krishnamoorthy et al. 2024). The components used are Tensorflow for building and training the model and Python libraries like numpy, pandas and scikit-learn for data processing and feature extraction from datasets.

3. Algorithms

3.1. Naive Bayes Classifier A Naive Bayes classifier is a supervised machine learning model which is used for classification of tasks. In this model, the major principle used is Bayes theorem.

3.1.1. Bayes theorem Naive Bayes is a classification technique. The main principle behind it is Bayes theorem which is used to extract all the features that predict the target value are independent of each other (Lakshmi et al. 2024). It calculates the probability of each class and finds the one with highest probability. This Naive Bayes classifier assumes that all the features we use to predict the target are independent of each other and does not affect each other.

$$P(A|B) = P(B|A)P(A))/P(B) \text{ is the Bayes theorem}$$

P (A|B) is the probability of hypothesis A given the data B. This is called the posterior probability.
P (B|A) is the probability of the data B given that hypothesis of A was true.
P (A) is the probability of hypothesis A is true. This is called the prior probability of A.
P (B) is the probability of the data B is true.

Naive Bayes classifier is simple, efficient, and effective algorithm used for many applications especially for text classification. The limitation of this model is that each and every word in the text is independent and have equal importance but each word is not treated equally because when it comes to language it contains articles and nouns which are not same. As Naive Bayes classifier is very efficient, it is used in combination with other language processing techniques.

3.2. Logistic Regression

Logistic Regression is supervised machine learning algorithm that can be used to predict the probability of each event or class based on its features, such as words, phrases and tone (Benerjee et al. 2023). Logistic regression also predicts the output of dependent variable. This Logistic regression is a simple and effective model used for binary classification. Therefore the outcome may be categorical or discrete value like it can be either Yes or No, 0 or 1, True or False, etc. It may not provide exact value as 0 or 1, it gives the probabilistic values which lie between 0 and 1.The probabilities calculated using sigmoid function are used to map real value into another value within the range of 0 and 1.We can't go beyond the limit of 0 and 1 so that it forms the curve of "S" form shape which is called Sigmoid function or Logistic function. If the predicted probability is above a threshold (e.g., 0.5), then it belongs to 1, otherwise it belongs to 0. It's referred as the linear regression but it is used for classification problems.

3.3. Support Vector Machine

Support Vector achine is a supervised learning algorithm used for classification and regression. In this algorithm, we plot each data item in n-dimensional space (n is the number of features) with each value of feature of a particular coordinate. Then we can find the best line or boundary that defines the two classes. There can be multiple lines or decision boundaries in n-dimensional space, but the main goal is to find the best decision boundary called as Hyperplane in SVM. The best hyperplane can be defined with one of the largest margin between the two classes. Here, the largest margin means that with the maximum width of the slab parallel to hyperplane with no interior data points (Tan and Kai 2023). These data points are support vectors that are closer to the separating hyperplane: these points are on the boundary of the slab. If the dimensions of the hyperplane depend on the features present in dataset, if there are 2 features the hyperplane form a straight line. And if there are 3 features the hyperplane will be 2-dimensional form. SVM can be of two types:

i. **Linear SVM:** Linear SVM is used for linearly separable data. In this method, the data can be classified into two classes by using a straight line then such data is called linearly separable data.
ii. **Non-linear SVM:** Non-linear SVM is used for non-linearly separable data in which data can't be classified by using a straight line. Such data is called non-linearly separable data.

3.4. Random Forest

Random Forest is a supervised machine learning algorithm based on concept of ensemble learning method. It can be used for both classification and regression problems. Ensemble learning is a process of combining multiple decision

trees on various subsets of dataset and finds the average to improve the accuracy. The random forest takes the prediction from each tree and finds the majority votes from the predictions and finalize the output. If it contains the greater number of trees in forest means there is high accuracy and prevents the overfitting problem. This random forest model follows the bagging technique. While forming decision trees, random forest adds the randomness to the model instead of searching for important feature it directly searches for the best feature from the subset of dataset features. It results in diversification which makes the model better (Thakur et al. 2023). This ensemble learning is effective in handling high-dimensional data and complex interactions between features.

3.5. Decision Trees

Decision Trees are supervised learning algorithms used for classification and regression tasks. The decision tree construction represents the sequence of decisions. Each node represents a decision based on the value of an input feature and each branch represents the outcome of that particular decision (Shreshta and Neeraj 2023). The tree's leaf nodes represent the final prediction and each link represents a decision rule. The main aim is to create a model which predicts the value of a target variable by decision rules from the dataset. The decision tree is used to classify each instance by sorting from root node to leaf node of the tree which provides a classification for each instance. An instance is classified from starting i.e., root node of the tree and testing each attribute by this node then moving down the tree branch to the attribute value in the given dataset. This process is then repeated for the subtree rooted at the new node.

4. Modules

4.1. Data Processing Module

In this module, the raw data undergoes several modifications for further processing. The main functions of this module include:

i. Data cleaning
ii. Merging of datasets
iii. Text Processing using NLP
iv. Conversion of text data into numerical data (feature vectors).
v. Splitting of data.

All the data processing is done using Pandas and NumPy libraries. Text processing and text conversion is done via NLTK and scikit-learn libraries.

4.2. Machine Learning Module

This is the primary module among all three modules. This module performs all tasks related to machine learning and result analysis. The functions of this module are:

i. Training of machine learning models.
ii. Testing of the model.
iii. Parameters of the respective values for each model.
iv. Keyword extraction.
v. Final output calculation

This module's output is passed to UI for providing visual response to user.

4.3. Feature Extraction

Feature extraction is the integral part of email spam detection using TensorFlow in Python. The following features can be extracted from emails:

1. **Bag-of-Words**: Represent an email as a bag or a set of word frequencies.
2. **Term Frequency-Inverse Document Frequency:** Weight word frequencies by their importance in the email and their rarity in the corpus.
3. **Word Embeddings:** Represent words as dense vectors capturing semantics, for instance, by Word2Vec or GloVe semantic relationships.
4. **Sender Info:** Extract features from the sender's email address, such as domain, username, and length.
5. **Subject Line:** Extract features from the subject line, such as length, keywords, and sentiment.
6. **Email Body:** Features extracted from the email body, such as length, keywords, and sentiment.
7. **Header Information:** Extract features from email headers, such as received headers, MIME types and attachments.
8. **Links and URLs:** Extract features from links and URLs, such as number, length, and domain.

9. **Attachments:** Extract features from attachments, such as number, size, and type.
10. **Spam Keywords:** Extract features from the known spam keywords and phrases.

All these features can be extracted using the following techniques:

i. **Text preprocessing:** Tokenization, Stopword removal, Stemming/Lemmatization
ii. **Feature extraction libraries:** Scikit-learn, NLTK, spaCy
iii. **Custom implementations:** Regular Expressions, String manipulation

Extracted features are then used to train a Machine Learning model. A Neural network or Support Vector Machine is used to classify emails as spam or legitimate.

5. Results and Discussion

Figure 2 shows a histogram that depicts the distribution of length of email text for spam and non-spam emails. From the figure it is identified that the length of the spam mails is shorter in majority of the cases. In contrast, other mails which are not spam have long length messages. The analysis indicates that a high percentage of non-spam emails are longer. This distribution is taken to classify spam emails based on length of email.

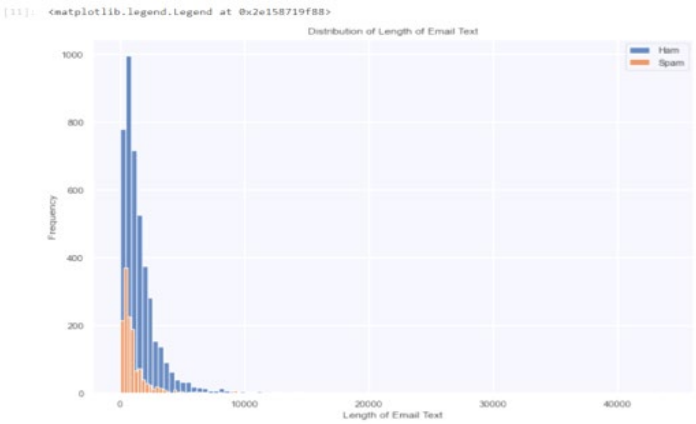

Figure 2. Distribution of Length of Email Text

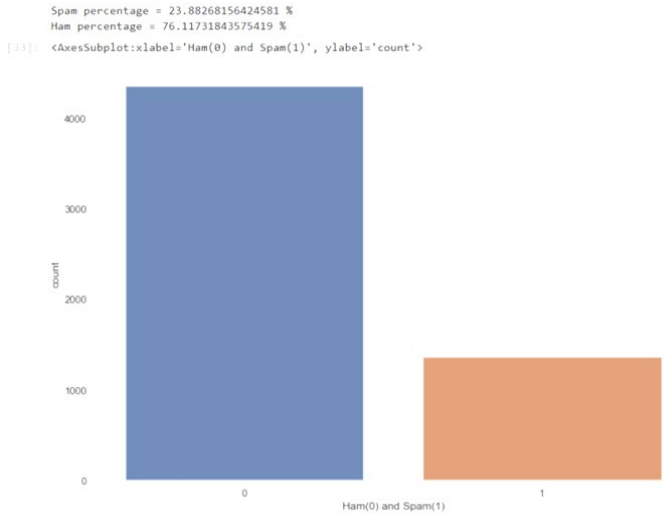

Figure 3. Classification of Ham and Spam

Figure 3 shows a bar chart that represents the distribution of ham and spam messages. It shows that there are significantly more ham messages (76.1%) than spam messages (23.9%). This is a typical distribution in a spam detection dataset.

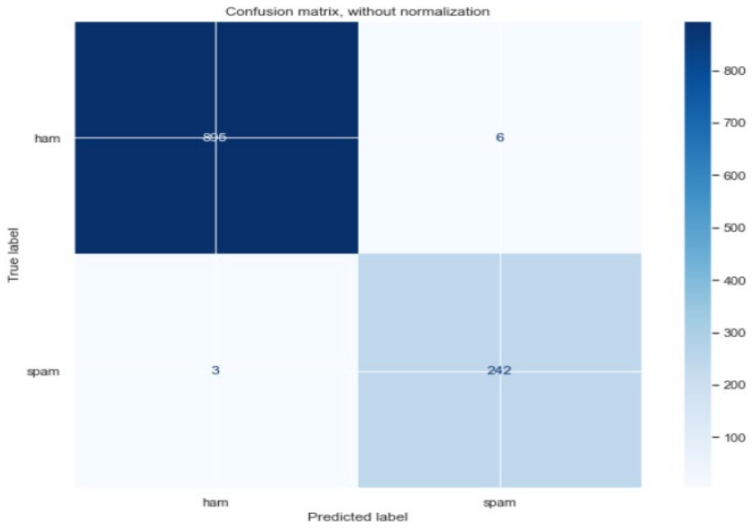

Figure 4. Confusion Matrix

The confusion matrix shown in Figure 4 shows that without normalization the model correctly classified 800 ham messages and 242 spam messages. But, it misclassified 3 ham messages as spam and 6 spam messages as ham. This suggests that the model shows a moderate accuracy in classifying both ham and spam messages.

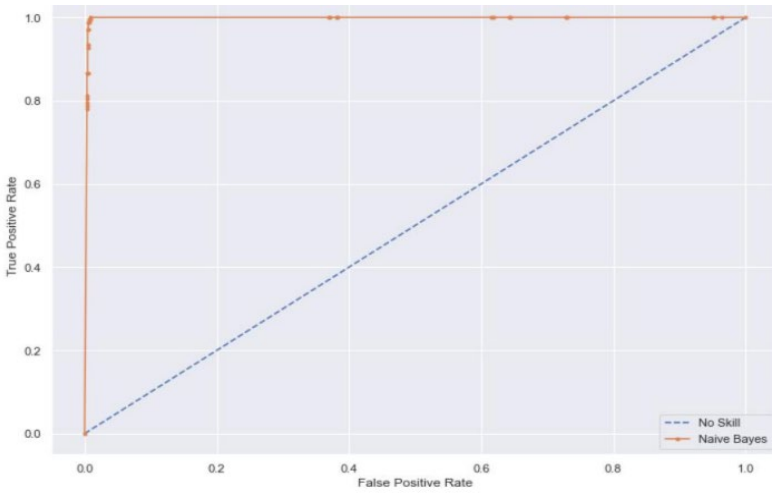

Figure 5. ROC Curve

The Receiver Operating Characteristic (ROC) curve for a Naive Bayes classifier is shown in Figure 5. The curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. An ideal spam detection system is one in which the TPR is high and the FPR is low; however, this is mainly a function of the different priorities and risk tolerance levels entertained within the email filtering context.

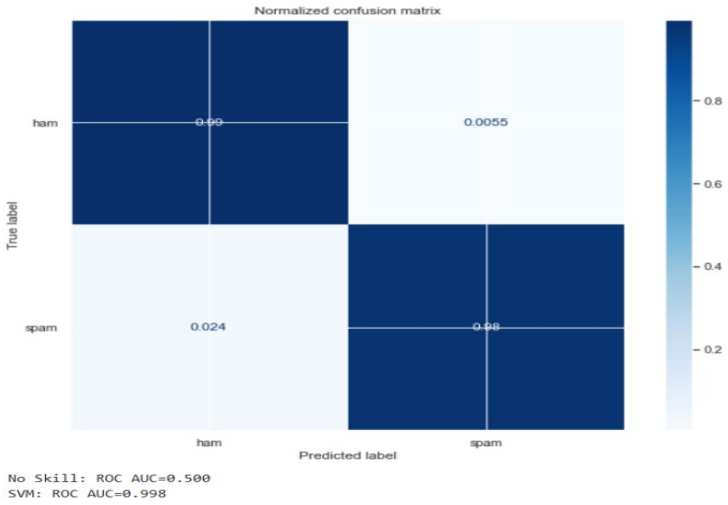

Figure 6. Normalized Confusion Matrix

Figure 6 shows normalized confusion matrix that shows the performance of an SVM classification model. The model classifies the messages as "ham" (not spam) or "spam". The confusion matrix shows that the model is very good at classifying messages correctly. The small values in the off-diagonal cells indicate that the model makes very few mistakes. The model only misclassifies 1% of "ham" messages as "spam" and 2% of "spam" messages as "ham".

The AUC scores (Area Under the Curve) are also shown with "No Skill" representing a random classifier and "SVM" representing the model being evaluated. The high AUC score for the SVM model (0.998) indicates that the model is very good at distinguishing between "ham" and "spam". Overall, this confusion matrix suggests that the model performs well at spam detection.

6. Conclusion

The Email spam detection system developed in Python using TensorFlow entails an end-to-end pipeline that introduces data preprocessing, feature extraction, model building, training, and evaluation. The second preprocessing step was further enhanced by using general stop words and spam-specific stop words to bring an increase in the model performance in classifying spam and non-spam emails. This helps the system realize that the text data is sequential, improving classification accuracy as an LSTM model. It is accurate and balances precision and recall, thereby reducing false positives and negatives. This renders a robust spam detection system that can be deployed for real-time classification of emails to ensure a cleaner and more secure inbox experience for users. An integration of TensorFlow and Python turns out to be an intense combination in meeting most of the challenges associated with spam email detection materializing the power of deep learning techniques in natural language processing application.

References

Banerjee, Sayan, Praveen Kumar, and Sayan Mandal. "E-MAIL & SMS SPAM CLASSIFIER." , 2023.

Farombi, Oluyinka E., Abigael B. Adetunji, and Funmilola A. Ajala. "Tensorflow Technique for SMS Spam Detection in Machine Learning, 2024.

Gadde, Sridevi, A. Lakshmanarao, and S. Satyanarayana. "SMS spam detection using machine learning and deep learning techniques." In 2021 7th international conference on advanced computing and communication systems (ICACCS), vol. 1, pp. 358-362. IEEE, 2021.

Govindan, Shalini, Ahmad Faisal Amri Abidin, Mohamad Afendee Mohamed, Siti Dhalila Mohd Satar, Mohd Fadzil Abdul Kadir, and Nazirah Abd Hamid. "Spam Detection model using tensorflow and deep learning algorithm." Malaysian Journal of Computing and Applied Mathematics 6, no. 2 (2023): 11-21, 2023.

Iqbal, Kashif, Salman A Khan, Shamim Anisa, Ayesha Tasneem, and Nazeeruddin Mohammad. "A preliminary study on personalized spam e-mail filtering using bidirectional encoder representations from transformers (bert) and tensorflow 2.0." International Journal of Computing and Digital Systems 11, no. 1: 893-903, 2022.

Krishnamoorthy, Parthiban, Mithileysh Sathiyanarayanan, and Hugo Pedro Proença. "A novel and secured email classification and emotion detection using hybrid deep neural network." International Journal of Cognitive Computing in Engineering 5: 44-57, 2024.

Lakshmi, H. N., Ratnam Dodda, Sanjana Reddy Vemula, Gayathri Vangala, and Sansya Natemmal. "Email Guard: Enhancing Security Through Spam Detection." In International Conference on Smart Data Intelligence, pp. 597-605. Singapore: Springer Nature Singapore, 2024.

Nandhini, S., and Jeen Marseline KS. "Performance evaluation of machine learning algorithms for email spam detection." In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), pp. 1-4. IEEE, 2020.

Navaney, Pavas, Gaurav Dubey, and Ajay Rana. "SMS spam filtering using supervised machine learning algorithms." In 2018 8th international conference on cloud computing, data science & engineering (confluence), pp. 43-48. IEEE, 2018.

Sethi, Paras, Vaibhav Bhandari, and Bhavna Kohli. "SMS spam detection and comparison of various machine learning algorithms." In 2017 international conference on computing and communication technologies for smart nation (IC3TSN), pp. 28-31. IEEE, 2017.

Shrestha, Neeraj. "A Novel Spam Email Detection Mechanism Based on XLNet." Master's thesis, University of Toledo, 2023.

Tan, Kai Qin. "Machine learning for email filtering and categorising." PhD diss., UTAR, 2023.

Thakur, Prazwal, Kartik Joshi, Shruti Jain, and Prateek Thakral. "Spam Detection in Emails using Machine Learning." 2023.

Toma, Tasnia, Samia Hassan, and Mohammad Arifuzzaman. "An analysis of supervised machine learning algorithms for spam email detection." In 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), pp. 1-5. IEEE, 2021.

Biographies

Mrs. T C Swetha Priya is pursuing her Ph.D under the Department of CSE at Jawaharlal Nehru Technological University, Hyderabad. She is currently working as Assistant professor in Stanley College of Engineering and Technology for Women under the Department of Information Technology. She has 8 years of Teaching experience and 3 years Research Experience. She has published over 16 papers in various International journals, National and International Conferences. She has attended several FDP's, Workshops and Seminars at National and International level.

Dr. R Sridevi is a Professor of CSE with 23 years of teaching experience. Presently working as Professor, Director, Directorate of Entrepreneurship, Innovation and Start-ups, JNTUH & Coordinator for Centre of Excellence in Cyber Security, JNTUH. She has lead various roles as Head of the CSE Department for nearly 3 years, Additional Controller of Examinations(EDEP), Additional Controller of Examinations(Result Processing). Worked as Chairman, Board of Studies for Department of CSE, JNTUHUCEH, Hyderabad. She has organized several Workshops, FDPs, curricular and extra-curricular events and 3 international conferences. Established three Research Labs : IoT Lab, Digital Forensics Lab & Big Data Analytics Lab and one smart classroom under TEQIP in the department. She guided 7 Ph.Ds & published several research papers in various national and international conferences and reputed journals with high indexing factor.

Dr. K. S. Sadasiva Rao is a Professor in the Department of CSE, Dean R&D at Sri Indu College of Engineering & Technology (A), Ibrahimpatnam, Hyderabad. He has 23 Years of Teaching Experience at the level of Head of the Department, Principal and Dean-Academics. He has rich experience in organizing seminars, workshop in Computer Science & Engineering topics. He is Convener for the International Conference ICICSET'24. He also worked as principal for a period of 8 years at Sri Indu PG College. He published 24 papers in several National and International Journals with high indexing factor and presented 6 papers in International Conferences. He is a Lifetime Member in the professional societies CSI, IETE, ISTE. He is also serving as a reviewer for many international journals.