**ABSTRACT:**

In the present digital era, the security of API is the most important due to the widespread use of various applications. This research explores the machine learning models to find the anomalies in the API access patterns and aims to improve cyber security measures. The dataset used in this includes many features of API like Sequence length, duration, number of users, number of unique API’s. This study utilizes various classification algorithms including Logistic regression, Decision Tree, Random Forests, Gradient Boosting, XGBoost, LightBGM, And Neural network to classify the API access patterns as normal or abnormal. The results show that machine learning models can efficiently classify normal and abnormal API access patterns, and improve the security and efficiency of API servers. The primary goal is to identify abnormal behaviour in that could indicate malicious activities.

***Keywords- API Security, Anomaly Detection, Machine Learning, Ensemble Methods, Cybersecurity.***

**I Introduction**

In the contemporary digital landscape, Application Programming Interfaces (APIs) have become pivotal components in the infrastructure of many services, enabling seamless communication between different software applications. APIs facilitate the integration of diverse systems, enhance functionality, and streamline processes by allowing different software programs to interact with each other. However, the extensiveuse of APIs has also introduced a myriad of security challenges, particularly concerning the detection of anomalous behaviors that could indicate potential security breaches or misuse. APIs serve as gateways to sensitive data and critical operations within applications, making them attractive targets for malicious activities. Cyber attackers often exploit vulnerabilities in API implementations to gain unauthorized access to systems, extract sensitive information, or disrupt services. Given the centrality of APIs in modern web services, ensuring their security is paramount to maintaining the integrity and reliability of digital ecosystems.

Traditional security measures, such as rule-based detection systems, often fall short in the dynamic and complex environment of API interactions. These systems typically rely on predefined rules and patterns to identify malicious activities, making them inadequate for detecting novel or sophisticated attack vectors that deviate from known patterns. This limitation underscores the need for more advanced and adaptive approaches to API security, capable of identifying subtle and evolving threats.

Machine learning (ML) offers promising solutions for enhancing API security through anomaly detection. Anomaly detection involves identifying patterns in data that do not conform to expected behavior. In the context of API access, anomalies may indicate potential security threats, such as unauthorized access attempts, unusual data extraction patterns, or deviations from normal usage patterns. ML algorithms are well-suited for this task as they can learn from vast amounts of data, identify complex relationships between variables, and adapt to new and unseen patterns. By leveraging historical data on API usage, ML models can build profiles of normal behavior and detect deviations that may signify anomalies. This capability makes ML a powerful tool for proactive and real-time security monitoring.

This study aims to explore the efficacy of various machine learning algorithms in detecting anomalies in API access behavior. By evaluating a range of models, from traditional methods like logistic regression to advanced ensemble techniques and neural networks, the research seeks to identify the most effective approaches for this task.

II. **Literature Review**

"Anomaly Detection in High-Performance API Gateways" this paper authored by Deshani Geethika, et.al[1], studied to find the behaviour of anomalies and taking corrective actions. API gateways are likely to be broadly deployed. therefore it is utmost important to detect behaviour or performance of anomalies in such high-performance API's. the main contribution of this paper is vichalana, a dataset is used to find the accuracy of anomaly detection aglorithms in API security access. in this they first classified anomalies into seven types. second, they provide detailed criteria for recreating them in API gateway environments. Third, they re-create these anomaly types in an API environment. finally they iluustrate the behaviour of anomaly types by using several example scenarios.

"Malware classification based on API calls and behaviour analysis" authored by Abdurrahman Pektaş, Tankut Acarman,[2]. this study shows the rumtime behaviour of classification proecdure for windows malware. this includes mining and searching n-gram over API call sequences is introdced to find episodes representing anomalies behaviour features of a malware. they used the voting experts algorithm to extract API pattrens over API calls. they used the dataset hodling a 17,400 malware samples and beloging to 60 distinct families.

"Machine Learning in Network Anomaly Detection:A Survey" By song wang, et.al[3], in this paper they intoduced the challenges to find the anomalies in the traditional network,as well as in the next generation network. anomalies could be threat to the network have ever happened. to protect API calls against malicious attacks they presents the machine learning procedure and methodologies. they also provide the comparision of different machine learning models.

"AIMS: A Predictive Web API Invocation Behavior Monitoring System" by Lanxuan Tong, et.al[4], in this study they come up with a new methodology called AIMS, this helps in automatically analyzing the user predictability, invocation and behaviuors. a context-aware K-nearest neighbor classifier is applied to detect the anomaly of user's behaviour.they did experiement on both real-world dataset and synthetic dataset.

"An Anomaly Detection Method to Detect Web Attacks Using Stacked Auto-Encoder" authored by Ali Moradi Vartouni, et.al,[5]. in this paper they proposed a method based on the deep neural network as feature learning method and isolation forest as a classfier. they compared this method with a method doesn't include feature extraction models on CSIC 2010 dataset.Aditionally they applied different activation functions.

"Analysis of HTTP requests for anomaly detection of web attacks" by Mikhail Zolotukhin, et.al[6], this study shows the analysis of HTTP logs for the detection of network intrusions. First, they train the set of HTTP requests which does not contain ant attacks is analyzed.they used different clustering and anomaly detection algorithms are trained to describe the model of normal users behaviour. After this model is used to detect network attacks as deviations from the usual in an online mode.

III. **Methodology**

the methodology used in this research includes the complete analysis and application of various machine learning algorithms to predict the anomalies in API access behaviour. the following steps entitles the entire process, from the data preprocessing to model evaluation.

1.DATA COLLECTION: the dataset used in this research was obtained from kaggle, the API security access behaviour analysis. the dataset is split into two categories supervised\_dataset.csv and remaining\_behaviour\_ext.csv. the dataset includes various labels related to API access behaviour, such as sequence lenght, session duration, number of users, unique API's and more.

2. DATA PREPROCESSING: data preprocessing is a vital step to ensure the quality and consistency of the data. the following steps were carried out:

2.1 loading data:pandas is used to load the datasets. initial inspection was performed to check for missing values and data types.

2.2handling missing values:coloumns inter\_api\_access\_duration(sec)and api\_access\_uniqueness contains the missing values, which were filled with their respective mean value.

2.3 feature selection:similar features were selected for training the models, ensuring the balance between having enough information for accurate prediction and avoiding overfitting.

3.EXPLORATORY DATA ANALYSIS(EDA):EDA was implemented to understand the relationship between different variables. correlation matrices were plotted using the seaborn to identify highly correlated features and potential multiple correlation issues. this helps in the selection of appropriate features for model training.

4.MODEL SELECTION AND TRAINING: several machine learning models were trained to predict the API access behviour, the models used in this research:

1.logistic regression: a linear model used to evaluate performance on the dataset.

2.Tree-based models:

2.1 Decision tree classifier:A simple tree-based model with entropy as condtions.

2.2 random forest classifier: it is an ensemble method used mutiple decision trees to improve the prediction accuracy.

2.3 gradient boosting classifier: another ensemble technique, this builds tree squentially to reduce errors.

2.4 Adaboost classifier:combines multiple weak classifiers to build a strong classifier.

2.5 bagging classifier:aggrigates the predictions of several base estimators to improve applicability.

3. boosting algorithms:

3.1 lightGBM: a gradient booting framework that uses tree-based learning algorithm.

3.2 XGBoost: it is an efficient and scalable implemenation of gradient boosting.

prediction and explores different models for making accurate predictions.

The main findings of the S.Rohatgi et. al. [13] paper is that machine learning algorithms, specifically the Gradient Boosted Trees model, can be used to predict the movement of the stock market with greater accuracy than traditional statistical methods. The study compared six different algorithms and found that Gradient Boosted Trees had the lowest relative error and standard deviation, making it the most efficient for forecasting stock market trends.

The study in the paper of C.Prasanth et. al. [14] suggests using an interactive system based on predictive analysis to prioritize loan approval based on customer eligibility and requirements. The Random Forest algorithm is used to accurately predict loan repayment and improve accuracy while reducing time complexity.

The study of Narayana Darapaneni et. al. [15] compares the performance and explainability of five different classification models, Decision tree, Random Forest, AdaBoost, GBoost, and XGBoost, using the credit default dataset from Kaggle. Tree-based models are found to be useful for visualizing the entire flow of conditions that lead to a particular outcome, making them helpful in explaining the results.

The study of Ch. Naveen Kumar et. al. [16] aims to predict client creditworthiness using ML approaches. The project focuses on data purifying, core attribute choice, and contrasts the success rates of several machine learning approaches. Results indicate that the blended decision tree with AdaBoost excelled other models in terms of forecasting customer loan eligibility.

In the study of Praveen Tumuluru et. al. [17] Machine learning algorithms such as the Support Vector Machine, Random Forest, K's Nearest Neighbour, and Logistic Regression were utilised to predict loan defaulters using customers' past data. The method known as random forest surpassed the other algorithms in terms of accuracy.

The study of Krishan Kumar Pandey et. al. [18] aimed to predict loan approval using historical data from an organization's loan processing using various categorization methods, including Random Forest, Logistic Regression, Support Vector Machine, Decision Tree, Gradient Boosting, and XG Boost. Each model's reliability was assessed using an ambiguity grid and judged using recall, accuracy, precision, and scores for F1. The results showed that Logistic Regression and XG Boost had the best performance, with F1 scores of 0.90 and 0.89 for class 1 (approved) and 0.62 and 0.64 for class 0 (not approved), respectively.

The study of Anon Mueankoo et. al. [19] aimed to develop a technique for predicting loan repayment status of As- Siddeek Islamic Cooperative Limited in Songkhla Province, Thailand. Feature selection techniques were optimised to discover important factors, and ML techniques such as Decision Tree, Logistic Regression, and Random Forest were used to fit the model. The precision of prediction models ranged from 60.0% to 63.1%

**OBJICTIVES:**

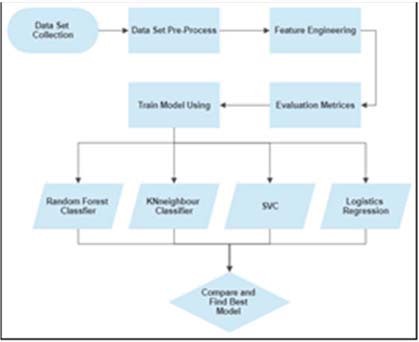
The intention of this study is to investigate the use of algorithms based on machine learning for predicting loan approval and to evaluate the performance of different algorithms using various metrics. The goal is to provide insights into the potential of these algorithms to improve the decision-making process for financial institutions and to identify the most accurate and reliable algorithm for predicting loan approval.

**Methdology**

determine the best algorithm for loan approval prediction, the study utilized the "Loan Eligibility Dataset" from Kaggle. The dataset underwent extensive preprocessing, including the removal of missing values, outliers, and redundant features. Exploratory data analysis was conducted to gain deeper insights into the dataset.

The information set was then divided into sets to be trained and tested using a 60/40 ratio, and four different ML techniques (Random Forest, Support Vector Machine, K- Nearest Neighbor, and Logistic Regression) trained on the training set. The act of each model was assessed using accuracy, F1 score, and ROC test scores.

The Random Forest model emerged as the most accurate model, achieving an accuracy score of 98.04%. The Logistic Regression model came in second place, with an accuracy score of 79.61%. Figure-1 depicts the entire process, from the beginning of dataset preprocessing to the determination of the best algorithm for loan approval prediction.

 Fig. 1. Approach to Project

1. ***Dataset Collection***

The loan approval dataset used in this study contains 614 rows and 13 columns, including features such as Gender, Dependents, Education, Self\_Employed, Applicant Income, Loan\_Status etc. The dataset was obtained from Kaggle, a platform for data science and machine learning enthusiasts.

1. ***Dataset Pre-Processing***

Before being used, data must typically undergo processing to ensure its cleanliness and suitability for machine learning algorithms. The dataset acquired from Kaggle required extensive preprocessing, including handling missing values, encoding categorical variables, and feature scaling.Initially, the dataset was scrutinized for missing values and empty cells. Any rows containing such values were removed from the dataset. Additionally, certain variables contained extraneous characters that were eliminated through cleaning procedures. Following the cleaning stage, the data were stored in a new CSV file for use in machine learning algorithms.

After pre-processing, info of the data set is given in figure-2. The data set, contains 598 entries with non-null values for Loan\_ID, Gender, Married, Education, Self\_Employed, ApplicantIncome, CoapplicantIncome, Property\_Area, and Loan\_Status.

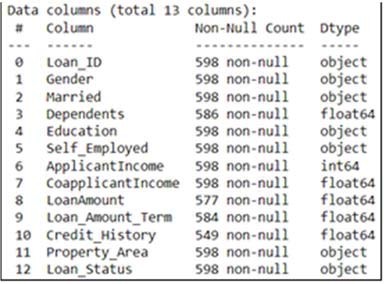


Fig. 2. Data set after Pre-Processing

C. **Feature Engineering**

During feature engineering, to encode the categorical values which were binary, the Label Encoder was used, and the values were converted to int data type. The Loan\_Status was taken as the target variable while the remaining columns considered as predictor variables.

1. **Evaluation Metrics**

The evaluation metrics used in this study were the accuracy score, F1 score, and ROC score. The accuracy score is a estimate of how much perfectly the model predicts the loan approval. The F1 score is a compromise between precision and recall and gauges the model's capacity to detect real-life positives while limiting fake positives and negatives. Finally, the ROC score is an evaluation of the model's capacity to compare variances among actual and false groups, and it offers an overall assessment of its efficacy. These criteria were employed to assess the results of the various machine learning models and choose the most reliable one for forecasting loan acceptance.

**Data Preparation**

To plan the dataset for machine learning calculations, it was to begin with isolated into two parts: the subordinate variable (advance status) and free factors. The subordinate variable was doled out to 'y', whereas the free factors were relegated to 'X'. The 'shape' work was at that point utilized to confirm the shape of both 'X' and 'Y'.Then utilized the "train\_test\_split" work in the "sklearn.model\_selection" module to part the information into preparing information and test information. The 'test\_size' alternative had a esteem of 0.4, demonstrating that 40% of the information was set aside for testing, whereas the leftover portion, or 60%, was set aside for preparing. To guarantee consistency over different runs, the 'random\_state' parameter was set to 1. Test information was doled out to "X\_test" and "Y\_test", whereas preparing information was alloted to "X\_train" and "Y\_train".

**Model Training**

Following the data split, we trained four different machine learning algorithms on the training set using the fit method from sklearn. To evaluate the performance of these algorithms, we employed the accuracy score function from the scikit-learn library.For each algorithm, the initial step involved defining the model along with its parameters. For instance, we specified the number of neighbors for the KNeighbors Classifier and the number of trees for the Random Forest Classifier. Subsequently, each model underwent training, and predictions were generated on the same training set to assess their accuracy.

**Random Forest Classifier**

The Random Forest Classifier was the first ML technique to be trained. Its performance was evaluated build on several benchmarks, including accuracy score, F1 score, and ROC score. The results demonstrated that the Random Forest Classifier achieved an impressive accuracy of 98.04%, an F1 rating of 0.85, and an ROC grade of 0.973. These findings are presented in table-I.

TABLE I. SCORES FOR RANDOM FOREST

|  |  |
| --- | --- |
| **Accuracy of Random forest**  **Classifier** | 98.044 |
| **F1 Score of Random forest**  **Classifier** | 0.9857 |
| **ROC Score of Random forest**  **Classifier** | 0.9739 |

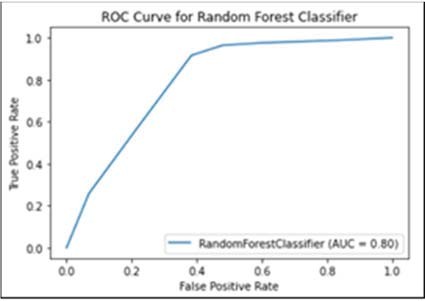


Fig. 3. ROC Bend of Random Forest Classifier

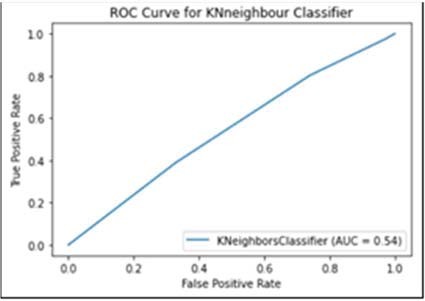
The ROC twist for the above Classifier in figure-3 shows giant correct actual rate and a little wrong actual rate, which specify that this classifier has a high accuracy in predicting both actual and untrue classes.

***KNneighbour Classifier***

The K-Nearest Neighbors classifier was the second model to be trained. According to the results presented in table-II, the K-Nearest Neighbors classifier achieved an validity of 78.49%, an F1 grades of 0.8533, and an ROC rating of 0.7090.

TABLE II. SCORES FOR KNNEIGHBOUR CLASSIFIER

|  |  |  |  |
| --- | --- | --- | --- |
| **Accuracy**  **Classifier** | **of** | **KNneighbour** | 78.491 |
| **F1 Score Classifier** | **of** | **KNneighbour** | 0.853 |
| **ROC Score of KNneighbour**  **Classifier** | | | 0.709 |

Fig. 4. ROC Bend of KNeighbors Classifier

The ROC bend for the KNeighborsClassifier in figure-4 shows a bottom correct positive rate and a greater wrong positive rate than the RandomForestClassifier, indicating lower accuracy in predicting both actual and false classes.

***SVC***

The third machine learning algorithm employed in this study was the Support Vector Classifier (SVC), which yielded an precision score of 68.71%, an F1 rating of 0.8133, and an ROC rating of 0.5087, as illustrated in Table III.

TABLE III.SCORES FOR SVC

|  |  |
| --- | --- |
| **Accuracy of SVC** | 68.715 |
| **F1 Score of SVC** | 0.8133 |
| **ROC Score of SVC** | 0.508 |

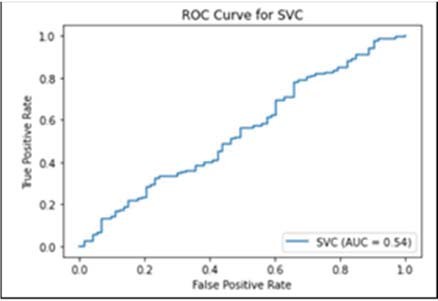


Fig. 5. ROC Bend of SVC

The ROC bend for the SVC in figure-5 shows a very small correct positive rate and a giant untrue positive rate, indicating that this classifier has a low accuracy in predicting actual and untrue classes.

**Logistics Regression**

The fourth and final algorithm implemented was the Logistic Regression, which achieved an precision score of 79.60%, F1 rating of 0.8666, and ROC rating of 0.6961. The results are presented in table-IV.

TABLE IV. SCORES FOR LOGISTICS REGRESSION

|  |  |
| --- | --- |
| **Accuracy of Logistics**  **Regression** | 79.608 |
| **F1 Score of Logistics Regression** | 0.866 |
| **ROC Score of Logistics Regression** | 0.696 |

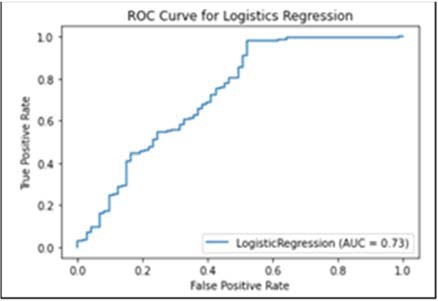


Fig. 6. ROC Bend of Logistics Regression

The ROC bend for the above classifier in figure-6 suggests a moderate real practical rate and false practical rate, indicating that this classifier has moderate accuracy in predicting actual and false classes.

TABLE V. COMPARISON BETWEEN SCORES OF DIFFERENT MODELS.

|  |  |  |  |
| --- | --- | --- | --- |
| **Machine Learning Model** | **Accuracy**  **Score** | **F1**  **Score** | **ROC**  **Score** |
| Random Forest Classifier | 98.04% | 0.85 | 0.973 |
| K-Nearest Neighbors | 78.49% | 0.8533 | 0.7090 |
| Support Vector Classifier | 68.71% | 0.8133 | 0.5087 |
| Logistic Regression | 79.60% | 0.8666 | 0.6961 |

As shown in the Table-V, the Random Forest achieved the greatest accuracy of 98.04%, along with Logistic Regression with an accuracy of 79.60%. The F1 score is a weighted average of the theory's accuracy as well as recall and is a measure of the model's general efficacy. In terms of F1 score, the Random Forest Classifier achieved the highest score of 0.85, followed by Logistic Regression with a score of 0.8666. The ROC score evaluates the model's ability to discriminate between actual and false classes. Here also, the Random Forest achieved the greatest score which is 0.973, along with Logistic Regression with a score of 0.6961.

RESULT

In this work, the act of several ML techniques for assessing credit approval was assessed. The accuracy score, F1 score, and ROC score of Random Forest Classifier, K- Nearest Neighbors Classifier, Support Vector Classifier, and Logistic Regression models were compared on a prepared dataset that underwent exploratory data analysis and feature engineering.

Our results showed that the Random Forest Classifier attained the greatest accuracy score of 98.04%, accompanied by Logistic Regression at 79.61%, K-Nearest Neighbors Classifier at 78.49%, and Support Vector Classifier at 68.71%. Additionally, the Random Forest Classifier defeated the remaining models in terms of ROC score with a score of 0.974.

These findings suggest that the Random Forest Classifier is the most accurate and reliable machine learning algorithm for predicting loan approval. The results of our study can be useful for financial institutions in improving their decision- making process and reducing the risk associated with loan approvals.

1. Conclusion

In conclusion, a detailed methodology for predicting level of credit acceptance status using ML techniques has been presented. A dataset from Kaggle was collected, which contains 599 rows and 13 columns, including predictor variables such as gender, married, dependents, loan amount term, credit history etc.

For feature engineering, LabelEncoder was used for all categorical columns, and the Loan\_ID column was dropped as it was not correlated with any other columns. The facts were split into tuition and trial sets, and four classification models were trained: Random Forest Classifier, KNN Classifier, SVC, and Logistic Regression.

The act of each model was assessed using accuracy score, F1 score, and ROC score. The Random Forest model achieved the greatest accuracy of 98.04%, along with Logistic Regression with an accuracy score of 79.61%. K-Nearest Neighbor and Support Vector Machine models had lower accuracy scores of 78.49% and 68.71%, respectively.

In terms of F1 score, the technique named Random Forest along with the Logistic Regression technique had the highest scores of 0.85 and 0.87, apiece. K-N Neighbor and Support Vector Machine techniques had F1 rating of 0.85 and 0.81, each.

ROC score, which indicates the trade-off between sensitivity and specificity, was highest for the Random Forest model (0.97) and lowest for the Support Vector Machine model (0.51). The Logistic Regression model had an ROC score of 0.70, while the K-Nearest Neighbor model had an ROC score of 0.71.

Overall, the Random Forest model performed the best in terms of accuracy, F1 score, and ROC score. Logistic Regression also performed well in terms of precision and F1 ratings. The results suggest that ML techniques can be used to predict loan eligibility, with the Random Forest and Logistic Regression models being the most suitable for this task.

References

[1] U. E. Orji, C. H. Ugwuishiwu, J. C. N. Nguemaleu and P. N. Ugwuanyi, "Machine Learning Models for Predicting Bank Loan Eligibility," 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), Lagos, Nigeria, 2022, pp. 1-5, doi: 10.1109/NIGERCON54645.2022.9803172.

[2] M. Alaradi and S. Hilal, "Tree-Based Methods for Loan Approval," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 2020, pp. 1-6, doi: 10.1109/ICDABI51230.2020.9325614.

[3] Alaraj, Maher & Abbod, Maysam & Hunaiti, Ziad. (2014). Evaluating Consumer Loans Using Neural Networks Ensembles.

[4] Supriya, P. Usha, M. Pavani, Nagarapu Saisushma, Namburi Vimala Kumari and Khullar Vikas. “Loan Prediction by using Machine Learning Models.” (2020).

[5] Bindal, R., P. K. Sarangi, G. Kaur, and G. Dhiman. "An approach for automatic recognition system for Indian vehicles numbers using k- nearest neighbours and decision tree classifier." (2019).

[6] D. K. Panag, N. Goel, P. K. Sarangi, A. K. Sahoo and B. P. Rath, "Prediction of Gold prices in India: Performance Analysis of Various Machine Learning Models," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-7, doi: 10.1109/ICRITO56286.2022.9964974.

[7] M. A. Sheikh, A. K. Goel and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," 2021 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 490- 494, doi: 10.1109/ICESC48915.2020.9155614.

[8] I. Parmar et al., "Stock Market Prediction Using Machine Learning," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 574-576, doi: 10.1109/ICSCCC.2018.8703332.

[9] S. Rohatgi, K. Kumar Singh and D. Jasuja, "Comparative Analysis of Machine Learning Algorithm to Forecast Indian Stock Market," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2021,

pp. 278-283, doi: 10.1109/ICACITE51222.2021.9404642.

[10] C. Prasanth, R. P. Kumar, A. Rangesh, N. Sasmitha and D. B, "Intelligent Loan Eligibility and Approval System based on Random Forest Algorithm using Machine Learning," 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 2023, pp. 84-88, doi: 10.1109/ICIDCA56705.2023.10100225.

[11] N. Darapaneni et al., "Tree Based Models: A Comparative And Explainable Study For Credit Default Classification," 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 2022, pp. 1-8, doi: 10.1109/UPCON56432.2022.9986411.

[12] P. Tumuluru, L. R. Burra, M. Loukya, S. Bhavana, H. M. H. CSaiBaba and N. Sunanda, "Comparative Analysis of Customer Loan Approval Prediction using Machine Learning Algorithms," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2021, pp. 349-353, doi: 10.1109/ICAIS53314.2022.9742800.

[13] K. K. Pandey, A. Giri, S. Sharma and A. Singh, "Predictive Analysis of Classification Algorithms on Banking Data," 2021 IEEE 4thInternational Conference on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, 2021, pp. 1-5, doi: 10.1109/GUCON50781.2021.9573792.