**Abstract**

The process of loaning credit by banks helps individuals and business to grow financially and make large payments and investments upfront that they might not have been able to make by themselves. This helps the economy by maintaining cash flows during the lean periods. However, crediting a loan comes with a risk that the client or the business might not be able to repay the loan in the future leading to loan defaults. To mitigate this issue, banks for a long time have conducted background financial checks on clients manually. In 21st century, with adequate data and resources machine learning models can be trained to perform the same task reducing time and effort to make the decision whether to disburse the loan or not.

Finance sector in India comprises mainly of commercial banks that generate most of their business through the interest received on loans disbursed to people and organisations. However, credit lending comes with risk of consumer defaulting the loan i.e., inability to pay back the due sum of money agreed upon by both the parties. Ensuring that the borrower will be able to pay back the loan on the proposed terms is the biggest challenge faced by banks in today’s ever-changing world. This paper proposes to leverage the use of Machine Learning (ML) algorithms to predict which clients are more likely to default on their loans using historical financial data like credit and debit card data, bank account transactions, previous credit history, current bank loans and income of the client.

This study makes use of the Credit Risk Model Stability data available on Kaggle provided by the Home Credit finance provider, founded in 1997. The research is a comparative analysis between different types of algorithms in machine learning. Results of this study can be scaled and applied to a real-world dataset and holds the immense potential to revolutionise the financial industry.

*Keywords – credit; loan default; gradient-boosting; bayesian-learning; machine-learning­*

**Introduction**

Commercial banks are the main players in Indian financial sector. Most of their revenue is earned from the interest on loans that is extended to individuals and corporates. However, granting loans has its own pros and cons. One of the cons include default risk. Default risk means when someone takes a loan but due to some reason is unable to pay back the agreed amount. To minimize this risk and ensure that the loan is repaid according to the accepted terms and conditions, gave us the motivation to work on this project.

To address the challenge, this paper leverages a combination of eight different Machine Learning (ML) and Deep Learning (DL) algorithms to predict the likelihood of loan default. These algorithms include Random Forest, Decision Tree, K- Nearest Neighbours, Gaussian Naïve Bayes, XGBoost, Light Gradient Boosting Machine (LGBM), Ada Boost and Long Short Term Memory(LSTM). Use of machine learning methods in this ever-changing financial sector can be attributed to two major reasons. The first reason is, with the advancement of technology and increasing online transactions, banks are now able collect more data than ever from internal and external data sources which can be easily processed by teaching a computer to do it. The second reason is the success of ML models in similar applications like stock price prediction and credit card fraud detection in the banking sector.

This explorative research uses real-world banking data provided by the Home Credit, an international consumer finance provider on Kaggle. The dataset consists of masked data of actual clients split into 32 training and 36 testing csv files. For each client id, there exists a dependent target class having values, 0 (client repays the loan) and 1 (client defaults the loan) that is to be predicted. After collecting the required data, we apply data mining, preprocessing and feature engineering methods to create a cleaned dataset to be fed to a machine learning model. We compare the performance of 8 ML & DL algorithms, out of which gradient boosting machines achieve nearly 99% accuracy, thus showcasing the promise of scaling these models to a real-world use case in the banking industry.

**Literature Survey**

A review of the researches previously done in this context help us to explore various aspects of loan default prediction using Machine Learning including statistical and probabilistic models, and comparative studies done using advanced and hybrid Machine Learning techniques. Some of which are discussed below.

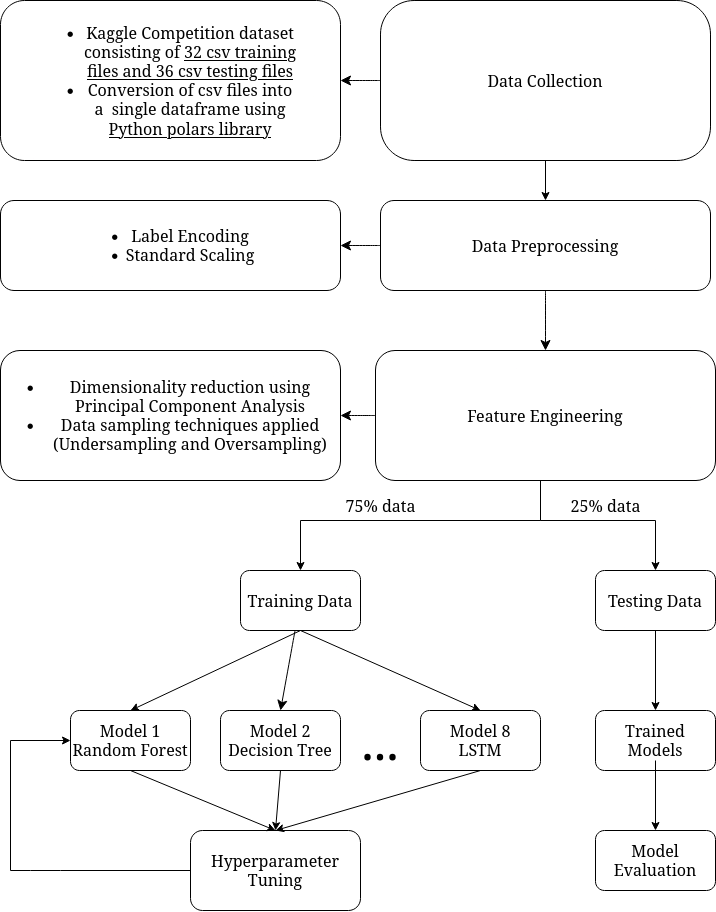
Asha RB et al. in [1] compared three algorithms named Support Vector Machines (SVM), K-Nearest Neighbour (KNN) and Artificial Neural Network (ANN). The conclusion of [1] reveals that ANN performed the best with an accuracy of 99.92% followed by KNN and SVM. Lin Zhu et al. in [2] compared Random Forest, Decision Tree, Support Vector Machine (SVM) and Logistic Regression. The experiment shows that Random Forest outperformed other algorithms with an accuracy of 98% followed by Decision Tree (95%) and SVM (75%). John O. Awoyemi et al. in [3] used hybrid sampling to handle imbalanced data. In [3] three algorithms named Naïve Bayes, KNN and Logistic Regression have been compared. The conclusion of [3] reveals that KNN performed the best with an accuracy of 97. 9% followed by Naïve Bayes (97.6%) and Logistic Regression (54%). Jing Gao et al. in [4] have used XGBoost and Long-Short Term Memory (LSTM) for their comparative research. The outcome indicated by [4] shows that XGBoost-LSTM model predicts credit card default with an accuracy of 95.4%, whereas XGBoost alone predicts with an accuracy of 89.5%. V.A. Kandappan et al. in [5] made use of Bidirectional LSTM to predict loan defaults. The conclusion of [5] reveals that LSTM achieved a promising accuracy of 94%. Anushi Jain et al. in [6] compared five algorithms named Logistic Regression, Support Vector Machine (SVM), Random Forest, XG Boost and Artificial Neural Network (ANN). The conclusion of [6] reveals that Logistic Regression is the best model to predict Loan default with an accuracy of 88.89% followed by Random Forest (88.85%) and XG Boost (88.57%). Md. Golam Kibria et al. in [7] Deep Learning model with two machine learning models Support Vector Machine (SVM) and Logistic Regression. The result of [7] reveals that the overall performance of deep learning (87.10%) is better than that of two machine learning models (86.23%). Huannan Zhang et al. in [8] compared Random Forest, Decision Tree and Logistic Regression algorithms to showcase the application of Random Forest Classifier in Loan default forecast. The conclusion of [8] is that the Random Forest Algorithm (≈ 86%) exceeds the decision tree (≈80%) and logistic regression classification (≈80%). Bhoomi Patel et al. in [9] compared four algorithms named Logistic Regression, Gradient Boosting, CatBoost Classifier and Random Forest to predict loan default. In [9] CatBoost Classifier outperformed other algorithms. It has an accuracy of 84.045%, whereas the other algorithms were 14.963%, 84.035%, and 83.514% respectively. Yanash Azwin Mohmad in [10] compared Long Short Term Memory(LSTM) model with four traditional machine learning algorithms: support vector machine, random forest, multi-layer perceptron neural network, and logistic regression. The results show that the LSTM model gives the highest accuracy of 82.4% in predicting late fees and mis-payments of loans. Abhishek Shivanna et al. in [11] used different algorithms including Deep Support Vector Machine (DSVM), Boosted Decision Tree (BDT), Averaged Perceptron (AP) and Bayes Ponit Machine (BPM). The results of [11] show that out all the models DSVM can best predict defaulters with an accuracy of 82.20%. Yue Yu in [12] compared four algorithms named Logistic Regression, Random Forest, Decision Trees and AdaBoost. The result of [12] shows that Random Forest gave the best accuracy of 82.12%. Saurabh Arora et al. in [13] compared six algorithms named K-Nearest Neighbour (KNN), Decision Tree, Random Forest, Logistic Regression, Support Vector Machine (SVM) and Naïve Bayes. The conclusion of [13] reveals that SVM is the best model to predict credit card default with an accuracy of 82% followed by Logistic Regression (81%) and Random Forest (80%). Theoneste Ndayisenga in his [14] has mentioned the use of Logistic Regression, Decision Tree, Support Vector Machines, Random Forest, KNN, Gausian Naive Bayes, Gradient Boosting and XG Boost. The result of the analysis of these algorithms shows that Gradient Boosting (≈ 81%) is the best model to predict bank default followed by XG Boost (≈ 80%). Mehul Madaan et al. In [15], compared Random Forest and Decision Tree algorithms to predict loan default. The conclusion of [15] is that Random Forest with an accuracy of 80% outperformed Decision Tree algorithm that gave an accuracy of 73%. The dataset that they used had biased data. Malik Mubasher Hassan et al. in [16] used Artificial Neural Networks to predict customer defaults. The result of [16] showed that ANN can predict the customer default with an accuracy of 77.9%.

Alžbeta Bačová and František Babič in their [17] have used Random Forest, AdaBoost and XGBoost for predictive analysis for credit card default. The results of [17] showed that the performance of these algorithms was very similar. Lili Lai in [18] has compared AdaBoost, XGBoost, Random Forest, KNN and Multi-Layer Perceptron algorithms to predict loan default. The conclusion of [18] is that AdaBoost outperformed all the other algorithms followed by XGBoost. Luca Barbaglia et al. in [19] used Penalized Logistic Regression, Gradient Tree Boosting and XGBoost for a highly unbalanced dataset of 12 million residential mortgages. The result of [19] revealed that XGBoost and Gradient tree Boosting outperformed Penalized Logistic Regression model. Hyeongjun Kim et al. in [20] made use of Support Vector Machines(SVM), Decision Tree and Artificial Neural Network(ANN). Three types of statistical analysis, Discriminant Analysis, Binary Response Models and Hazards Models, have been used in [20].

Abhishek Agarwal et al. in [21] have mentioned about Logistic Regression, Random Forest, Decision Trees, Naïve Bayes and KNN. The main motive of [21] is to compare measures between the original dataset before and after applying the Principal Component Analysis. The conclusion of [21] is that the accuracy of Logistic Regression was best in both the cases and Decision Tress was not affected much. Mohammad Ahmad Sheikh et al. in [22] have used Principal Component Analysis (PCA) to analyse its importance. The conclusion of [22] Is that the model is marginally better after applying PCA.

**Methodology**

In the recent years Machine learning algorithms have revolutionized the finance sector by offering financial institutes a powerful data-driven tool to help with the enormous tasks of predicting loan defaults by a client. The workflow of the proposed methodology is explained using the flowchart be.

Fig 1. Flowchart representation of the proposed methodology

* **Data Collection**

The dataset used in the model is the Home Credit – Credit Risk Model Stability provided by the Home Credit organisation on Kaggle. It consists of 32 training files and 36 testing files. All the files are available in csv and parquet format. The files consist of data collected for a particular case id from two types of sources: internal and external data sources.

The training files are used to train and test the model. These are further divided into two parts in the ratio 75:25 for training and testing purposes respectively.

* **Data mining**

The csv files extracted from the dataset are needed to be merged and converted into a format that can fed into the machine learning models. To merge all the files together, a “train\_base.csv” file is provided that contains the target variable. The files were merged into a single dataset using the python library, polars. Polars is a python library written in Rust and uses a multi-threaded query engine for fast and effective parallel execution.

* **Preprocessing**

After converting all the csv files into a single polars dataframe, the data consists of null values, categorical string data and dates. This data needs to be pre-processed and converted into a usable format. Firstly, the dataset is transformed back into pandas dataframe for label encoding the columns with string data. Label Encoding is a preprocessing technique which converts string data into numerical data by assigning a unique index value to each unique string value in a column.

Another data preprocessing technique known as standard scaling is used. Standard scaling is also known as Z-score normalisation. The numerical features of the dataset are scaled to have a mean of 0 and standard deviation of 1. The formula for standard scaling is:

z = (x−μ) / σ

where, x = original value of the feature

μ = mean of the feature values

σ = standard deviation of the feature values

Standard scaling is used to normalise distribution of values, reduces dominance of a single feature or multiple features in the dataset and improves converge rate of the model.

* **Feature Engineering**

The final dataset after cleaning and preprocessing consists of 303 independent features, 1 target variable and over 15 lakh records. The size of the dataset was reduced from 15 lakh records to 1 lakh records so that dataset is balanced and there is equal representation between the two classes. Faster convergence and improved training times can be achieved using an algorithm called Principal Component Analysis (PCA). PCA is a dimensionality reduction technique that is used to reduce the number of features in a dataset while preserving all or most of the essential information. PCA can be lossy or lossless in nature depending on the dataset. Using PCA, top 100 features were extracted to train the machine learning models.

|  |  |
| --- | --- |
| **Feature Name** | **Feature Description** |
| credamount\_770A | Loan amount or credit card limit |
| cntpmts24\_3658933L | Number of months with any incoming payment in last 24 months |
| bankruptcy\_history | Bankruptcy history of the client |
| age | Age of the client |
| education\_level | Education level of the client |
| credit\_score | Credit score of the client |
| employment\_history | Employment history of the client |
| debt\_to\_income\_ratio | Debt-to-income ratio of the client |
| income | Income of the client |
| payment\_history | Payment history of the client |
| late\_payment\_history | Late payment history of the client |

*Table 1. Some important features of the dataset*

* **Proposed Model**

Given the nature of the dataset, the machine learning algorithms used for comparative analysis in this study are: ensemble learning, gradient boosting, bayesian learning, lazy learning and deep learning algorithms. The dataset is divided into 75:25 training and testing data respectively.

* **Decision Tree**

Decision tree is a heirarchical structure that can be used for both, regression as well as classification tasks. Each internal node represents a decision taken based on a feature and leaf nodes represent the output of the model. It splits the dataset features recursively into subsets based on the feature that best divides the data i.e., providing the maximum information gain into separate classes. The dataset is divided at each step such that it maximises the information gain.

E(S) = -p\_{(+)} \log p\_{(+)} - p\_{(-)} \log p\_{(-)}

*Fig 2. Formula for entropy calculation*

\text{Information Gain} = E(Y) - E(Y \mid X)

*Fig 3. Information gain calculation*

* **Random Forest**

Random forest is an ensemble learning algorithm that builds multiple decision trees during model training. The output of a random forest model for a classification task is based on voting of the various decision trees and for a regression task, the output is the mean of the output of the internal decision trees. It is indifferent to noisy data and generalises the model reducing overfitting.

* **Gradient Boosting**

Gradient boosting is a boosting technique in which a strong model is built by sequential learning of multipe weak learning models. It combines weak learner models and optimises them to minimise the value of a loss function. Gradient boosting algorithms used in this study are: XGBoost, Ada Boost and LightGBM. The aim of these algorithms is to minimise the loss function, in this case, for binary classification is log loss function. The log loss function heavily penalises the wrong classifications.

\[

-\frac{1}{N} \sum\_{i=1}^{N} y\_i \cdot \log(p(y\_i)) + (1 - y\_i) \cdot \log(1 - p(y\_i))

\]

*Fig 4. Log loss function*

* **Gaussian Naive Bayes**

Gaussian Naive Bayes is a probabilistic classifier algorithm that is based on the Bayes’ theorem. It takes into consideration that the input features are independent and they follow a Gaussian (normal) distribution curve. It works by calculating the probability for each class for a set of input features and returns the class with maximum probability.

P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) \cdot P(B|A)}{P(B)}

*Fig 5. Bayes theorem*

* **K-Nearest Neighbours**

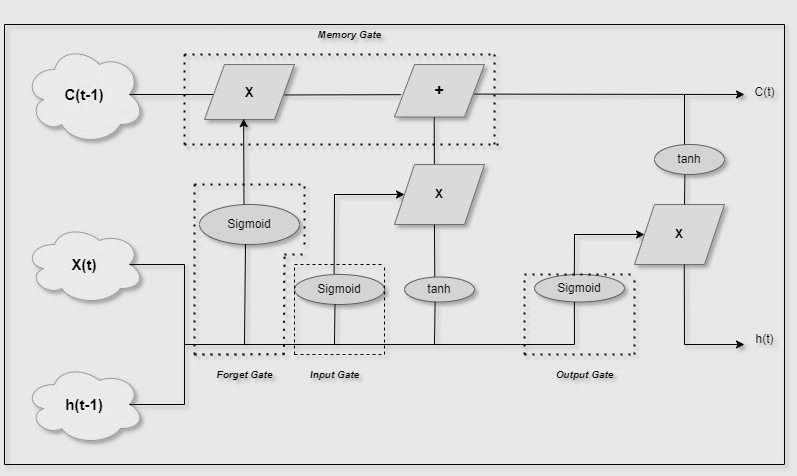
K-nearest neighbours is a lazy learning algorithm that allots a class label to a data point based on the majority of its k-nearest neighbours calculated using a distance metric, in this case, euclidean distance from the given point. It does not store the dataset in memory and is simple to implement.

d(x, y) = \sqrt{\sum\_{i=1}^{n} (y\_i - x\_i)^2}

*Fig 6. Euclidean distance*

* **Long-Short Term Memory (LSTM)**

LSTM is a type of Recurrent Neural Network (RNN) architecture that helps to capture long term and sequential data. It consists of memory cell, input gate, output gate and a forget gate to help retain important information throughout the model training phase.

*Fig 7. Architecture of LSTM Neural Network*

|  |  |  |
| --- | --- | --- |
| **Layer (type)** | **Output Shape** | **Param #** |
| lstm (LSTM) | (None, 304, 256) | 264,192 |
| lstm\_1 (LSTM) | (None, 128) | 197,120 |
| flatten (Flatten) | (None, 128) | 0 |
| dropout (Dropout) | (None, 128) | 0 |
| dense (Dense) | (None, 32) | 4,128 |
| dense\_1 (Dense) | (None, 1) | 33 |

*Fig 8. LSTM Model Architecture*

* **Hyperparameter Tuning**

Hyperparameter refers to parameters of a machine learning model that are set prior to model training process. Hyperparameter tuning is the process of finding optimal values for the hyperparameters of the machine learning models.

The hyperparameter selection for machine learning was done using Grid Search using the scikit-learn library in python. This method allows a single machine learning model to train on a different number of combinations of hyperparameters and returns the best possible combination having the highest accuracy. In LSTM model, the hyperparameter selection was done using random search and techniques early stopping and model checkpoints along with dropout layers were used to prevent overfitting during model training.

**Results**

This study is a comparative analysis of the various machine learning models used to predict whether a given customer will be able to repay the loan disbursed to them given their historical and current financial credit data. The machine learning models used in this paper are trained on two types of datasets: standard cleaned dataset having 303 features and a smaller dataset having top 100 features from the original dataset after applying Principal Component Analysis (PCA) technique.

|  |  |  |
| --- | --- | --- |
| **Confusion Matrix** | **Predicted Values** | |
| **Actual Values** | **True Negatives** | **False Positives** |
| **False Negatives** | **True Positives** |

*Fig 9. Classification Matrix*

The methodology used for evaluating the models is accuracy score. It is a simple model evaluation metric that calculates the ratio of correctly classified inputs (True positives and True negatives) and the total number of inputs classified by the model.

\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}

*Fig 10. Accuracy Score*

Out of the models experimented, gradient boosting models performed the best, closely followed by random forest and decision tree. Gaussian Naive Bayes and K-Nearest neighbours gave exact same results on model evaluation. LSTM model performed well as it understood the current and historical records of the clients by retaining the information but as all deep learning models, it took the longest time to train on 10 epochs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Models** | **Before PCA** | | **After PCA** | |
| **Accuracy** | **Training Time (in seconds)** | **Accuracy** | **Training Time (in seconds)** |
| XGBoost | 98.52% | 4.74 | 97.70% | 0.56 |
| Ada Boost | 97.99% | 36.75 | 98.10% | 4.56 |
| LightGBM | 97.30% | 2.95 | 97.86% | 1.50 |
| Random Forest | 96.62% | 7.27 | 96.18% | 18.54 |
| Decision Tree | 95.12% | 6.68 | 96.20% | 0.70 |
| Gaussian Naive Bayes | 93.89% | 0.26 | 98.09% | 2.84 |
| K-Nearest neighbours | 93.89% | 0.30 | 98.09% | 3.84 |
| LSTM | 95.49% | 671 | 96.00% | 622 |

*Table 2. Comparison of accuracies*

**Conclusion and Future Work**

The banking sector has supported entire economies through the process of disbursing loans to the people. It is one the major sources of income for a bank, hence, the task of accurately predicting whether a given customer will be able to repay the loan or will default the loan becomes even more crucial. Complex problems like these can be automated by harnessing the power of machine learning algorithms. Machine learning algorithms when fed with correct and sufficient amount can make up to 100% correct predictions.

In this experimental study, the performance of 8 different kinds of machine learning models is compared using accuracy as metric for evaluation. Given the nature of dataset used, gradient boosting algorithms performed the best. XGBoost and AdaBoost models performed the best (reaching almost 99%) before and after applying PCA respectively. This study showcases that machine learning models can be applied on real-world banking data to automate the process of predicting loan defaults and solve the complex problem revolutionising the banking industry forever.

Future research on this topic has the potential to scale these models to be practically applied in a real-world banking institution. Using different validation techniques and diversifying model choices, the accuracy can be improved further. Based on the results in this experimental study, it can be proved that machine learning and deep learning models have immense prospect in the financial sector.

**References**

[1] RB, A., & KR, S. K. (2021). Credit card fraud detection using artificial neural network. In Global Transitions Proceedings (Vol. 2, Issue 1, pp. 35–41). Elsevier BV. <https://doi.org/10.1016/j.gltp.2021.01.006>

[2] Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. In Procedia Computer Science (Vol. 162, pp. 503–513). Elsevier BV. https://doi.org/10.1016/j.procs.2019.12.017

[3] Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. In 2017 International Conference on Computing Networking and Informatics (ICCNI). 2017 International Conference on Computing Networking and Informatics (ICCNI). IEEE. <https://doi.org/10.1109/iccni.2017.8123782>

[4] Gao, J., Sun, W., & Sui, X. (2021). Research on Default Prediction for Credit Card Users Based on XGBoost-LSTM Model. In A. Farouk (Ed.), Discrete Dynamics in Nature and Society (Vol. 2021, pp. 1–13). Hindawi Limited. <https://doi.org/10.1155/2021/5080472>

[5] Kandappan, V. A., & Rekha, A. G. (2021). Machine Learning in Finance: Towards Online Prediction of Loan Defaults Using Sequential Data with LSTMs. In Soft Computing: Theories and Applications: Proceedings of SoCTA 2020, Volume 2 (pp. 53-62). Singapore: Springer Singapore.

[6] Jain, A., Gupta, S., & Narula, M. S. Loan Default Risk Assessment using Supervised Learning.

[7] Kibria, M. G., & Sevkli, M. (2021). Application of deep learning for credit card approval: A comparison with two machine learning techniques. *International Journal of Machine Learning and Computing*, *11*(4), 286-290.

[8] Zhang, H., Bi, Y., Jiang, W., Luo, C., Cao, S., Guo, P., & Zhang, J. (2020). Application of random forest classifier in loan default forecast. In *Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, July 17–20, 2020, Proceedings, Part III 6* (pp. 410-420). Springer Singapore.

[9] Patel, B., Patil, H., Hembram, J., & Jaswal, S. (2020). Loan Default Forecasting using Data Mining. In 2020 International Conference for Emerging Technology (INCET). 2020 International Conference for Emerging Technology (INCET). IEEE. https://doi.org/10.1109/incet49848.2020.9154100

[10] Mohmad, Y. A. (2022). Credit Card Fraud Detection Using LSTM Algorithm. In Wasit Journal of Computer and Mathematics Science (Vol. 1, Issue 3, pp. 26–35). Wasit University. <https://doi.org/10.31185/wjcm.60>

[11] Shivanna, A., & Agrawal, D. P. (2020). Prediction of Defaulters using Machine Learning on Azure ML. In 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE. <https://doi.org/10.1109/iemcon51383.2020.9284884>

[12] Yu, Y. (2020). The Application of Machine Learning Algorithms in Credit Card Default Prediction. In 2020 International Conference on Computing and Data Science (CDS). 2020 International Conference on Computing and Data Science (CDS). IEEE. <https://doi.org/10.1109/cds49703.2020.00050>

[13] Arora, S., Bindra, S., Singh, S., & Kumar Nassa, V. (2022). Prediction of credit card defaults through data analysis and machine learning techniques. In Materials Today: Proceedings (Vol. 51, pp. 110–117). Elsevier BV. https://doi.org/10.1016/j.matpr.2021.04.588

[14] Ndayisenga, T. (2021). *Bank loan approval prediction using machine learning techniques* (Doctoral dissertation).

[15] Madaan, M., Kumar, A., Keshri, C., Jain, R., & Nagrath, P. (2021). Loan default prediction using decision trees and random forest: A comparative study. In IOP Conference Series: Materials Science and Engineering (Vol. 1022, Issue 1, p. 012042). IOP Publishing. <https://doi.org/10.1088/1757-899x/1022/1/012042>

[16] Hassan, M. M., & Mirza, T. (2020). Credit card default prediction using artificial neural networks. *GIS Science Journal*, *7*, 383-390.

[17] Bacova, A., & Babic, F. (2021). Predictive Analytics for Default of Credit Card Clients. In 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI). 2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI). IEEE. <https://doi.org/10.1109/sami50585.2021.9378671>

[18] Lai, L. (2020). Loan Default Prediction with Machine Learning Techniques. In 2020 International Conference on Computer Communication and Network Security (CCNS). 2020 International Conference on Computer Communication and Network Security (CCNS). IEEE. https://doi.org/10.1109/ccns50731.2020.00009

[19] Barbaglia, L., Manzan, S., & Tosetti, E. (2021). Forecasting Loan Default in Europe with Machine Learning. In Journal of Financial Econometrics (Vol. 21, Issue 2, pp. 569–596). Oxford University Press (OUP). <https://doi.org/10.1093/jjfinec/nbab010>

[20] Kim, H., Cho, H., & Ryu, D. (2020). Corporate Default Predictions Using Machine Learning: Literature Review. In Sustainability (Vol. 12, Issue 16, p. 6325). MDPI AG. <https://doi.org/10.3390/su12166325>

[21] Agarwal, A., Rana, A., Gupta, K., & Verma, N. (2020). A Comparative Study and enhancement of classification techniques using Principal Component Analysis for credit card dataset. In 2020 International Conference on Intelligent Engineering and Management (ICIEM). 2020 International Conference on Intelligent Engineering and Management (ICIEM). IEEE. https://doi.org/10.1109/iciem48762.2020.9160230

[22] Sheikh, M. A., Goel, A. K., & Kumar, T. (2020). An Approach for Prediction of Loan Approval using Machine Learning Algorithm. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE. https://doi.org/10.1109/icesc48915.2020.9155614