Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Outlier Detection Using Gaussian Mixture Model Clustering to Optimize XGBoost for Credit Approval Prediction

De Rosal Ignatius Moses Setiadi [1,2]*, Ahmad Rofiqul Muslikh [3], Syahroni Wahyu Iriananda [4], Warto [5], Jutono Gondohanindijo [6], and Arnold Adimabua Ojugo [7]

[1] Faculty of Computer Science, Dian Nuswantoro University, Semarang, Central Java 50131, Indonesia; e-mail : moses@dsn.dinus.ac.id

[2] Research Center for Quantum Computing and Materials Informatics, Faculty of Computer Science, Dian Nuswantoro University, Semarang 50131, Indonesia

[3] Faculty of Information Technology, University of Merdeka, Malang, East Java 65146, Indonesia; e-mail : rofickachmad@unmer.ac.id

[4] Department of Informatics Engineering,Universitas Widya Gama Malang, Indonesia; e-mail : syahroni@widyagama.ac.id

[5] Informatics Department, Faculty of Dakwah, UIN Profesor Kiai Haji SaifuddinZuhri, Purwokerto, Indonesia; e-mail : warto@uinsaizu.ac.id

[6] Faculty of Technics and Informatics, AKI University, Semarang, Central Java 50136, Indonesia; e-mail : jutono.gondohanindijo@unaki.ac.id

[7] Department of Computer Science, Federal University of Petroleum Resources Effurun, Nigeria; e-mail : ojugo.arnold@fupre.edu.ng

* Corresponding Author : De Rosal Ignatius Moses Setiadi

Abstract: Credit approval prediction is one of the critical challenges in the financial industry, where the accuracy and efficiency of credit decision-making can significantly affect business risk. This study proposes an outlier detection method using the Gaussian Mixture Model (GMM) combined with Extreme Gradient Boosting (XGBoost) to improve prediction accuracy. GMM is used to detect outliers with a probabilistic approach, allowing for finer-grained anomaly identification compared to distance- or density-based methods. Furthermore, the data cleaned through GMM is processed using XGBoost, a decision tree-based boosting algorithm that efficiently handles complex datasets. This study compares the performance of XGBoost with various outlier detection methods, such as LOF, CBLOF, DBSCAN, IF, and K-Means, as well as various other classification algorithms based on machine learning and deep learning. Experimental results show that the combination of GMM and XGBoost provides the best performance with an accuracy of 95.493%, a recall of 91.650%, and an AUC of 95.145%, outperforming other models in the context of credit approval prediction on an imbalanced dataset. The proposed method has been proven to reduce prediction errors and improve the model's reliability in detecting eligible credit applications.

Keywords: Credit Approval Prediction; Data Preprocessing; Ensemble Learning; Gaussian Mixture Model; Imbalanced Data; Outlier Clustering; Outlier Detection; XGBoost.

Received: September, 30th 2024
Revised: October, 30th 2024
Accepted: October, 31st 2024
Published: November, 1st 2024

1. Introduction

Credit approval is one of the most critical aspects of the financial industry, directly affecting financial institutions' business risk and stability [1]. Accurate credit decisions are essential to reduce the risk of default, which can harm the financial condition of institutions and limit borrowers' access to needed credit services. Therefore, an accurate and efficient credit evaluation process is a top priority in various financial sectors, aiming to minimize risk and expand access to credit for qualified borrowers. In this context, machine learning techniques have been shown to improve the accuracy and speed of credit approval predictions[2], [3].

Credit datasets often contain outliers due to large demographic variations among borrowers or errors in data input. For example, outliers such as very high income or unusual credit history can affect the credit evaluation results. It is important to detect and handle such

DOI : 10.62411/jcta.11638

outliers before the machine learning model is trained[4], [5]. Outliers can reduce the accuracy of predictions due to the bias introduced into the model[6]–[9], and mishandling outliers can result in incorrect credit decisions. Several studies have shown that proper outlier handling can improve model performance in various other fields, such as healthcare[10], security[11], [12], agriculture[13], industrial monitoring[14], etc.

Outlier detection has become a significant focus in data preprocessing because it ensures that the data used is accessible from the influence of extreme values that do not reflect the general pattern. Some commonly used outlier detection methods are Local Outlier Factor (LOF)[15]–[17], Cluster-Based Local Outlier Factor (CBLOF)[11], Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[10], [13], [16], Isolation Forest (IF)[13], [15], [17], K-Means[16], [18], and Gaussian Mixture Model (GMM)[16], [17]. LOF identifies outliers based on local density around data points, but this method is often sensitive to specific parameters and is less effective on datasets with many dimensions. CBLOF, which works by dividing the data into large and small clusters, is more effective at detecting outliers in small clusters but still has weaknesses in handling complex datasets. K-Means is efficient clustering[19], [20] but has significant weaknesses in detecting outliers due to its sensitivity to Euclidean distance and the assumption that the data is evenly divided into clusters.

DBSCAN is a density-based clustering method that effectively detects outliers as data points, not in dense clusters. Its advantage is its ability to identify outliers without specifying the number of clusters in advance[21]. However, its sensitivity to parameters such as the epsilon radius (ϵ) and the minimum number of points can also be problematic. IF is an ensemble-based method that detects outliers by isolating the data through repeated partitioning. IF has the advantage of handling high-dimensional data with relatively fast computational speed[22], but it can be susceptible to parameter settings such as sample size and number of trees. GMM offers a more flexible solution for outlier detection using a probabilistic approach. GMM divides the data into several Gaussian distributions, allowing for more refined outlier identification by considering probabilistic distributions rather than relying solely on distance or density[23], [24]. With this flexibility, GMM excels in detecting outliers in complex datasets such as credit datasets, where non-linear and diverse patterns often emerge.

After evaluating several outlier detection methods, GMM gave the best results, especially when combined with the XGBoost model. XGBoost is a highly efficient gradient boosting-based algorithm known for its high accuracy. With its ability to iteratively improve each decision tree's performance, XGBoost can capture prediction errors from previous models and correct them in the next step[25]. This makes it an ideal choice for complex datasets with many features, such as credit datasets, which contain diverse variations[26], [27]. GMM serves to detect outliers probabilistically, which allows for more precise identification of anomalies in the dataset so that XGBoost can focus on more relevant data patterns. The combination of XGBoost with GMM is predicted to give better results than other outlier detection methods, such as LOF, CBLOF, or K-Means, which tend to have limitations in handling data complexity and uneven distribution.

Random Forest (RF), another classification model often compared to XGBoost, offers advantages in stability and robustness to small variations in data. RF uses an ensemble approach by combining many independent decision trees, thus reducing the risk of overfitting[28], [29]. Its advantage is handling non-linear and complex data[30]–[33]. Still, RF can be less efficient than XGBoost, especially on very large datasets, because it does not explicitly capture the relationships between trees as XGBoost does. On the other hand, models such as Support Vector Machine (SVM)[34], K-Nearest Neighbors (KNN)[26], Logistic Regression (LR)[34], [35], Decision Tree (DT)[26], [36], Naive Bayes (NB)[36], and neural networks such as BiLSTM and BiGRU have also been evaluated. SVM is known for its ability to classify high-dimensional data but often requires complex parameter tuning[37]. KNN is easy to implement but is very sensitive to outliers and is often inefficient on large datasets. Conversely, LR is simple and easy to interpret but is less effective in handling data with non-linear relationships. DT offers a simple and easy-to-interpret tree-based approach but is prone to overfitting, especially on imbalanced datasets. Although fast and efficient, NB operates under the assumption of independence between features, which is often unrealistic in complex datasets. Meanwhile, BiLSTM and BiGRU, as recurrent neural network models, offer strong capabilities in capturing data sequence dependencies[10], [32], [38]–[40], but they require longer training time than decision tree-based models such as RF and XGBoost.

This hypothesis leads to the conclusion that XGBoost with outlier detection using GMM is more effective than other methods, including RF with various outlier detection approaches, especially in credit approval prediction. The higher performance in the initial trials indicates that XGBoost is superior in handling complex and outlier-prone datasets, while RF shows good stability but may require further optimization techniques to compete with the accuracy of XGBoost. This study contributes by showing that combining XGBoost and the Gaussian Mixture Model can significantly improve the accuracy of credit approval prediction. The structure of this paper consists of several main sections: first, this section; second is a literature review related to outlier detection methods and classification techniques; third, the methodology used in this study; fourth, experimental results and in-depth analysis of model performance; fifth, comparison with related studies, and finally, conclusions and recommendations for future research.

2. Related Works

Several studies have focused on improving the performance of machine learning models for loan approval prediction using various techniques and models. An important approach is to use hybrid models that combine feature selection, instance selection, and classification techniques. Weng and Huang [41] proposed a hybrid model that integrates decision trees with clustering techniques such as Expectation Maximization (EM) to improve prediction accuracy by handling irrelevant features and refining the process. Their model performed better than traditional machine learning techniques.

Another approach to loan approval prediction emphasizes the importance of handling outliers and class imbalance in a credit dataset[35] and used logistic regression to predict defaulting borrowers, focusing on personal attributes of the borrowers such as age, credit history, and income. Their model successfully reduced non-performing assets by accurately predicting loan defaults. However, the simplicity of logistic regression in handling linear relationships limits its ability to model complex and non-linear patterns in large datasets.

Prastyo et al.[42] proposed a Naïve Bayes-based model, using Information Gain (IG) for feature selection to improve classifier performance. When their model was tested, the accuracy reached 86.29% and proved that preprocessing steps, such as feature selection and discretization, are crucial in improving the model performance. Similarly, Kadam et al. [43] used SVM and Naïve Bayes for loan prediction, with Naïve Bayes outperforming the other models regarding loan forecasting. However, SVM often struggles with high-dimensional and noisy datasets, making it less suitable for complex credit datasets.

Recent work has also investigated ensemble learning techniques. Viswanatha et al. [44] explored various ensemble models, including RF and KNN, to predict loan approval status. They found that Naïve Bayes provided the highest accuracy of 83.73%, although Random Forest showed robustness in handling non-linear and high-dimensional data. However, these studies did not integrate advanced clustering methods, such as DBSCAN and K-Means, which can further improve the model robustness by detecting outliers and underlying patterns. In another study, Diwate et al. [45] used data mining techniques for loan approval prediction, applying models such as RF and DT to identify safe customers for loan disbursement. Their findings highlight the need to balance simplicity and accuracy, showing that ensemble methods such as RF can outperform simpler classifiers such as DT in more complex credit datasets.

The research gap shows that although ensemble methods such as Random Forest (RF) have been widely used, little research has explored the combination of probabilistic outlier detection methods such as the Gaussian Mixture Model (GMM) with classification models. Techniques like K-Means and DBSCAN are less flexible in handling complex data distributions, while GMM offers a more effective probabilistic approach. The combination of RF with GMM has potential, but the use of XGBoost, which has been proven efficient and has high accuracy in various studies, needs to be explored when combined with GMM. Therefore, XGBoost and GMM are worth considering as more optimal approaches.

3. Proposed Method

Based on the previous analysis, it was found that outlier detection and feature optimization play an essential role in improving the accuracy of credit approval prediction. The hypothesis is that combining an optimal outlier detection method, followed by a robust ensemble-based classification algorithm, can produce a more accurate and efficient prediction model. This study uses the GMM as a flexible outlier detection method to identify anomalous data probabilistically. The cleaned data is then fed into the XGBoost model, a gradient boosting-based algorithm, which is known to excel in handling complex and diverse datasets. The combination of GMM and XGBoost is expected to improve prediction accuracy by capturing relevant patterns in credit datasets containing outliers. As an illustration, the proposed method is shown in Figure 1, which shows the main stages in preprocessing, outlier detection with GMM, and classification with XGBoost.

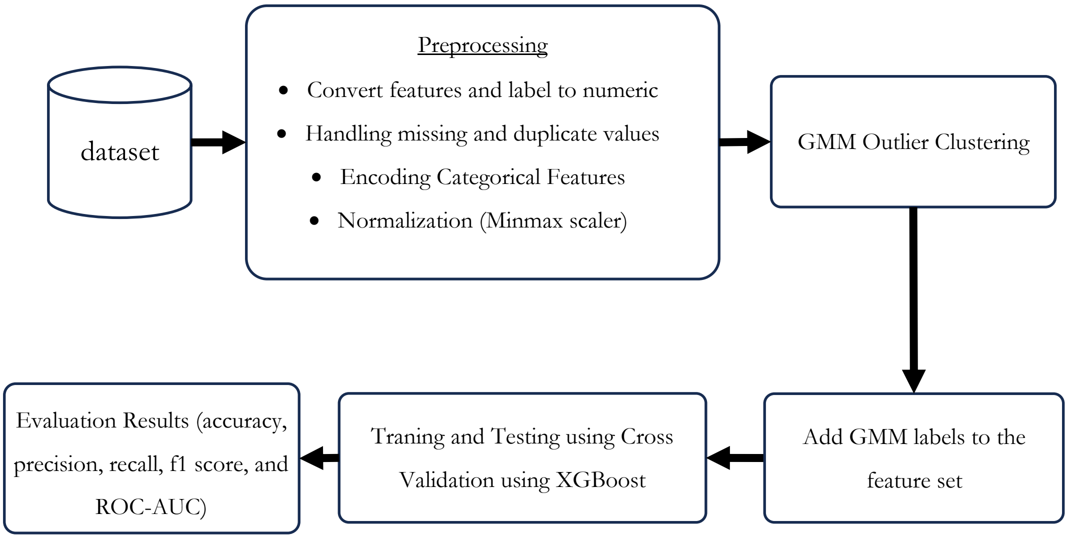
dataset
Preprocessing
Convert features and label to numeric
Handling missing and duplicate values
Encoding Categorical Features
Normalization (Minmax scaler)
GMM Outlier Clustering
Add GMM labels to the feature set
Traning and Testing using Cross Validation using XGBoost
Evaluation Results (accuracy, precision, recall, f1 score, and ROC-AUC)

Figure 1. Proposed method flow.

3.1. Preprocessing

Preprocessing is essential to preparing the dataset for model training and evaluation. The dataset consists of both categorical and numerical features, requiring transformation before further analysis. The preprocessing steps include:

1. In the credit approval dataset, the class labels indicating the credit approval status are initially denoted by the symbol '+' for approval and '-' for rejection. These labels must be converted to numeric values for the machine-learning model to work properly. In this case, '+' is converted to 1 (approval) and '-' to 0 (rejection)
2. Some columns in the dataset may contain non-numeric values that must be converted to numeric format for further processing. For example, columns 'A2' and 'A14' are converted to numeric values using the pd.to_numeric function. If there are any errors or invalid values, they are converted to NaN (missing values) and then removed in the next step.
3. Records with missing values are removed using df.dropna() and duplicates are removed using df.drop_duplicates(). This ensures that the model is trained on clean and relevant data.
4. Feature Encoding: All categorical variables are encoded into numerical values using the LabelEncoder, transforming text labels into integers. Let x_{cat} represent a categorical feature; it is converted as Equation (1).

$$x_{\text{encoded}} = \text{LabelEncoder}(x_{\text{cat}}) \tag{1}$$

5. Normalization: The numeric features are normalized to ensure all values are within a standard range, which helps improve the efficiency of the clustering and classification models. MinMaxScaler transforms each numeric feature into the range [0, 1]. This process is formulated in Equation (2).

$$\hat{x}_i = \frac{x_{\text{i}} - \min(x)}{\max(x) - \min(x)} \tag{2}$$

3.2. Outlier Detection with Gaussian Mixture Model (GMM)

After preprocessing, the dataset is analyzed to detect outliers using GMM. GMM is a probability-based clustering method that models data as a mixture of several Gaussian distributions. Unlike distance-based clustering methods, GMM is more flexible because it can identify complex distributions and capture variations in the data.

3.2.1. GMM Clustering for Outlier Detection

GMM divides the dataset into several different Gaussian distributions, and each data point has a probability of belonging to one of these distributions. In this way, GMM allows the identification of outliers based on the low probability that a data point belongs to any distribution. The Gaussian distribution is represented in Equation (3).

$$P(x) = \sum_{k=1}^{K} \phi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{3}$$

Where $P(x)$ is the probability that a data point x belongs to one of the distributions, ϕ_k is the weight for the k-th distribution, $\mathcal{N}(x|\mu_k, \Sigma_k)$ is a Gaussian distribution with mean μ_k and covariance Σ_k, K is the number of Gaussian distributions in the model.

Outliers are identified as data points with low probability in any distribution. After GMM is applied, labels indicating the probability of the data point belonging to a distribution are added to the dataset as additional features. These labels improve the accuracy of subsequent classification models.

3.2.2. Addition of GMM as a Feature

The clustering results from GMM are added to the dataset as new features. Each data point is given a label indicating which distribution is most likely associated with that point, which is then used in the XGBoost classification model. Thus, XGBoost can handle not only the original features but also additional information from GMM, which helps handle outliers more effectively.

3.3 Model Training and Cross-Validation

After adding features from GMM, the dataset is split into training and testing sets using Stratified K-Fold Cross-Validation. Cross-validation helps ensure that each fold has the same distribution of class labels, thereby reducing potential bias during model evaluation. XGBoost is a highly efficient gradient-boosting algorithm known for superior classification performance. XGBoost works by building a series of decision trees, where each tree improves on the prediction error of the previous tree. This process is done iteratively to minimize the overall prediction error. XGBoost uses the loss function defined in Equation (4) in each iteration. The final prediction is made by combining the results of all the decision trees using Equation (5).

$$L = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{5}$$

Where L is the total loss function, $\ell(y_i, \hat{y}_i)$ is the loss between the actual label y_i and the prediction $\hat{y}_i$, $\Omega(f_k)$ is the complexity penalty for the k-th decision tree, $\hat{y}_i$ is a prediction for input x_i, $f_k(x_i)$is the output of the k-th decision tree for the input x_i, K is the number of decision trees in the model.

3.4. Model Evaluation

The model's performance is evaluated based on several metrics: accuracy, precision, recall, F1-score, and AUC-ROC. These metrics are computed for each fold in the cross-validation process, and the average value is reported. The confusion matrix is also generated to evaluate the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN)[46]. These evaluation values are used in several evaluation metrics used to comprehensively assess model performance, namely:

1. Accuracy is the proportion of correct predictions to all predictions. In the context of an imbalanced dataset, for example, if there are more eligible borrowers than unqualified borrowers, accuracy does not fully describe the model's performance. The formula for accuracy can be calculated by Equation (6).

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

2. Precision measures how accurate the model is in predicting loan approvals. This metric is important because it indicates the proportion of correct predictions among all positive predictions made by the model. The higher the precision, the less likely the model is to provide a wrong loan approval. Precision is calculated by Equation (7).

$$\text{precision} = \frac{TP}{TP + FP} \tag{7}$$

3. Recall measures how well the model is in detecting loan-worthy borrowers. In the context of loan approvals, the higher the recall, the more effectively the model can identify eligible borrowers. Recall can be calculated by Equation (8).

$$\text{recall} = \frac{TP}{TP + FN} \tag{8}$$

4. F1-score is the balance value between precision and recall. This metric provides a more balanced picture of the model's performance, especially when there is an imbalance between loan approvals and rejections. F1-score is calculated by Equation (9).

$$\text{f1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{9}$$

5. Specificity measures how well the model predicts correct credit rejections, i.e., the ability to detect TP among all cases of credit rejection. In the context of credit approval, specificity is important to minimize the risk of credit approval errors, which can be calculated by Equation (10).

$$\text{specificity} = \frac{TN}{TN + FP} \tag{10}$$

6. Area Under the Curve - Receiver Operating Characteristic (AUC-ROC) measures the ability of the model to distinguish between credit approvals and rejections. AUC is the area under the ROC curve, which describes the relationship between the True Positive Rate (Recall) and False Positive Rate (Specificity). A higher AUC value indicates that the model can better separate eligible and unworthy borrowers. This value can be calculated by Equation (11).

$$\text{AUC} = \int_0^1 \text{ROC curve} \tag{11}$$

4. Results and Discussion

The dataset used in this study is sourced from the UCI Machine Learning Repository and concerns credit card approval[47]. It contains 690 instances and 15 features, comprising a mix of categorical, integer, and real-valued attributes. Several features represent nominal data, some with a small number of categories and others with more varied values. A few features represent discrete numeric data, including financial figures or counts. Some continuous features represent numerical values related to the applicant's credit history or financial status. The target variable in this dataset is the class label, which indicates whether a credit card application was approved (+ change to 1) or rejected (- change to 0). Table 1 provides a summary of the dataset’s features.

Next, the dataset distribution can be seen in Figure 2. Class distribution is crucial to understanding the balance between approved and rejected applications. In some cases, class imbalance can affect model performance, leading to biased results where the majority class dominates. Data distribution analysis was also carried out in this study to enrich the results.

Table 1. Dataset Features Summary.

Feature	Type	Description
A1	Categorical	Nominal, representing some category with limited values
A2	Real	Continuous, representing a numerical attribute
A3	Integer	Discrete numeric value
A4-A14	Categorical	Nominal features representing various characteristics of the applicant
A15	Integer	A discrete numeric value representing some count or score
Class	Binary	Target variable: 1 (Approved), 0 (Rejected)

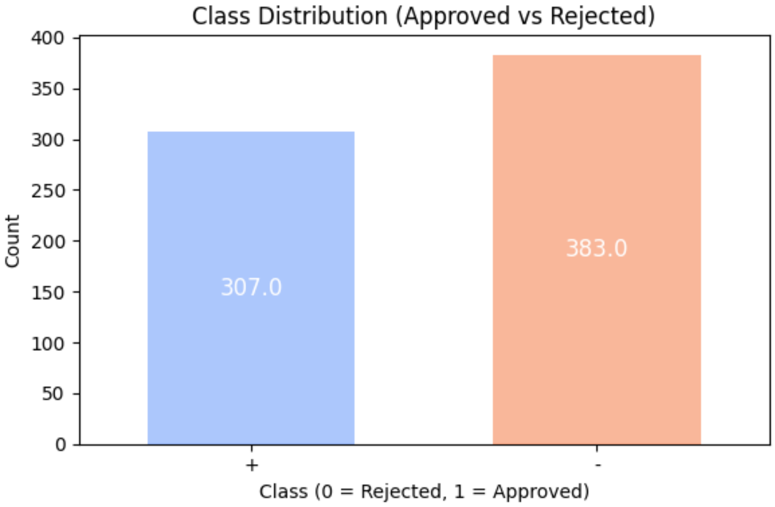

Figure 2. Dataset Class distribution.

In GMM and XGBoost modeling, parameters must be adjusted to achieve optimal performance. In GMM, n_components is set to 3 to separate the data into three Gaussian distributions, while covariance_type is set to 'full' to increase flexibility in handling data variability. Random_state is used to ensure consistency of results. Meanwhile, in XGBoost, learning_rate is set to 0.1 to control the step size of each iteration, with max_depth set to 6 to maintain a balance between complexity and overfitting. The n_estimators parameter is set to 100 to ensure the model has enough trees to learn effectively. In addition, eval_metric uses 'logloss' to measure the classification error. Numerical data is normalized with MinMaxScaler to ensure uniform scaling across features, which is important for model stability. Cross-validation using StratifiedKFold maintains a proportional distribution of classes across folds, ensuring a representative evaluation. Furthermore, the first experiment was carried out by comparing the performance of the proposed method by comparing XGBoost with RF without and with various clustering methods for outlier detection, such as LOF, CBLOF, DBSCAN, IF, K-Means, and GMM. The comparison results are presented in Table 2.

Table 2 presents the performance comparison results between XGBoost and RF with and without clustering-based outlier detection. Generally, the model's performance without outlier detection shows lower results than when the outlier detection method is applied. With XGBoost, the accuracy without outlier detection only reaches 86.331%, while RF is slightly better with an accuracy of 87.081%. In addition to accuracy, other metrics such as recall, precision, F1 score, and AUC also show that XGBoost's performance is not better than RF without outlier handling. When outlier detection methods based on LOF, CBLOF, and IF are applied, RF still outperforms XGBoost in most metrics. For example, with LOF, RF's accuracy reaches 87.381%, higher than XGBoost's, which only reaches 85.883%. This shows that for simpler outlier detection methods or those that rely on locality, such as LOF and CBLOF, RF is better able to handle outliers than XGBoost. However, different results emerge when density-based and probabilistic clustering methods are used. When using DBSCAN, K-Means, and especially GMM, XGBoost shows significant performance improvements and begins to outperform RF. With DBSCAN, XGBoost's accuracy jumps to 89.333%, higher than RF's 87.313%. The same trend is seen in the K-Means method, where XGBoost achieves an accuracy of 91.590% compared to RF's 91.140%. The best overall performance is obtained when XGBoost is combined with GMM, with an accuracy of 95.493%, outperforming RF's 95.192%. XGBoost also shows advantages in recall, F1 score, specificity, and AUC metrics when using GMM. However, in the precision metric, RF with GMM is slightly better, with a value of 98.262% compared to 98.248% on XGBoost, although this difference is insignificant. These findings show that GMM is very effective in detecting outliers in credit data, allowing XGBoost to leverage this information and achieve superior performance compared to other models. Overall, these results confirm that appropriate outlier detection methods, such as GMM, provide significant performance improvements, especially in the XGBoost model, which consistently outperforms RF under complex and varied dataset conditions.

Table 2. Comparison of XGBoost and RF with various clustering methods.

Method	Accuracy	Precision	Recall	F1 Score	Specificity	AUC
RF only	87.081	86.024	85.954	85.733	88.004	86.979
XGBoost only	86.331	85.950	84.271	84.760	88.001	86.135
RF+LOF	87.381	86.938	85.627	85.963	88.819	87.223
XGBoost+LOF	85.883	85.468	83.943	84.326	87.453	85.698
RF+CBLOF	87.230	86.789	85.294	85.736	88.818	87.056
XGBoost+CBLOF	86.933	86.592	84.949	85.493	88.548	86.749
RF+DBSCAN	87.313	84.127	88.333	86.178	86.486	87.409
XGBoost+DBSCAN	89.333	89.348	86.966	87.973	91.281	89.123
RF+IF	87.531	87.389	85.276	86.046	89.371	87.323
XGBoost+IF	86.632	85.361	85.616	85.238	87.460	86.538
RF+K-Means	91.140	93.318	86.644	89.719	94.827	90.735
XGBoost+K-Means	91.590	92723	88.644	90.487	93.998	91.321
RF+GMM	95.192	**98.262**	90.977	94.417	98.641	94.809
XGBoost+GMM	**95.493**	98.248	**91.650**	**94.784**	**98.641**	**95.145**

Furthermore, in Table 3, various machine learning model classifiers are tested, including Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Gated Recurrent Unit (BiGRU), deep learning models combined with GMM clustering for outlier detection.

Table 3. Comparison of GMM clustering outlier detection with various classifiers.

Method	Accuracy	Precision	Recall	F1 Score	Specificity	AUC
NB	85.134	92.285	73.248	81.493	94.820	84.034
KNN	90.389	90.313	88.650	89.248	91.829	90.239
LR	83.925	80.163	86.277	82.817	82.014	84.145
SVM	92.639	96.066	87.311	91.296	97.004	92.158
DT	92.043	91.447	91.299	91.188	92.651	91.975
BiLSTM	90.387	89.285	89.649	89.353	91.003	95.005
BiGRU	91.591	92.443	88.983	90.438	93.739	**95.439**
RF	95.192	**98.262**	90.977	94.417	98.641	94.809
XGBoost	**95.493**	98.248	**91.650**	**94.784**	**98.641**	95.145

Table 3 shows that in the context of credit approval, especially in imbalanced data, metrics such as recall, precision, F1 score, and AUC are more relevant than accuracy only because the main goal is to minimize errors in detecting applications that are potentially wrongly approved or rejected. XGBoost performed best regarding recall (91.650%) and F1 score (94.784%), which ensure that the model can detect as many viable applications as possible without sacrificing overall accuracy. The BiGRU model is also noteworthy because its AUC value is the highest but unsupported in other metrics. Meanwhile, XGBoost's AUC (95.145%) is the second highest, indicating the model's ability to consistently distinguish between approved and rejected applications. Although superior in precision (98.262%), RF is less effective in recall, which is an important measure in credit applications because it is more important to minimize the risk of wrong rejections. Therefore, although RF and other models have advantages in certain aspects, XGBoost with GMM is more suitable for credit approval, especially in the context of balanced decision-making between accepting and rejecting applications.

5. Comparison

In this section, this study presents a comparison with several related studies. It should be noted that the comparison was conducted with a study on credit approval prediction that used the same dataset. The results of the comparison are presented in Table 4.

Table 4. Comparison with related research.

Method	Accuracy	Precision	Recall	F1 Score	Specificity	AUC
Ref [42]	86.29	86.33	86.29	86.30	-	91.52
Ref [48]	87.10	87.91	89.26	88.60	-	-
Ref [41]	94	95	91	93	-	-
Proposed	95.493	98.248	91.650	94.784	98.641	95.145

Table 4 shows that the proposed approach outperforms previous studies in key metrics. The recall reaches 91.650%, higher than Ref [42] (86,29%) and Ref [48] (89,26%), indicating the effectiveness of the model in detecting credit-worthy applications. Although Ref [41] has a strong recall of 91%, the proposed approach still outperforms with a better balance between recall and precision. In terms of precision, the proposed model achieves 98.248%, close to Ref [13] which has a precision of 95, but still higher. The F1 score of the proposed method (94.784%) also shows a balanced performance between positive and negative detection compared to Ref [41] with an F1 score of 93%.

In addition, the AUC of 95.145 of the proposed approach surpasses the AUC of Ref [42] (91,52%), confirming that this model is better at distinguishing between approved and rejected applications. With a specificity of 98.641%, this model is also more effective in rejecting ineligible applications, which is rarely discussed in other studies, including Ref[41]. Overall, this approach improves performance in almost all metrics, making it more suitable for credit approval applications that require a balance between precise detection and overall accuracy.

6. Conclusions

This study aims to improve the accuracy of credit approval prediction by combining GMM-based outlier detection with the XGBoost algorithm. The experiment results show that the combination of GMM and XGBoost can provide the best performance compared to other methods, with an accuracy of 95.493%, a recall of 91.650%, and an AUC of 95.145%. This shows that the proposed approach effectively handles outliers and can improve the ability to detect eligible credit applications while minimizing the risk of wrong rejections, especially in imbalanced datasets. The main finding of this study is that GMM, with its probabilistic approach, successfully detects outliers more finely than distance- or density-based methods. Integrating outlier clustering results from GMM into the XGBoost model significantly improves prediction accuracy, especially on credit data that tends to contain non-linear patterns and anomalies. This proves that choosing the right outlier detection method, especially GMM, can significantly optimize the credit prediction model. The limitation of this study lies in the complexity of the GMM model, especially in determining the optimal number of Gaussian components (n_components). In addition, applying this method to very large datasets may require longer computational time. This study is also limited to only one credit dataset, so further research is needed to test the generalization of this model to various datasets and domains. For further research, it is recommended to explore the combination of GMM with other more complex classification methods, such as deep learning models, and test this approach on datasets with higher dimensions or more dynamic data.

Author Contributions: Conceptualization: D.R.I.M.S.; methodology: D.R.I.M.S.; software: A.R.M.; validation: S.W.I, W. and J.G.; formal analysis: A.A.O; investigation: S.W.I, W. and J.G; resources: A.R.M.; data curation: D.R.I.M.S.; writing—original draft preparation: D.R.I.M.S.; writing—review and editing: all; visualization: S.W.I.; supervision: D.R.I.M.S.; project administration: A.R.M.; funding acquisition: all.

Funding: This research received no external funding.

Data Availability Statement: The code and mirror dataset can be downloaded via the URL https://github.com/MosesdeRosal/CC-ApprovalXGB-GMM.git.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] B. Casu, L. Chiaramonte, E. Croci, and S. Filomeni, "Access to Credit in a Market Downturn," *J. Financ. Serv. Res.*, vol. 66, no. 2, pp. 143–169, Oct. 2024, doi: 10.1007/s10693-022-00388-x.

[2] Y. Abakarim, M. Lahby, and A. Attioui, "Towards An Efficient Real-time Approach To Loan Credit Approval Using Deep Learning," in *2018 9th International Symposium on Signal, Image, Video and Communications (ISIVC)*, Nov. 2018, pp. 306–313. doi: 10.1109/ISIVC.2018.8709173.

[3] M. F. Faisal, M. N. U. Saqlain, M. A. S. Bhuiyan, M. H. Miraz, and M. J. A. Patwary, "Credit Approval System Using Machine Learning: Challenges and Future Directions," in *2021 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, Dec. 2021, pp. 76–82. doi: 10.1109/CoNTESA52813.2021.9657153.

[4] Y. Wang, M. Wang, Y. Pan, and J. Chen, "Joint loan risk prediction based on deep learning-optimized stacking model," *Eng. Reports*, vol. 6, no. 4, Apr. 2024, doi: 10.1002/eng2.12748.

[5] P. S. A. B. Reddy, M. G. Reddy, and R. S. Ponmagal, "An approach for prediction of loan approval using ML algorithm," in *4th International Conference on Internet of Things, ICIoT2023*, 2024, p. 020120. doi: 10.1063/5.0217401.

[6] E. B. Wijayanti, D. R. I. M. Setiadi, and B. H. Setyoko, "Dataset Analysis and Feature Characteristics to Predict Rice Production based on eXtreme Gradient Boosting," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 299–310, Feb. 2024, doi: 10.62411/jcta.10057.

[7] B. I. Igoche, O. Matthew, P. Bednar, and A. Gegov, "Integrating Structural Causal Model Ontologies with LIME for Fair Machine Learning Explanations in Educational Admissions," *J. Comput. Theor. Appl.*, vol. 2, no. 1, pp. 65–85, Jun. 2024, doi: 10.62411/jcta.10501.

[8] F. Omoruwou, A. A. Ojugo, and S. E. Ilodigwe, "Strategic Feature Selection for Enhanced Scorch Prediction in Flexible Polyurethane Form Manufacturing," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 346–357, Feb. 2024, doi: 10.62411/jcta.9539.

[9] J. A. Ingio, A. S. Nsang, and A. Iorliam, "Optimizing Rice Production Forecasting Through Integrating Multiple Linear Regression with Recursive Feature Elimination," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 96–108, Aug. 2024, doi: 10.62411/faith.2024-17.

[10] N. R. M and S. Satheeskumaran, "An efficient multi-disease prediction model using advanced optimization aided weighted convolutional neural network with dilated gated recurrent unit," *Intell. Decis. Technol.*, vol. 18, no. 2, pp. 769–798, Jun. 2024, doi: 10.3233/IDT-240368.

[11] Z. S. Dhahir, "A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLOF-Based Feature Engineering and XGBoost," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 174–190, Sep. 2024, doi: 10.62411/faith.2024-33.

[12] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Enhancing Critical Infrastructure Security: Unsupervised Learning Approaches for Anomaly Detection," *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, p. 236, Sep. 2024, doi: 10.1007/s44196-024-00644-z.

[13] A. R. Muslikh, P. N. Andono, A. Marjuni, and H. A. Santoso, "Ensemble IDO Method for Outlier Detection and N2O Emission Prediction in Agriculture," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 7, 2024, doi: 10.14569/IJACSA.2024.0150737.

[14] R. Wei, Z. Li, L. Geng, M. Wuken, and Y. Liu, "Industrial image anomaly detection based on multi Gaussian discriminant model and robust core set," *Meas. Sci. Technol.*, vol. 35, no. 11, p. 116009, Nov. 2024, doi: 10.1088/1361-6501/ad6c76.

[15] E. F. Agyemang, "Anomaly detection using unsupervised machine learning algorithms: A simulation study," *Sci. African*, vol. 26, p. e02386, Dec. 2024, doi: 10.1016/j.sciaf.2024.e02386.

[16] K. A. ElDahshan, G. E. Abutaleb, B. R. Elemary, E. A. Ebeid, and A. A. AlHabshy, "An optimized intelligent open-source MLaaS framework for user-friendly clustering and anomaly detection," *J. Supercomput.*, vol. 80, no. 18, pp. 26658–26684, Dec. 2024, doi: 10.1007/s11227-024-06420-2.

[17] S. K. Nanda and N. J. Borah, "Development of Novel Framework for Identifying Anomalies in High Volume of Data Using Robust Machine Learning Algorithm," *SN Comput. Sci.*, vol. 5, no. 5, p. 500, Apr. 2024, doi: 10.1007/s42979-024-02681-z.

[18] B. Zhao, X. Zhou, and Z. Wen, "Bank Customer Profile Based on Classification Algorithm," in Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence, Jan. 2024, pp. 366–371. doi: 10.1145/3675417.3675478.

[19] J. Lwin, "Enhancing Cloud Task Scheduling with Multi-Objective Optimization Using K-Means Clustering and Dynamic Resource Allocation," *J. Comput. Theor. Appl.*, vol. 2, no. 2, pp. 202–211, Oct. 2024, doi: 10.62411/jcta.11337.

[20] S. Sharma, J. Tandukar, and R. Bista, "Generating Harmonious Colors through the Combination of n-Grams and K-means," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 140–150, Dec. 2023, doi: 10.33633/jcta.v1i2.9470.

[21] A. A. Bushra and G. Yi, "Comparative Analysis Review of Pioneering DBSCAN and Successive Density-Based Clustering Algorithms," *IEEE Access*, vol. 9, pp. 87918–87935, 2021, doi: 10.1109/ACCESS.2021.3089036.

[22] W.-R. Chen, Y.-H. Yun, M. Wen, H.-M. Lu, Z.-M. Zhang, and Y.-Z. Liang, "Representative subset selection and outlier detection via isolation forest," *Anal. Methods*, vol. 8, no. 39, pp. 7225–7231, 2016, doi: 10.1039/C6AY01574C.

[23] D. Kim, J. Park, H. C. Chung, and S. Jeong, "Unsupervised Outlier Detection using Random Subspace and Subsampling Ensembles of Dirichlet Process Mixtures," *arXiv*. Jan. 01, 2024. [Online]. Available: http://arxiv.org/abs/2401.00773

[24] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier Detection with Globally Optimal Exemplar-Based GMM," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, Apr. 2009, pp. 145–154. doi: 10.1137/1.9781611972795.13.

[25] M. B. Teferi and L. A. Akinyemi, "Deep Learning-Based Cross-Cancer Morphological Analysis: Identifying Histopathological Patterns in Breast and Lung Cancer," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 3, pp. 235–248, Oct. 2024, doi: 10.62411/faith.3048-3719-36.

[26] A. Alagic *et al.*, "Machine Learning for an Enhanced Credit Risk Analysis: A Comparative Study of Loan Approval Prediction Models Integrating Mental Health Data," *Mach. Learn. Knowl. Extr.*, vol. 6, no. 1, pp. 53–77, Jan. 2024, doi: 10.3390/make6010004.

[27] D. R. I. M. Setiadi, K. Nugroho, A. R. Muslikh, S. W. Iriananda, and A. A. Ojugo, "Integrating SMOTE-Tomek and Fusion Learning with XGBoost Meta-Learner for Robust Diabetes Recognition," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 23–38, May 2024, doi: 10.62411/faith.2024-11.

[28] J. Martin, S. Taheri, and M. Abdollahian, "Optimizing Ensemble Learning to Reduce Misclassification Costs in Credit Risk Scorecards," *Mathematics*, vol. 12, no. 6, p. 855, Mar. 2024, doi: 10.3390/math12060855.

[29] A. Bhaskar *et al.*, "Automatic credit card approval prediction system," in *2nd International Conference on Computing and Communication Networks, ICCCN 2022*, 2024, p. 050007. doi: 10.1063/5.0184623.

[30] F. O. Aghware *et al.*, "Enhancing the Random Forest Model via Synthetic Minority Oversampling Technique for Credit-Card Fraud Detection," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 407–420, Mar. 2024, doi: 10.62411/jcta.10323.

[31] M. D. Okpor *et al.*, "Pilot Study on Enhanced Detection of Cues over Malicious Sites Using Data Balancing on the Random Forest Ensemble," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 109–123, Sep. 2024, doi: 10.62411/faith.2024-14.

[32] D. R. I. M. Setiadi, D. Marutho, and N. A. Setiyanto, "Comprehensive Exploration of Machine and Deep Learning Classification Methods for Aspect-Based Sentiment Analysis with Latent Dirichlet Allocation Topic Modeling," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 12–22, May 2024, doi: 10.62411/faith.2024-3.

[33] D. R. I. M. Setiadi, H. M. M. Islam, G. A. Trisnapradika, and W. Herowati, "Analyzing Preprocessing Impact on Machine Learning Classifiers for Cryotherapy and Immunotherapy Dataset," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 39–50, Jun. 2024, doi: 10.62411/faith.2024-2.

[34] K. Babu, S. Prabhakaran, P. Marikkannu, M. S. Roobini, P. Rai, and A. Pratap Singh, "Smart Credit Card Approval Prediction System using Machine Learning," *E3S Web Conf.*, vol. 540, p. 13001, Jun. 2024, doi: 10.1051/e3sconf/202454013001.

[35] M. A. Sheikh, A. K. Goel, and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Jul. 2020, no. Icesc, pp. 490–494. doi: 10.1109/ICESC48915.2020.9155614.

[36] K. K. Karthik and D. B. David, "A novel approach for enhancing the performance accuracy of loan prediction by comparing Naive Bayes with Decision Tree algorithm," in *International Conference on Advanced Communication Computing and Material Sciences, ICACCMS 2022*, 2024, p. 050030. doi: 10.1063/5.0228264.

[37] F. S. Gomiasti, W. Warto, E. Kartikadarma, J. Gondohanindijo, and D. R. I. M. Setiadi, "Enhancing Lung Cancer Classification Effectiveness Through Hyperparameter-Tuned Support Vector Machine," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 396–406, Mar. 2024, doi: 10.62411/jcta.10106.

[38] D. R. I. M. Setiadi, S. Widiono, A. N. Safriandono, and S. Budi, "Phishing Website Detection Using Bidirectional Gated Recurrent Unit Model and Feature Selection," *J. Futur. Artif. Intell. Technol.*, vol. 2, no. 1, pp. 75–83, 2024, doi: 10.62411/faith.2024-15.

[39] A. Imtiaz, N. Pathirana, S. Saheel, K. Karunanayaka, and C. Trenado, "A Review on the Influence of Deep Learning and Generative AI in the Fashion Industry," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 3, pp. 201–216, Oct. 2024, doi: 10.62411/faith.3048-3719-29.

[40] A. Pathirana, D. K. Rajakaruna, D. Kasthurirathna, A. Atukorale, R. Aththidiye, and M. Yatiipansalawa, "A Reinforcement Learning-Based Approach for Promoting Mental Health Using Multimodal Emotion Recognition," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 124–141, 2024, doi: 10.62411/faith.2024-22.

[41] C.-H. Weng and C.-K. Huang, "A Hybrid Machine Learning Model for Credit Approval," *Appl. Artif. Intell.*, vol. 35, no. 15, pp. 1439–1465, Dec. 2021, doi: 10.1080/08839514.2021.1982475.

[42] P. H. Prastyo, S. E. Prasetyo, and S. Arti, "A Machine Learning Framework for Improving Classification Performance on Credit Approval," *IJID (International J. Informatics Dev.*, vol. 10, no. 1, pp. 47–52, Jun. 2021, doi: 10.14421/ijid.2021.2384.

[43] A. S. Kadam, S. R. Nikam, A. A. Aher, G. V Shelke, and A. S. Chandgude, "Prediction for Loan Approval Using Machine Learning Algorithm," *Int. Res. J. Eng. Technol.*, vol. 8, no. 4, pp. 4089–4092, 2021.

[44] V. Viswanatha, A. C. Ramachandra, K. N. Vishwas, and G. Adithya, "Prediction of Loan Approval in Banks using Machine Learning Approach," *Int. J. Eng. Manag. Res.*, vol. 13, no. 4, pp. 7–19, 2023, doi: 10.31033/ijemr.13.4.2.

[45] Y. Diwate, P. Rana, and P. Chavan, "Loan Approval Prediction Using Machine Learning," *Int. Res. J. Eng. Technol.*, vol. 8, no. 5, pp. 1741–1745, 2021.

[46] S. Fanijo, "AI4CRC: A Deep Learning Approach Towards Preventing Colorectal Cancer," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 143–159, Sep. 2024, doi: 10.62411/faith.2024-28.

[47] J. R. Quinlan, "Credit Approval - UCI Machine Learning Repository," *UCI Machine Learning Repository*, 1987. https://archive.ics.uci.edu/dataset/27/credit+approval

[48] M. G. Kibria and M. Sevkli, "Application of Deep Learning for Credit Card Approval: A Comparison with Two Machine Learning Techniques," *Int. J. Mach. Learn. Comput.*, vol. 11, no. 4, pp. 286–290, Aug. 2021, doi: 10.18178/ijmlc.2021.11.4.1049.