# Bio-Inspired Optimized Prediction of Invoice Payment Delays in MSMEs

Akash Raghuvanshi[1], Harsh Upadhyay[2], Arpit[3]
{akash.raghuvanshi@gmail.com, harshupadhyay323@gmail.com, arpit.22scse1010963@galgotiasuniversity.edu.in}
School of Computer Science & Engineering, Galgotias University, Greater Noida, India[1]
School of Computer Science & Engineering, Galgotias University, Greater Noida, India[2]
School of Computer Science & Engineering, Galgotias University, Greater Noida, India[3]

**Abstract.** Invoice payment delays are a major challenge in small and medium-sized enterprises (SMEs). In this paper, a novel intelligent and optimized predictive framework is proposed by combining CatBoost with the Grasshopper Optimization Algorithm (GHO). The solution offers real-time guidance by considering historical invoice data, client likelihood to pay, and custom-engineered features such as payment consistency, macroeconomic trends, and seasonality. Experimental results on actual SME invoice datasets show that the metaheuristic-optimized ensemble model achieves 93% AUC and outperforms other models. The proposed model effectively improves convergence rate and prediction accuracy.

**Keywords:** Invoice Payment Delay, Artificial Intelligence, CatBoost, Accuracy, Grasshopper Optimization Algorithm (GOA), Corrugated Box Manufacturing, SME Finance.

## 1 Introduction

Small and medium-sized enterprises (SMEs) play a major role in the manufacturing sector, including the corrugated box industry that supports packaging needs for areas like FMCG, textiles, pharmaceuticals, and e-commerce. One issue that almost every SME in this sector struggles with is delayed

IICTDS 2025, December 19-20, Chandigarh, India
Copyright © 2026 EAI
DOI 10.4108/eai.19-12-2025.2362547

invoice payments. Even a small delay can affect their cash flow, make it harder to buy raw materials on time, and create day-to-day operational difficulties. These issues have increased in recent years because customers no longer follow steady payment habits, and most SMEs still don't have any proper data tools to keep track of such risks.

In many cases, SME owners rely on their own experience or simple rule-based checks before giving credit or deciding payment terms. This approach was fine earlier, but the market today is very different. Prices of raw materials change often, demand goes up and down depending on the season, and customer behaviour is not always consistent. Because of these factors, manual judgement can easily lead to mistakes. Although machine learning has been used widely in financial risk prediction, very little work has focused specifically on predicting invoice payment delays for manufacturing SMEs. Many of the existing models are also difficult for business owners to interpret or trust.

**Research Gap:** Invoice management has become a challenging in developing predictive risk analysis framework despite its advantages. Mostly issues such as mixed data formats, lack of established risk assessment criteria, and limited integration across enterprise systems that slow down the construction of accurate and scalable risk predictive models. In recent years, there has been a significant progression in the use of machine learning resulting in a significant increase in the amount of data typically available to organizations. Therefore, research in this area is important for developing a framework for predicting the risk accurately also focus on integrating financial metrics, payment behaviour tracking, and invoice-level data. Implementing such risk predictive framework can improve the financial control and enhance business decisions. Based on prior research, the majority of studies tend to overlook the common billing delays faced by SMEs in favor of analyzing credit ratings or loan defaults. The fact that many models prioritize increasing their accuracy over ensuring their output is easy to understand and implement further adds complexity.

**Contribution of This Work:** It is our intention to combine CatBoost with GHO in this study. The idea was to design a model that works accurately and still offers clear reasons behind its predictions. This method is not only theoretical; it was built with practical SME needs in mind, so it can be applied to day-to-day financial operations

## 2 Literature Review

### 2.1 Credit Risk Assessment and Prediction

The possibility that a borrower will not repay a loan, putting the company out of pocket, is called credit risk [1]. The majority of the prior research in this field used very simplistic statistical approaches. In their classical research, Altman, Fisher, Wiginton, and Crook [2, 3, 4, 5] examined financial ratios and borrower behavior history to detect probable defaulters. They relied extensively on approaches like Logistic Regression and Linear Discriminant Analysis (LDA). Back when market conditions and consumer tastes were more predictable, those tactics made a lot of sense.

Many changes, though, have occurred within the last many decades. Numerous variables, such as economic climate, seasonal demands, and market trends, can impact consumer behavior in modern

SMEs, and these elements can cause changes to occur quickly. Thus, traditional linear methods may not work in practice since they presuppose that all interactions are static and predictable. Many SMEs have realized that they can not depend solely on their financial records. The behavior of customers is becoming increasingly unpredictable, and new risks can emerge in ways that older models failed to predict. More and more, businesses are turning to data-driven approaches for credit risk analysis and forecasting.

In an effort to further improve dependability, some researchers have investigated clustering algorithms in conjunction with prediction models. There has been a lot more focus on loan defaults than on invoice payment delays. Anticipating potential payment delays is another challenge. Invoice delay forecasting techniques in the past were more involved and difficult to measure due to these real-world issues.

### 2.2 Invoice Management Challenges in SMEs

Traditional machine-learning approaches have been widely applied to credit-risk assessment, invoice payment prediction, and default-risk modelling. Early studies on invoice or debtor behaviour predominantly relied on classical classifiers such as Logistic Regression (LR), Support Vector Machines (SVM), Decision Trees (DT), Random Forests (RF), Naïve Bayes (NB), and K-Nearest Neighbours (KNN) to identify late payments or defaulting customers. These models became popular due to their interpretability, low computational cost, and ability to work effectively with structured financial attributes such as transaction history, payment terms, credit scores, and financial ratios.

Zeng et al. [6] used various supervised machine-learning techniques (DT, NB, LR, Boosted Stumps, and rule-based learners) to improve the invoice-to-cash process. This study formulated the invoice payment outcome prediction as a multi-class classification task, the classes ranging from "on time" to "90+ days late." They have used real-world invoice data from four large firms to use invoice-level, historical, and customer-level features to capture payment behaviour. Their results demonstrate that incorporating historical behavioural features significantly increases prediction accuracy. The authors further examine cost-sensitive learning to detect high-risk (90+ days) invoices better.

Hu [7] evaluated traditional methods such as LR, SVM, DT, RF, and AdaBoost on a real dataset of more than 1,00,000 invoices from a Fortune 500 corporation. By engineering customer-level behavioural features—including past delay ratio, average delay days, payment-term patterns, and historical invoice volumes—the work demonstrates that supervised models, particularly RF, can achieve high predictive accuracy for both binary (on-time vs delayed) and multi-class delay predictions. The thesis also highlights the importance of addressing class imbalance and shows that customer behaviour is a stronger predictor of payment outcomes than invoice-level attributes alone. Although there has been some success with artificial intelligence and deep learning models for identifying intricate patterns in financial data, the majority of these systems were developed for bigger banks, rather than small firms [8]. For that reason, they may not be the best choice in environments teeming with SMEs. In addition, most AI models just display a numerical value without providing any context for it. SME owners usually want to understand the reason behind a delay prediction

instead of just getting a score [9]. If they cannot see the logic behind the prediction, it becomes hard for them to trust the tool or use it confidently in their daily decisions.

### 2.3 Research Gap and Contributions

Although plenty of studies have explored credit risk prediction in general, only a small amount of research looks specifically at invoice payment delays in SMEs. Many of the existing methods focus mainly on improving accuracy and often overlook things like how easy the model is to understand, how practical it is to use, or the fact that SME data is usually limited and sometimes inconsistent.

- This study aims to fill this gap by using a combined approach that brings together CatBoost and the Grasshopper Optimization Algorithm (GHO). The main contributions of this work are:
  - Building a model that is designed specifically to predict invoice payment delays for manufacturing SMEs.
  - Using a nature-inspired method (GHO) to help tune the model's parameters in a more reliable and efficient way.

## 3 Meta-Heuristic Machine Learning Model

In recent years, machine learning and different types of optimization methods have become more common, and because of this, people have started using them for problems that were earlier handled with very simple statistical approaches. Client payment habits can be highly unpredictable, throwing SMEs off track with unexpected delays. Cash flow management becomes challenging. Older linear models struggle with real-world data messiness, assuming neat patterns that rarely exist. Modern models like XGBoost[10], Random Forest, and CatBoost[11] are built to handle that complexity better. Grasshopper Optimization Algorithm (GHO) [12] is a great example of this. The Grasshopper Optimization Algorithm (GOA) is inspired by how grasshoppers swarm together when they sense food, and it's being used to optimize complex problems. GHO's approach is highly effective – it's more adaptive and adaptable, which lets it explore options effectively and find some effective solutions.

Corrugated box companies need payments on time. Late payments are a major challenge for SMEs. difficulty in procuring raw materials and disruptions in production activities, and cash flow is disrupted. They are highly vulnerable to even tiny setbacks because their financial resources are limited. If they could just predict which bills will be late, they'd be better prepared. The proposed framework addresses this challenge the ensemble model with GHO optimization. to support SME financial risk management.

### 3.1 Data Preprocessing

It is necessary to clean the dataset before training any models. Our first order of business was to search for duplicate or missing records, and we were lucky not to find any. A number of numerical attributes, such as the invoice amount and the number of days of delay, displayed notably increased values. The dataset had categorical variables like company name and location, which needed some preprocessing. We used label encoding to prep them for CatBoost, while XGBoost and Random Forest needed a bit more work. We also made the numerical features consistent to keep things uniform.

### 3.2 Optimization Approach

By optimizing CatBoost's hyperparameters with the Grasshopper Optimization Algorithm (GHO), we were able to boost its performance. The cooperative behavior of grasshoppers is the basis for GHO, which means that it does away with the need to manually test each parameter or do a tiresome grid search. Despite first scattering, they swiftly regroup to probe a possible danger. Similarly, mathematics takes into account multiple choices while deciding on the optimal course of action. The swarm's attitude will shift at each stage depending on how much it likes or dislikes the final product.

In our case, GHO was used to adjust four main settings of the CatBoost model:

- learning rate,
- tree depth,
- number of estimators,
- L2 regularization.

The aim was to reduce the validation loss in cross-validation. Once GHO found the best combination, we used those parameter values to train the final CatBoost model.

### 3.3 CatBoost Algorithm

In 2018, the boosting model CatBoost was developed by Prokhorenkova and others [11]. It serves datasets well when the variables include both numerical and category values. When it comes to processing data using categories, CatBoost truly excels. It employs a technique known as ordered target statistics rather than conventional encoding techniques like label encoding. Remember that the model is limited to using data from unseen rows, thus it will not be able to find the target value. This minor adjustment greatly improves the accuracy of predictions and aids in the prevention of leaks.

Among CatBoost's numerous useful characteristics is the ability to build symmetric or "oblivious" trees. At any given depth in these trees, the rule applies uniformly to all branches. Despite the seeming limitation, this really helps the model learn and guarantees more consistent behavior. When working with structured, tabular data—the kind that most SMEs encounter—the format usually performs effectively for structured SME datasets.

As follows is the definition of the goal function of CatBoost:

$$L = \sum_{i=1}^{N} \ell\big(y_i, F_{m-1}(x_i) + \eta h_m(x_i)\big) + \Omega(h_m) \tag{1}$$

**Symbols in Equation (1) — CatBoost Objective**

- $L$ — overall loss that the model tries to minimise.
- $y_i$ — actual target value for the $i$-th training example.
- $F_{m-1}(x_i)$ — model prediction before adding the $m$-th tree.
- $h_m(x_i)$ — output of the new weak learner at iteration $m$.
- $\eta$ — learning rate controlling how strongly the new tree influences the prediction.
- $\ell(\cdot)$ — loss function that measures prediction error.
- $\Omega(h_m)$ — regularisation term to prevent overfitting.

Equation (1) shows how CatBoost updates the model at each step by adding a new tree and adjusting the prediction using the learning rate and regularization.

**Input:** Training data $(X_{\texttt{train}}, y_{\texttt{train}})$, categorical features *catFeatures*, hyperparameters (iterations, depth, learning_rate)

**Output:** Trained CatBoost model

**Step 1: Encode Categorical Features**

Use ordered target statistics to convert categorical variables into numerical form.

**Step 2: Initialise Predictions**

Set initial prediction as the mean target value: $\hat{y}_i^{(0)} = \texttt{mean}(y_{\texttt{train}})$

**for** $t = 1$ **to** *iterations* **do**

Compute gradients using the ordered boosting scheme;

Build an oblivious (symmetric) decision tree with depth $d = \texttt{depth}$;

Scale tree predictions using the learning rate $\eta$;

Update predictions: $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot h_t(x_i)$;

Evaluate loss on validation set;

**end**

**return** *CatBoost model with the best validation performance*

**Algorithm 1:** CatBoost Training Procedure

### 3.4 Grasshopper Optimization Algorithm (GHO)

A metaheuristic approach that attempts to simulate the cooperative behavior of grasshoppers, the Grasshopper Optimization Algorithm (GHO) was presented by Saremi et al. (2017). The algorithm's goal is to discover a happy medium between exhaustively testing every conceivable answer and progressively honing the ones that show promise. It redirects the swarm's attention from a wide to a narrow search by employing a decreasing comfort coefficient. Every grasshopper's location is monitored continuously by this system:

$$\mathbf{X}_i = c\left(\sum_{j=1, j\neq i}^{N} s(d_{ij})\hat{d}_{ij}\right) + \mathbf{g} \tag{2}$$

**Meaning of the Terms in Equation 2:**

- $\mathbf{X}i$ —Here is the most recent location of grasshopper $i$.
- $dij$ —distance between grasshoppers I and j has been calculated $j$.
- $s(d_{ij})$ — evaluates the influence's attractiveness or repulsiveness using an interaction function.
- $\hat{d}_{ij}$ — direction, represented as a unit vector, from $i$ to $j$

- **g** — the most effective solution that the swarm has arrived at thus far.
- $c$ — a scaling factor (comfort coefficient) that, as the number of repeats increases, subtly decreases exploration

Looking at Eq. (2), the movement of each grasshopper depends on two things: the push or pull from its neighbours and the pull toward the current best position.
The comfort coefficient decreases linearly:

$$c = c_{\max} - \frac{t}{T_{\max}}(c_{\max} - c_{\min}) \tag{3}$$

**Symbols in Equation (3) — Comfort Coefficient**

- $c_{\max}$ — initial comfort coefficient, used at the beginning of the search.
- $c_{\min}$ — final comfort coefficient, used when the search becomes more focused.
- $t$ — current iteration number in the optimization process.
- $T_{\max}$ — total number of iterations allowed for the algorithm.

Equation (3) shows how the comfort level gradually decreases during the run so that the search shifts from broad exploration to more focused local refinement.

**Table 1:** GHO Parameters Used in This Study

| Parameter | Description | Actual Value / Range Used |
|---|---|---|
| $N$ | Number of grasshoppers in the population | 30 |
| $c_{\max}$ | Starting comfort coefficient | 1.0 |
| $c_{\min}$ | Ending comfort coefficient | 0.00004 |
| $T_{\max}$ | Maximum number of iterations | 50 |
| $s(r)$ | Social interaction function | $fe^{-r/l}e^{-r}$ |
| $lb, ub$ | Search ranges for tuned parameters | Iterations: (50, 200)<br>Learning rate: (0.01, 0.3)<br>Tree depth: (3, 10) |

These settings were the exact values used during our experiments (1). A population size of 30 grasshoppers and a limit of 50 iterations gave a good balance between search quality and running time. The ranges for the learning rate, tree depth, and number of boosting rounds were selected after a few trial experiments and by referring to values that are usually effective in gradient boosting models. This gave the algorithm enough room to search for good parameters while still keeping the tuning process manageable for our dataset.

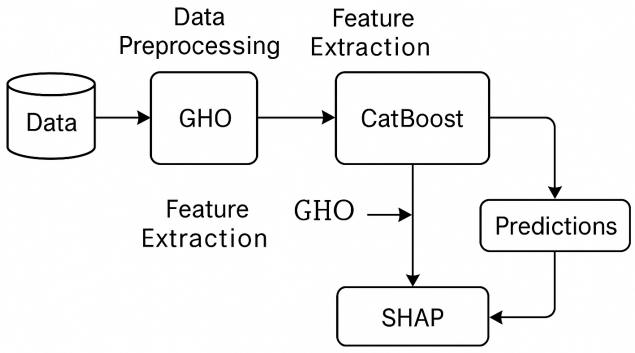



**Fig. 1.** Block Diagram of the Grasshopper Optimization Algorithm (GHO)

*Application in This Study:*

In this study, GHO was used to tune several important CatBoost parameters. The parameters included the

- learning rate,
- the depth of the trees,
- the number of boosting rounds,
- $L_2$ regularization.

Properly setting these parameters is critical as they affect the model's learning capability. The model was able to select more optimal parameters by using GHO for tuning and avoiding early search stagnation. Overall, it made training easier and, in the end, the forecasts were better. By tweaking these settings, the CatBoost-GHO model consistently generated more comprehensible results, facilitating the identification of invoices posing a high risk of delay in SME production setups.

## 4 Results and Discussion

### 4.1 Dataset Description

Invoices from a number of corrugated companies totaling slightly more than**1,00,000 invoices** were included in the dataset that was utilized. A varied spectrum of clients receive packing materials from

the ten frequent customers whose invoices are included in the dataset. The customers belong to a variety of industries, including textiles, automotive, electronics, FMCG, retail, e-commerce, food and beverages, and pharmaceuticals.

In total, the dataset includes 17 features. These cover financial details, customer behaviour, and a few economic indicators. Some of the commonly used fields are invoice amount, payment terms, order history, previous delays, consistency of payments, GST compliance, kraft paper price movement, CPI growth, unemployment rate, and interest rate. One convenient thing about this dataset is that it had no missing values, which made the initial cleaning process much easier.

Because the records do not include specific dates, the dataset is best viewed as a general snapshot of recent business activity rather than a continuous time-based series. The target label was created from the payment delay days column. Any invoice with a delay greater than zero was marked as a late payment. In the final dataset, about **57.20%** of the invoices were paid on time, while **42.79%** were late. Among the delayed invoices, the average delay was roughly **14.50 days**, and in extreme cases it went up to around **40.17 days**.

An equally distributed dataset is also present in other places. All six major cities (Chennai, Bangalore, Pune, Delhi, Mumbai, and Ahmedabad) contributed between 16,000 and 17,000 invoices to the fairly evenly dispersed dataset.

**Table 2:** Dataset Summary and Characteristics

| **Characteristic** | **Value** |
|---|---|
| Total Records | 1,00,000 |
| Total Features | 17 |
| Number of SMEs | 10 |
| Missing Values | 0 (0%) |
| Non-Delayed Invoices | 57,205 (57.20%) |
| Delayed Invoices | 42,795 (42.79%) |
| Mean Payment Delay | 14.50 days |
| Maximum Payment Delay | 40.17 days |
| Industries Covered | 8 |
| Regions Covered | 6 |

### 4.2 Experimental Setup

Python 3.10 ran on a machine with an Intel Core i9 CPU and 32 GB of RAM for all the trials in this study.We put many models through their paces, including LSTM, SAINT, Random Forest, XGBoost, CatBoost, and versions of each.Prior to training any of them, the dataset was cleaned according to industry standards. This included removing outliers, normalizing the numerical values, and converting the categorical columns into formats that the models could use. We fine-tuned CatBoost with 5-fold cross-validation and GHO to dial in the accuracy. It helped us lock in consistent performance across all data splits while tweaking the parameters.

### 4.3 Feature Importance Analysis

SHAP values from the CatBoost-GHO model showed us what drives payment delays, where behavioural traits were identified as major contributing factors. The model's decision-making was heavily influenced by factors related to the customer's payment history, payment frequency, and other relationship-based variables. Taken together, these aspects of behavior were these behavioural features contributed more than 48% of the overall feature importance. Some other macroeconomic reasons were also significant, as indicated before. Two indicators that reveal how the state of the economy impacts consumer spending are the economic risk index and the rate of change in the consumer price index.Given the cyclical nature of both demand and raw material supply, it should come as no surprise that we also observed seasonal changes. Following these modifications, the model's predictions seemed more credible because they incorporated this additional context.

### 4.4 Performance Metrics

We generated confusion matrices for three versions of the model: the basic CatBoost setup, the CatBoost model tuned using GWO, and the CatBoost model tuned using GHO. A side-by-side view of these results is presented in Table 3. .

**Table 3:** Comparison of Confusion Matrices for CatBoost Variants

| CatBoost Baseline | Pred 0 | Pred 1 |
|---|---|---|
| **True 0** | 9703 | 1738 |
| **True 1** | 1352 | 7207 |

*(a) CatBoost Baseline*

| CatBoost + GWO | Pred 0 | Pred 1 |
|---|---|---|
| **True 0** | 9669 | 1772 |
| **True 1** | 1409 | 7150 |

*(b) CatBoost + GWO*

| CatBoost + GHO (Proposed) | Pred 0 | Pred 1 |
|---|---|---|
| **True 0** | 9710 | 1731 |
| **True 1** | 1352 | 7207 |

*(c) CatBoost + GHO*

*Confusion Matrix Interpretation.*

The basic CatBoost model managed to catch most of the delayed invoices, but it also triggered a lot of false alerts, marking 1738 on-time invoices as delayed. The CatBoost + GWO version did not help much and actually made slightly more errors on both sides. The CatBoost + GHO model gave the most balanced output. It correctly picked up 9710 invoices that were not delayed and had the lowest number of false positives, with 1731 cases. Overall, these results show that using GHO helped the model settle on a better set of hyperparameters. .

### 4.5 Comparative Analysis of Models

Table 4 gives an overview of how all the models performed. The CatBoost-GHO model topped the charts with 0.8458 accuracy, 0.8238 F1-score, and a **93%** AUC. Figures 2–5 this show how the different models perform in accuracy, precision, recall, and F1-score, and we can see that they show the improvement trend.

**Table 4:** Model Performance Summary

| Model | Accuracy | Precision | Recall | F1 | AUC |
|---|---|---|---|---|---|
| XGBoost Baseline | 0.8427 | 0.8037 | 0.8369 | 0.8200 | 0.9265 |
| Random Forest Baseline | 0.8368 | 0.8081 | 0.8114 | 0.8098 | 0.9178 |
| CatBoost Baseline | 0.8455 | 0.8057 | 0.8420 | 0.8235 | 0.9274 |
| XGBoost + GWO | 0.8442 | 0.8056 | 0.8382 | 0.8216 | 0.9268 |
| Random Forest + GWO | 0.8075 | 0.7573 | 0.8099 | 0.7827 | 0.8961 |
| CatBoost + GWO | 0.8412 | 0.8030 | 0.8352 | 0.8189 | 0.9236 |
| XGBoost + GHO | 0.8448 | 0.8054 | 0.8402 | 0.8224 | 0.9273 |
| Random Forest + GHO | 0.8367 | 0.8060 | 0.8143 | 0.8101 | 0.9183 |
| LSTM | 0.8399 | 0.7900 | 0.8526 | 0.8201 | 0.9245 |
| SAINT | 0.8410 | 0.8079 | 0.8246 | 0.8162 | 0.9238 |
| **CatBoost + GHO (Proposed)** | **0.8458** | **0.8063** | **0.8420** | **0.8238** | **0.9300** |

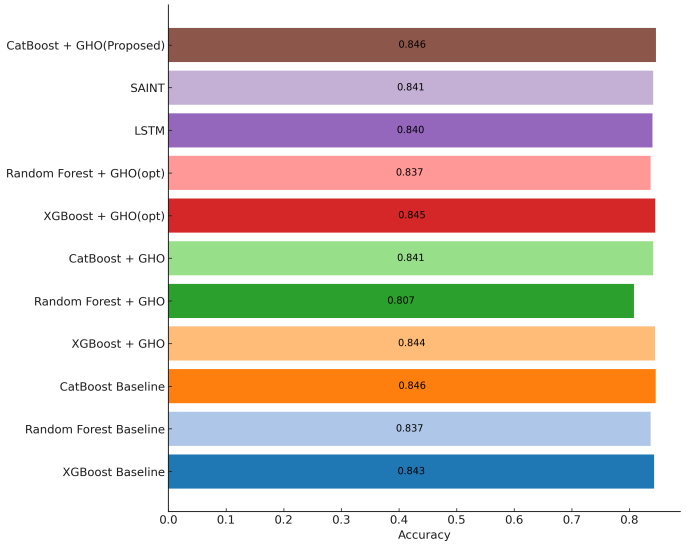



**Fig. 2.** The CatBoost-GHO approach is used for predicting invoice payment delays.

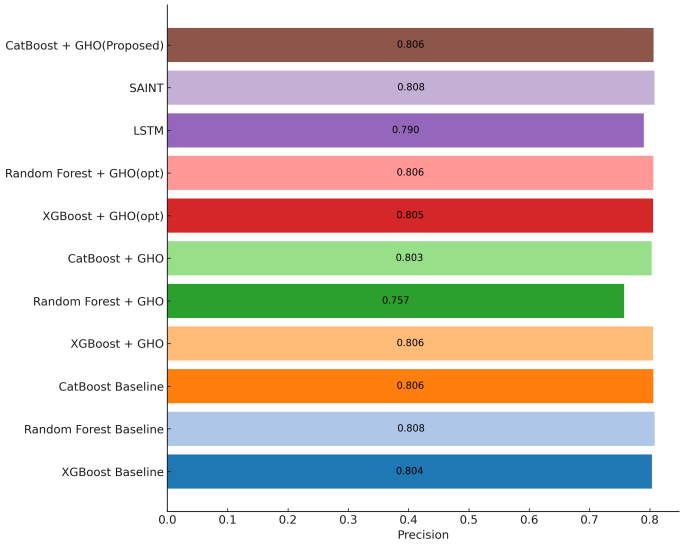



**Fig. 3.** Precision Comparison Across Models

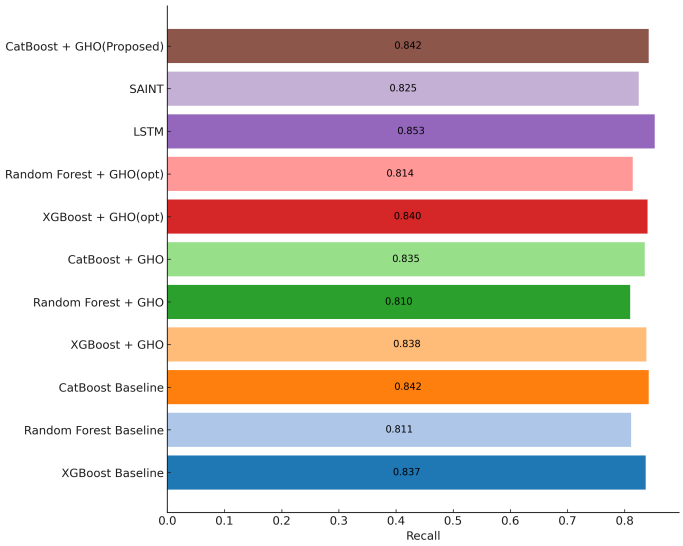



**Fig. 4.** Recall Comparison Across Models

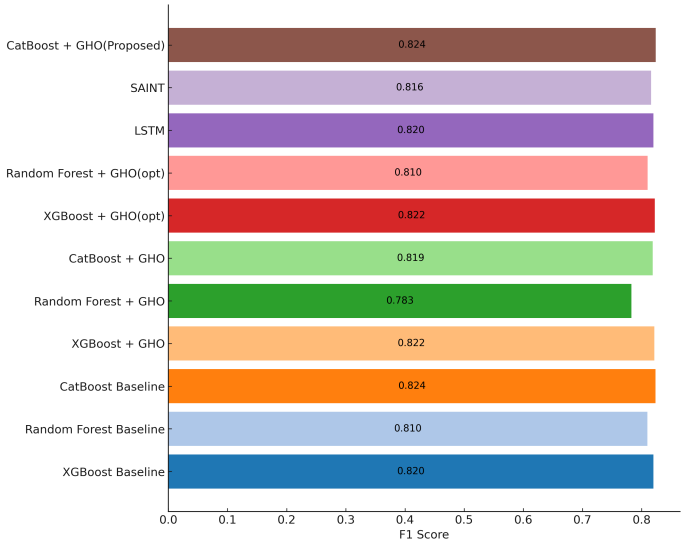



**Fig. 5.** F1-Score Comparison Across Models

### 4.6 Model Interpretability

SHAP values helped us understand what was driving the model's predictions. The analysis showed that a customer's past payment behavior was the biggest factor influencing the model's predictions. Things like average delay, payment consistency, and the length of their relationship with the supplier had the most influence on the predictions. Along with these, transaction-level features, general economic indicators, and seasonal patterns also played a role in deciding whether an invoice might be delayed.

It is also worth mentioning that some of the transaction-related features in the dataset, like *invoice amount*, *amount ratio*, and *order frequency*, tend to be linked to each other. This is expected because they each describe a different part of the customer's overall buying behaviour. In the presence of such multicollinearity, SHAP tends to distribute importance across related variables rather than assigning it to a single feature. This does not affect the overall interpretation of the model; instead, it indicates that these features jointly contribute to the prediction of payment delays. Therefore, the SHAP results should be viewed in terms of groups of related behaviours rather than standalone features.

Table 5 lists the most influential features.

**Table 5:** Top 10 Features by Relative Importance in CatBoost–GHO Model

| Rank | Feature | Importance (%) | Category |
|---|---|---|---|
| 1 | Historical average delay | 19.8 | Customer History |
| 2 | Payment consistency | 16.5 | Customer History |
| 3 | Invoice amount (log) | 14.2 | Transaction |
| 4 | Relationship score | 12.3 | Customer Tenure |
| 5 | Amount ratio | 8.7 | Transaction |
| 6 | Order frequency | 7.4 | Customer History |
| 7 | Economic risk composite | 6.2 | Macro |
| 8 | Previous orders count | 5.1 | Customer History |
| 9 | CPI growth rate | 4.8 | Macro |
| 10 | Seasonal factor | 5.0 | Seasonal |

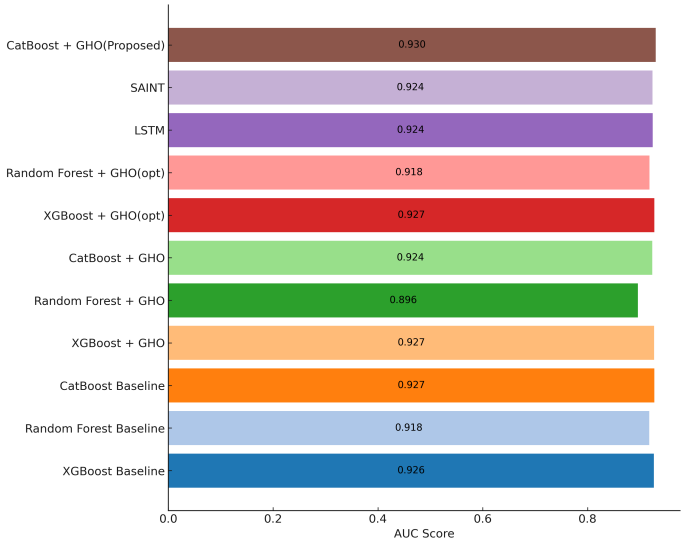



**Fig. 6.** AUC Comparison of Evaluated Models

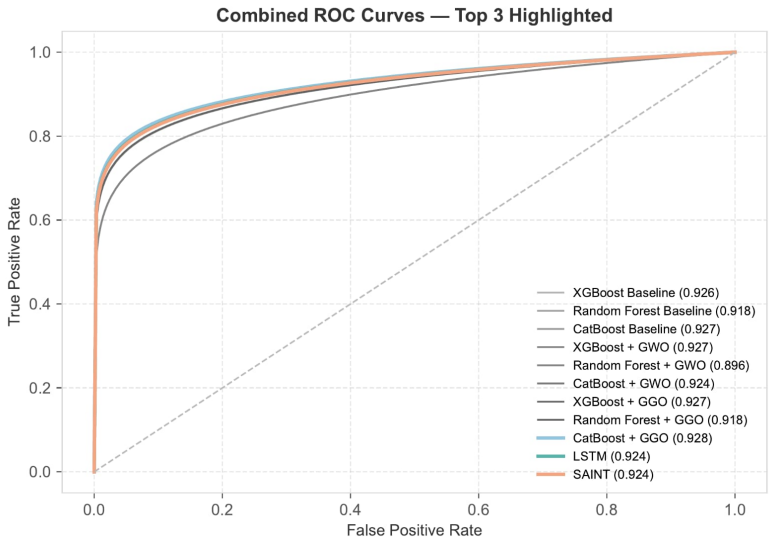



**Fig. 7.** ROC Curve Comparison of CatBoost Variants

### 4.7 Practical Implementation for SMEs

The Proposed model can be used quite easily in a real SME environment. In a normal workflow, the finance or accounts team enters the invoice details into their billing or ERP system. These details include the customer name, invoice amount, payment terms, and the customer's previous order behaviour. The trained model then gives a simple output, such as a delay risk score or a "high/medium/low" delay warning for that invoice.

SME managers can use this information to decide which customers may need earlier reminders, stricter credit periods, or follow-up calls before releasing new goods. The SHAP results also help them understand why a particular customer is at risk, for example due to long past delays or inconsistent payment habits. This makes the model's suggestions easier to trust and act upon.

The system can be set up in different ways depending on the SME's resources. It can run on a small local computer in the office, be added as a feature inside the existing billing software, or be accessed through a simple web page. Since the model is light after training, it does not need high-end hardware. Ultimately, we want to provide SMEs a tool that makes them feel more confident when managing their cash flow and making loan decisions.

## 5 Conclusion

In order to predict when MSME manufacturers will get payment for their bills, this study provided a model that combines CatBoost and Grasshopper Optimization (GHO). Out of all the methods we tested, the CatBoost-GHO model proved to be the most effective. In terms of robustness and stability, this model outperformed all other models.

Based on the area under the curve (AUC) of 0.9300, the model achieved better results than regular CatBoost (0.9274), XGBoost (0.9265), and Random Forest (0.9178). Its accuracy was 0.8458 and F1-score was 0.8238, both of which were excellent. On the other hand, CatBoost and the GWO-modified Random Forest showed comparatively lower performance.

These results are further bolstered by confusion analysis. While the number of true positives remained at 7207, the number of false positives decreased from 1738 in the original CatBoost model to 1731 in the CatBoost-GHO model. Thus, the model was able to use fewer spurious delay predictions while retaining all of the accurate ones, thanks to the GHO modification.

In a search for bills with a high probability of late payment, the CatBoost-GHO model emerged as the most viable and trustworthy choice. Small and medium-sized enterprises (SMEs) can use it to better predict their cash flow, identify clients who may require more frequent follow-up, and change their loan limits as appropriate.As for what is to come, we are planning to make it more user-friendly for SME owners, integrate it with payment data in real-time, or bring the idea to other industries.

## References


[1] Hand P, Adams N. Defining and Measuring Credit Risk: The Evolving Role of Data Analytics. Journal of Banking and Finance. 2014;49:221-31.

[2] Altman EI. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. The Journal of Finance. 1968;23(4):589-609.

[3] Fisher RA. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics. 1936;7(2):179-88.

[4] Crook J, Edelman D, Thomas LC. Recent Developments in Consumer Credit Risk Assessment. European Journal of Operational Research. 2007;183(3):1447-65.

[5] Wiginton JC. A Note on the Comparison of Logit and Discriminant Models of Consumer Credit Behavior. Journal of Financial and Quantitative Analysis. 1980;15(3):757-70.

[6] Zeng S, Boier-Martin I, Melville P, Murphy C, Lang CA. Using Predictive Analysis to Improve Invoice-to-Cash Collection. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08). Las Vegas, Nevada, USA: ACM; 2008. p. 492-501.

[7] Hu P. Predicting and Improving Invoice-to-Cash Collection through Machine Learning. Massachusetts Institute of Technology; 2015. Available from: https://dspace.mit.edu/handle/1721.1/98711.

[8] Mai M, Tian S, Lee C, Ma L. Deep Learning Models for Default Risk Analysis: A Comparison. European Journal of Operational Research. 2020;283(1):270-82.

[9] Kvamme J, Sellereite T, Aas K, Sjursen A. Predicting Mortgage Default Using Convolutional Neural Networks. Expert Systems with Applications. 2018;102:207-17.

[10] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016. p. 785-94.

[11] Prokhorenkova A, Gusev G, Vorobev A, Dorogush A, Gulin A. CatBoost: Unbiased Boosting with Categorical Features. In: Advances in Neural Information Processing Systems; 2018. p. 6638-48. Available from: https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

[12] Saremi S, Mirjalili S, Lewis A. Grasshopper Optimization Algorithm: Theory and Application. Advances in Engineering Software. 2017;105:30-47.