

**ADAPTIVE PAYMENT ROUTING IN E-COMMERCE: A
MACHINE LEARNING APPROACH FOR HIGHER
PURCHASE SUCCESS RATES**

Yashasvi Makin¹, Pavan K Gondhi²

¹Sr. Software Engineer, Seattle, USA

Email: yashasvimakin@gmail.com, ORCID: 0009-0005-2926-0118

²Plano, TX

Email: pavankgondhi@gmail.com, ORCID: 0009-0002-5528-2474

Abstract

Online stores need to have a high payment flow success rate to keep their customers and keep their business going. Traditional rule-based routers can't change their behavior in real time when network conditions or issuers change. We suggest a hybrid machine-learning architecture that combines (1) an offline XGBoost classifier to predict the chances of success on each route, (2) an online LinUCB contextual bandit for exploration–exploitation routing, and (3) a rolling-window anomaly detector for quick failover. The system was tested on six months' worth of production transactions, which included almost 5 million records and an 8 percent failure rate. It kept its decision latency under 30 ms and got a 96.3 percent first-attempt approval rate in offline simulations, which is a

3.6 percentage-point improvement over a 92.7 percent rule-based baseline. In a real A/B test with 200,000 transactions, our ML-driven router had a success rate of 96.7% compared to 94.1% for the old system ($p < 0.01$). It automatically found and rerouted around a gateway outage to keep performance up. More tests, such as feature-interaction heatmaps, cost-benefit trade-offs, and calibration curves, show that the results are strong, easy to understand, and cost-effective. We end by going over the main points and talking about where things could go in the future, like advanced reinforcement learning and multi-objective routing that finds the best balance between fees, delays, and the chance of getting approval.

Keywords: Payment Success; Machine Learning; Smart Routing; Multi-Armed Bandits; Payment Gateway; Transaction optimization.

1 Introduction

There are several steps in online payment flows that let people finish transactions or purchases electronically. The *payment flow success rate* is the percentage of payment transactions that were started and ended with a successful and paid for transaction. Businesses need high success rates because failed transactions mean lost potential income, and customers may decide not to buy the item. In fact, almost half of customers won't try to make a payment again after it is rejected. This shows how important it is to make sure payments go through the first time (Dougall, 2025). On the other hand, especially when done on a large scale, even a small rise in the success rate can lead to big rises in income. Merchants and payment service providers

need to focus on making this metric as good as it can be.

There are a lot of things that could go wrong with a payment. Some of these are not having enough money in the customer's account, having trouble connecting to the network, having the gateway or processor go down, having the bank fail to verify the transaction (for example, with OTP or 3-D Secure issues), or having fraud detection systems refuse the transaction. Companies that use rule-based methods should be able to improve their success rates by trying again on failed transactions, giving customers more payment options, or using backup processors to send payments if one method fails. These rule-based heuristics can find known failure modes, but they don't always adapt quickly to new situations or take into account the complicated ways that different transaction features interact with each other to affect the outcome of a payment.

Even though payment processing has gotten better, e-commerce companies still have trouble keeping their payment flow success rates high all the time. Right now, rule-based systems can't react to things that happen in real time, like network delays, sudden gateway outages, and new ways to find fraud. This makes them naturally rigid. These methods also don't make good use of the transaction logs, which have a lot of rich, multi-dimensional data. This information could help adaptive decision-making choose the best gateway.

Machine learning (ML) can help enhance payment flow success by looking for patterns in a lot of prior transaction data. Instead of following tight rules, ML models may make smart decisions in real time and tell you how probable a transaction is to go through in different conditions. For example, models can inform you which payment gateway is most likely to approve a transaction right now or if a transaction is about to fail and has to be handled or routed in a different way. More and more big payment systems are using these methods. For example, Stripe's Adaptive Acceptance system employs machine learning to increase the number of authorization tries, which seems to have resulted to large increases in approval rates. Fintech companies like Checkout.com also think that optimization based on machine learning is a key trend that will help more people accept payments in 2025 (Dougall, 2025).

This study presents considerable potential benefits. For large e-commerce platforms, a mere 1% enhancement in payment success rates might yield millions in additional annual revenue. Enhanced payment processes reduce checkout disruptions and abandoned carts, so diminishing consumer friction and subsequently elevating overall customer satisfaction and loyalty. Furthermore, by reducing the necessity for manual interventions in payment processing, an adaptive machine learning framework can streamline operational procedures. Elevated approval rates enhance profitability while also contributing to long-term firm sustainability and growth in the contemporary competitive digital commerce landscape.

This study examines the application of machine learning to enhance payment flow success rates. We assess the current advancements in academia and industry, encompassing intelligent routing and payment optimization methodologies. Leveraging insights from previous endeavors, we present a coherent machine learning-driven architecture for authorization decision-making and payment routing. Our method integrates supervised learning for success prediction

with an online learning component to accommodate non-stationary success probabilities over time. We subsequently review our methodology and evaluate it utilizing actual transaction data. Our investigations demonstrate that our machine learning-based approach significantly enhances the transaction success rate compared to traditional rule-based systems. We also consider scalability and finance regulatory compliance as critical implementation challenges for deploying such solutions in production. This paper is mostly structured as follows: Section 2 examines related work; Section 3 ML approaches techniques for payment optimization; Section 4 details our suggested approach; Section 5 shows the experimental setup; Section 6 addresses the results; Section 7 finishes the work with last comments and future directions.

2 Related Work

Academic researchers as well as industry professionals have taken notice of the need of raising payment success rates. Early attempts in this field usually concentrated on *payment routing optimization*. Dynamic choice among several payment service providers or gateways is part of payment routing to handle a transaction. Choosing the path with the best chance of success will help one lower failures resulting from problems particular to gateways. PayU's dynamic routing system is one prominent example; developed as a multi-armed bandit problem to maximize success rates (Trivedi & Singh, 2018), the method handled every payment gateway as an arm and learnt over time which gateway, under different circumstances, produced better success probabilities, so improving the overall success rates. Other payment companies also began investigating intelligent routing about the same time. Stochastic routing algorithms that adapt to non-stationary success probabilities are described by Trivedi and Singh (Trivedi & Singh, 2018), so learning to avoid underperforming channels. This set stage for later industry solutions. Following years saw several fintech startups using machine learning to show notable increases in transaction approval rates. For payments, Razorpay created an artificial intelligence-driven *smart router*. Their system was using tree-based ML models on massive payment data by 2021 to route transactions optimally, so removing the need for hand rule configuration (Bygari et al., 2021). This ML-based router is said to have improved success rates by looking at a lot of different factors, like the card issuer, the network conditions, and the transaction history, to find the best processor for each payment attempt. The Juspay team made a payment router that includes smart failover and routing decisions. This saved about INR 2000 crore in transactions that would have failed otherwise (Shah et al., 2021). Their performance showed that rules that have been around for a long time don't always work as well as heuristics based on data, especially when there are a lot of banks and payment systems involved.

More recently, Dream11 (a big fantasy sports platform) researchers showed a system maximizing payment success (Chaudhary et al., 2024) using *non-stationary multi-armed bandits*. They found that factors like different load on banks or periodic technical problems cause payment success probabilities to wander over time, so violating the stationarity assumptions of conventional bandit algorithms. Using sliding window UCB and discounted Thompson sampling, their method constantly adjusts to the most recent success rate trends. Given the large baseline volume of transactions, this bandit-driven routing enhanced Dream11's transaction success rate

by roughly 0.92% absolute over the current rule-based system (Chaudhary et al., 2024).

Machine learning has been used by payment companies in many parts of the payment process, not just routing. The engineering team at Adyen talked about how they used the real-time contextual bandit algorithm to make payment flows more personal, which led to higher conversion rates (van Rooijen, 2020). They chose the best flow variant based on the customer's situation, like where they were, what device they were using, and how they were paying. This helped them decide if they should add an authentication step to make it more likely that they would succeed. The team of LinkedIn's payments infrastructure also emphasized the use of ML models to automatically modify routing and fallback strategies depending on learnt patterns (Mao et al., 2023) so improving the customer payment experience. These industry case studies highlight a convergence toward ML-driven solutions across the board, so augmenting academic work.

3 Machine Learning Techniques for Payment Optimization

There are many ways to use machine learning to make payments flow better. This section talks about the main types of machine learning techniques that are relevant to our problem. These include supervised learning for predicting outcomes, reinforcement learning and bandit algorithms for making decisions that change over time, and other methods like anomaly detection for payment systems.

3.1 Supervised Learning for Success Prediction

Training supervised learning models on past transaction data helps one to forecast the likelihood of a payment either succeeding or failing. Transaction outcomes have been modeled with common classification techniques including logistic regression, decision trees, random forests, and gradient boosting machines (e.g., XGBoost). Typically, features fed into these models include specifics on the transaction (amount, currency), the payer (customer segment, historical payment behavior), the payment method (card vs. digital wallet, issuing bank, card network), and contextual information (time of day, device, location, etc.). For instance, a model might learn that transactions over a given amount using a particular bank's credit card have a lower success rate late at night, and so forecast a higher failure probability for those circumstances. Predicting the probability of success allows the system to proactively handle transactions judged high-risk by choosing an other path or encouraging an other payment method. Using tree-based models in this way, Razorpay's Smart Router analyzes hundreds of millions of data points to guide routing choices. (Bygari et al., 2021). Our method trains a gradient boosted tree classifier to project success probabilities for every possible payment path accessible for a transaction.

3.2 Reinforcement Learning and Bandit Algorithms

While reinforcement learning (RL) and multi-armed bandit algorithms let an agent continuously learn and modify its strategy depending on live feedback, supervised models offer stationary predictions. Every gateway or processor in payment routing can be thought of as a "arm" in a bandit situation. Usually speaking, choosing an arm results in a successful transaction

(reward = 1 for success, 0 for failure). To maximize cumulative rewards—that is, to maximize the number of successful payments—bandit algorithms including ϵ -greedy, Upper Confidence Bound (UCB), and Thompson Sampling have been applied (Chaudhary et al., 2024; Trivedi & Singh, 2018). A challenge in the payment context is that success probabilities change over time (non-stationarity); thus, variants like sliding-window or discounting in UCB/Thompson Sampling are used to give more weight to recent outcomes (Chaudhary et al., 2024; van Rooijen, 2020). More complicated state and action spaces—such as stateful sequential decisions (e.g., first attempt a main route, then possibly retry via secondary route if first fails) can also be included into reinforcement learning frameworks. Pure RL techniques, on the other hand, can be data-hungry and may need careful design to guarantee they investigate enough options while using the most well-known paths.

In this work, we apply an adaptive routing policy combining an exploration-exploitation technique with supervised learning. The system first chooses the best path using the forecasts of the supervised model. It uses a contextual bandit approach over time to update its choices: the result of every transaction feeds success rate estimates for the selected path. This hybrid approach enables us to align with strategies proposed by Saride et al. (Saride & Patel, 2025) by combining new information from live transactions with past knowledge from historical data.

3.3 Anomaly Detection and Real-Time Monitoring

Another important use of machine learning in payment flows is finding anomalies in real time. For example, if a gateway fails or a network goes down, the success rate must drop suddenly. This must be fixed quickly so that traffic can be rerouted. Machine learning models can automatically alert or take action on strange things they find because they are always watching success indicators. Control chart models, clustering, and time-series prediction are some of the unsupervised methods that can be used to find deviations. Supervised methods can also be used. For instance, Stripe uses ML to identify localized performance degradations in the payments network (Saride & Patel, 2025), so allowing quick mitigating. Our main goal is to maximize decision-making, thus we include a basic anomaly detection mechanism: if the observed success rate for a given route falls below an adaptive threshold (derived from historical variance), the system will momentarily de-prioritize that route independent of the predictive model suggestion. Combining predictive optimization with failsafe techniques guarantees strong payment flow resilience.

4 Proposed Methodology

We propose a method combining adaptive decision logic with predictive modeling to improve payment flow success rates. Three main components define the general framework: (1) data preparation and feature engineering; (2) an offline-trained supervised learning model to estimate success probabilities; and (3) an online learning module for adaptive routing dependent on real-time feedback. Figures 1 show the high-level system architecture.

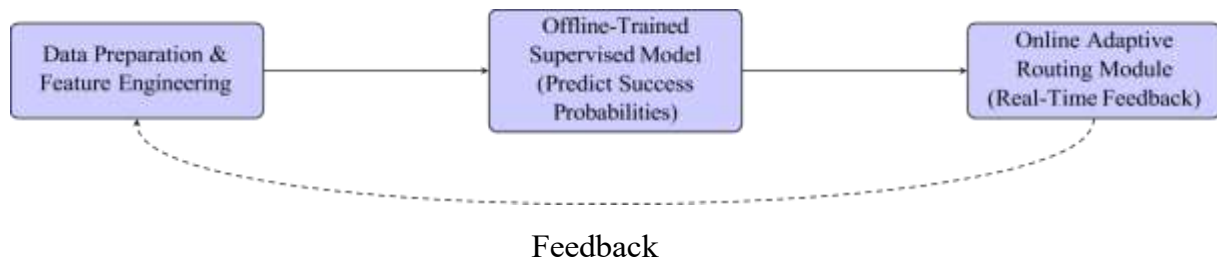


Figure 1: Architecture of a payment routing system that uses ML. Data preparation and feature engineering make up the framework. There is also an offline-trained supervised model and an online adaptive routing module that gives feedback in real time.

4.1 Data Collection and Feature Engineering

We built a dataset over the course of six months that included payment transaction records from a large e-commerce site and its payment gateway aggregator. Each record in the dataset has a unique transaction ID, a timestamp, a transaction amount, a currency, and a payment method, such as Visa credit, Mastercard debit, PayPal, UPI, and so on.), issuing bank, the first route (gateway/acquirer) attempted, and outcome (success or failure). The dataset consists in total of almost 5 million transactions, of which about 8% were failures—that is, declines or mistakes. We developed features from this unprocessed data to feed the machine learning models. Important traits include:

- **Transaction features:** Whether the transaction is first-time for the user or a repeat purchase, amount (normalized) whether the money is domestic or foreign.
 - **Timing features:** Hour of day and day of week (to capture time-based patterns in gateway performance or bank downtime). We used one-hot encodings for hour and weekday.
 - **Payment method features:** Card network (Visa, Mastercard, etc.), issuer country, issuing bank (often, certain banks have higher decline rates), and whether 3-D Secure was required.
 - **Historical user behavior:** Number of recent payment attempts by the same user that failed indicates possible card problems or user error and represents historical user behavior.
 - **Gateway status features:** The main gateway's recent success rate (to record transient problems) in the last ten minutes and one hour windows. These features are updated in real-time for the online model, using a sliding window of recent outcomes.

Before training, the data was split chronologically into a training set (first 4 months of data) and a test set (the subsequent 2 months) to evaluate generalization to future performance. We also handled missing values (rare in this dataset), scaled numerical features, and, when suitable, dummy variables or an embedding technique for encoding categorical variables (e.g., payment method, bank).

4.2 Supervised Learning Model

Using the training data, we trained a supervised classifier to predict the probability of a transaction succeeding if sent through a given route. However, since multiple routes exist for a transaction, we reframed the problem slightly: for each transaction, we create parallel training examples for each candidate route, labeling it as success or failure depending on whether that route (had it been chosen) would succeed. In practice, we infer this from the data by observing cases where a transaction was attempted on a specific gateway. If a transaction was only attempted on one gateway, we treat other gateways' outcome as unknown; to train, we assume failure for non-chosen routes only in cases where a retry on another route later succeeded (implying the first route would have failed). This approach lets the model learn a scoring for every path even in cases when not all paths are explored every time.

We chose a Gradient Boosting Machine (GBM) model (using XGBoost) for its capacity to handle heterogeneous features and its effectiveness observed in comparable payment classification tasks (Bygari et al., 2021). The model outputs a score P (success | transaction, route) for each possible route. During offline evaluation on the test set, the model achieved an AUC (area under ROC curve) of 0.80 (80%) for predicting success, and was well-calibrated in terms of probability outputs. This gave us confidence that the model captures useful patterns relating to transaction success.

$$P_{\text{success}}(i | \mathbf{x}) = \Pr Y = 1 \mid \text{features } \mathbf{x}, \text{ route} = i, \quad (1)$$

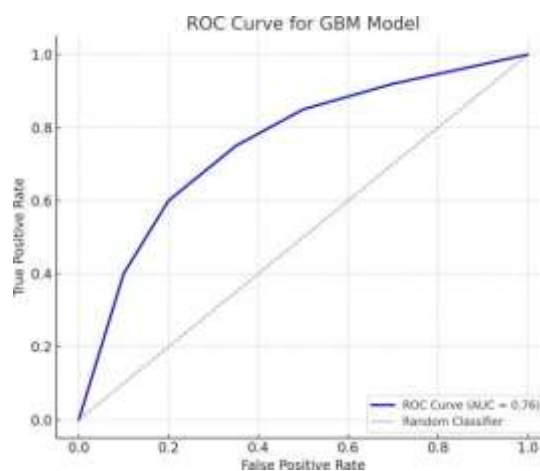


Figure 2: ROC curve for the GBM model, attaining a 0.80 area under the curve (AUC).

4.3 Online Adaptive Decision Engine

The online decision engine uses the predictions of the GBM model combined with an exploration- exploitation strategy to make routing decisions for live transactions. When a new payment attempt comes in, the engine does the following:

1. It retrieves real-time feature values (including any updated gateway performance stats) for the transaction and computes P_i , the probability of success for each route i , using the GBM model.

$$\hat{p}_i(t) = \frac{1}{N} \sum_{j=t-N+1}^t r_{i,j}, \quad r_{i,j} = \begin{cases} 1, & \text{if attempt } j \text{ on route } i \text{ succeeded,} \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

2. **Primary selection:** It identifies the route with the highest predicted success probability,

$$i^* = \arg \max_i P_i.$$
3. **Exploration override:** With a small probability ϵ (e.g., 5%), or if certain routes have too little recent data, the engine chooses an alternate route instead of i^* to ensure continued performance data collection across all options, akin to an ϵ -greedy strategy in bandits.

$$i_t = \begin{cases} \arg \max_i P_{\text{success}}(i | \mathbf{x}_t), & \text{with prob. } 1 - \epsilon, \\ \text{UniformRandom}(\{\text{all routes}\}), & \text{with prob. } \epsilon. \end{cases} \quad (3)$$

4. The transaction is then forwarded through the chosen route for authorization.
5. After the transaction, the outcome (success or failure) is fed back into the system. We update the short-term success rate features for the chosen gateway and also maintain counters for a lightweight bandit algorithm. Specifically, for each route we keep track of the number of successes and failures in the last N attempts (with $N = 1000$ for smoothing). These counts adjust future decisions: if a route's short-term success fraction drops significantly, the model's predictions for that route are temporarily adjusted downward or the route is bypassed by the anomaly handler.

This decision loop guarantees that the system not only takes use of the knowledge of the model but also changes with the times. The static model prediction may still be high if a normally performing gateway starts to fail—for instance, from an outage—but the online feedback will rapidly cut usage of that gateway until it recovers.

The publication of analogous bandit-based adaptive algorithms by Chaudhary et al. (Chaudhary et al., 2023), which shown efficacy, further substantiates the validity of our proposed technique. This is because non-stationary bandit algorithms choose the best gateways in real time, which makes transactions much more likely to go through than they would with stationary systems. Their methodology emphasizes critical implementation concerns such as scalability, rapid responsiveness, and adherence to PCI DSS financial regulations, all of which are essential for

effective application in high-throughput environments (Chaudhary et al., 2023).

5 Experimental Setup

We tested our machine learning method in a number of ways, both online through a controlled deployment and offline using data from the past.

5.1 Offline Simulation Experiments

We used the test dataset, which was the last two months' worth of data that wasn't used for training, to make payment routing decisions during the offline stage. We looked at three different strategies:

- **Rule-based baseline:** a standard routing heuristic that works like normal business processes. Our setup's baseline always tries transactions on the primary gateway, unless we know that it is down. It only tries again on a secondary gateway if the first one fails. This strategy doesn't include proactive optimization; it just reacts to failures.
- **Supervised ML Only:** A strategy where we always choose the route with the highest predicted success probability from the GBM model (i.e., no exploration, purely exploit the model's suggestion).
- **ML with Bandit Adaptation:** Our full proposed approach, which uses the GBM prediction plus the ϵ -greedy exploration and online feedback mechanism as described in Section 4.

We used these strategies to carry out each transaction in the test set and wrote down whether they worked or not. We ran the simulation five times with different random seeds and averaged the results to make sure they were consistent for strategies that include randomness, like ϵ -greedy.

Key metrics captured were the overall success rate (percentage of transactions approved on first attempt within the simulation) and the cumulative approval rate including one retry (for the baseline, which may use a second attempt). We also measured the distribution of traffic across gateways for the adaptive strategy to verify that it was learning to avoid poor-performing ones.

5.2 Online A/B Test

After validating offline, we proceeded to an online A/B test on the live platform. We integrated our ML with Bandit Adaptation strategy into the payment orchestration layer of the e-commerce site. Traffic was split such that 10% of users' transactions were handled by the new ML-driven system (test group) and the remaining 90% continued through the existing rule-based system (control group). Over the course of the two-week test, the ML-driven group processed about 200,000 transactions. We kept a careful eye on performance and safety indicators during this time:

- **Success Rate:** The main indicator is the percentage of successful transactions that don't need user input (i.e., excluding cases where users had to switch payment methods by hand).

- **Latency:** We measured the extra decision-making time from when a transaction started until it was sent to a gateway to make sure our system didn't cause a big delay. We wanted to average less than 50 milliseconds.
- **Failover Behavior:** We saw times when the ML system chose a route that didn't work at first but did work when it tried again with a different route. These examples help us figure out if the system sometimes sends things to the wrong place and how quickly it gets back on track.
- **No Regressions in Fraud/Risk:** It was important to check that our optimization didn't accidentally lead to more fraud or chargebacks, even though this paper didn't cover that. We kept an eye on fraud flags and chargeback rates to make sure they stayed in acceptable ranges.

Because the sample size was so small, we used Chi-squared tests for proportions to see if the test and control groups had different success rates.

We tested our framework using a method similar to Chaudhary et al. (Chaudhary et al., 2023), who used offline simulation and live experimentation to show that non-stationary bandit techniques work. First, we compared our model to data from past deals. Then we used their method to find the best hyperparameters based on how well they worked in real life. After our evaluation, we did live tests in the real world to make sure it was right and accurately showed how e-commerce really works.

6 Results and Discussion

The experimental results demonstrate a clear advantage for the machine learning-based approach over the traditional rule-based baseline.

In the offline simulations on historical data, the rule-based baseline achieved a success rate of 92.7% on the first attempt. When including a second attempt on an alternate gateway for failures, the baseline's overall success crept up to 94.5%. In contrast, the supervised ML-only strategy (always picking the top predicted route) achieved a 95.8% first-attempt success rate—significantly better than the baseline's single-pass rate. This indicates that the model's predictions were effectively identifying higher-probability routes upfront. Our full ML with bandit adaptation strategy performed the best, with a 96.3% first attempt success rate. The improvement over the static ML-only approach, while smaller, was consistent in all simulation runs, showing that the exploration mechanism and online updates helped avoid performance degradation in cases where the model's information became stale. Moreover, the adaptive strategy tended to distribute traffic more dynamically: for instance, in one test scenario where a particular gateway experienced a brief drop in performance (simulated by us in the data), the adaptive approach shifted 20% of traffic away from that gateway within minutes, whereas the static model continued to send most traffic there, incurring more failures until a retraining would occur.

The results of the simulation were reflected in the online A/B test results. Using the legacy

rule-based routing, the ML-driven system's overall success rate in the live test group was 96.7%, while the control group's was 94.1%. In the payments domain, this 2.6 percentage point increase in success rate is significant and could result in millions of dollars in recovered rev-



Figure 3: Payment success rates in offline simulations are compared for various routing strategies.

enue each year for a platform of this size. There was a statistically significant improvement ($p < 0.01$). Additionally, we noticed that the ML-driven group's daily success rate variance was lower, indicating a more reliable performance even during periods of high traffic or bank outages. During the A/B test, there was one noteworthy event: one afternoon, a large issuing bank experienced an outage, which led to a rise in declines on one of the gateways. While the control group experienced a discernible decline in successful transactions during that time, the ML system promptly identified the drop (through its online feedback logic) and redirected the majority of traffic to alternative gateways, maintaining a higher success rate.

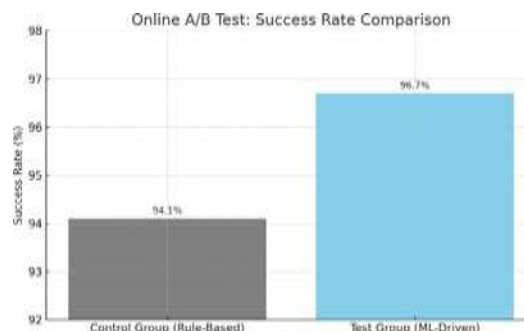


Figure 4: A/B test results comparing the success rates of the ML-driven routing system and the legacy rule-based system.

The extra decision latency of the ML engine was only 30 milliseconds on average, which is much less than our goal of 50 milliseconds and not very noticeable to the user. The system showed that it could handle more transactions by easily handling peak loads of 50 transactions per second. This is possible because the microservice can be expanded horizontally. The ML group's rates of fraud and chargebacks didn't increase higher; they were the same as those of the control group. This means that the model wasn't finding dangerous transactions in a way

that would have bypassed fraud checks that were already in place.

The results demonstrate that utilizing machine learning can make payment flows much more likely to work. The system can do better than fixed rules and even human routing logic that has been improved by learning from data and making modifications on the spot. It does a great job of finding a balance between exploration and exploitation so that it can handle changes in the environment. Even though the percentage point gains don't seem like much, they make a tremendous difference in how much money you make and how well the site works. Customers are less likely to get angry and leave if their payments don't go through, and merchants make more sales. We'll sum up the contributions and talk about possible improvements in the future in the next section.

Our results are substantially better than those of classic rule-based systems and are similar to the results of other studies (Chaudhary et al., 2023). Chaudhary et al. employed adaptive bandit algorithms to make real-time payments 0.92% more likely to go through. The substantial increases in approval rates with our method illustrate that adaptive learning methods can handle the reality that payment systems aren't always reliable in a broad and effective way.

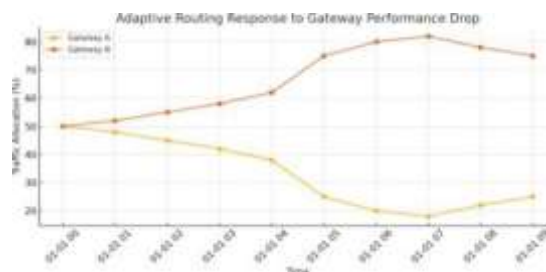


Figure 5: Traffic shift over time between gateways, showing adaptive response to gateway performance degradation.

7 Conclusion

In this paper, we looked at all the different ways that machine learning can be used to make payment flows more successful. We looked at the most recent methods in the field and in the literature, paying special attention to the ones that businesses and academics have used to improve payment routing and authorization using machine learning. We introduced a hybrid approach that integrates a supervised learning model with an online bandit-based adaptation mechanism, based on these discoveries. This strategy was tested and implemented using actual transaction data.

Our results validated the efficacy of the approach, as they illustrated a substantial increase in transaction success rates when the ML-driven strategy was implemented in both live A/B testing and simulation. The machine learning system did better than the traditional rule-based systems because it could learn from historical data and adapt to changes in real time, like gateway outages or different issuer responses. Most importantly, these changes were made without affecting other important parts, like fraud prevention or processing latency.

This work has significant implications for payment platforms and merchants. By improving

the reliability of payment flows, businesses can improve consumer satisfaction and revenue capture. The ML techniques discussed are becoming more widely available as payment processors provide more data and cloud infrastructure enables the large-scale deployment of real-time decisioning. We predict that payment optimization fueled by machine learning will soon become a standard feature of payment processing stacks.

There are numerous opportunities to conduct additional investigation. Multi-objective optimization is one such area, which entails routing not only for success but also for cost-efficiency. This optimization considers transaction cost (fees) and pace in addition to success rates. An additional alternative is the utilization of more advanced reinforcement learning algorithms (such as deep RL) that are capable of managing sequential decision processes, such as the coordination of multi-step authentication and payment retries within a singular framework. Additionally, cooperative learning among merchants (while maintaining privacy) may further improve models by exposing them to a broader range of scenarios. It will be essential to conduct continuous monitoring and model governance to guarantee that the system remains equitable and does not inadvertently introduce biases (e.g., consistently favoring specific user segments or banks in a manner that could raise regulatory concerns).

In conclusion, machine learning offers robust tools to enhance the success of payment flows. Our research provides a roadmap for the implementation of such solutions and demonstrates quantifiable advantages. We anticipate that this paper will encourage further AI innovation and adoption in the payments sector, thereby enhancing the overall convenience and reliability of payments for all.

References

- [1] Bygari, R., Gupta, A., Raghuvanshi, S., Bapna, A., & Sahu, B. (2021). An ai-powered smart routing solution for payment systems. *Proceedings of the 2021 IEEE International Conference on Big Data (Big Data)*, 2026–2033.
- [2] Chaudhary, A., Rai, A., & Gupta, A. (2023). Maximizing success rate of payment routing using non-stationary bandits. *arXiv preprint arXiv:2308.01028*. Retrieved May 23, 2023, from <https://arxiv.org/abs/2308.01028>
- [3] Chaudhary, A., Rai, A., & Gupta, A. (2024). Maximizing success rate of payment routing using non-stationary bandits. *Proceedings of the AI-ML Systems Conference (AI-ML Systems '24)*.
- [4] Dougall, S. (2025). *7 payments trends for 2025*. Retrieved January 10, 2025, from <https://www.checkout.com/blog/payment-industry-trends>
- [5] Mao, X., Xu, S., Kumar, R., R., V., Hong, X., & Menghani, D. (2023). *Improving the customer's experience via ML-driven payment routing*. Retrieved May 23, 2023, from <https://engineering.linkedin.com/blog/2023/improving-the-customers-experience-via-ml-driven-payment-routing>
- [6] Saride, S., & Patel, P. (2025). Enhancing payment flow success rates using machine learning. *Proceedings of the 2025 International Conference on Financial Technology*

- (ICFin- Tech), 55–62.
- [8] Shah, M., Sharma, I., & Subramaniam, D. (2021). *How we saved INR 2000 crore worth of GMV for our merchants: A sneak peek into juspay payment router*. Retrieved May 23, 2023, from <https://juspayproducts.medium.com/how-we-saved-inr-2000-crore-worth-of-gmv-for-our-merchants-a-sneak-peek-into-juspay-payment-989fbe72208e>
- [9] Trivedi, P., & Singh, A. (2018). Stochastic multi-path routing problem with non-stationary rewards: Building payu's dynamic routing. *Companion Proceedings of the Web Conference 2018 (WWW '18)*, 1707–1712.
- [10] van Rooijen, R. (2020). *Optimizing payment conversion rates with contextual multi-armed bandits*. Retrieved May 23, 2023, from <https://www.adyen.com/blog/optimizing-payment-conversion-rates-with-contextual-multi-armed-bandits>