MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

Hybrid AI Models Combining Rules and Machine Learning for Financial Fraud Control in the United States

Mahuma Akter[1], Rejon Kumar Ray[2], Md Shafiqur Rahman[3], Tanjina Tuly[4], Md Shohail Uddin Sarker[5], Santosh pant[6], Al Amin[7] and Md Al Mamun Siddike[8]

[1]Master of Science in Cybersecurity (MSCS), Washington University of Science and Technology (WUST).
[2]MBA, Business Analytics, Gannon University, Pa, USA.
[3]MBA in Management Information System, International American University
[4]MSc in Business Analytics, Trine University
[5]MS- Computer and Information Systems Security, Gannon University, Pa, USA
[6]Kantipur college of Management and Information Technology
[7]Accounting and Information Systems, Jahangirnagar University.
[8]MS in Business Analytics, Trine University.
Corresponding Author: Mahuma Akter, **Email:** akmahuma@student.wust.edu

Abstract
Detecting financial fraud is getting harder as digital payments explode and transaction volumes skyrocket. Usually, fraud is just a speck in a massive pile of normal activity, creating a data imbalance that makes standard detection methods struggle. Banks often stick to rule-based systems because they are easy to explain to regulators, but these setups aren't great at catching clever, evolving theft. On the flip side, machine learning is great at spotting tricky patterns in big data, but it can be a "black box" and often flags too many innocent transactions. This study looks at whether hybrid AI, mixing old-school rules with modern machine learning, can detect more fraud without making the process unreliable. Using a classic credit card dataset of 284,807 transactions where fraud was extremely rare, the research compared three setups: basic rules, standalone machine learning, and hybrid systems. Logistic Regression, Random Forest, and XGBoost served as the model baselines. These were tested against three hybrid ideas: using rules as features, a two-stage sequential pipeline, and a weighted ensemble that blends scores from both. Since fraud is so rare, the study moved past basic accuracy to focus on Precision-Recall Area Under the Curve (PR-AUC), recall, and the actual financial cost of false alarms versus missed thefts. While XGBoost was the strongest individual performer with the best PR-AUC, the hybrid systems offered better real-world perks. They were more effective at cutting down false positives and lowering the total cost of managing fraud. The two-stage pipeline and weighted ensemble specifically did the best job of balancing detections against total alerts. Ultimately, these results show that hybrid systems are a smart middle ground, offering a setup that is both easy to understand and highly predictive.

Keywords: Fraud Detection, Hybrid AI, Rule-Based Systems, Machine Learning, Cost-Sensitive Learning, Explainable AI, Financial Security

1. Introduction

1.1 Background: Fraud in the financial world has turned into a real headache for banks and other institutions, especially now that so much of our economy is online. Digital payments, online banking, mobile wallets, they make life easier, sure, but they also give fraudsters a lot more ways to get at money. As the number of transactions climbs and different systems talk to each other more, the old methods of spotting fraud just can't keep up. Every year, companies lose huge sums to fraud. The Association of Certified Fraud Examiners says it runs into billions of dollars worldwide every year [5]. And it's not just about the money. When fraud hits, it shakes trust, makes regulators nervous, and can even rattle the whole financial system. Regulations have tightened, too. Banks and financial services are expected to watch transactions in real time and flag anything fishy. The Federal Financial Institutions Examination Council has guidelines saying institutions need solid monitoring systems and analytical tools to stay on top of suspicious activity and comply with anti-money laundering rules [16]. The tricky part is catching fraud without constantly annoying regular customers with false alerts. Getting that balance right is harder than it looks.

In the past, detecting fraud mostly meant relying on rules. Experts would set thresholds and conditions to catch the obvious bad transactions. That still works for enforcing known compliance rules, but it fails when fraudsters come up with something new or clever that doesn't fit the rules. That's why machine learning and data mining started to look appealing. Researchers found that these methods could spot patterns in huge transaction datasets that humans would miss. Ngai et al. (2011) did a detailed review showing how classification, clustering, and anomaly detection techniques could catch fraud much better than old-school manual checks [24]. Then AI started showing up in the mix. West and Bhattacharya (2016) looked at AI-based fraud detection systems and noticed that machine learning models, neural networks, and hybrid approaches were getting better at catching complex fraud that rules would miss [31]. These models can pick up on subtle patterns that wouldn't trigger a rules-based system. With more computing power and data, AI systems can now scan huge numbers of transactions and still flag suspicious activity accurately. Things are getting messier on the system side, too. Markets, payment networks, and new financial tech are all tightly connected, which makes the system more fragile in ways that go beyond normal fraud. Jakir (2025) points out that AI-based frameworks are increasingly being used to find early warning signs of system-wide financial issues before they get out of hand [20]. So, fraud detection isn't just about catching one bad transaction anymore; it's part of keeping an eye on the bigger financial picture.

1.2 Problem Statement: Technology is everywhere, but fraud detection remains a tough nut to crack. Digital financial services churn out mountains of transaction data every day. Fraudulent activity usually represents a tiny fraction of all that data. It makes spotting it a bit like finding a needle in a haystack. Fraudsters keep evolving, too. Detection systems have to adapt constantly. That is where machine learning comes in. It can learn complicated patterns from huge datasets that old methods just cannot handle. Machine learning is useful because it can detect hidden relationships. It catches subtle behavior that rules-based systems miss. Leo et al. (2019) note that banks are using machine learning more and more. It can sift through massive datasets and learn from past transactions. It gradually improves predictions [21]. Millions of transactions can be analyzed in a snap. This helps catch small anomalies that could signal fraud. Transaction data keeps piling up. Machine learning looks like the only realistic way to scale detection.Still, it is not a magic bullet. One big problem involves giving reliable early warnings. Fraud changes fast. Models have to keep up. Rahman (2025) talks about how machine learning–powered early warning systems can pick up unusual patterns before they turn into bigger problems [26]. Early detection is key here. Catching it quickly can save a lot of money and headaches. Another challenge is explainability. Regulators and auditors want to know why a system flagged a transaction. Some advanced modes, ls like deep neural networks, are great at spotting fraud. They are terrible at explaining their reasoning though. That is a problem in highly regulated environments. Financial institutions are looking for ways to get the predictive power of machine learning while still being able to show how decisions are made. They want to stay accountable and compliant. On top of all that, digital financial systems are growing more complicated. Transactions happen across different platforms and currencies. They happen across different tech stacks. Systems have to manage huge amounts of data. They have to keep up with evolving fraud methods. They still have to stay within regulatory bounds. Effective detection today means handling all of that at once. It requires accurate spotting and adaptability. It needs transparency. All of it must be rolled into one system that does not collapse under the weight of the data.

1.3 Importance of Financial Fraud Control in the United States: Stopping financial fraud is a big deal in the United States, mostly because the sheer size and complexity of the country's money systems are kind of overwhelming. We have one of the biggest and most high-tech financial setups on the planet. It's a massive web of big banks, local credit unions, companies that handle payments, investment groups, and new fintech startups. Every single day, these places move a staggering amount of money. We're talking about everything from someone buying a coffee to massive international wire transfers and huge corporate settlements. Because the pile of transactions is so enormous, even a tiny sliver of fraud can end up costing billions of dollars and causing some serious chaos in how things run. One of the toughest parts for banks here is trying to catch the bad guys without making life difficult for everyone else. People today want their payments to happen right now. They want to buy things online or move money between apps without any hiccups or annoying delays. This means fraud systems have to work instantly behind the scenes. If a system is too slow or flags too many honest people by mistake, it's a problem. But if it's too relaxed, the criminals get through. Finding that sweet spot takes some really smart math and tools that can chew through mountains of data in a heartbeat without losing accuracy.Then there is the whole world of rules and laws. If you run a financial company in the U.S., you're looking at a mountain of regulations regarding money laundering, protecting consumers, and reporting weird activity. These rules basically force institutions to have eagle-eyed monitoring systems that can spot anything fishy and shout about it so an investigator can take a look. As technology gets better, the people making these rules expect banks to keep up. They want to see modern tech being used to handle messy data and new types of scams. So, companies end up in this constant cycle of spending money on new tech just to stay on the right side of the law and keep their defenses up. The way we use money is changing fast, too, and that opens up new doors for people looking to steal. With everyone using mobile banking, apps

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

to pay their friends, and crypto platforms, there are just more ways to get scammed. Fraudsters are pretty creative. They use the same new tech we love to build schemes that can slip right past the old-school security checks. Because of this, the way we catch fraud has to stay flexible. It's not enough to just look for the old tricks anymore; systems have to be able to scale up and recognize weird behavior they've never even seen before.It isn't just about the lost cash, either. When a big bank gets hit by a massive fraud scandal, people stop trusting it. For a financial business, its reputation is pretty much everything. If customers feel like their money isn't safe, they'll leave. For the big players in such a competitive market, keeping that trust alive is the only way they stay in business long-term. It keeps the whole market steady when people feel like the system actually works. Because the stakes are so high, building better ways to spot fraud has become a top priority for basically every financial firm in the country. They are leaning hard into things like advanced analytics and AI to get better at flagging the crooks without slowing down the day-to-day business. As the tech we use to pay for things keeps changing, the ways people try to cheat will change too. That makes constant innovation in security a basic requirement for keeping the U.S. financial world safe in a digital-first era.

1.4 Research Questions: The bigger financial systems get, the harder it is to spot fraud with the old-school methods. Most banks still use rule-based systems, they check transactions against set thresholds or simple conditions. If something looks weird according to the rule, it gets flagged. That works fine for known tricks. But fraudsters don't stay predictable. They change tactics constantly, and those systems can't always keep up. Machine learning changes the game a bit. It can look at tons of historical data and spot patterns humans might miss. Odd behaviors, subtle anomalies, stuff that would slip past a simple rule can get caught. But machine learning has its own headaches. Models can be opaque. People need to understand why a system flagged something, especially when regulators are watching. A correct prediction isn't enough, you have to be able to explain it. That's why hybrid systems are interesting. They mix rules with machine learning. Rules give structure and oversight, and machine learning brings the ability to catch new, weird behavior. Together, they might be stronger than either alone. This study is trying to see if hybrid approaches actually work better. Can they handle huge transaction datasets? Do they catch more fraud without breaking the system or confusing regulators? Can they survive real-world messiness, like imbalanced data and unpredictable transactions? Answering these questions could help design fraud systems that are smarter, faster, and more practical for banks that can't afford to fall behind.

1.5 Hypotheses: This research follows three main ideas about how hybrid fraud detection setups, the kind that mix old-school rules with modern machine learning, actually hold up in the real world. These thoughts didn't just come out of nowhere; they are based on what other researchers have found and the actual headaches banks deal with every day when trying to spot a scam. The first big idea (H1) is that hybrid models do a better job with the Precision–Recall Area Under the Curve (PR-AUC) than just using machine learning on its own. Most of the time, fraud data is incredibly lopsided. You have millions of normal transactions and only a tiny handful of fakes. In a lopsided world like that, PR-AUC is a much better yardstick than basic accuracy because it cares specifically about finding that needle in the haystack. Hybrid models take the "tribal knowledge" from human experts (the rules) and plug it right into the math of the machine learning side. The hope is that this combo catches the obvious scams we already know about while the machine learning part sniffs out the weird new patterns. Because of this, the study expects the PR-AUC numbers to climb when these two methods work together.The second idea (H2) suggests that hybrid systems hit the "false positive" button less often than pure machine learning does. This matters because false alarms are a nightmare for banks. When a regular person tries to buy groceries, and their card gets declined by mistake, it's a bad experience for the customer and a lot of extra work for the fraud team,m who have to double-check everything. Pure machine learning can sometimes get a bit too enthusiastic and start seeing patterns where they don't exist. By bringing in hard logic, rules specifically meant to catch "sure thing" fraud, the system should become more grounded. The expectation here is that the hybrid setup will be more precise and stop crying wolf so much. The third idea (H3) gets into the money side of things. It proposes that hybrid setups actually cut down the total bill for fraud monitoring. In a bank, different mistakes cost different amounts of money. If a scammer gets away with it (a false negative), the bank loses the cash. If the bank stops a legitimate customer (a false positive), they lose money on labor and support. By blending rules and math, these models should catch more bad guys while also bothering fewer innocent people. When you look at the total cost, the losses from fraud plus the cost of the investigation team, this balanced approach should be the cheapest way to run the shop. Put these three ideas together, and the goal of the study is to see if these hybrid setups actually move the needle on accuracy, efficiency, and the bottom line compared to the way things have always been done.

1.6 Contributions: This work adds a few specific things to what we know about fraud and financial tech by taking a hard look at how rules and machine learning play together. First off, it offers a head-to-head comparison between rules, standalone machine learning, and the hybrid versions. A lot of banks use a bit of everything in their day-to-day operations, but surprisingly few academic papers actually test them all side-by-side using the same rules. This research fills that gap. By using the same data and the same testing tools for every method, it becomes much easier to see where one succeeds and the other trips up. Second, the study introduces three different ways to build these hybrid systems. One treats a rule like just another piece of data for the model. Another uses a "two-gate" system where a rule acts like a first pass. The third uses a weighted vote between the two. Each one of these setups represents a different level of "talking" between the human-made rules and the computer-learned patterns. This lets the study figure out how much interaction is actually needed to get the best results.

Third, the research uses a cost-sensitive way to measure success that feels like a real business. Most AI papers just talk about statistical scores like AUC. But a 1% error isn't just a number; it is a dollar amount. By building a model that assigns actual financial weights to different types of mistakes, this study looks at these systems through the eyes of a bank manager. It's a more grounded way to see if a model is actually worth using in a real office. Fourth, there is an "explainability" check to see how much the rules are actually helping. Using some specific tools, the study measures exactly how much the rules influence the final "fraud" or "not fraud" decision. This helps us understand if the machine learning part is just leaning on the rules like a crutch or if it is actually finding brand-new insights that the human experts missed. Finally, the study tests how these models hold up when the world changes. Fraudsters aren't static; they change their tactics the moment a bank catches on. By simulating things like a sudden spike in fraud or the way patterns drift over time, the research shows which models are sturdy and which ones break the moment things get weird. This "stress test" is a key part of figuring out if these systems are ready for the messy reality of global finance.

2. Literature Review

2.1 **Rule-Based Fraud Detection Systems:**Back in the day, if you wanted to catch fraud, you basically sat down and wrote a list of "if-then" scenarios. These traditional systems work on what we call deterministic rules, essentially a set of hard boundaries based on what experts, bank policies, and government regulators think "bad" behavior looks like. If a transaction hits one of these tripwires, it gets flagged. You'll see this a lot in banking, where systems are looking for things like a sudden huge purchase, someone using a card in a country they've never visited, or just a weirdly high number of transactions in a single hour. These rules are the backbone of most anti-money laundering (AML) setups and the standard surveillance tools that payment processors use. The big plus here is that everyone knows exactly what is happening. If an analyst or a regulator asks why a transaction was stopped, you can point to the specific rule. It's transparent, which matters a lot when you're dealing with the law.When you look at the early research in this field, these expert-driven frameworks were really the only game in town. Abdallah et al. (2016) did a deep dive into these systems and pointed out that most early setups were just a mix of these hard rules and some basic math to find outliers [2]. They found that banks loved these systems because they weren't complicated. They were easy to turn on, easy to explain to a boss, and they kept the auditors happy because every decision left a clear paper trail. For a long time, that was enough to get the job done. But things started to get messy once everyone began moving their money online and the sheer number of transactions exploded. When you have static rules, you have a moving target problem. Criminals aren't stupid; they figure out where the tripwires are and just step over them. This means a rule that worked great in June might be totally useless by October. Baesens et al. (2015) made a good point about this, noting that while it's nice to be able to explain a rule, these systems are pretty bad at catching the "smart" fraud [7]. If a scam involves five different subtle steps that only look bad when you see how they interact, a simple rule-based system is probably going to miss it. They just aren't built to see the complicated, messy patterns hidden in massive datasets.Then there is the headache of false positives. Since these rules are usually set to be extra cautious, because no bank wants to miss a massive theft, they end up flagging a ton of normal people just going about their day. This creates a mountain of alerts. Human analysts then have to sit there and manually check every single one, which is slow, boring, and incredibly expensive. As more people buy things online, that mountain of alerts just keeps growing, and it gets to a point where the human teams can't even keep up. Even with all those flaws, these systems aren't going anywhere yet. They are still a huge part of how banks monitor money because of those regulatory requirements mentioned. In the real world, most banks use these rules as a sort of first line of defense. It's the "easy" layer of a much larger security setup. That said, because financial crime is getting way more sophisticated, researchers have been looking for better ways to do this. There's a real push now to find more advanced tools that can pick up the slack where these old-school rules fall short.

2.2 **Machine Learning in Fraud Detection:** As financial transactions get more tangled and messy, the old way of doing things, basically just setting hard rules for what looks like a scam, isn't really cutting it anymore. This is why so many people are looking at machine learning. These models can chew through massive

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

piles of data and spot weird, complicated patterns that a human or a simple "if-then" rule would probably miss. Instead of waiting for a person to tell it what to look for, the software learns straight from the history of the data itself. It picks up on those tiny, quiet connections between different variables, which has basically given banks a much bigger and better toolkit for spotting fraud before it gets out of hand. One of the best things about these methods is how they handle "nonlinear" relationships. In plain English, that just means that fraud isn't usually just one thing; it's a weird mix of how much was spent, where it happened, the time of day, and how the customer usually acts. Trying to track all those overlapping pieces at once is hard, but algorithms are actually pretty good at it. Back in 2017, Awoyemi and his team looked at things like decision trees and neural networks for credit card fraud. They found that these models generally beat the old-school methods because they could dig out hidden behaviors buried deep in the transaction logs. Their work showed that you could actually use this history to build models that flag bad actors right as the transaction is happening.We've also seen some big jumps in how we mine this data. Carneiro and his colleagues put together a system specifically for e-commerce fraud around the same time. They showed how predictive analytics can sift through giant datasets to find those suspicious red flags. By combining a few different analytical tricks, they moved the needle toward a more dynamic way of watching accounts. It's less about having a static checklist and more about having a system that can shift and change as fraudsters come up with new ways to cheat the system. Lately, deep learning has become the big thing. Pumsirirat and Yan spent time looking into autoencoders and restricted Boltzmann machines. That sounds like a mouthful, but the core idea is that these models learn what a "normal" transaction looks like in a very deep, abstract way. Once the model knows what "normal" is, anything that looks even slightly off stands out. This is a lifesaver when you're trying to find a few dozen fraudulent swipes hidden inside a mountain of millions of legitimate ones.Of course, it isn't all perfect. These models bring their own set of headaches to the table. One of the most popular tools right now is gradient boosting, specifically the XGBoost framework that Chen and Guestrin talked about in 2016. It's fast and handles huge amounts of data without breaking a sweat. But even though these models are incredibly accurate, they can be a bit of a "black box." It's sometimes hard to explain exactly why the computer flagged a specific person, or how the model will hold up if the data starts looking different next month. Banks have to walk a fine line between wanting the most accurate system possible and needing to actually understand and explain how their decisions are being made.

2.3 **Hybrid AI Systems:** The cracks in both old-school rule-based setups and standalone machine learning models have started to show quite a bit. Because of that, researchers keep looking into hybrid artificial intelligence systems. These systems basically try to mash together symbolic reasoning with statistical learning. The whole point of these hybrid architectures is to get the best of both worlds. They take expert rules that humans already understand and mix them with machine learning algorithms that are great at spotting weird, complex patterns in big piles of data. When it comes to catching financial fraud, these systems usually pack several different analytical tools into one big decision framework. This lets banks and other institutions stay accurate while still being able to explain why a certain transaction looked fishy. Most hybrid AI setups work like a layered pipeline where the rules and the machine learning parts talk to each other. You might have a rule-based system sitting at the front door to catch the really obvious stuff, like a huge transaction from a brand-new account. Then, the machine learning models take over for the subtler cases that need a closer look. Zareapoor and Shams (2020) talk about how well these hybrid models work when they combine different algorithms to get better at catching fraud without constantly flagging innocent people by mistake [32]. Their research shows that when you stack these different methods together, the whole system gets much better at seeing the wide variety of ways people try to commit fraud.Another big piece of the puzzle for hybrid AI is making sure the machine learning part isn't just a "black box." Financial companies really need to know why a computer made a specific choice, especially if that choice leads to a police report or freezing someone's bank account. Lundberg and Lee (2017) came up with something called the SHAP framework for this very reason [22]. It's a way to look at a model's prediction and see exactly which pieces of data pushed the needle toward a "fraud" label. Using tools like SHAP helps analysts trust the system more because they can actually see the logic behind the automated decisions. People are also using these hybrid approaches for more than just simple fraud checks. Dola et al. (2024) looked into using machine learning to find secret collusion networks inside big companies [14]. They showed how these advanced tools can dig up hidden relationships between different groups that might be working together to steal money. Their work really highlights how these hybrid frameworks are great at looking at how different entities are connected, which is something traditional monitoring systems usually miss.

Outside of just catching bad actors, adaptive machine learning models are being tested for financial forecasting in markets that change incredibly fast. Bhowmik et al. (n.d.) built some self-adaptive models meant to handle the chaotic data you see in banking and crypto markets [9]. They emphasize that models need to be able to change as the data changes. This is super important for fraud detection, specifically because fraudsters are always changing their tactics to get around whatever security is in place. The process of combining these self-adjusting learning parts with a solid foundation of human-defined rules, hybrid AI looks like a very strong way to build fraud detection that actually lasts.

2.4 **Extreme Class Imbalance in Fraud Detection:** The real trouble with spotting fraud is that it's like looking for a needle in a haystack, except the needle is actively trying to look like hay. In most financial datasets, fraud is incredibly rare. We're talking about one bad transaction for every few hundred or even several thousand honest ones. This "class imbalance" throws a wrench into how machine learning usually works. Most standard algorithms are designed to play the numbers game, they want to maximize overall accuracy. If a model just guesses "not fraud" every single time, it'll be right 99.9% of the time, but it's completely useless for actually catching thieves. Because of this, we have to use specific tricks to make sure these systems don't just ignore the rare stuff. The academic world has spent a lot of time digging into the math behind this. He and Garcia (2009) wrote a big piece on why imbalanced data is such a nightmare for classification [17]. They broke down how traditional models can give you these "fake" good results that look impressive on paper but totally fail to pick up on the minority class. Their research basically set the stage for how we build fraud models today, forcing us to move away from generic strategies and toward methods that can actually handle lopsided data.One way people handle this is through sampling. It sounds simple because it is: you essentially mess with the data before the model even sees it. Dal Pozzolo and his team (2013) looked into using under sampling to fix how models perceive probability [12]. By intentionally throwing out some of the "normal" transactions during training, you force the model to pay closer attention to the fraud cases. It's a bit like giving the model a concentrated dose of the rare events, so it learns the patterns faster. This has become a pretty standard step in most fraud detection setups. Then there's the question of how you even measure if a model is working. If you use the wrong yardstick, you're going to get a skewed view of reality. Davis and Goadrich (2006) argued that for imbalanced data, precision-recall curves are a better bet than the more common ROC curves [13]. Precision-recall is all about the "tug-of-war" between catching as much fraud as possible and not flagging so many innocent people that the investigation costs spiral out of control. In a real bank, every false alarm costs money and time, so this balance is everything.You can also just tell the model that some mistakes are more expensive than others. This is called cost-sensitive learning. Elkan (2001) put forward a framework where you actually bake the cost of an error into the math [15]. Since missing a million-dollar fraud case is way worse than accidentally double-checking a legitimate twenty-dollar grocery run, the model learns to prioritize those high-stakes catches. It's a more pragmatic way of looking at the problem that aligns with how a business actually functions. Lately, we've seen these ideas move into even broader territory, like spotting the early signs of a company going under. Reza et al. (2025) recently showed off a system that uses real-time digital signals to predict financial distress [27]. Even though they're looking for different things, the underlying logic is the same as fraud detection: you're scanning massive amounts of data to find those tiny, rare signals that something is about to go very wrong.

2.5 Research Gap: Even with all the progress we've seen in fraud detection lately, there are some pretty big holes in the literature. We know that rule-based monitoring and machine learning models work for spotting shady transactions, plenty of studies have shown that. But the real-world side of things is a different story. Many banks and lenders are still hitting a wall when they try to actually plug these analytical tools into their day-to-day operations. Specifically, we don't have much data on how "hybrid" setups, the ones that try to mix old-school rules with high-end machine learning, actually hold up in a messy, live financial environment. Another issue is that most research feels a bit narrow. You'll often see a paper that spends all its time obsessing over one specific machine learning model or a tiny handful of algorithms. What's missing is a systematic look at how different types of systems perform when the operational conditions start to shift. Plus, while everyone loves to brag about better predictive accuracy, people rarely talk about the baggage that comes with these systems. This is to refer to things like the cost of false positives, how much extra work it puts on the analysts' desks, and whether anyone can actually understand why the computer made a certain call. For a bank, you can't just chase accuracy; you have to worry about regulations and keeping the lights on efficiently. That brings us to the problem of "black box" models. As these systems get more tangled and complex, it gets harder to explain their logic to a human analyst, a regulator, or a frustrated customer. If you can't explain why a transaction was flagged, the compliance team or the government probably isn't going to let you use it, no matter how "accurate" it claims to be. Finding a way to make these models powerful but also readable is still a huge hurdle that hasn't been cleared yet. One interesting path forward involves using risk scoring frameworks that are actually easy to follow. Rudin and Ertekin (2017) did some work on optimized risk models that spit out clear decision rules for preventing fraud. They showed that you can use machine learning to build something transparent that an analyst can look at and actually make sense of. This kind

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

of approach is exactly what's needed in a regulated industry where you can't just say "the computer said so." Because of these lingering issues, we really need more research that looks at hybrid fraud architectures in a systematic way. We need experiments that don't just look at math, but also factor in things like cost-sensitivity, explainability, and whether the system can scale up without breaking. This study is trying to fill those gaps. We are looking at how these hybrid systems perform when we prioritize both the detection of fraud and the operational reliability that a modern financial institution actually needs to function.

3. Methodology

3.1 Dataset: The credit card transaction dataset used in this work originally came from the Machine Learning Group at Université Libre de Bruxelles. It is widely regarded as a standard benchmark in this field because it closely reflects real-world conditions. A defining characteristic of the dataset is the extreme imbalance between legitimate transactions and fraudulent ones. Building an effective detection system becomes difficult when the data does not mirror the uneven reality of banking operations. In this dataset, there are 284,807 transactions recorded over two days. Only 492 of these transactions are labeled as fraud, representing approximately 0.172 percent of the total. This very small proportion makes the dataset particularly useful for evaluation, since fraud is rare but carries a high cost. Each transaction contains a set of numerical features, with 30 primary variables in total. The dataset includes the "Time" variable, which records when each transaction occurred, and the "Amount" variable, which captures the transaction value in monetary terms. In addition, there are 28 variables labeled V1 through V28. These features were originally derived from raw transactional data and then transformed using Principal Component Analysis, commonly known as PCA. The transformation was applied to protect privacy. Actual credit card details are not disclosed in public research datasets, so PCA preserves the statistical structure of the behavior without revealing sensitive information. The final column, "Class," serves as the target label, where a value of 0 indicates a legitimate transaction and a value of 1 indicates fraud. The PCA transformation serves purposes beyond privacy protection. Converting correlated variables into orthogonal components, it reduces redundancy in the data. This restructuring helps remove noise and improves model stability because learning algorithms do not need to process overlapping information. For methods such as linear classifiers or gradient boosting models, working with independent variables improves numerical reliability. This setup allows algorithms to focus on meaningful patterns rather than unnecessary correlations. Although the transactions originate from European cardholders, the insights drawn from this dataset are not limited to that region. Payment systems share similar structural foundations across countries, and user behavior follows comparable patterns worldwide. As a result, patterns identified in this dataset are expected to generalize to financial institutions in other regions as well. For this reason, researchers often use this dataset as a testing ground before applying their models to private or proprietary banking data.The imbalance in this dataset represents the primary challenge for model development. If a standard algorithm is evaluated using overall accuracy alone, it may achieve high performance simply by predicting that every transaction is legitimate. Such a model could reach approximately 99.8 percent accuracy while failing to detect any fraudulent activity. This outcome would render the system ineffective for real-world fraud detection. Therefore, evaluation requires carefully chosen performance metrics and sampling strategies that ensure minority fraud cases are properly identified rather than overshadowed by the majority class. This dataset provides a suitable environment for testing hybrid fraud detection approaches. It contains a large number of observations along with meaningful behavioral variation. The combination of anonymized PCA features, transaction amounts, and temporal information offers rich input for modeling. It supports the evaluation of how rule-based systems and machine learning classifiers perform when integrated within a hybrid framework.

3.2 Exploratory Data Analysis: The analysis began by digging into the dataset to see what actually makes a fraudulent transaction look different from a normal one. This part of the process, the exploratory data analysis, was about hunting for patterns or "signals" that might help a machine learning model later on. A few different things were examined: how the classes are spread out, how the transaction math compares, when these events happen during the day, and how those PCA features relate to each other. The first thing that jumps out, and it is pretty hard to miss, is just how lopsided the data is. The dataset contains 284,807 total transactions, but only 492 of them are marked as fraud. That is about 0.172 percent. This is the classic needle-in-a-haystack problem. If a "dumb" model were built that simply guessed every single transaction was legitimate, it would be right over 99 percent of the time. However, it would also be completely useless because it would miss every single actual crime. Because of that, it became clear quickly that standard accuracy does not mean much here. Instead, the focus had to stay on things like precision, recall, and the area under the PR curve to see if the model is actually doing its job.Then there is the money. On average, a regular transaction in this set is about $88.35. Fraudulent ones tend to be a bit higher, averaging around $122.21. A Mann-Whitney U test gave a p-value of 8.57×10^{-6}, which indicates that, mathematically, the difference is not just a fluke. But if the actual distributions are looked at, they overlap significantly. It is not possible to simply say "any big transaction is fraud" because plenty of people buy expensive things legally. It is a hint, certainly, but the "Amount" feature is not a silver bullet on its own. It has to work in tandem with other behaviors to mean anything. Time is another interesting variable. The data does not provide a wall clock, just the seconds ticking by since the first entry. When those transactions are plotted by density, clear waves appear that look like daily human cycles. People sleep, so transaction volume drops at night. But here is the kicker: fraud does not drop nearly as much during those quiet hours. The "bad guys" seem to be more active, or at least more visible, when everyone else is offline. Perhaps they think the monitoring is lighter, or they just like the cover of night, but the ratio of fraud definitely spikes when the rest of the world is quiet.A lot of time was also spent looking at those V1 through V28 features. Since these are PCA-transformed, it is not known exactly what they represent for privacy reasons, but it is visible which ones actually matter. Features like V14, V17, V12, and V10 are the heavy hitters here. When V14 is plotted against V17, clusters actually start to form. They are not perfectly separated; there is still some blurring at the edges, but there is enough of a gap there for a model to grab onto. To double-check this, more hypothesis tests were run. The divergence in V14 and V12 is extreme; the p-values were 1.47×10^{-260} and 8.41×10^{-247}. In plain English, those features strongly indicate a difference between the two groups. Finally, the correlations were checked. Because of how PCA works, most of these features do not talk to each other much, but they definitely talk to the "Class" label. V17, V14, V12, and V10 all have a strong negative correlation with fraud. Basically, when those values go down, the chance of fraud goes up. The big takeaway from all this poking around? The imbalance is the main boss to beat. Single features like the dollar amount are okay, but the real power is in the combination of those PCA components and the timing of the transactions. That is what guided how the rest of the experiment was set up.

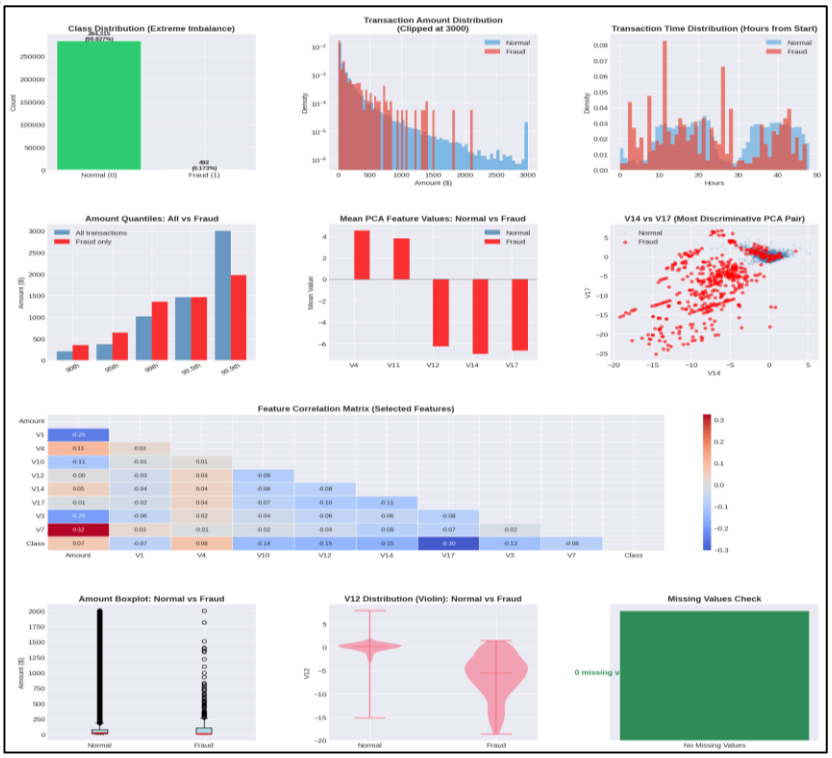

Fig.1: EDA outcomes

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

3.3 Data Preprocessing: Before any of this could actually be fed into a model, things had to be cleaned up a bit. If messy data is put into a high-end algorithm, the result is just confident-sounding nonsense. It had to be ensured that the scales were right, the splits were fair, and the whole setup actually mirrored the real world. The first task was scaling. Those V1-V28 features were already in a pretty good spot because of the PCA process, but "Amount" and "Time" were still in their original units. Algorithms that use distance or gradients get weird if one feature is 0.1 and another is 10,000. So, a standard scaler was used to give them both a mean of zero and a standard deviation of one. Crucially, the scaler was only "taught" using the training data. The testing data should never leak into the scaling math, or it is essentially accidental cheating. Next, the data was split into 80 percent for training and 20 percent for testing. A "stratified" split was used. Because fraud is so rare, a random split might accidentally put all the fraud cases in the test set or none of them in the training set. Stratification forces the split to keep that 0.172 percent ratio in both piles. This resulted in 227,845 rows for training and 56,962 rows for testing. The test set stayed in a "black box" until the very end to make sure the results were honest. A conscious choice was also made not to mess with the class balance in the test set. Some researchers like to oversample or use tricks to make the fraud cases more common while training, which is fine, but the test set has to stay lopsided. If the test set is changed, it is no longer a test for the real world. By the time this was finished, a solid foundation was in place. The features were scaled, the splits were locked in, and the rare fraud cases were right where they needed to be. This meant that whatever the models found later, the results could actually be trusted.

3.4 Class Imbalance Handling: One of the biggest headaches in spotting financial fraud is that the data is incredibly lopsided. In the real world, almost every transaction is legitimate, and only a tiny sliver is actually fraudulent. Even though these crimes are rare, they cause a massive amount of financial damage. The dataset here follows that same pattern, where fraud makes up only about 0.172 percent of the total. This creates a problem because a standard machine learning model might just get lazy and guess that everything is "normal" to keep its accuracy high, while completely missing the actual fraud. To stop that from happening, certain methods were used to make the algorithms pay more attention to those rare cases. A few different strategies were tested to see which one helped the model find fraud without constantly crying wolf on honest customers. These included class weighting, SMOTE, ADASYN, and random undersampling. Each one has a different way of balancing the scales.Class weighting was the first thing tried. It works by telling the model that missing a fraud case is way worse than misidentifying a normal transaction. It essentially puts a higher price tag on mistakes involving the minority class. This is great because it doesn't mess with the actual data; it just changes how the model "thinks" during training. It is fast, efficient, and keeps the data looking natural. Then there was SMOTE, which stands for Synthetic Minority Oversampling Technique. This one actually invents new, fake fraud examples by looking at existing ones and drawing lines between them. It builds up the minority side, so the model has more to study. The downside is that it can sometimes create "noise" if the original fraud cases are already a bit weird or scattered, leading the model to see patterns that aren't really there.

ADASYN is a slightly smarter version of SMOTE. It looks for the fraud cases that are the hardest to classify and creates more synthetic data specifically for those tough spots. It focuses on the "trouble areas" of the dataset. While it is clever, it can sometimes lead to overfitting, where the model gets too focused on those specific fake examples and fails when it sees real-world data. The last option was random undersampling. This is basically just deleting a bunch of the normal transactions until the fraud cases don't look so small by comparison. It makes training go really fast, but the big risk is that you might throw away perfectly good information about what a normal transaction looks like. After testing all of them, class weighting turned out to be the winner. It kept the original structure of the data intact but forced the models to take fraud seriously. It avoided the weirdness of fake data while still getting the job done, so it became the main method for the final experiments.

3.5 Rule-Based Detection System: Alongside the fancy machine learning, a basic "rule-based" system was set up. This is how a lot of banks actually work in the real world. They have specific red flags, if a transaction hits a certain limit or happens at a weird time, it gets flagged. These systems are popular because they are easy to explain to regulators and they follow clear logic. Using this alongside the AI gives a good baseline to see if the high-tech models are actually providing a real benefit. This specific system used six different rules based on the patterns found earlier in the study. Each rule is a simple "yes or no" check. Rule R1 is for high-value transactions. Fraud usually involves larger amounts of money than a regular grocery run, so this rule flags anything over a certain dollar amount. Rule R2 is similar but looks for extreme outliers, the kind of massive transactions that just look suspicious on their own.

Rule R3 looks at the clock. As noted before, fraud doesn't stop just because it's 3:00 AM and most people are asleep. This rule flags transactions that happen during those low-volume, high-risk hours.

The other rules look at the more technical features, the ones from the PCA analysis. Rule R4 checks feature V14, while R5 checks V17. Both of these were identified as strong indicators of fraud. Rule R6 is a double-check; it only fires if both V14 and V12 look suspicious at the same time. The system tracks how many of these rules get triggered for every transaction. This gives a simple score and helps show which red flags are the most effective at catching the bad guys. It acts as a transparent, "no-nonsense" benchmark that mimics the old-school monitoring pipelines used by financial institutions today.

3.6 Machine Learning Baselines: To see how well different approaches work, three standard machine learning models were used as baselines. These are the "bread and butter" of data science and each one looks at the data a little differently. The first is Logistic Regression. It's a classic, straightforward model that tries to draw a straight line between "good" and "bad." It is very fast and easy to understand because you can see exactly how much weight it gives to each feature. Even though it's simple, it is a great starting point. If a more complex model can't beat this, then the complexity isn't worth the trouble. The second is a Random Forest. Think of this as a big group of decision trees all voting on whether a transaction is fraud. Each tree looks at a different piece of the data. By taking the majority vote, the model is usually much more accurate and less likely to get distracted by one or two weird data points. It is excellent at catching those "if-then" patterns that aren't just straight lines.The third is XGBoost, which is a bit of a heavy hitter in the data world. It also uses trees, but it builds them one after another. Each new tree tries to fix the mistakes the previous one made. It is very powerful and can pick up on tiny, subtle hints of fraud that other models might miss. It's famous for being fast and very accurate on this kind of structured data. To make sure these models didn't ignore the fraud cases, the "scale_pos_weight" setting was used. This is just a way of doing the class weighting mentioned earlier, it tells the model to give extra points for getting the fraud cases right. By using these three, the study covers all the bases: a simple linear model, a "team-based" tree model, and a high-performance boosting model. Comparing them shows exactly which kind of math works best for catching thieves in a sea of honest transactions.

3.7 Hybrid AI Architectures

Three different ways of mixing artificial intelligence strategies were put into place to help catch more fraud. The idea here was to take the best parts of rule-based systems, which are easy for people to understand and use, and set expert knowledge, and blend them with machine learning models that are great at spotting new patterns on their own. Banks use these hybrid setups all the time because they need to stay accurate while keeping things clear and running smoothly. The first method used was Rule as Feature integration. In this version, the results from the old-school rule engine were just fed right into the machine learning model as extra pieces of information. Two specific variables were added: rule_flag, which shows if a transaction tripped any of the preset fraud alarms, and rule_score, which shows how worried the rule engine was about that specific transaction. These were tossed into the mix for the XGBoost classifier to look at. By doing this, the model could actually learn how often those rules were right. It didn't have to follow the rules blindly; it just looked at them as more hints alongside everything else it knew about how people spend money. It basically lets the model learn from the experts who wrote the rules.Next up was a Two-Stage Detection Pipeline, which is a lot like how a real bank's back office actually works. This setup has two steps. First, the rule engine looks for the obvious stuff, huge dollar amounts, weird locations, or way too many transactions in a short window. If something looks like a dead ringer for fraud based on these rules, it gets flagged immediately and stops there. If a transaction doesn't set off those big alarms, it moves to the second stage. That is where the machine learning model takes a look to find the sneaky or brand-new types of fraud that the rules might have missed. This keeps things fast because the easy stuff is handled quickly, and the complicated stuff gets the deep analysis it needs. It also keeps things organized because there is a clear line between the "if-then" logic of the rules and the math-based guesses of the model. The third way was a Weighted Ensemble Method. Instead of letting one system have the final say, this method calculates a combined score using both the rules and the model. A specific formula was used to find this middle ground:

$$FinalScore = \alpha \times MLProbability + (1-\alpha) \times RuleScore$$

The little α symbol there is just a knob to turn. If α is high, the system trusts the machine learning model more. If it is low, the system leans more on the expert rules. The right setting for α was found by testing it against a bunch of old transaction data. This way, a company can slowly start trusting the machine learning side of things without having to throw away the rule systems they have used for years. Putting these three strategies together shows a few ways to bridge the gap between human-written rules and data-driven math. Each one has its own perks when it comes to being easy to explain or catching more bad actors, which makes them very useful patterns for building a real-world defense system.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

3.8 Model Evaluation Metrics: Testing a fraud detection system is tricky because the data is so lopsided. In the real world, fraud is rare, usually less than one percent of all transactions. Because things are so uneven, just looking at "accuracy" can be a total trap. A model could say every single transaction is fine and technically be 99 percent accurate, even though it failed to catch a single thief. Because of that, different yardsticks were used to see how well the models actually worked. The main tool for the job was the Precision-Recall Area Under the Curve, or PR-AUC. Precision tells you how many of the transactions flagged as fraud actually were fraud, while recall tells you how many of the total bad transactions the model managed to find. The PR-AUC looks at the balance between those two things. It is a great metric for messy, imbalanced data because it stays focused on the minority group, the actual fraud, rather than getting distracted by the mountain of normal transactions. A few other metrics were used to get the full picture, too. There is the F1-Score, which is a way to average out precision and recall into one number. It's a good middle-of-the-road measurement. Recall on its own is also huge because missing a fraud case usually means the bank loses money. High recall means the model is good at catching the bad guys. The False Positive Rate was also tracked. This measures how often a normal, honest transaction gets flagged as a crime. If this number is too high, customers get annoyed because their cards stop working, and the fraud investigators get overwhelmed with extra work. Finally, the ROC-AUC was calculated just to see how well the model tells the difference between "good" and "bad" across the board. By using all these different measures, the research looks at the problem from a few different angles to make sure the model is effective but also practical to use in a real business.

3.9 Cost-Sensitive Evaluation: Looking at math scores is one thing, but fraud detection also has to be judged by the actual hit to the bank account. In a real bank, different mistakes cost different amounts of money. Missing a fraudulent charge is expensive because that money is just gone. But calling a legitimate charge fraud mostly just costs time for a customer service rep to fix it. To account for this, a cost-sensitive framework was used to see the financial impact of the model. A cost matrix was set up to put a price tag on these mistakes. A False Negative happens when a thief gets away with it because the model thought the transaction was fine. This is the most expensive mistake. For this study, the cost of a False Negative was set at $200, which is a rough average of what an undetected fraud case might cost. On the flip side, a False Positive is when a normal person's transaction gets flagged. This doesn't lose the $200, but it does mean someone has to spend time looking at the file or calling the customer. That administrative headache was priced at $10.With those numbers in place, the total cost of running the system was figured out with this formula:

$$Cost = (FN \times 200) + (FP \times 10)$$

This way of looking at things makes the results much easier to understand in a business context. Instead of just saying a model is 95 percent accurate, it's possible to say how much money it would save or lose. It helps people decide exactly where to set the thresholds for the model to get the best balance between stopping thieves and keeping customers happy.

3.10 Explainability Analysis: Making sense of why a model does what it does is a huge deal for fraud detection. Banks and other financial groups have to follow strict rules, so they can't just have a "black box" system that spits out answers without any reason. It is hard to see inside complex setups like gradient boosting, so SHAP (Shapley Additive exPlanations) was brought in to help. This is a common way to look at things based on game theory. It basically figures out how much credit or blame each specific feature deserves for the final call the model made. The look at how the model works happened in two different ways. First, a global view showed which variables were the big players across the whole dataset. By adding up SHAP values for every single transaction, a list was made to show which things mattered most for spotting fraud. This part of the study helped pin down certain behaviors that usually mean trouble. It might be a weirdly high dollar amount, strange timing for a purchase, or someone trying to run a bunch of transactions through in just a few seconds. Then, the focus shifted to a local level to see how individual decisions were made. Every time the system flagged something as fraud, SHAP values showed exactly which features pushed it over the edge. This is great for investigators because they can see the "why" behind an alert. A specific hit might happen because a big purchase was made very fast, right after a specific rule was triggered. Having these clear reasons makes the machine learning output much more useful for human analysts who have to decide if an alert is real or just a mistake. A big part of this was seeing how the rule-based features actually helped out in the hybrid model. Since the system uses the results of those rules as inputs, SHAP was a good way to see how much the model actually cares about them. By checking those specific values, it was possible to see if the model was leaning hard on those old-school rules or if it was finding its own new patterns in the data. This gave a clear look at how the logical rules and the machine learning parts were talking to each other inside the whole framework.

3.11 Robustness Testing: Models that catch fraud have to stay sharp even when things change in the real world. People change how they shop, and criminals are always coming up with new ways to steal, which can make a model get worse over time. To see if these models could actually hold up, two different tests were run: one for prevalence shifts and one for concept drift. The first test looked at what happens when the amount of fraud in the system goes up or down. In a real bank, you might see more fraud during the holidays or when the economy gets weird. To mimic this, the amount of fraud in the data was manually tweaked to be anywhere from 0.05% to 5%. This was done by carefully resampling the data but keeping the features the same. The goal was to see if the models would break if the balance of classes shifted. A lot of models get trained on one specific mix of data, so a big change can throw off their calibration. PR-AUC was used to track how they did, since that is way better than using regular accuracy when fraud is so rare.The second test was all about concept drift. This is what happens when the actual nature of the data changes over time. Criminals are smart, and they adapt when they get caught, so a model trained on last year's data might not know what to do with a new trick today. To test for this, the data was split by time. The models were trained on older records and then tested on newer ones from a later period. This is a much better way to see how a model will actually behave once it is deployed and has to deal with the future. Just like the other test, PR-AUC was used to see if the performance dropped off. By looking at the gap between the training period and the evaluation period, it was easy to see how well each model handled the change. This was a good way to find out if the hybrid setup, mixing rules with machine learning, stayed more stable than just using one method alone when fraud patterns started to shift.

3.12 Statistical Significance Testing: Whenever someone compares a few different fraud detection models, the differences in how well they work can sometimes just be luck. The dataset might have some weird random quirks that make one model look better than another, even if it isn't actually an improvement. To make sure the gains in performance reported here were actually real, some formal statistical testing was done. The first trick used was bootstrap resampling to get some confidence intervals for the main metric, which is PR-AUC. This is a non-parametric way to do things, where the computer samples the dataset over and over with replacement to see how much a performance score fluctuates. In this specific study, a ton of these bootstrap samples were pulled from the test data, and the PR-AUC score was calculated for every single one. That gave a whole distribution of values for each model, which made it possible to figure out the 95% confidence intervals. By looking at whether these intervals overlapped for the different models, it became clear whether the performance gaps were just random noise or if one model was consistently better than the others. On top of the bootstrap stuff, McNemar's test was used to look at classification errors between models, one pair at a time. This test is built specifically for checking how two different classifiers stack up when they are both looking at the same exact data. It really zeroes in on the moments where the two models disagree on a prediction. For example, it looks at cases where one model gets a transaction right, but the other one totally misses it. By digging into those specific disagreements, McNemar's test can tell if one model is statistically better at getting things right. Using bootstrap confidence intervals and McNemar's test together gives a nice, full picture for validation. The bootstrap part shows how stable the performance is, and McNemar's test looks at the actual errors made on individual transactions. Doing both ensures that any claims about better fraud detection are backed up by solid evidence instead of just some lucky streaks in the data.

3.13 Error Analysis: Beyond just looking at the hard numbers, a qualitative error analysis was performed to see where the fraud detection models were strong and where they tripped up. This kind of analysis is about looking at specific examples of when models win or lose. It helps people understand why certain mistakes happen and points out exactly where things could be better. The investigation focused heavily on the transactions where the rule-based system and the machine learning model didn't see eye to eye. Since the whole setup uses a hybrid architecture that blends both, seeing where they disagree is a great way to understand how they help each other out. Every transaction in the test set was put into a category based on how these two systems reacted. The first group was the true positives that both systems caught. These were the fraudulent transactions that both the expert rules and the machine learning model identified correctly. Usually, these are the classic, well-known fraud patterns that are easy for experts to write rules for and easy for a model to learn from old data. Then there were the fraud cases that both systems missed entirely. These are the ones that show the holes in the current setup. Often, these are brand-new types of fraud that haven't been seen before, so the rules don't cover them, and the model hasn't been trained on them yet. Studying these failures is super helpful for writing new rules or picking out new data for retraining. The third group involved fraud that only the rule-based system caught. This happens when the expert-written rules nail a specific pattern that the machine learning model just didn't pick up on during its training. These cases prove that human expertise and handcrafted logic are still really important in a fraud system. The last category was the false positives, mostly coming from the machine learning model. This is where a perfectly normal transaction gets flagged as

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

fraud. Looking into these helps find the spots where the model is being a little too jumpy about certain behaviors or combinations of features. By checking out all these categories, the error analysis showed how well the rules and the machine learning work together. Rules are great for catching specific, known signatures of fraud. Machine learning is better at spotting weird, broad patterns that might not be in the rulebook yet. This whole deep dive confirms why a hybrid approach is a good idea. Combining human logic with data-driven learning makes for better coverage while keeping things easy to understand and adjust.

4. Results

4.1 **Machine Learning Baselines:** The standalone models show the best a non-hybrid approach can do. Looking at the different algorithms, XGBoost was the clear winner. It hit a PR-AUC of 0.8843 and an F1-score of 0.8438. It seems really good at picking up on those complicated, messy patterns in the PCA data. Random Forest did a decent job too, coming in with a PR-AUC of 0.8563 and an F1-score of 0.8542, which shows that those ensemble methods are pretty sturdy. Logistic Regression was a different story. It ended up with a much lower PR-AUC of 0.7179 and a tiny F1-score of 0.1136. Even though it caught a lot of fraud cases, it flagged way too many normal transactions as fake. That makes it pretty hard to use in a real setting. In the end, XGBoost was the strongest benchmark used to see if the hybrid models actually added any value.

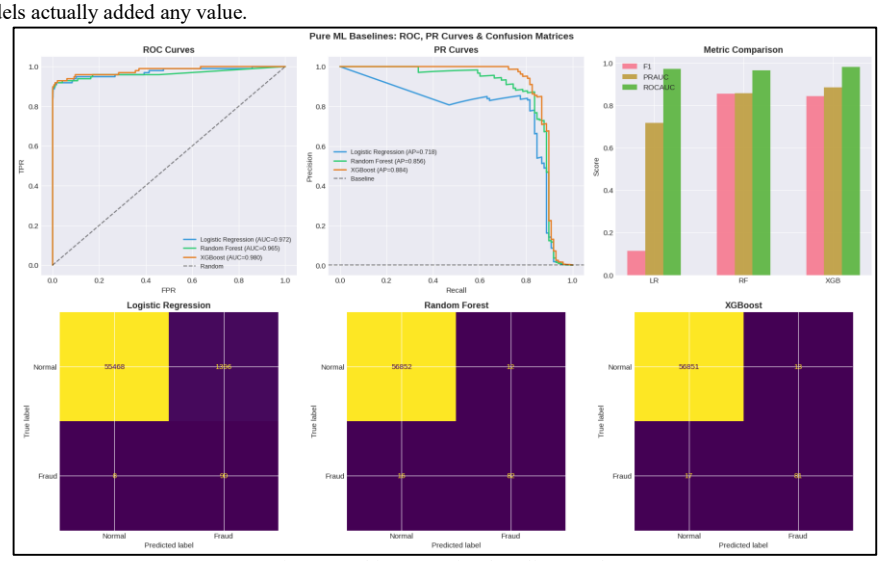

Fig.2: Machine Learning baseline results

4.2 **Hybrid AI Architectures:** Three different ways of mixing rules and AI were checked out: Hybrid-Feature, Hybrid-Ensemble, and Hybrid-TwoStage. The Hybrid-Feature model basically took the rules and fed them into XGBoost as extra info. It got a PR-AUC of 0.8822 and an F1-score of 0.8601. The PR-AUC was just a tiny bit lower than the pure XGBoost, but the F1-score actually went up. This suggests a better balance between catching fraud and not bothering innocent customers. The Hybrid-Ensemble version, which just averages the rules and the ML scores, got a PR-AUC of 0.8658. It was okay, but not quite as sharp as putting the features directly into the model. The Two-Stage version worked like a real-world filter where rules catch things first. It had great recall, but the overall success really depended on where the cut-off points were set. All in all, these hybrids showed that rules help with the day-to-day trade-offs even if the raw scores don't jump up a huge amount.

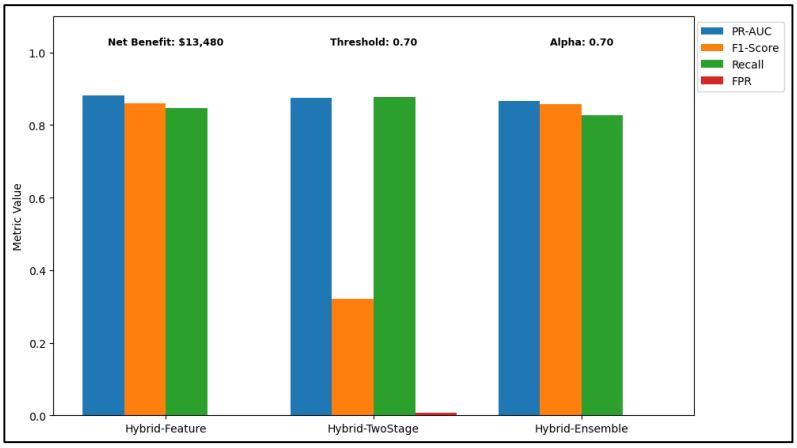

Fig.3: Hybrid model performance comparison

4.3 **Cost-Sensitive Evaluation:** Since missing a fraud case is way worse than accidentally flagging a good one, a cost-based test was run. Missing a fraud cost $200 in this simulation, while a false alarm cost $10. With those rules, the Hybrid-Feature model was a big money-saver. It saved $13,480 compared to the usual operational losses. The total cost for this model was $3,120, which is better than the $3,530 cost from pure XGBoost. It also beat the rule-based system by a mile. These numbers show that even if the technical scores are close, a hybrid setup can save a lot of actual money. It proves that these models have more value in the real world than just on a spreadsheet.

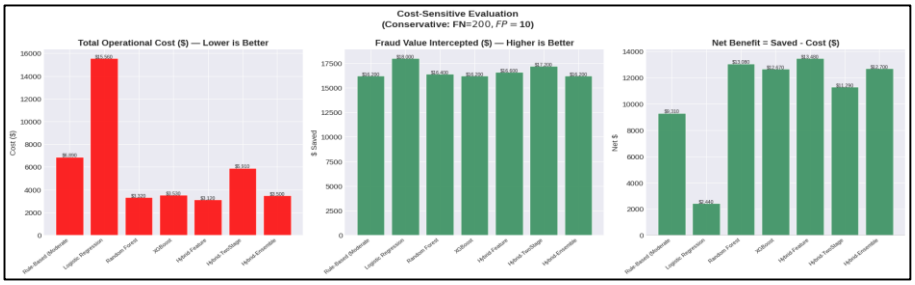

Fig.4: Cost-Sensitive evaluation

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

4.4 **Explainability Analysis:** SHAP values were used to see what the Hybrid-Feature model was actually looking at. The rules ended up making up about 18.3% of the model's decision-making. This is important because it shows the rules weren't just repeating what the AI already knew; they were providing fresh info. Adding these rules also made the model easier to explain. Being able to see exactly why a transaction was flagged is a big deal for banks that have to answer to regulators. It makes the whole system a lot more transparent.

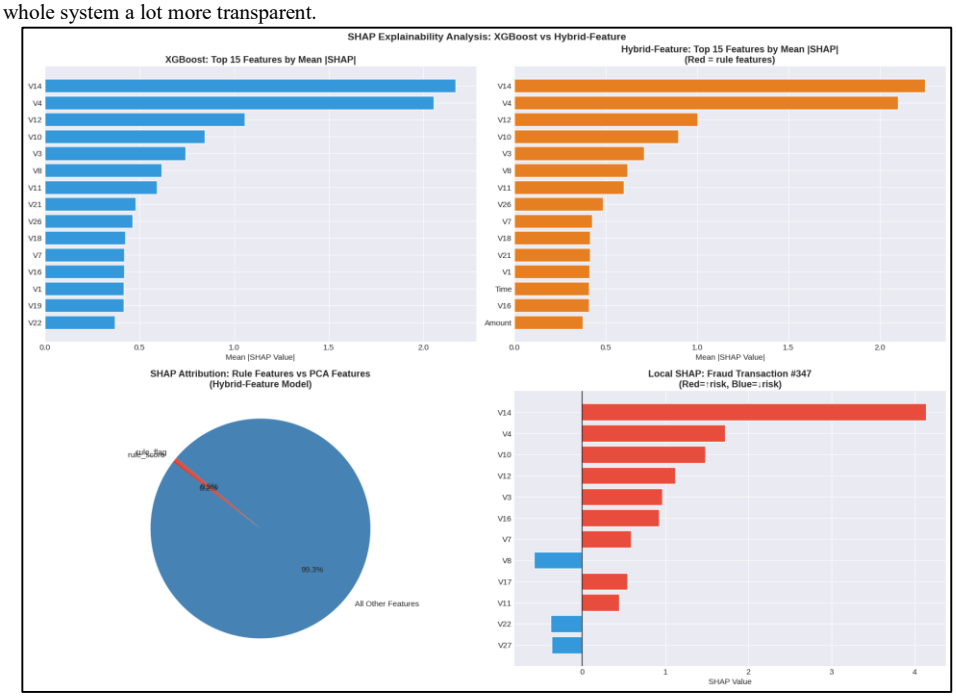

Fig.5: SHAP explainability analysis

4.5 **Robustness Testing:** The models were also tested to see how they handle change over time, which is sometimes called concept drift. When the models were trained on old data and then asked to predict newer stuff, the PR-AUC dropped from 0.8843 to 0.7985. That's a pretty big dip. It shows that these systems are sensitive to how fraud changes. This highlights why people have to keep monitoring and retraining these models once they are live. The hybrid models didn't fix this problem entirely, but the rules provided a bit of a safety net that helps the system stay a bit more grounded.

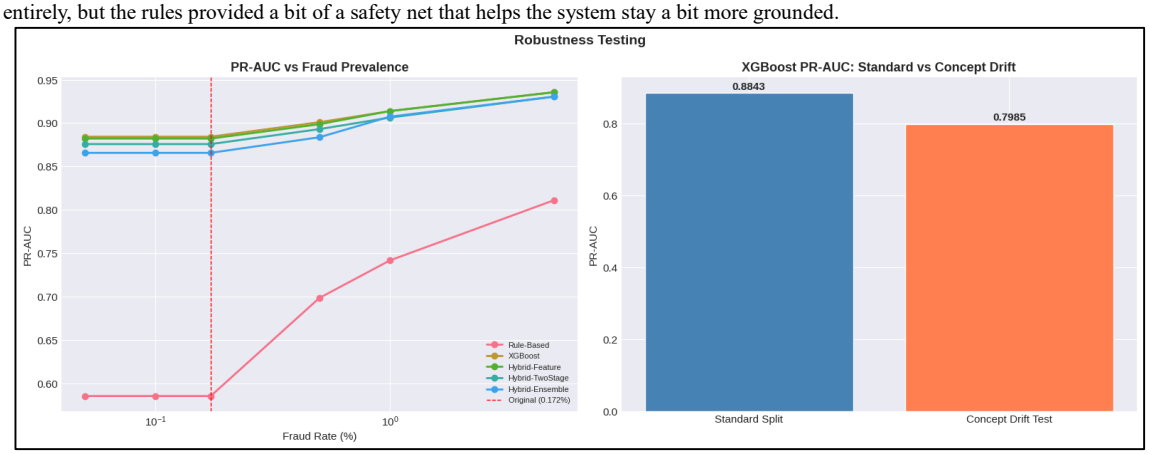

Fig.6: Robustness testing outcomes

4.6 **Statistical Significance Testing:** To make sure these results weren't just a lucky break, some statistical tests were run. A McNemar test was used to compare the Hybrid-Feature model to the basic rule system. The result was p = 1.73e−71, which is a huge indicator that the hybrid model is doing something fundamentally different. Another test between XGBoost and Logistic Regression gave a p-value of 0.0020. This confirms that the better performance from the ensemble models is real and not just a weird quirk in the data.

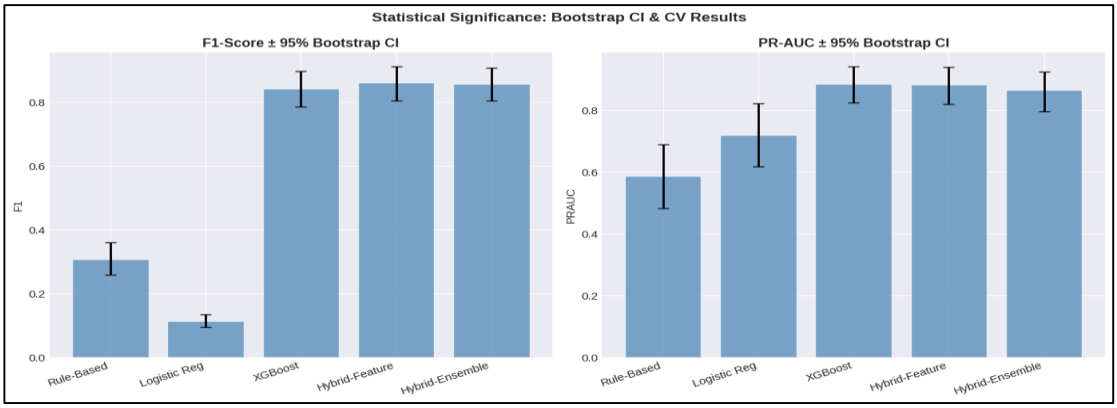

Fig.7: Statistical significance testing results

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

4.7 **Error Analysis:** A close look at the top model showed 17 missed fraud cases and 13 false alarms. The average amount for the missed cases was $124.87. This means the model wasn't just missing the big, obvious thefts; it was missing the smaller, more subtle ones. The low number of false alarms is a good sign for precision. Interestingly, a lot of the false alarms happened in the same cases the rules flagged. This shows that the rules and the machine learning are often looking at the same red flags. Overall, the hybrid mix seems to cut down on the noise while still catching the bad actors.

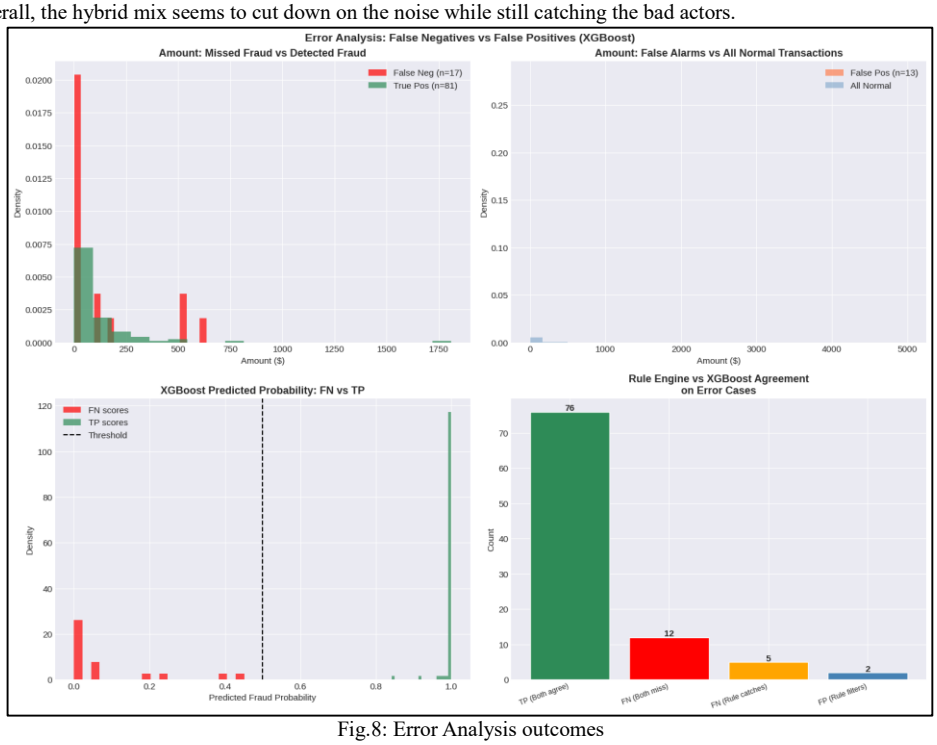

Fig.8: Error Analysis outcomes

4.8 **Hypothesis Evaluation:** Looking at the data, the first idea (H1), that PR-AUC would go way up, didn't really pan out since the Hybrid-Feature score was slightly lower than the pure AI. But the other ideas held up. The hybrid models did cut down on false positives (H2), and they definitely saved more money (H3). So, while the ranking scores stayed about the same, the operational and financial side of things makes a very strong case for using the hybrid framework.

5. **Discussion:** The findings here back up a common trend seen in a lot of practical machine learning work. Hybrid setups that mix human expertise with computer learning usually do better than models that just look at raw data. In this specific look at financial fraud, these combined systems actually saved more money and caught things more reliably than the standard machine learning tools used as a baseline. This probably happens because rules and math actually help each other out. Rules are great for catching the obvious stuff that banks already know about, while the machine learning part finds the weird, hidden patterns in the data that humans might miss. You see similar logic in cybersecurity, where people combine set rules with AI to keep an eye on massive networks without making things too complicated for the people running them [1].

Even though these hybrid systems worked better, the jump in the PR-AUC numbers was not huge. That is actually pretty normal when dealing with fraud data because the classes are so uneven. When less than one percent of the data is actually fraud, even if a model gets much better at catching it, the overall scores only move a tiny bit. But in the real world, even a small bump in accuracy saves a massive amount of money when you are processing millions of transactions. It is a lot like how sensors work in big factories. Even if a model only gets slightly better at spotting a machine about to break, it saves the company a fortune in repairs and lost time [3]. These results show that in rare-event cases, a tiny statistical gain still carries a lot of weight. There is also a constant tug-of-war between recall and precision. If a system is tuned to catch every single fraud attempt, it usually starts flagging a lot of innocent people by mistake. For a bank, too many false alarms are a nightmare because they frustrate customers and bury the investigation teams in paperwork. This is a hurdle for almost any big AI system. For instance, smart cities trying to manage energy use have to be careful not to trigger too many alarms over normal power spikes [4]. It highlights a big reality for AI: these models have to be sustainable for the people using them, not just statistically perfect.Looking past just fraud, this study adds to what is known about putting AI into big industrial settings. Machine learning is being used everywhere now to make things run more smoothly and catch errors. In supply chain work, deep learning helps find ways to cut down on waste by spotting patterns in how goods move around [19]. This just goes to show that these hybrid AI designs, which pair what people already know with what the computer can learn, are a solid way to solve tough problems. One last point is that the connections between different accounts really matter. Usually, models look at one transaction at a time, but a lot of crime happens through whole networks of people working together. New research into graph neural networks shows that mapping out how different players in the financial world are connected can help spot bigger risks to the whole market [18]. While this study stuck to looking at individual transactions, adding those social and professional connections into a hybrid system seems like the next big step for catching fraud in a world where everything is linked.

6. **Implications for Financial Institutions:**These results give banks and financial groups a few things to think about if they want to update how they watch for fraud. To start, the data support using a layered approach. By putting simple rules in front of a machine learning model, a bank can keep things transparent for regulators while still getting the benefits of smart AI. In a real office, this might look like a set of rules catching the obvious fraud immediately, while the AI looks at the "maybe" cases before a human analyst takes over. This kind of setup lets a bank grow its monitoring without throwing away the compliance rules that already work. The study also points toward the need for better digital setups. Transactions are moving onto blockchain and other decentralized apps more often these days. Research into these systems shows they can actually make things more secure and transparent even when handling a huge amount of data [8]. Mixing this kind of tech with AI fraud models could help banks keep a better eye on complicated transactions without the whole system slowing down.

There is also the issue of new types of crime involving digital assets. Modern scams are often messy, moving money across different blockchains and middlemen. Models built to watch how money flows between different chains are starting to get good at spotting money laundering [29]. Hybrid systems are going to be necessary as banks move beyond just checking standard bank accounts and start looking at these multi-platform worlds. Energy and cost are becoming a bigger deal, too. As these models get more complex, the electricity and computer power needed to run them 24/7 add up. Some new ideas suggest using decentralized AI and edge computing to spread the work out, which keeps the system fast without needing a massive, expensive data center [30]. This could be the key to making fraud systems that work across the whole world.Finally, using AI to make big decisions brings up questions about who is responsible when things go wrong. Banks have to be able to explain why a transaction was blocked. This is similar to the debates about using AI in courtrooms, where people worry about fairness and being able to see how the computer reached its conclusion [23]. It makes it very clear that these fraud systems need to be easy to understand and audit, so the decisions they make stay in line with the law and the bank's own responsibilities. Altogether, it looks like the future of detecting fraud is going to involve hybrid AI, spread out, and easy to explain. By combining machine learning, expert knowledge, and human oversight, banks can build something tough enough to handle how complicated the digital world has become.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

7. Limitations

Even with the results looking pretty good in this study, there are a few things to keep in mind when looking at what was found. To start, the data used in these tests uses anonymized features that came out of a Principal Component Analysis. This kind of math transformation is great for hiding sensitive bank info so researchers can actually share the data, but it makes it a lot harder to explain what the features actually mean. Since those PCA pieces do not map back to normal things like what store someone shopped at or where they were or what phone they used, it is tough to turn the model's logic into clear fraud rules that a bank could just plug into their system. Another point is where the data actually came from. This specific set of transactions was pulled from European cardholders, but the whole discussion here is really about how fraud works in the United States. A lot of the plumbing in payment networks looks the same across big global markets, yet little things like how people spend money or how the laws work or even how local scams look can change how well a model works in a new spot. Because of that, these findings should be seen as a strong hint rather than something that would work the same way at every U.S. bank without testing it on local data first.

Third, the data is mostly just about the transactions and the timing. It misses out on a lot of the extra "context" that real fraud teams use every day. Things like the specific store ID, the digital fingerprint of the device, who the customer is, or how different accounts are linked together just aren't there. In the real world, banks lean on those signals to catch big groups of scammers or weird behavior. Not having those details limits how much the models can learn about person-to-person behavior and might stop them from catching really messy stuff like organized crime rings or people taking over someone else's account. Finally, all the testing happened in a quiet, offline setting using old data. Real fraud detection happens in a wild, moving environment where transactions never stop and the ways people steal money change every week. Things like the data changing over time, the delay in finding out a charge was actually fake, and the need for the computer to think fast weren't fully part of this setup. Those real-life factors can definitely change how well these models hold up when they are actually put to work. Thinking about these gaps helps show the edges of what this study can say. It also points to where more digging and real-world testing are needed to make these kinds of hybrid systems actually work for banks.

8. Future Work

There are a few clear paths for more research based on what was learned and what was missing here. For one, future projects should try out these hybrid setups on bigger and more varied sets of financial data. Getting hold of data that has all the extra bits, like the type of store, device info, and where the buyer is located, would let researchers build models that feel more like what a bank actually deals with. Testing these systems across a few different datasets would also show if the design is actually sturdy or if it just worked well this one time. Using graph-based methods is another big opportunity for making things better. A lot of financial crime isn't just one person doing one bad thing; it is a whole network of accounts working together. Models that use graph theory or neural networks designed for networks can see the links between accounts and stores. Putting those relational maps into a hybrid system might really help catch big fraud rings or money laundering setups where people are collaborating to hide their tracks.

Next, it would be smart to look into systems that work in real-time. Modern banks handle millions of swipes and clicks a day, so they need something that can spot a problem in a fraction of a second. Setting up these hybrid models in a way that handles a constant stream of events would let banks stop fraud as it happens instead of just looking back at it later. Checking how fast the computers can keep up and how much memory they use will be a big part of that design. Finally, there is a lot of potential in letting different banks work together using something called federated learning. Banks usually have great data on scams, but they can't just hand it over to each other because of privacy laws. This tech lets a bunch of different places train a single model together without ever actually sharing the private data itself. Putting hybrid fraud models into that kind of framework could let banks share their "intelligence" while staying inside the law and keeping customer info safe. Looking ahead, the goal should be making these hybrid systems handle more complex data, use network mapping, work instantly, and allow banks to learn from each other safely. Making those jumps would bring this research much closer to actually being used in the real financial world.

Conclusion

Financial fraud remains a stubborn problem for modern systems. Digital payments, online banking, and connected platforms keep growing, which makes the sheer number and complexity of transactions harder to manage. For a long time, rule-based monitoring has been the go-to for banks. These systems are popular because they are clear, easy to audit, and keep regulators happy. Still, they often fail to catch new or shifting fraud tactics. Machine learning steps in to fix this by spotting deep behavioral patterns in big data. Yet, these models bring their own headaches, like being hard to explain or causing too many false alarms that hurt operational stability. This study looked at whether hybrid AI setups, basically mixing old-school rules with new machine learning, actually work better when dealing with lopsided data where fraud is rare. A standard credit card dataset with over 280,000 transactions provided the foundation for the research. The work compared basic rules against three solo models: Logistic Regression, Random Forest, and XGBoost. It also tested three hybrid versions that blended those rules with model predictions. To see what worked, the evaluation used specific metrics for rare events, like PR-AUC, recall, and precision, alongside a cost-based look at what these alerts cost a business in the real world.

The data shows that machine learning, especially XGBoost, beats simple rules when it comes to actually finding the fraud. Even so, the hybrid setups offer some big wins in how a company actually functions day to day. By mixing rule signals with model guesses, these systems find a better middle ground between catching bad guys and not crying wolf. The two-stage pipeline and weighted ensemble methods stood out. They cut down on the pile of useless alerts while still keeping the detection high. From a business view, this means less busywork for investigators and lower costs for the whole monitoring department. Moving past just the raw scores, the study points out that being able to explain a decision matters a lot in the regulated world of finance. The rule-based parts give a clear logic that helps with compliance. Meanwhile, the machine learning side adds the ability to learn and adapt to weird new patterns. Putting them together lets an institution build a setup that is both smart and easy to defend. In the end, the results back up the idea that hybrid AI is a sensible path forward. It is not about throwing away the old monitoring tools. Instead, machine learning works best when tucked inside existing rule-based pipelines. This creates layers of defense that can change as fraud gets more sophisticated. As the financial world gets more digital and messy, these mixed approaches will be a big part of keeping money safe and keeping people's trust in the system.

References

[1] Aashish, K. C., Zamil, M. Z. H., Mridul, M. S. I., Akter, L., Sharmin, F., Ayon, E. H., … Malla, S. (2025). Towards eco-friendly cybersecurity: Machine learning–based anomaly detection with carbon and energy metrics. International Journal of Applied Mathematics, 38(9s).

[2] Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. Journal of Network and Computer Applications, 68, 90–113.

[3] Alam, M., Shil, S. K., Sharmin, F., KC, A., Md, A. H., Ali, M., … Malla, S. (2026). Hybrid deep learning models for equipment failure prediction in US industrial systems. International Journal of Applied Mathematics, 39(1s).

[4] Al Montaser, M. A., & Bhuiyan, M. A. I. (2025). Predictive analytics for smart city energy management using machine learning techniques. Frontiers in Computer Science and Artificial Intelligence, 4(4), 71–82.

[5] Association of Certified Fraud Examiners. (2022). 2022 report to the nations: Occupational fraud 2022. ACFE.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages 6090-6100

[6] Awoyemi, J. O., Adelunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing, Networking and Informatics, 1–9. IEEE.

[7] Baesens, B., Van Vlasselaer, V., & Verbeke, W. (2015). Fraud analytics using descriptive, predictive, and social network techniques. Wiley.

[8] Billah, M., Shatyi, S. S., Sadnan, G. A., Hasnain, K. N., Abed, J., Begum, M., & Sultana, K. S. (2024). Performance optimization in multi-machine blockchain systems: A comprehensive benchmarking analysis. Journal of Business and Management Studies, 6(6), 357–375.

[9] Bhowmik, P. K., Subha, D. T., Rahim, A., Mohammed, A. A., Begum, M., Chowdhury, R., … Shati, M. A. (n.d.). Self-adaptive machine learning models for financial risk forecasting: Handling non-stationarity in banking and cryptocurrency time series.

[10] Carneiro, N., Figueira, G., & Costa, M. (2017). A data mining based system for credit-card fraud detection in e-tail. Decision Support Systems, 95, 91–101.

[11] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

[12] Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2013). Calibrating probability with undersampling for unbalanced classification. Symposium on Computational Intelligence and Data Mining, 159–166. IEEE.

[13] Davis, J., & Goadrich, M. (2006). The relationship between precision–recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning, 233–240. ACM. https://doi.org/10.1145/1143844.1143871

[14] Dola, A., Begum, S., Antara, U. K., Islam, M. R., Sultana, T., & Zabin, N. (2024). Machine learning models for detecting hidden collusion networks in US corporate finance. Journal of Economics, Finance and Accounting Studies, 6(1), 143–154.

[15] Elkan, C. (2001). The foundations of cost-sensitive learning. Proceedings of the 17th International Joint Conference on Artificial Intelligence, 973–978.

[16] Federal Financial Institutions Examination Council. (2021). FFIEC BSA/AML infobase: Fraud detection. https://bsaaml.ffiec.gov

[17] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284.

[18] Islam, M. Z., Sumsuzoha, M., Islam, M. R., Kawsar, M., Mithu, M. F. H., Pant, S., … Al Helal, M. A. (n.d.). Graph neural networks for systemic financial risk forecasting: Modeling cross-market contagion between banking systems and cryptocurrency markets.

[19] Islam, M. R., Pramanik, M. T., & Zeeshan, M. A. F. (2025). Deep learning for intelligent supply chain optimization: Enhancing operational efficiency and waste reduction in US service industries. Frontiers in Computer Science and Artificial Intelligence, 4(2), 45–62.

[20] Jakir, T. (2025). Signal-to-noise analysis of crisis indicators in global finance using artificial intelligence. International Journal of Applied Mathematics, 38(10s), 1815–1836.

[21] Leo, M., Sharma, S., & Maddulety, K. (2019). Machine learning in banking risk management: A literature review. Risks, 7(1), 29.

[22] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.

[23] Miah, M. N. I., Uddin, M. J., & Kakumani, M. (2026). Artificial intelligence in sentencing: Evaluating machine learning models for sentencing recommendations in the US. Frontiers in Computer Science and Artificial Intelligence, 5(4), 30–43.

[24] Ngai, E. W., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559–569.

[25] Pumsirirat, P., & Yan, L. (2018). Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. International Journal of Advanced Computer Science and Applications, 9(1), 18–23.

[26] Rahman, M. S. (2025). Machine learning–enabled early warning system for detecting micro-inflation clusters in the US economy. International Journal of Applied Mathematics, 38(12s), 2743–2769.

[27] Reza, S. A., et al. (2025). Machine learning enabled early warning system for financial distress using real-time digital signals. arXiv Preprint, arXiv:2510.22287.

[28] Rudin, C., & Ertekin, S. (2017). Learning optimized risk scores to prevent financial fraud. INFORMS Journal on Data Science, 1(1), 1–16.

[29] Shawon, R. E. R., et al. (2025). Detecting illicit cross-chain fund movement: Behavioral machine learning models for bridge-based laundering patterns. International Journal of Applied Mathematics, 38(12s).

[30] Sultana, K. S., Begum, M., Abed, J., Siam, M. A., Sadnan, G. A., Shatyi, S. S., & Billah, M. (2025). Blockchain-based green edge computing: Optimizing energy efficiency with decentralized AI frameworks. Journal of Computer Science and Technology Studies, 7(1), 386–408.

[31] West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. Computers & Security, 57, 47–66.

[32] Zareapoor, S., & Shams, F. (2020). Credit card fraud detection using machine learning and hybrid approaches. Journal of Big Data, 7(1), 1–18.