Flagging Payments for Fraud Detection: A Strategic Agent-Based Model

Katherine Mayo, Shaily Fozdar, Michael P. Wellman

University of Michigan
{kamayo, sfozdar, wellman}@umich.edu

Abstract

Credit card fraud occurs when a malicious party, the fraudster, makes unauthorized charges on a credit card. Fraud can result in large losses, leading banks to adopt fraud detectors. The use of these detectors can be costly; thus banks must be strategic in deciding which payments to flag for review by the detector. The *flagging problem* is how to balance costs of the detector with potential losses from undetected fraud. To study this problem, we introduce a flagging game played by bank and fraudster nodes in a financial credit network. Our model includes two types of banks, strong and weak, characterized by the accuracy of their fraud detectors. Bank nodes select a strategy that determines the probability each payment is flagged and sent to the detector based on its various attributes. Fraudster nodes select a strategy that sets the frequency and value of attempted fraudulent payments. We analyze this flagging game using empirical game-theoretic analysis to identify strategic equilibria under various cost configurations. We find that increasing costs of using the detector affects only strong bank nodes, who flag fewer payments when costs are high. Our analysis shows that the costs for weak bank nodes are primarily from the liability for fraud committed, while the cost of fraud detection plays a more important role for strong bank nodes. We find the number of victims of attempted fraud slightly increases when all payments are flagged compared to the strategic flagging case.

1 Introduction

Credit card fraud occurs when a malicious party, the ***fraudster***, makes unauthorized purchases using a credit card that does not belong to them. The fraudster may obtain the card's details in a variety of ways such as physically stealing the card, or through methods such as electronic database breeches. In 2020, $28.58 billion worldwide was lost to payment card fraud, which includes credit card fraud (Nilson Report 2021). Losses are expected to increase in the future with projected losses of $39.89 billion in 2025.

The majority of fraud losses are borne by card issuers. As a result, many banks have developed systems for catching fraudulent charges, called ***fraud detectors***. Much current research focuses on the development and improvement of these detectors. Our work complements this by seeking to understand the decision context and broader effects of fraud detection on the payment system. Specifically, we note that fraud detection often carries a cost for human and/or technical capital. Therefore, we are interested in studying how such costs might impact the adoption of fraud detector technology and the resulting effect on patterns of fraud in the network.

We introduce payment ***flagging*** as a bank action sending a credit card payment to the fraud detector. Thus, the bank's ***flagging problem*** is, in an environment in which invoking the fraud detector is costly, to decide which payments to send to the detector. To investigate this problem, we develop an agent-based model where fraudsters and customers route payments through a banking network. Each bank node in our model has access to its own fraud detector, which we treat as a black box defined only by its accuracy and cost characteristics. We refer to bank nodes with high fraud detection accuracy as strong bank nodes and those with lower fraud detection accuracy as weak bank nodes. Using the model, we define a ***flagging game*** played by strategic bank and fraudster nodes. Bank nodes decide which payments to flag, trading off costs of missing fraud and using the fraud detector. Fraudster nodes decide the value of their payments and how often to attempt fraud. We analyze the resulting game under several cost configurations using ***empirical game theoretic analysis*** (EGTA) (Tuyls et al. 2020; Wellman 2016), a method for identifying equilibrium configurations among heuristic strategies using agent-based simulation.

From our experiments, we find that high costs for using the fraud detector deter strong bank nodes from flagging many payments, but this is not the case for weak bank nodes. This aligns with our analysis of the costs of fraudulent payments in the environment. For strong bank nodes, these costs are more balanced between liability costs and costs of fraud detection than the costs for weak bank nodes, whose primary costs are the losses from liability for fraud. When compared to the scenario where banks flag all payments, we find customer nodes may be worse off due to an increased number of customer nodes experiencing impersonation attempts by the fraudster node.

2 Related Works

Previous work in the fraud detection space has focused mainly on designing systems for detecting fraud more quickly and accurately than human auditors. Methods for fraud detection have evolved from rule-based to statistical data mining and machine learning based techniques. Both supervised and unsupervised models have been used with supervised methods making use of labeled data sets and unsupervised methods relying on techniques such as anomaly detection (Ryman-Tubb, Krause, and Garn 2018; West and Bhattacharya 2016). Recent literature focuses on leveraging the power of deep neural networks for detecting fraud (Lin et al. 2021; Zheng et al. 2020). To our knowledge, such works do not incorporate possible constraints that might limit the use of these systems.

In contrast, Dervovic et al. (2021), introduces the constraint of resources problem to fraud detection by enforcing a constraint on the number of payments reviewed. They investigate the problem of selecting a subset of jobs with random arrival times, in which the decision to select a job must be made immediately. Each job carries a reward for being selected with the goal to maximize the total reward received. The authors propose the Non-parametric Sequential Allocation Algorithm and test its effectiveness at selecting the most valuable fraudulent transactions using a public fraud data set.

Prior work has used the credit network model to study topics ranging from auctions (Ghosh et al. 2007) to liquidity (Dandekar, Goel, and Govindan 2011; Dandekar et al. 2015). The model uses a directed graph of nodes connected by weighted, directed edges representing the capacity for connected nodes to transact with one another. The model was extended by Cheng et al. (2016) to include interest rates creating the financial credit model. We further extend this model to include fraud edges to model credit card fraud in the network.

3 Payment Model

3.1 Financial Credit Networks

The interactions between card-issuing banks and customers are modeled using a graph based on the ***financial credit network*** formalism (Cheng et al. 2016). The network contains a set of nodes $C = \{1, \dots, n\}$ representing customers and a set of nodes $B = \{1, \dots, b\}$ representing banks, with $b \ll n$. The nodes in the network are connected by a set of weighted, directed edges E. Edge $(i, j, debt, label, v) \in E$ represents node i owing value v to node j. Debt edges representing payments also include a boolean variable $label$ denoting whether the payment is fraudulent ($label = True$) or not ($label = False$). Edge $(i, j, credit, v) \in E$ represents a node i extending a line of credit with value v to node j. Such edges exist in our network only when $i \in B$ and $j \in C$ or $i, j \in B$. Thus, credit can be extended only by bank nodes. We refer to the total debt owed by node i to node j as $d_{ij} = \sum_{(i,j,debt,label,v)} v$ and the total credit node i extends to node j as $c_{ij} = \sum_{(i,j,credit,v)} v$.

Banks extend credit to customers in the form of credit limits on a credit card, which customers use to make payments within the network as shown in Figure 1. In the first image, bank node B_1 has extended a line of credit to customer node C_1 of 100 units, denoted by the solid red edge. A similar relationship exists between bank node B_2 and customer node

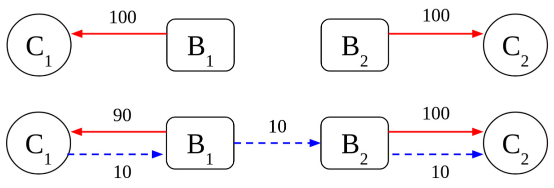

Figure 1: Customer node C_1 draws on its line of credit to make a payment of 10 units to another customer node in the network, C_2.

C_2. Customer node C_1 then draws on its credit line to make a payment to another node in the network, C_2. The edges in the figure show only the total debt, d_{ij}, and total credit, c_{ij}, values for compactness. As a result of drawing on its credit line, C_1 now owes a debt of 10 units to B_1 represented by the dashed blue edge from C_1 to B_1. The amount of credit available to C_1 to make additional payments in the network decreases by the value of the payment as captured by the value on the solid red credit edge from B_1 to C_1 decreasing to $c_{B_1C_1} = 90$ units. To route payments for their customers, bank nodes use a series of credit and debt edges connecting all bank nodes to one another referred to as the ***interbank network***. We assume bank nodes in our model have an infinite willingness to route payments for their customer nodes and set the credit lines between bank nodes to an arbitrarily large value. For simplicity, these credit edges are left out of Figure 1. Bank node B_1 routes the payment through the interbank network to the payment receiver's bank, B_2, creating a debt between the banks equal to the value of the payment. Finally, C_2 receives the payment as represented by the debt edge from B_2 to C_2 with value $d_{B_2C_2} = 10$ units.

3.2 Fraudster

Our network also contains a set of nodes $F = \{1, \dots, m\}$ representing the fraudsters. We refer to a fraudster node's use of a customer node's credit line as ***impersonating*** the customer node. The addition of a fraudster node adds a new type of non-directed edge to the network $(i, j, fraud, v)$ where $i \in F$ and $j \in C$. This edge denotes that fraudster node i is impersonating customer node j and using j's credit line to make a payment of v units to some other customer node in the network. We use $f_{ij} = \sum_{(i,j,fraud,v)} v$ to represent the total value of fraud committed by i using j's credit line.

We model the fraudster nodes in our network as shown in Figure 2. Bank node B_1 has extended customer node C_1 100 units of credit, which C_1 has not yet used. The impersonation of C_1 by the fraudster node is represented by the dotted black edge between fraudster node F_1 and customer node C_1. When the fraudster node F_1 impersonates customer node C_1 to make a payment to node C_2, we represent the impersonation with a value of 10 units on the dotted black edge between F_1 and C_1. Since the payment value and receiver are the same as in Figure 1, the remaining changes to the network appear the same. However, the debt edges added to the network from C_1 to B_1 and from B_1 to B_2 have $label = True$, since this payment originated with the

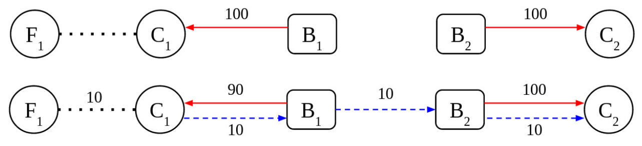

Figure 2: Fraudster node F_1 uses customer node C_1's credit line to make a payment of 10 units to customer node C_2.

fraudster node, not a customer node. Again, the edge values in the figure show the total debt, credit, and fraud values as in earlier examples and we omit the interbank credit edges for simplicity.

4 The Flagging Game

We introduce our flagging game, which is played over T time steps by banks and fraudsters in a financial credit network. Fraudsters aim to maximize successful fraud volume by selecting a strategy defining the frequency and value of fraudulent payments attempted. Banks process credit card payments on behalf of their customers, while addressing the flagging problem. To do so, they select from a set of strategies that determine the probability a given payment is flagged, seeking to balance the cost of using the fraud detector and the cost of missing a fraudulent payment.

4.1 Initialization

The game begins with the initialization of bank, customer, and fraudster nodes. As part of initialization, banks and fraudsters select their strategies from the sets defined in Section 4.4.

Bank Nodes Bank nodes are assigned a type, either strong or weak, which dictates the bank node's fraud detection capabilities. Let γ_i denote the probability that bank node B_i's fraud detector accurately labels a payment. That is, if a payment is truly fraudulent, B_i's fraud detector would label it fraudulent with probability γ_i, and as non-fraudulent with probability $1 - \gamma_i$. True non-fraudulent payments are likewise labeled as such with probability γ_i. These values are drawn for each bank node on initialization, with $\gamma_i \sim U[0.75, 0.9]$ if B_i is a strong bank node and if it is a weak bank node, $\gamma_i \sim U[0.5, 0.65]$.

Customer Nodes We calibrate the payment behavior of our customer nodes using the JP Morgan AI Research (JPMAIR) payments data for fraud detection synthetic data set. The data set contains each payment's sender, receiver, value, and a variety of other variables for both fraudulent and non-fraudulent payments. For our model, we use only the non-fraudulent payments in the data set, leaving us with a set of 47,570 transactions.

We randomly select a subset of customers from the JPMAIR data set and associate them with customer nodes in our network by *datasetID*. For example customer node C_5 in our network may have *datasetID* = 127352 used only for interaction with the data set. We then set a payment frequency parameter, λ, for each customer node calculated from the behavior of their assigned JPMAIR data set customer. Customers nodes are also assigned to a bank node uniformly at random. Customer node C_j is extended a line of credit from its assigned bank node B_i, creating the edge $(B_i, C_j, \mathit{credit}, v = 2P)$, where P is the total value of payments C_j's *datasetID* sends in the JPMAIR data set.

Fraudster Nodes Fraudster nodes are initialized under the assumption they could potentially compromise the credit card information of any customer node and discover their bank node assignments. Additionally, we assume the means by which the fraudster obtains the credit card details does not deter the customer from continuing to use their card while undetected fraudulent transactions are made. Fraudster nodes initially select a customer to impersonate uniformly at random.

Fraudster nodes are particularly inclined to use their stolen credit card information to make payments to certain customer nodes. We refer to this set as the ***suspicious set***, TS, and initialize it by uniformly randomly selecting from all customer nodes in the network. The size of TS is also uniformly randomly selected so that $|TS| \sim U\{10, \ldots, 20\}$. While the true identity of the suspicious set is unknown to the banks, we assume they know that such a group exists and have a noisy estimate of its size. Bank nodes maintain their own estimates of the list, which we we call a bank node's ***perceived set***, PS. The size of the perceived set of bank node B_i is uniformly randomly drawn so that $|PS_i| \sim U\{|TS|+1, \ldots, |TS|+5\}$. The accuracy of a bank node's perceived list is based on the bank node's type. For each customer node in TS, a strong bank node B_i adds the customer node to its set PS_i with probability 0.9 and a weak bank node adds it with probability 0.6. The remaining spots in the set are filled by a random draw of customer nodes in the network who are not members of TS.

Payment Queue Lastly, we initialize the ***payment queue***, which is used to track the schedule of payments to be attempted in the network. Each slot in the queue is equivalent to one time step in the game, thus the queue length is T. The T time steps are comprised of alternating ***active*** and ***non-active*** time periods, each of length A. *Active* time periods are those in which the fraudsters are more likely to be active, as described in Section 4.2 below. To initialize the queue, we calculate each customer node's initial arrival time using its frequency λ and place them in the queue at the time of first payment. Finally, we calculate the initial arrival time of each fraudster node according to their selected strategy and place them in the queue.

4.2 The Game

After initialization, the game proceeds with each time step t consisting of the same set of steps: the next payment sender is removed from the queue, a payment is created, the sender's bank node uses its strategy to make the flagging decision, the payment is processed, and the sender node is placed back in the queue for its next payment.

Customer Nodes We create customer node payments in the network using the JPMAIR data set. For a customer node, C_i, a payment sent by C_i's *datasetID* is randomly selected from the data set. The true network payment value is drawn from a normal distribution with μ equal to the value of this data set payment and $\sigma = 10$. The receiver of the data set payment is not guaranteed to match a *datasetID* of a customer node in our network, so we use a hashing function to map the receiver's ID to the index of a customer node in our network. Specifically, the receiver in the network has an index equal to $datasetID \mod n$, where n is the number of customer nodes in the network. For example, if the data set has the payment received by a customer with the ID 128372, the payment is routed to C_{172} in our network.

The payment is then processed by the customer node's bank node. The processing of a payment by bank node B_i for its customer node C_i is completed as a series of steps.

1. If the payment's value is larger than C_i's total available credit ($v > c_{B_iC_i}$), the payment attempt is terminated and processing is complete.
2. B_i uses its strategy to determine the probability ρ the payment is flagged.
3. The payment is sent to the detector with probability ρ and with probability $1 - \rho$ processing continues.
   - If sent to the detector, the payment is correctly labeled as fraudulent or not with probability γ_i.
4. If the payment is:
   - not sent to the detector or labeled not fraudulent by the detector, the payment is added to the network as described in Section 3.
   - labeled as fraudulent, the payment attempt is terminated regardless of the true label.

The next arrival time of C_i is calculated as for initialization and C_i placed back in the payment queue.

Fraudster Nodes When a fraudster node attempts a payment, the value is determined from its selected strategy. The strategy defines an interval, as described in Section 4.4, from which the payment's value is chosen uniformly. The receiver of the payment is selected from the customer nodes in the network such that with probability 0.8 the receiver is in the *suspicious set* and with probability 0.2 the receiver is some other customer node in the network. The attempted payment is then processed as described for customer node payments above.

The next arrival time, a, of the fraudster node is calculated subject to its selected strategy, as well as the designation of the time period a falls in. If a is calculated to be in a *non-active* time period, then with a low probability (0.2) we schedule the fraudster to arrive at a, and with remaining probability (0.8) we calculate a new a. This process continues until we find a time in which we can schedule the fraudster node in the queue. When a falls in an *active* time period, the fraudster is always placed in the queue at a.

A fraudster node will select a new customer node to impersonate only when the fraudster node's previously attempted payment is terminated by a bank node for any reason. When selecting a new customer to impersonate, fraudster nodes prefer to target bank nodes where they have historically been most successful, but are indifferent between customer nodes of such a bank. Therefore with 0.7 probability the fraudster node selects a random customer node of the bank node where it has committed the most fraud, and with probability 0.3 selects another customer node. For each new customer node, the fraudster node chooses a new frequency with which to make payments in accordance with its selected strategy.

4.3 Payoffs

After T time steps, the fraudster and bank nodes receive payoffs based on the performance of their selected strategies. Fraudster nodes receive payoffs equal to the total value of fraud they commit in the network. Fraudster node F_i's payoff is equal to:

$$payoff_{F_i} = \sum_j f_{F_ij}$$

The payoff to bank nodes is the sum of costs they are subject to given the presence of fraudster nodes in the network with the best strategy minimizing these costs. Since bank nodes are liable for fraud that occurs, the first cost is the sum of the value of B_i's undetected fraudulent payments, which we write $FC = \sum_{j,j \in C} \sum_{(j,B_i,debt,label=True,v)} v$. The second cost to bank nodes is attributed to employing fraud detection measures. This can be broken into two subcosts, the first of which is the cost of sending a payment through the fraud detector. We model this as a flat fee for each payment, β, that represents costs such as paying employees or paying for the use of computer servers. Fraud detection technology is also imperfect, causing the side effect of inaccurately labeling a payment as fraudulent. These false positives can create costs for credit card issuers in the form of lost transaction fees and inconvenience to customers that may translate to a loss of business for the bank. We define α_1 to represent the lost transaction fee, a percentage of the payment's value the bank node would have received if it had not terminated the payment. The inconvenience to customer nodes is modeled as a flat cost per terminated payment, α_2. Thus, for a bank node B_i, the payoff for its selected strategy is:

$$payoff_{B_i} = -FC - \alpha_1 VFP - \alpha_2 NFP - \beta NFD, \quad (1)$$

where VFP is the total value of false positive payments, NFP is the number of false positive payments, and NFD is the number of payments B_i flagged.

4.4 Strategies

Fraudster Node Strategies The goal of fraudster nodes in the network is to maximize the value of fraud they can commit. To achieve this, a fraudster node may take actions to evade the fraud detectors or may attempt to take advantage of imperfect fraud detectors such as by making many payment attempts to increase the odds of success. We model such ideas as 8 strategies, each comprised of two components: payment frequency and payment value. Each of these components is defined by an interval of possible values from which the fraudster node's behavior is drawn as described in Section 4. The payment frequency can be either *low* or *high*. The payment values are drawn from an interval considered:

- *low*: an interval with low values
- *high*: an interval with high values
- *random*: an interval over the combined low and high values
- *alternate*: alternates between a payment from the *low* interval and a payment from the *high* interval

We refer to the fraudster node strategies in the format *[frequency, value]*. For example *[low, high]* refers to the strategy with payment attempts low in frequency and high in value.

Bank Node Strategies The strategic decision for a bank node in our network is which payments to flag when invoking the fraud detector is costly. We consider the flagging process a quick review of payments and thus do not use customer-specific information such as the sender's average behavior which is often used for a more in-depth analysis by the fraud detector. The strategies of the banks nodes are modeled as logistic functions in the form of:

$$\rho = \frac{1}{1 + e^{-k(v-(cx))}}, \tag{2}$$

where v is the value of the payment being evaluated and k is a game configuration. The logistic function returns ρ, the probability the payment is flagged.

When determining whether a payment may be flagged for review, bank nodes consider 3 aspects of the payment: its value, the receiver, and the timing of the payment. If a payment's value is high, the bank node may prefer it be reviewed by the fraud detector, since bank nodes are liable for fraud that occurs. Consider $c = 1$ in Equation 2 for now. The value of x in the equation determines the tipping point at which payments are considered high and are more likely to be flagged. We consider 3 possible ways to calculate the value of x:

- *500*: the average of possible payment values in our model
- *pay_avg*: the average payment value of all customer nodes of the bank node
- *credit_est*: an estimated payment value calculated using the average ratio of payment value to initial credit line

The second part of the bank node's strategy focuses on the receiver and timing of the payment. Consider a payment sent at time t to customer node C_j by a customer of bank node B_i. The bank node considers the receiver of the payment to be suspicious if C_j is in B_i's perceived set ($C_j \in PS_i$). If the time in which the payment was sent is an *active* period when fraudsters are more likely to attempt payments, the bank node may also have reason for additional scrutiny. While any one of these conditions many trigger the bank node to favor review of the payment, if any combination or all 3 are present, bank nodes may be more likely to want to flag the payment. This notion forms the ***suspicion matrix***, which assigns a different c value for different combinations of these suspicious events. As a result, the logistic function is shifted such that if, for example, a payment is of high value and the receiver is suspicious there is a greater change of the payment going to the detector than if the payment value was merely high.

Our strategies include 4 such matrices, each characterized by the general behavior in the presence of suspicious conditions.

- *most*: sends the majority of payments to the detector solely based on the payment's value
- *reg*: as more suspicious conditions are present, it becomes increasingly likely a payment goes to the detector
- *rare_sus*: sends only payments with a high degree of suspicion to the detector
- *rare*: has a low probability of sending any payment to the detector and only uses the payment's value

To understand the difference in these strategies, consider a payment of 800 units sent to a suspicious receiver during an *active* period. The *most*, *reg*, and *rare_sus* matrices return a very high probability of flagging this payment, while the *rare* matrix returns a very low probability. Now consider a similar payment of 800 units sent to some non-suspicious receiver, during a *non-active* period. The *most* matrix is still very likely to flag the payment and the *rare* is still very unlikely to flag the payment. However, with only a suspiciously high payment value, the *reg* matrix flags this payment with a probability less than in the original example payment. The absence of other suspicion markers results in the *rare_sus* matrix returning a low probability for flagging this payment. Finally, if the payment was sent to a suspicious receiver *or* during an *active* period, only the *reg* matrix's result would change. With only one suspicious condition true, the payment would be sent to the detector with a likelihood between the outcomes in the two previous examples.

This creates a total of 12 bank strategies, which we refer to in the form *[x, matrix]*. For example, the strategy *[500, reg]* refers to the strategy in which a bank node calculates the probability a payment is flagged by considering all suspicious conditions that may occur and considers a payment value high, absent other information, if it is greater than 500.

5 Empirical Game-Theoretic Analysis

We analyze the flagging game using EGTA, a method that extensively simulates various strategy combinations called ***strategy profiles***, to identify Nash equilibria of a game. Each ***strategy profile*** specifies the number of players employing each strategy in the game. We test strategy profiles by first assigning nodes in our network to play a strategy such that the total number of nodes assigned to each strategy matches the given strategy profile. Nodes play the game using their assigned strategy and payoffs are calculated for each strategy profile as described in Section 4. The payoff to nodes for playing a strategy of a given strategy profile is calculated as the sample average of payoffs observed over many simulation runs of the profile in the game. EGTA selects strategy profiles to simulate using an iterative procedure with the goal of identifying symmetric mixed-strategy Nash equilibrium as done in previous studies (Cassell and Wellman 2013; Wellman, Kim, and Duong 2013)

We perform our experiments on games consisting of $n = 200$ customer nodes, $b = 4$ bank nodes (2 strong, 2 weak), and $m = 1$ fraudster node. The game is played over $T =$

4, 320 time steps, broken into *active* and *non-active* time periods of length $A = 72$. For the bank node strategy's logistic function, we use $k = 0.002$. We use a value of $\alpha_1 = 0.01$ for the transaction fee bank nodes obtain for routing payments. We test 9 configurations of the flagging game defined by $\alpha_2 = \{0.5, 1, 2\}$ and $\beta = \{0.1, 1, 2\}$.

Our results show the fraudster node selects the strategy *[high, high]*, which results in frequent high value payment attempts, in all game configurations. This suggests that in this environment, the best strategy for the fraudster node is not to try and disguise its payment attempts, but to take advantage of the inability for all payments to be detected. It makes playing these odds worth while with large payment values.

Bank nodes in our game tend to prefer to use strategies with x set to *500* in all game configurations. Thus, absent other information, bank nodes find that payments with values larger than 500 should be more likely to be flagged. The prevalence of this indicates that knowledge of the possible payment values in the network alone may be enough to judge if a payment value warrants suspicion. It may be that the other methods that measure against average behavior are less useful, as they are based on historical information rather than present behaviors.

For suspicion matrices, we find that weak bank nodes prefer the *most* matrix in all configurations, while the strong bank nodes prefer to use the *most* matrix only when $\beta < 2$. Thus, when costs are lower, all bank nodes flag most payments for review by the detectors. However, when costs are high, strong bank nodes switch to preferring the *rare_sus* matrix, which requires a payment's value, receiver, and timing to all be suspicious for the payment to have a high probability of being flagged. In this case, strong bank nodes are willing to use their detectors, but in a limited capacity.

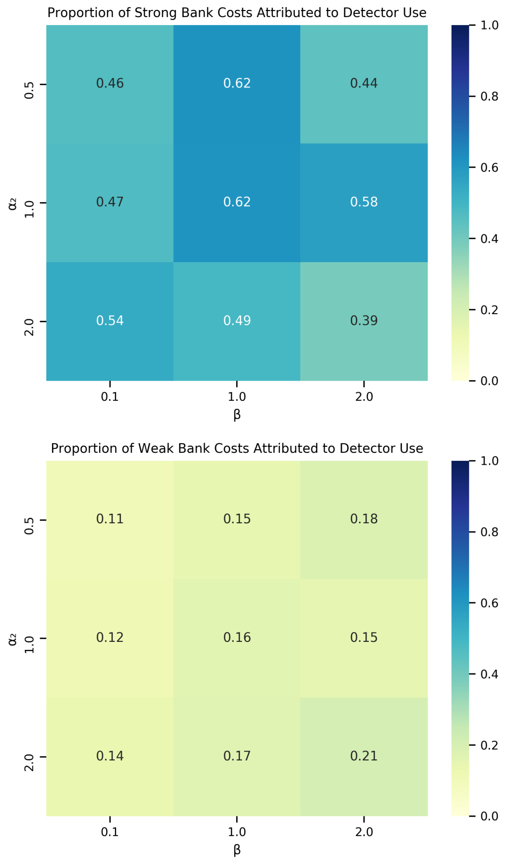

Figure 3: The costs to weak bank nodes of fraud are predominantly due to missing fraudulent payments, while the costs to strong bank nodes are more balanced.

6 Equilibrium Analysis

To understand the effect strategic flagging has on the network, we analyze outcomes when nodes play according to the equilibria. In the event of a mixed strategy equilibrium, nodes are randomly assigned to play a pure strategy by a weighted draw according to the probability distribution defined by the equilibrium. Outcomes are averaged over 100 runs of the game and in the event there are multiple equilibria of a game configuration, the results are also averaged across equilibria. To aid our analysis, we also compare outcomes to the case where bank nodes flag all payments and the fraudster node plays according to the equilibrium strategy *[high, high]*.

We find the fraudster node in our game has a generally low success rate of 35% in committing fraud, but is still able to commit on average 23,083 units of fraud. This is attributed to the fraudster's strategy of attempting frequent, high value payments. When banks send all payments to the detector, the success of the fraudster is slightly lower with an average of 25% and the fraudster commits an average of 18,051 units of fraud.

The existence of fraudster nodes imposes two costs on the bank nodes: liability for fraudulent payments and costs incurred for adopting fraud detectors. In Figure 3, we analyze the proportion of total costs (Equation 1) attributed to fraud liability, FC in Equation 1, for each bank type. We find that weak bank nodes in our game incur the majority of their costs from fraud liability. This aligns with the their strategic choice to flag the majority of payments, as the liability from fraudulent payments remains the greatest cost in all game configurations. Conversely, strong bank costs are balanced between the two costs. Thus, strong bank nodes are able to consider a trade-off in managing costs in this environment.

Fraud is not widely experienced by customers in the network with an average of 12% of all customers, the majority of whom belong to a weak bank, being impersonated by the fraudster. In contrast, when bank nodes flag all payments an average of 15% of all customers are victims of fraudster impersonation. This phenomenon is explained by the behavior of the fraudster node. Each time the fraudster is not successful in a payment attempt, it chooses a new customer to impersonate. The fraudster node's success rate is lower when all payments are flagged and thus, the fraudster node switches which customer they are impersonating more often in this case. As a result, customers may be slightly worse off if experiencing attempted fraud is deemed costly for the customer.

7 Conclusion

In this work, we propose the flagging problem in which banks incur a non-negligible cost for using credit card fraud detectors. As a result, banks must strategically flag payments for review by their detectors so as to balance the cost of using the detector with potential losses from fraudulent payments. To analyze the flagging problem's affect on the adoption of fraud detectors and patterns of fraud, we introduce the flagging game played by fraudster and bank nodes in a financial credit network model. The bank nodes in our model differ in their detection capabilities, but all experience the same costs for use of their detectors. Bank nodes consider these costs when selecting a strategy that determines the probability a given payment is flagged. Fraudster nodes, with the singular goal of maximizing the amount of fraud they can commit, select a strategy defining the frequency and value of their payment attempts.

Our experiments show that fraudster nodes prefer to frequently attempt payments of high value, likely to take advantage of the imperfect fraud detectors employed by bank nodes. When costs for using the detector are low, all bank nodes send most payments to the detector. However, when costs become high, strong bank nodes select a strategy with limited use of the fraud detector. Thus, with high costs, strong bank nodes find the threat of fraud to be minor compared to the incurred costs of invoking the detector. This is confirmed in our analysis of costs of fraudulent payments in the network where we find weak bank nodes costs attributed primarily to not caught fraud and strong bank nodes experiencing more balanced costs. Our analysis shows that customer nodes may be a little worse off if all payments are flagged by banks, as the fraudster node is forced to attempt to impersonate more customers in the network.

While this work provides some insight into the flagging problem and the resulting effects on patterns of fraud, expansions of the model could broaden our understanding of these interactions. For example, we could explore additional dimensions along which the flagging and fraudster decisions are made. We might also consider expanding our study to different cost schemes for banks and adding a penalty to fraudster nodes for acquiring customer card details.

8 Acknowledgments

This publication includes or references synthetic data provided by J.P. Morgan.

References

Cassell, B.-A.; and Wellman, M. P. 2013. EGTAOnline: An experiment manager for simulation-based game studies. In *Multi-Agent Based Simulation XIII*, volume 7838 of *Lecture Notes in Artificial Intelligence*, 85–100. Springer.

Cheng, F.; Liu, J.; Amin, K.; and Wellman, M. P. 2016. Strategic payment routing in financial credit networks. 721–738.

Dandekar, P.; Goel, A.; and Govindan, R. 2011. Liquidity in credit networks: A little trust goes a long way. In *12th ACM Conference on Electronic Commerce*, 147–156.

Dandekar, P.; Goel, A.; Wellman, M. P.; and Wiedenbeck, B. 2015. Strategic Formation of Credit Networks. In *ACM Transactions on Internet Technology*, volume 15, 1–41.

Dervovic, D.; Hassanzadeh, P.; Assefa, S.; and Reddy, P. 2021. Non-Parametric Stochastic Sequential Assignment With Random Arrival Times. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 4214–4220.

Ghosh, A.; Mahdian, M.; Reeves, D. M.; Pennock, D. M.; and Fugger, R. 2007. Mechanism Design on Trust Networks. In *Third International Workshop on Internet and Network Economics*, 257–268.

Lin, W.; Zhong, L. S. Q.; Liu, C.; Feng, J.; Ao, X.; and Yang, H. 2021. Online credit Payment Fraud Detection via Structure-Aware Hierarchical Recurrent Neural Network. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 3670–3676.

Nilson Report, T. 2021. Card Fraud Losses Worldwide. *Nilson Report*, 1209: 5–7.

Ryman-Tubb, N. F.; Krause, P.; and Garn, W. 2018. How Artifical Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artifical Intelligence*, 76: 130–157.

Tuyls, K.; Perolat, J.; Lanctot, M.; Hughes, E.; Everett, R.; Leibo, J. Z.; Szepesvári, C.; and Graepel, T. 2020. Bounds and dynamics for empirical game-theoretic analysis. *Autonomous Agents and Multiagent Systems*, 34(7).

Wellman, M. P. 2016. Putting the agent in agent-based modeling. *Autonomous Agents and Multi-Agent Systems*, 30: 1175–1189.

Wellman, M. P.; Kim, T. H.; and Duong, Q. 2013. Analyzing Incentives for Protocol Compliance in Complex Domains: A Case Study of Introduction-Based Routing. In *12th Workshop on the Economics of Information Security*.

West, J.; and Bhattacharya, M. 2016. Intelligent financial fraud detection: A comprehensive review. *Computers Security*, 57: 47–66.

Zheng, W.; Yan, L.; Gou, C.; and Wang, F.-Y. 2020. Federated Meta-Learning for Fraudulent Credit Card Detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 4654–4660.