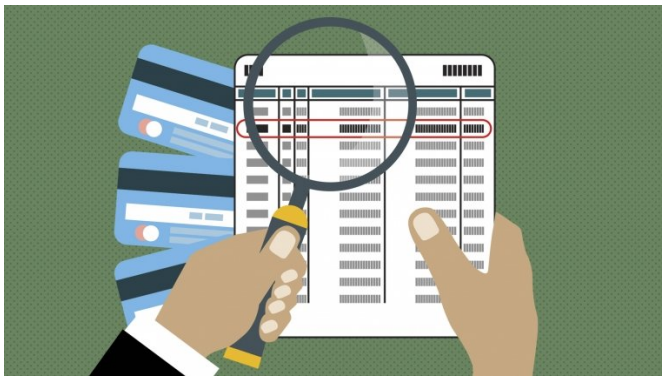
# Reducing false positives in credit card fraud detection

20 September 2018, by Rob Matheson



MIT researchers have employed a new machine-learning technique to substantially reduce false positives in fraud-detecting technologies. Credit: Chelsea Turner

Have you ever used your credit card at a new store or location only to have it declined? Has a sale ever been blocked because you charged a higher amount than usual?

Consumers' credit cards are declined surprisingly often in legitimate transactions. One cause is that fraud-detecting technologies used by a consumer's bank have incorrectly flagged the sale as suspicious. Now MIT researchers have employed a new machine-learning technique to drastically reduce these false positives, saving banks money and easing customer frustration.

Using machine learning to detect financial fraud dates back to the early 1990s and has advanced over the years. Researchers train models to extract behavioral patterns from past transactions, called "features," that signal fraud. When you swipe your card, the card pings the model and, if the features match fraud behavior, the sale gets blocked.

Behind the scenes, however, data scientists must dream up those features, which mostly center on blanket rules for amount and location. If any given customer spends more than, say, $2,000 on one purchase, or makes numerous purchases in the same day, they may be flagged. But because consumer spending habits vary, even in individual accounts, these models are sometime inaccurate: A 2015 report from Javelin Strategy and Research estimates that only one in five fraud predictions is correct and that the errors can cost a bank $118 billion in lost revenue, as declined customers then refrain from using that credit card.

The MIT researchers have developed an "automated feature engineering" approach that extracts more than 200 detailed features for each individual transaction—say, if a user was present during purchases, and the average amount spent on certain days at certain vendors. By doing so, it can better pinpoint when a specific card holder's spending habits deviate from the norm.

Tested on a dataset of 1.8 million transactions from a large bank, the model reduced false positive predictions by 54 percent over traditional models, which the researchers estimate could have saved the bank 190,000 euros (around $220,000) in lost revenue.

"The big challenge in this industry is false positives," says Kalyan Veeramachaneni, a principal research scientist at MIT's Laboratory for Information and Decision Systems (LIDS) and co-author of a paper describing the model, which was presented at the recent European Conference for Machine Learning. "We can say there's a direct connection between feature engineering and [reducing] false positives. … That's the most impactful thing to improve accuracy of these machine-learning models."

Paper co-authors are: lead author Roy Wedge, a former researcher in the Data to AI Lab at LIDS; James Max Kanter '15, SM '15; and Santiago Moral Rubio and Sergio Iglesias Perez of Banco Bilbao Vizcaya Argentaria.

## Extracting "deep" features

Three years ago, Veeramachaneni and Kanter developed Deep Feature Synthesis (DFS), an automated approach that extracts highly detailed features from any data, and decided to apply it to financial transactions.

Enterprises will sometimes host competitions where they provide a limited dataset along with a prediction problem such as fraud. Data scientists develop prediction models, and a cash prize goes to the most accurate model. The researchers entered one such competition and achieved top scores with DFS.

However, they realized the approach could reach its full potential if trained on several sources of raw data. "If you look at what data companies release, it's a tiny sliver of what they actually have," Veeramachaneni says. "Our question was, 'How do we take this approach to actual businesses?'"

Backed by the Defense Advanced Research Projects Agency's Data-Driven Discovery of Models program, Kanter and his team at FeatureLabs—a spinout commercializing the technology—developed an open-source library for automated feature extraction, called Featuretools, which was used in this research.

The researchers obtained a three-year dataset provided by an international bank, which included granular information about transaction amount, times, locations, vendor types, and terminals used. It contained about 900 million transactions from around 7 million individual cards. Of those transactions, around 122,000 were confirmed as fraud. The researchers trained and tested their model on subsets of that data.

In training, the model looks for patterns of transactions and among cards that match cases of fraud. It then automatically combines all the different variables it finds into "deep" features that provide a highly detailed look at each transaction. From the dataset, the DFS model extracted 237 features for each transaction. Those represent highly customized variables for card holders, Veeramachaneni says. "Say, on Friday, it's usual for a customer to spend $5 or $15 dollars at Starbucks," he says. "That variable will look like, 'How much money was spent in a coffee shop on a Friday morning?'"

It then creates an if/then decision tree for that account of features that do and don't point to fraud. When a new transaction is run through the decision tree, the model decides in real time whether or not the transaction is fraudulent.

Pitted against a traditional model used by a bank, the DFS model generated around 133,000 false positives versus 289,000 false positives, about 54 percent fewer incidents. That, along with a smaller number of false negatives detected—actual fraud that wasn't detected—could save the bank an estimated 190,000 euros, the researchers estimate.

## Stacking primitives

The backbone of the model consists of creatively stacked "primitives," simple functions that take two inputs and give an output. For example, calculating an average of two numbers is one primitive. That can be combined with a primitive that looks at the time stamp of two transactions to get an average time between transactions. Stacking another primitive that calculates the distance between two addresses from those transactions gives an average time between two purchases at two specific locations. Another primitive could determine if the purchase was made on a weekday or weekend, and so on.

"Once we have those primitives, there is no stopping us for stacking them … and you start to see these interesting variables you didn't think of before. If you dig deep into the algorithm, primitives are the secret sauce," Veeramachaneni says.

One important feature that the model generates, Veeramachaneni notes, is calculating the distance between those two locations and whether they happened in person or remotely. If someone who buys something at, say, the Stata Center in person and, a half hour later, buys something in person 200 miles away, then it's a high probability of fraud. But if one purchase occurred through mobile phone, the fraud probability drops.

"There are so many features you can extract that characterize behaviors you see in past data that relate to fraud or nonfraud use cases," Veeramachaneni says.

**More information:** Paper: "Solving the false positives problem in fraud prediction using automated feature engineering" www.ecmlpkdd2018.org/wp-conten … oads/2018/09/567.pdf

*This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology