An Executive's Guide to the Evolution of Fraud Detection

Julian Wong
Architect | DataVisor

Julian is a fraud and security detection industry veteran. As Head of Trust & Safety at Indiegogo and Etsy, and Risk Management leader at Upwork, he developed scalable systems and teams for mitigating fraud and risks. Prior to that, Julian led Google's engineering team responsible for building algorithms to prevent fraud on its ad platform. Julian holds a Bachelor's Degree in Engineering from the University of California, Berkeley and an MBA from NYU Stern School of Business.

With attacks coming from every angle, fraud detection is critical in today's world. Companies employ an arsenal of defenses, including rules engines, machine learning models, and ID verification as well as reputation lookups such as email, IP blacklists and whitelists that have been around for a long time.

But which methods make sense?

Each one provides value so these solutions, combined with domain expertise, will help build a fraud management system that will protect your business, products and users.

Rules engines and learning models are two of the major foundational components of many organizations' fraud detection architecture.

In this paper, we will outline how they work, and discuss the benefits and limitations of each. In addition, we will underscore the growing recognition of unsupervised analytics as an area that can overcome the limitations of learning models and rules engines.

Unsupervised analytics is a burgeoning segment that can enable an organization to go beyond simple rules engines and learning models by combing through mounds of data to identify and stop new fraud before it has a chance to even take shape.

Why should you care?

The headlines are everywhere. Fraud is an increasingly popular way to exploit a company's weak defenses. It can bring an operation to its knees.

Here's what the fraudsters are thinking: *Pile up a few thousand fake credit card transactions in short order, cover up the tracks, then move on to the next mark before they even know what him them. Damage done.* All before you even know what's happened. Sounds scary, doesn't it?

So, what does this mean to you? You have a business to run. Regulations to adhere to. And a bottom line to safeguard.

Every minute your operation is exposed to fraud is a minute too long. So if there was a way to keep your business running safe and sound, wouldn't you take a look? There is—and its unsupervised analytics. This paper provides an overview. Don't be the next headline. Take the right precautions so that your business can prosper.

BACKGROUND

Rules Engines

Rules engines separate operational business logic from the application code, enabling non-engineering fraud domain experts (e.g., trust and safety or risk analysts) with SQL/database knowledge to manage the rules themselves.

Rules engines can take blacklists of IP addresses, and other such lists derived from consortium databases, as input data. An analyst can also add a new rule as soon as a new fraud/risk scenario crops up. As a result, rules engines give businesses the control and capability to handle one-off brute force attacks, seasonality and short-term emerging trends.

However, rules engines do have scale limitations. Fraud perpetrators don't rest on their laurels. They change approach after being caught, so rules can go bad in as little as a few days. The process of adding, removing, and updating rules and weights every few days—especially with hundreds or thousands of rules to run and test—is impractical and would require operational and financial resources few organizations could muster.

Learning Models

Supervised machine learning is the most widely used learning approach for fraud detection. Using techniques such as decision trees, random forests, nearest neighbors, Support Vector Machines (SVM) and Naive Bayes, machine learning models often solve complex computations with hundreds of variables (high-dimensional space) to accurately determine cases of fraud.

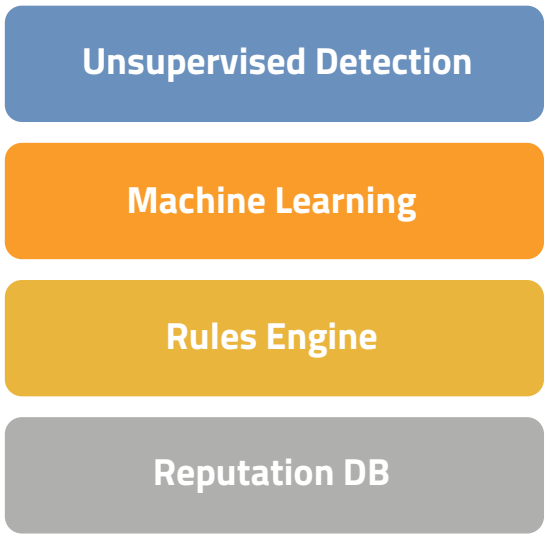
Because of their ability to predict the label for a new unlabeled data set, trained learning models fill in the gap and bolster the areas where rules engines may not provide great coverage. But trained learning models, although powerful, do have their limitations.

Fraud evolves quickly, and there are often no labeled examples of a given fraud type. Schemes change and fraud purveyors undertake new types of attacks looking for fresh vulnerabilities around the clock. A new fraud attack pattern doesn't register with the established training data, so the learning models probably won't return accurate results.

So What Can You Do?

Rules engines and learning models are important components of a fraud detection program. But now companies looking to deal reliably with the uncertain world of fraud have a new choice. It's unsupervised analytics—a burgeoning field that doesn't rely on prior knowledge of the fraud patterns. It requires no training data. The core component of the algorithm is the unsupervised attack campaign detection which leverages correlation analysis and graph processing to discover the linkages between fraudulent user behaviors, create clusters and assign new examples into one or the other of the clusters.

DataVisor Fraud Detection Technology Stack



GOING BEYOND RULES ENGINES AND LEARNING MODELS

Unsupervised campaign detection provides both attack campaign group info and self-generated training data, both of which can be fed into machine learning models to bootstrap them. With this data, the supervised machine learning will pick up patterns and find the fraudulent users that don't fit into these large attack campaign groups.

This framework enables Datavisor to uncover fraud attacks perpetrated by individual accounts, as well as organized mass scale attacks coordinated among many users such as fraud and crime rings – adding a valuable piece to your fraud detection architecture with a "full-stack."

Datavisor's correlation analysis groups fraudsters that act similarly into the same cluster. Anomaly detection, on the other hand, finds outliers which do not fit to an expected pattern or other users in the group. It looks for the handful of users that stick out. Such an approach may miss catching a fraud ring where the users in that ring are correlated and linked by a number of similar behaviors.

Unsupervised analytics works with rules engines and machine learning models. Organizations gain insights to provide to their fraud analysts to create new rules. When unsupervised analytics finds fraud that has not been encountered by a customer previously, the data from the unsupervised campaign detection can serve as early warning signals and training data for their learning models, which creates new and valuable dimensions to their model's accuracy.

The key difference between unsupervised analytics and supervised analytics is that unsupervised analytics does not require any training data. Unsupervised analytics automatically discovers and learns patterns from a huge amount of data without any prior knowledge.

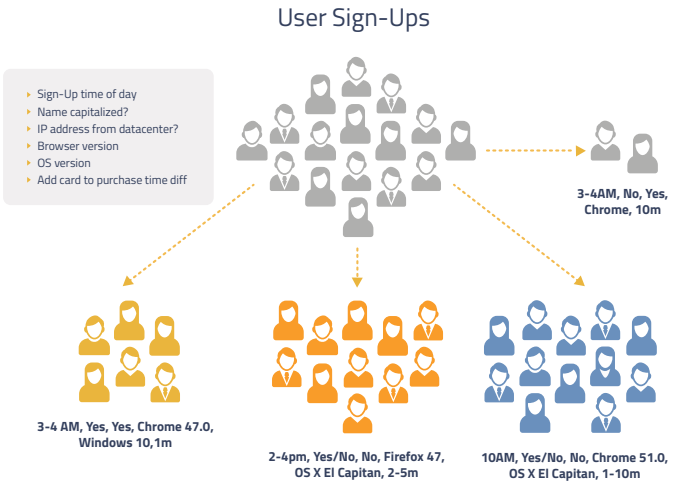
There are two main categories of unsupervised algorithms:

- The first is **clustering analysis**: It identifies—on a high level—similar users or events in a high dimensional space. In the fraud space, because one attacker usually creates many different identities to conduct fraud, the account behaviors of these accounts naturally cluster. That's why the clustering algorithm is able to identify them.

- **Graph analysis** works differently. It locates the associations among the users to identify suspicious connections. Algorithms such as connected component computations and graph cut identify suspicious subgraph components.

In clustering analysis, each data point correlates to a user in the high dimensional space. For each one of those users, unsupervised analytics can extract a number of features. For example: what time they signed up, whether the name is capitalized, if the IP address comes from a data center or proxies, the browser they used, and their operating system. Depending on the behavior, there can be many other defining characteristics (see Exhibit 1).

Exhibit 1: Defining Behavior Characteristics

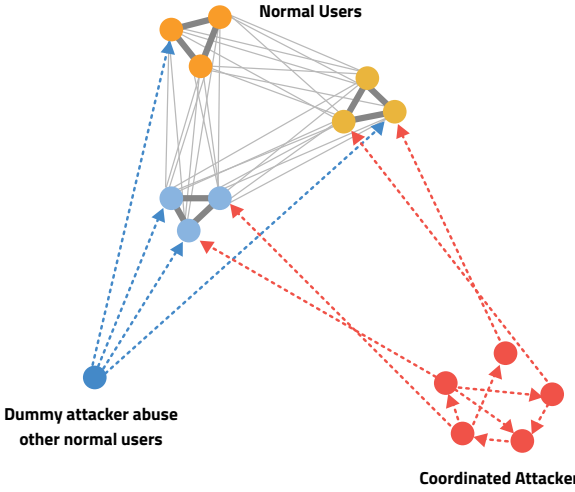


Using clustering algorithms in the example from Exhibit 1, unsupervised analytics bundles these into groups, including normal users who signed up between 2:00 pm and 4:00 pm and users who signed up for accounts from 3:00 am to 4:00 am. In contrast to the other group, the cluster that signed up from 3:00 am to 4:00 am used a data center IP address, Chrome and the same Windows version. In addition, the purchase time is an extraordinarily short period of less than one minute.

One user with this kind of odd behavior might actually be normal because some people work late and may like to conduct these kinds of activities in the middle of the night. But a cluster of these users with precisely the same behavior is highly suspicious and has all the markings of a set of fraudulent users. This is the essence of the clustering.

Let's look at another set of unsupervised algorithms—graph analysis (see Exhibit 2). At the top is a set of normal users—they're loosely connected with each other, some closer than others (they might be close friends or family members).

Exhibit 2: Graph Analysis at Work



The dummy attacker at the bottom left tries to spam all these good users. The attacker is very aggressive, and gets a very low response rate. By looking at the behavior of this attacker on the graph, we can easily identify this as the work of a spammer.

But not all spammers use such obvious behavior. Some are smarter attackers that actually form cliques among themselves that could masquerade as normal users. They send links to each other, they're friends with each other, and they also send a low rate attack to normal users. This kind of attack doesn't often raise any flags. So to uncover this stealthy type of attacker, unsupervised analytics runs graph algorithms to detect these strongly connected components.

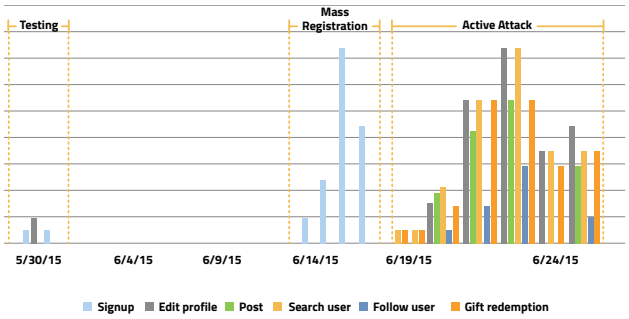
One challenge is defining the links between those users. There are two types of actions: explicit and implicit. For example, explicit actions could include User A sending a message to User B. They could also friend each other. Implicit include two users attending the same event or liking the same post.

The next challenge is scalability: how to efficiently presenting a relationship among and between users and then processing a graph with hundreds of millions of notes to identify the suspicious subgraphs. This is the key to the graph-based analysis.

Real World Attacks

Let's look at a real-world example of attackers incubating a promotional abuse campaign (see Exhibit 3). The timeframe runs along the bottom of the chart. The vertical axis measures the number of users. The attackers first incubate their plan early on in the timeline. They conduct testing with a small number of users in May. Finding an opening in the system, in May and June, they prepare a script to massively register users accounts. They only register users but don't conduct many activities as those users; they're essentially sleeper cells.

Exhibit 3: Incubating a Real-World Attack



But after two weeks, in late June, those sleeper cells become active. They start to add locations, add hobbies, signatures and phone numbers and then they connect with each other. After that, they launch an attack.

So how do we detect this kind of attack?

Traditional blacklists won't recognize this kind of activity as bad because the accounts appear legitimate and haven't exhibited bad behavior. It would be nearly impossible to detect that these users are bad based just on their IP address reputations.

Rule-based systems would show that these two-week old users have good reputations, haven't exhibited aggressive behavior and have profiles that appear to be perfect. They look normal, so the rule-based system would not uncover them.

Machine learning approaches wouldn't detect this kind of threat because the attacks would happen so suddenly. It's very hard to collect labels and start training in such a short period of time. By the time the machine is trained, the attackers would have shifted their behavior.

This is where unsupervised analytics comes in. It's able to detect this kind of attack because of the very strong coordinated behavior.

The key advantage of unsupervised techniques is that it does not require any training data. It can discover new forms of fraud and also identify large-scale fraud rings. Because it can identify a group of bad users, instead of suspecting one user as fitting a label, it has very low false positive rates and can identify attacks before the actual damage occurs.

TAKEAWAYS

Unsupervised analytics blazes a new trail and provides companies with a new weapon in their arsenal to keep fraudsters at bay. It offers the following advantages:

Requires no training data: Unlike rules engines and learning models, unsupervised analytics doesn't require any training data. A new fraud attack pattern doesn't register with the established training data, so the learning models won't return accurate results. Rules engines and learning models are important components of a fraud detection program but now companies looking to deal reliably with the uncertain world of fraud have a new choice. It's unsupervised analytics—a burgeoning field that doesn't rely on prior knowledge of the fraud patterns. It requires no training data.

- **Discovers new forms of fraud:** New fraud patterns elude traditional technologies because they rely on established definitions and training. On the other hand, unsupervised analytics can spot fraud patterns quickly without any previous exposure to the patterns it sees.

- **Finds large scale fraud rings:** The ability of unsupervised analytics to comb through piles of data quickly enables it to spot previously unseen patterns. In the process, it can stop large fraud rings before they can even get started.

- **Delivers low false positive rates:** Rules engines and learning models by themselves throw off false positives—often to the point where they're ignored. Unsupervised analytics cuts through the noise to deliver very low false positives so enterprises can focus on the most pressing issues and not chase their tails in search of phantom problems.

- **Detects problems before damage occurs:** With unsupervised analytics, dangerous patterns are isolated and stopped before any damage occurs.