EPISODE 373

[INTRODUCTION]

[0:00:00.4] JM: Coinbase is a platform for buying and selling digital currency; Bitcoin, Ethereum, and Litecoin. Every payments company deals with fraud, but a cryptocurrency company has a harder job than most payments companies because Bitcoin transactions are anonymous and nonreversible. This is in contrast to a bank which deals with a regulated reversible transaction system.

Soups Ranjan is the director of data science at Coinbase. In this episode he walks through the challenges of preventing fraud and he describes how machine learning and humans in the loop are used to deal with bad actors. From the data ingestion, to the data engineering, to the data science, this episode is a great overview of antifraud at Coinbase and it's a nice complement to the presentation that we previously aired from Soups. This was a while ago; three or four days ago, on Sunday, where we talked about the same antifraud stuff we'll talk about today, but in that episode it was a presentation that Soups gave from his own mouth. In this episode we go a little bit further.

This is the second episode in our series of shows about Coinbase. Yesterday, we discussed how Coinbase makes cryptocurrencies easier to work with, and tomorrow we're going to dive into the security infrastructure of Coinbase. We'd love you your thoughts on this series. We'd love to hear any other suggestions or feedback you have. You can send me an email, jeff@softwareengineeringdaily.com. I would love to hear from you.

[SPONSOR MESSAGE]

[0:01:49.9] JM: To build the kind of things developers want to build today, they need better tools, super tools, like a database that will grow as your business grows and is easy to manage. That's why Amazon Web Services built Amazon Aurora; a relational database engine that's compatible with MySQL or PostgreSQL and provides up to five times the performance of standard MySQL on the same hardware.

[INTERVIEW]

[0:03:29.5] JM: Soups Ranjan is the director of data science at Coinbase. Soups, welcome to Software Engineering Daily.

[0:03:34.4] SR: Thanks. Thanks for having me, Jeff.

[0:03:36.7] JM: As director of data science at Coinbase, much of what you study is fraud. Describe some of the flavors of fraud that you see from attackers.

[0:03:46.5] SR: The biggest fraud problem we are solving here is related to payment fraud, in which essentially attackers, they attempt to use a Bitcoin exchange, such as ours, as an ATM. An attacker could purchase a whole bunch of stolen credit cards online or they could purchase stolen credentials for bank accounts, and then they would come to Coinbase, they would essentially create a fake account and link that stolen credit card or that stolen bank account to this fake coinbase account. Then they would attempt to purchase digital currencies using Coinbase. Digital currencies we support right now are Bitcoin, Ethereum and Litecoin.

Once they purchase that, then they essentially move it out of the Coinbase exchange to a private wallet that only they have access to. Then a few months down the road, whenever the victim is taking a look at his or her bank statements or credit card statements they would realize, “What is this Coinbase entry in my statement?” Then they would call up the bank or the credit card company and they will reverse the transaction.

In these scenarios, the attacker takes off the digital currencies they purchased. The victim, under some circumstances, we have to return the funds back to them, and Coinbase is essentially left holding the bag.

Now, the payment fraud problem at a digital currency exchange such as Coinbases, I would argue, one of the hardest payment fraud problems in the world right now because of the fundamental nature of digital currencies. Digital currencies such as Bitcoins are fungible, as in one Bitcoin is equivalent to another and they are instantly transferable. You can actually move it out of an exchange immediately, it’s like digital cash.

The third piece over here is that they’re not reversible. If I send a transaction to someone, I can’t get that transaction reversed until I could essentially course that other person to return it back to me. It’s kind of like cash, which makes a digital currency exchange a pretty attractive target for attackers. I would argue that compared to even other e-commerce merchants out there or payment processes out there because of the nature of the goods that we are selling. It's, I would say, one of the most lucrative ways of converting —

[0:06:30.2] JM: Go into that in more detail, because it’s quite interesting because this is not the exact same problem that PayPal or Venmo or Stripe has to solve. It's made more difficult by the nature of cryptocurrencies. Explain why that is.

[0:06:44.9] SR: Right. One way of thinking about it would be that payment processes, like Stripes or PayPal as well in that same category, for them the way — If I were a scammer, the way I would actually have to perpetrate payment fraud would be I would have to essentially create lots of customer accounts right, link stolen cards with each one of them. Then I have to purchase something in order to move the money out of those systems.

Therefore, all these payment processes essentially have a collusion type of a fraud. The scammer, once it’s charging the stolen credit cards via these customer accounts, they’re going to actually charge it against another fake merchant account that they create on that same platform. Essentially, they pull the funds by buying these fake goods. Then after that they will move the fiat out of the merchant account. That's the payout mechanism.

[0:07:48.8] JM: You have to launder a PayPal transaction through fake goods.

[0:07:53.7] SR: Yeah. In some circles we also just call it the bank drop attack.

[0:07:58.5] JM: Okay. That creates another hurdle of complexity for the attacker that you could potentially stop in the PayPal scenario.

[0:08:04.7] SR: Right. It creates another hurdle. It's a different detection mechanism because we're looking for collusion more than anything. It's also a little bit harder for the scammers to perpetrate fraud using merchants of even processes like that, because they have to essentially have a way to buy some fake goods. They have to go through a few of the hurdles. Whereas with digital currency [inaudible exchange is they don't really have to do that. They can actually be off at the district currency instantaneously.

Before we proceed any further, I just wanted to state this at the beginning of the interview that Coinbase has been able to stay ahead of the scammers pretty well, and we consider the fraud loss as one of our USPs, or unique selling propositions which differentiate us from other digital currency exchanges. We've been able to consistently keep it below are OKR metric. If we're not able to keep it under control, then it could easily and quickly escalate to be a redline item against our profitability.

The other caveat I wanted to add before we proceed to further down the interview is that majority of folks who are transacting on a digital currency exchange such discretization such as ours are actually engaging in good behavior. They are either using digital currencies to purchase online goods or they are using it to transfer money instantaneously to their family or friends, etc.

It's these small, very very small percent of users who my team deals with, and I just wanted to highlight that. It's very small percent of users, and therefore we should not associate — When

I'm talking through the rest of this interview, we should not think about digital currency exchanges as only having this behavior.

[0:10:00.3] JM: Right. It’s not a black market filled with scummy people.

[0:10:03.5] SR: Yeah. Right.

[0:10:04.9] JM: Now that we've illustrated the typical fraud problem, let’ start to talk about how you detect that fraud. I think of Coinbase as an event stream of transactions where people are buying cryptocurrencies and selling cryptocurrencies. Can you describe how the data that you’re ingesting looks, and then we’ll get into the data infrastructure and how you manage those different types of data.

[0:10:36.9] SR: Sure. Yeah. As soon as you create an account on Coinbase, literally, everything you do during the sign-up flow or everything that you do on the site when you’re doing a transaction is essentially fed in as a signal to our machine learning model which assigns a risk score to every single user. We use that risk score to determine how much purchase power the user has.

Our system, or risk-based assessment of how much a user is allowed to purchase, etc., is one piece of the puzzle over here. A whole fraud program, it's comprised of humans, human analysts who are really really skilled, as well as a machine learning system and they both have to work together. You could not solve this problem by just asking our human analysts to take a look at every single transaction because, A; it wouldn't scale. B; it would be prone to human error once in a while.

Likewise, you couldn’t really just only use machine learning, because one of the limitations of machine learning system in this case is that the label data that we are using in machine learning, it can come a little too late. Let me give an example, like we use in the broader fee cycles as well. Suppose a scammer comes in, creates an account using a stolen credit card belonging to Alice, and then a second thing the scammer does — Because we require [inaudible 0:12:10.3] account to do it, is they have to then upload an ID. Most accounts have to upload an ID. Let’s say they upload an ID belonging to Bob. Then thirdly, all accounts on Coinbase have to provide a phone number in order to do a purchase on Coinbase. Thirdly, let's say they link the phone number belonging to Carl. They have mismatched identities belonging to three people; Alic, Bob and Carl.

Now, the real reason why machine learning works over here is that we're not going to take a look at the information from the credit card or from the bank statement linked by Alice. We're going to take a look at extract information around the ID if he did an SSN-based verification belonging to Bob. Then we also going to take a look at the phone number we're going to look up who is this phone number registered to. What is the name and address of that person?

Then we bring all these pieces of information together and we will basically look for mismatches between names across these data sources. There's few other data sources. For instance, to create an account of Coinbase, you have to also provide an email address. We will take that email address and also look up old social media profiles related to that email address. Again, extract names and addresses behind that social media profile.

The number one reason why machining learning works over here is because we are doing this name, address mismatches across these data sources. The question you could ask is why even use machine learning? Why not just use a rules-based system over here? Why couldn't I just say, "Hey, if names across — If the name Alice doesn't match the name bob, then don't even allow that person to purchase on Coinbase."

But then we would have way too many false positives. You would not be allowing a lot of good uses from transacting on our platform, because I used to live in Houston. I travelled to California. I may have forgotten to update my information with social security. They may still think I in Houston. Therefore, my address would be different across two data sources. My full name is really long, [inaudible 0:14:24.2] engine, but I also use my short name in a lot of places. If you just looked at name mismatches, then that would be a false positive as well. That's one of the reasons why we use machine learning, because machine learning is better in making a judgment than there' a gray area.

The second reason why machine learning works to detect fraud is because of the velocity base signals, which is essentially based on the Broken Window Theory. The Broken Window Theory says that all these scammers, they are constantly talking amongst each other. They're on online forums. They're sharing ideas. Sometimes these scammers they even create tools that they sell to each other, because like the people who made money during gold rush or the people who were selling the shovels, right? They sell these tools where, essentially, they're encoding all the tricks.

Sooner or later when all scammers discover the latest trick, they're all going to use it. Therefore, velocity based signals are really really good at helping catch fraud as well. What ends up happening is suppose all the scammers are using a remote desktop protocol, Windows remote desktop protocol in order to pretend that they are coming from a particular machine.

We had a case where we saw that a particular screen resolution, 1364 x 768, the normal probability of it occurring is less than .1%. Normal, people don't use it. Our admins, our human analysts, they were actually banning lots and lots of accounts which are using that screen resolution because they use this sixth sense and they were like, "Something looks fishy over here." Then later on, this got picked up by a machine learning model because that particular screen resolution had a really really high band rate. The earl reason was that all these scammers that were using Windows RBD protocol which had a bug because of it, the screen resolution was off by one pixel on each side. The screen resolution would have been 3066 x778.

[0:16:31.0] JM: Your point here is that —This was a case where machine learning picked up it but it was too late basically. It was later than you would have had if you didn't have the humans in the loop.

[0:16:41.7] SR: If you didn't have humans in the loop then, yes, it would have been later. You're absolutely right. Yeah. Humans are actually o constantly banning accounts before chargebacks can roll in, because chargebacks can occur in credit cards for up to six months, and in ACH, for up to two months; 60 days.

If we've waited for the that these accounts, Alice has to call and say that they're bad and therefore apply the label bad, then that would be too late. Whereas if you have humans in the loop and they’re constantly labeling accounts which they are findings suspicious, then you can short-circuit the training process for machine learning.

[0:17:16.3] JM: Okay.

[SPONSOR MESSAGE]

[0:17:25.5] JM: Hosting this podcast is my full-time job, but I love to build software. I'm constantly writing down ideas for products; the user experience designs, the software architecture, and even the pricing models. Of course, someone needs to write the actual code for these products that I think about. For building and scaling my software products I use Toptal.

Toptal is the best place to find reasonably priced, extremely talented software engineers to build your projects from scratch or to skill your workforce. Get started today and receive a free pair of Apple AirPods after your first 20 hours of work by signing up at toptal.com/sedaily. There is none of the frustration of interviewing freelancers who aren’t suitable for the job. 90% of Toptal clients hire the first expert that they're introduced to. The average time to getting matched with the top developer is less than 24 hours, so you can get your project started quickly. Go to toptal.com/sedaily to start working with an engineer who can build your project or who can add to the workforce of the company that you work at.

I've used Toptal to build three separate engineering projects and I genuinely love the product. Also, as a special offer to Software Engineering Daily listeners, Toptal will be sending out a pair of Apple AirPods to everyone who signs up through our link and engages in a minimum of 20 work hours. To check it out and start putting your ideas into action, go to toptal.com/sedaily.

If you’re an engineer looking for freelance work, I also recommend Toptal. All of the developers I’ve worked with have been very happy with the platform. Thanks to Toptal for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:19:29.6] JM: We've given an overview of how this works in terms of the overall system, but I want to understand how it works in terms of the data infrastructure. You've got data from a variety of sources that you need to aggregate and you probably need to buffer them somehow. I don't know if you're using Kafka or something. Where you're storing — You're storing the data probably in different databases, and you need to have data infrastructure that makes this accessible to the people that are building applications within Coinbase. Can talk about how the data infrastructure looks and maybe what open source tools you're using, what AWS tools you're using?

[0:20:09.2] SR: Sure. Yeah. A lot of what we do here is built on open source technology. I'll talk about both our data engineering pipeline and machine learning pipeline.

[0:20:19.1] JM: Yes. Perfect.

[0:20:19.4] SR: Which one do you want me to start with?

[0:20:20.7] JM: Start with data engineering.

[0:20:21.5] SR: Data engineering cool. A data engineering pipeline — First of all, our production databases is Mongo. For business intelligence purposes, we use Redshift, which is essentially PostgreSQL. Now, we have data engineering pipeline which essentially at the moment recreates that Redshift tables from scratch every single day. As we've continued to grow and scale, this process is not going to scale, so we are in the process of rewriting it.

We are now essentially tailing the oplog, or the operation log of Mongo which essentially has all the inserts, updates, deletes, etc. We tail that log and then, in a streaming fashion, we then apply those updates to Redshift so that the BI, business intelligence, which in our case we're using Looker on top of Redshift. Looker, in a few months from now, we would be able to guarantee that the data in Looker or in Redshift is only, let's say, 15 minutes stale compared to the production database, which is Mongo. That is our data engineering pipeline.

[0:21:25.3] JM: I've done a number of shows at this point where people are talking about Looker. I don't know what it does. Can you describe what Looker is?

[0:21:32.5] SR: Yeah. Sure. Think of Looker as Tableau, which is —

[0:21:37.8] JM: Okay. I know Tableau.

[0:21:38.5] SR: Yeah, it's really popular. Tableau is really popular. The differences between Looker and Tableau are that Looker has essentially created another language which they call LookML using which people who don't even know SQL, they could essentially very easily create dashboards.

We have folks in our business operations and support compliance organization or the antifraud people. All of those folks are now, essentially Looker experts because they don't have to rely on a data analyst for all the queries. They're able to actually create simple dashboards very quickly on their own.

Looker also allows people to share a dashboard that they have created with each other. Another one — I don't know if you or your listeners are familiar with IPython Notebook or Jupyter Notebooks.

[0:22:36.4] JM: Yeah.

[0:22:37.9] SR: Jupyter allows you to have code along with charts, etc., but that's for much more advanced users. Looker allows you to have reusable dashboards where you can also go in and see what were the filters that were applied and how is this dashboard being generated. You can actually — If you have doubts about it, you can question it and assert whether it is doing the right thing as you wanted to do.

[0:23:07.4] JM: It might not be intuitive for people listening why you would want a production database that's in Mongo and then you've got this event log — Every database has an event log of changes that happen to it for a variety of reasons. We've done shows about this. Why do you want different databases? Why do you want a Mongo database that's like the production database that gets updated and you have this Redshift database that's tailing it by 15 minutes or whatever it is, 15 seconds. I can't remember what you said.

[0:23:37.0] SR: 15 minutes.

[0:23:37.5] JM: 15 minutes. Yeah. That you're doing analytics on.

[0:23:41.1] SR: Yeah. The main reason is that Mongo in NoSQL, so it allows us to very quickly create — We use Ruby on Rails, and we very quickly create what we call models. Essentially, are like classes in Java or C++.

It allows us to create these models where we can very easily add a new field and without having to specify or change the schema. It allows us to make iterations very quickly. Now, you can't really use Mongo for cross-table joints, so to say. It's not going to scale. You can't even, therefore, use it to do these heavy-duty queries which, let's say, where we need to slice and dice. Let's say in an accounting problem, could be how much of Bitcoin has moved into the platform and how much of Bitcoin has moved out of the platform, where you have to essentially tally up all these transactions and use FIFO or LIFE accounting. All those things get harder to do in Mongo. Whereas they're much more natural to do in Redshift.

For business intelligence purposes, you don't want to, anyway, stress your production databases, because you'd rather have the production systems up and running and when you want queries for BI purposes, that you don't really have the strict SLA requirements as your users of the product have. You can actually have a second database which in this case is Redshift which allows us to run these complex long-running queries which where we don't really have to have a very strict SLA.

[0:25:26.2] JM: Right. Okay. I'd love to go in more detail in data infrastructure, but we should talk some about data science. Once you've got that —Is that Redshift database, is that the source of truth if you're doing data science for all transactions and all users and all everything, you can just query it for everything you need in Coinbase if you're a data scientist?

[0:25:47.5] SR: Yeah. Is your question that what is the source of truth?

[0:25:51.6] JM: The source of truth, if I'm a data scientist building something a Coinbase, am I typically querying just that giant Redshift database?

[0:26:00.0] SR: Yeah. Exactly. Yeah. Redshift is basically the tool that our data analysts and data scientists are primarily interacting with. Yeah.

[0:26:08.6] JM: Okay. Give an example of — Or I guess describe in more detail what people build on top of that, because you describe at the beginning this conversation, the machine learning pipeline or kind of the different things that you're looking at as you're assigning a risk score to use. I guess the main objective of this data science pipeline is to assign metrics to users so that you're evaluating those users. You could tell me if I'm wrong. Okay.

[0:26:39.6] SR: Yeah, the machine learning pipeline is separate. The machine learning pipeline — The moment it is using that data engineering pipeline, but we are going to move a bit. That's why on a purely abstraction point of view, from purely abstraction point of view, I think of them as separate. The machine learning pipeline, their intention is to use all the signals that we are interested in about our users and take the label data that we have about users, as in is this use a fraudulent or this uses not fraudulent, and then train a supervised machine learning algorithm.

At the moment, our machine learning pipeline relies on the data engineering pipeline, which is we take the data which is in Redshift and build a model using another open source software called Vorpal Rabbit, which is essentially very fast rabbits built incorrectly and spelled very fast. Vorpal Rabbit is a really great tool. It allowed us to stand up our machine learning platform within two months with just myself and an intern.

Now, what we do is essentially we take the data which is in Redshift, create our dataset using SQL queries and then use Pandas, Python Pandas data frames to massage the data and then we have what we call in machining learning feature engineering, which is essentially we create lots of transforms, etc. For instance, you can have a transformer which says what is the city of the user? If the City is Berkeley, or is the city San Francisco etc., then you got to convert these categorical features into [inaudible 0:28:20.0] features so you would encode saying that, "Okay. City:Berkeley would have the value one for whoever has the place of residence as Berkeley, and all of the cities, like City:San Francisco or city: San Diego, or city:Palo Alto would be 0 for that user.

A transform like that that are — Other more complicated all transforms, they are all encoded in Python. Then we provide this data to Vorpal Rabbit and we're using a stochastic gradient descent which is another name for logistic regression to train a supervised band re-classifier.

[0:29:01.8] JM: I did a series of these interviews at Stripe a while ago and I talked to them about machine learning. One of the things they said was you have to play with this knob of accepting some amount of transactions that you know are fraud because it helps you train to classify fraud more quickly. Do you have that knob at Coinbase?

[0:29:23.9] SR: We don't, actually. We've taken a very different approach, and I can explain why, but let me first explain the approach. Our approach is if we had a knob, or when we had a knob — We used to have a knob like that, a year and a half, two years ago. Then what that meant was any user who's machine learned risk score was, let's say, greater than .8, you would say, "Hey, we think you're bad. You can't purchase from Coinbase at all."

Then that meant that those users never had a second chance to prove themselves innocent. We were losing good users as well. This is a cost to pay for false positives. Then what we did was we said, "Okay, based on your risk score, we will assign your purchase power, and even if you're really really risky, then we will allow you to purchase, let's say, at a very very small amount. Let's say $5 a week or something like that."

Now, then you can think of it as, okay, essentially if that user was truly a scammer, then we are paying to learn the behavior, $5 dollars a week of a cost. If that user was actually good, then we would observe that user over a long period of time and the machine learning system and the risk system will see, "Yeah, this user never really charged back, or this account never charged back for the next six months, or a year." Therefore, the risk score would evolve and then they would transition into being a good user with the descent limits. That had allowed us to look on the problem of preventing false positives.

[0:30:56.5] JM: Can we talk more about the idea of a risk score, and I guess the different metrics that are connected to the idea of a risk score?

[0:31:05.8] SR: Sure, yeah. Risk score itself is just a continuous value between zero and one where highly risk users have a value as high as one and the least risky users have a value as low as zero. Now is your intention to go in deeper into the machine learning metrics, like [inaudible 0:31:24.2] curve or log loss, etc., or you're driving towards the business metric?

[0:31:28.2] JM: Whatever's interesting.

[0:31:28.9] SR: Sure. Metrics is always very interesting when it comes to machine learning. On the business side, we measure fraud rate as our key business metric over here. Fraud rate, we define it as losses caused because of chargebacks as the numerator, and denominator is our total purchase volume.

That's the fraud, the business metric. The machine learned metric is of two kinds, [inaudible 0:31:59.3] curve and log loss. Let me explain that. First of all, log loss is essentially measuring, when you're doing a training, it's essentially measuring for all the users that your system thought scammy or fraudsters, how high did the machine learning algorithm assign a risk score to them. Vice versa, for the uses who are truly good, how low was the risk score assigned to those good users? How good is the system at giving bad users high scores in good users low scores.

In a lot of ways, log loss is actually as a machine learned metric going to be highly indicative of the fraud rate on the business side because for two reasons. Again, if the machine learned metric was really good at assigning a high risk score to the bad guys, then my fraud rate, the numerator on the fraud rate, the loss would be low. Therefore, the fraud rate would be low.

If my machine learned metric was really good at giving high limits to the good users, then my purchase volume in the denominator would be high, and therefore fraud rate will be low. That's why when we're training a machine learning model, we use log loss as our key metric to evaluates its efficacy and efficiency. If a model is doing well from a log loss perspective, then we promote it to be the production model. Then afterwards we, again, monitor how is this model doing in production via an A-B test.

[SPONSOR MESSAGE]

[INTERVIEW CONTINUED]

[0:34:56.8] JM: Talk about the production in more detail. When you're releasing —The process of releasing and A-B testing and machine learning model, because this is something that people who are building machine learning models in all domains could probably gain from hearing about it, because that's what you always need to do. Whenever you deploy your new model, you need to make sure it's actually an improvement over the previous model. Explain the deployment process for machine learning models.

[0:35:26.4] SR: Yeah, sure. Yeah. It's an important point that sometimes it is not taught in academia, etc., and this is something that you can only learn —

[0:35:35.2] JM: I don't think I heard the word deployment in undergrad.

[0:35:37.4] SR: Yeah. Exactly. I wish that we could actually teach kids what it really means to deploy a machine learning model.

[0:35:46.4] JM: Or anything, or Ruby on Rails application.

[0:35:49.7] SR: Yeah. That's why internship are there and Coinbase has gained an internship program as well. I actually really enjoy teaching my interns, what it means to deploy models. A few things before I talk about deploying a model that I should first talk about. First, it can be literally very scary to deploy a model because you are literally then — The risk score for lots of people can change very quickly. We take lots of precautionary steps such that we are not going to move risk scores and thereby purchase limits of lots of users.

It is also very important to deploy a new model as we learn new behavior because we're going to stay ahead of the fraudsters. We want to quickly extrapolated, identify and extrapolate a new fraud pattern that we're seeing.

Therefore, what we do is we would run a model in production and then we do a shadow deployment where we deploy or challenger model in shadow mode. The challenger model is going to score every single user, but we are not going to use the score from the challenge model to assign limits or assign the purchase power. We will only use the production model to do it.

Later on, we will essentially do sort of a — Call it a data exercise or call it a simulation of all these users over, let's say, a period of time, a couple of weeks or a month. What was the score saying by the production model? What was the score saying by the challenger model? What does the distribution of scores look like for good uses and what is a distribution of scores look like for bad users? Also, what is the score distribution look like for veils?

We are very interested in making sure veils or high-value users will purchase a lot. These scores won't change. Sometimes models can make mistakes. Then what we do is view it essentially present the current production score and the challenges learned score to our analysts to look at for veils and they would go, "Yeah, this model is right," or "That model is wrong," or "This model is right," or "That model is wrong," and they will go in and essentially manually override the scores for those users or lock the scores for those users.

Then next step, if machine learned metrics like log loss makes sense for a challenger model, and also the distribution of scores for good and bad uses — For good users, it didn't change too much. For bad users, changed a lot. That makes a lot of sense then to promote this challenges model to be a production model.

[0:38:23.6] JM: Right. Interesting. That model being deployed, what actually does that mean? Is that like in a micro-service or something? I guess this would be a good place to talk about, like the infrastructure and how requests are handled. Where in a request path is a machine learning model getting a request to some micro-service request, or how is it being deployed and where is that request coming from?

[0:38:54.6] SR: You mean where is the request coming from to deploy a new model?

[0:38:57.9] JM: A lot of people have talked to you, okay, you've got your machine learning model running in a container somewhere and when you have a question for the machine learning model, you make a request to that container.

[0:39:07.2] SR: Yup. Is your question then about the scoring side?

[0:39:14.4] JM: My question is more about when I'm deploying a machine learning model, what is that actually mean? Where is the code I'm deploying? Is it in a container? Is it in an EMI, or AMI? What's the unit of runtime?

[0:39:28.3] SR: Yeah. At Coinbase, we have a really good infrastructure team which has essentially codified our infrastructure. In the sense that every single micro-service that we run is integrated inside a Docker container and then deployed in AWS.

When we deploy a new machine learning model, it's essentially changing the name of the model, so changing the pointer to the S3 path where the new model is being hosted. That's a config change, and then you hit deployed again, and then a new Docker container is built and then it's essentially launched an our AWS cloud.

[0:40:10.0] JM: What is the request path for — When is a machine learning model being hit? Is it being hit every time a user makes a request to buy or sell Bitcion?

[0:40:20.3] SR: Yeah. On the scoring side, throughout the user journey, we score a user. We rescore them. You've signed up for an account, you complete our quick start flow, which is you provide us your email address, phone number, link a bank account, then immediately after that you get a risk score. Then afterwards, as you continue to purchase or sell, etc., then we continue to rescore.

Now, of course, there has to be a good balance over here. We can't be rescoring users too often either because then the risk score — Or the limits could change quite often, and our users don't really like the limits to change that often. We take lots of steps to make sure that we can correct for any drastic correct for what I would call hysteresis type behavior where these quick increases and quick decreases, quick increase and quick decrease in the risk for a user. We prevent that hysteresis by essentially saying that only change a user's risk score if the new score is actually greater than the old score by threshold. That's a very simple way of preventing a hysteresis like that.

[0:41:34.2] JM: How often are you updating these risks scores?

[0:41:37.9] SR: Pretty often. They are being computed every time you do a buy or a sell or you add a payment method or you add a new ID, or you go any additional verification. It could be too frequent, and that's why we do the sort of dampening to make sure that your scores don't change too often.

[0:42:01.3] JM: What's the path for a change like that? If I make a purchase on Coinbase, the Mongo database gets updated, 15 minutes later the Redshift databases is going to be updated. How is that going to propagate to my risk score?

[0:42:17.2] SR: Yeah, Redshift doesn't come in the path of scoring, because that would be too late then. What we do in the moment is we basically use the version AWS of Mongo to create a feature vector and then we send it to a micro-service which then essentially does a join or a .product between the feature vector and the machine learned model file, which has the [inaudible 0:42:43.2] for each feature.

Therefore, this is the current architecture. I'm happy to talk about the one upcoming in the future. In the current architecture, our feature generation is being done twice, once at scoring time, at run time when we want to score uses. For that, we do it inside Mongo, because production database is the only one which is going to have the least latency. For training, we're using Redhift which has a big latency. The feature generation has to be done in Redshift as well.

[0:43:20.2] JM: Got it.

[0:43:20.9] SR: Going forward, we are moving towards an architecture which would be more like a micro service where only the data that we are interested, not this multi-terabyte database that we have. Only the data that we're interested in for a machine learning model generation purpose, we are going to send those data points to a database in a micro-service and then we would use a tool like, let's say, Apache Spark, or maybe we continue using Vorpal Rabbit to essentially do queries against this data store and generate a model. Then we need to actually get a score for a user. Instead of generating a feature vector in Mongo and sending it to somewhere to get a score, we'll just have to send the user ID to this micro-service because this new micro-service will have the data store which will be kept up-to-date with production and therefore it can solve two problems with one. We will not have code duplication we would be able to do not only a model generation and experimenting with different models whenever we want, but also score using that same database and the future engineering code will be the same.

[0:44:30.3] JM: When you take a user's model — You've got a machine learning model that's pre-computed, and if a user makes a request to purchase something, the Mongo database gets updated and then the user's data gets copied from the Mongo database into a feature vector and the feature vector gets the micro-service that hosts the machine learning model does something with that user feature vector in order to decide, "Okay, is this user —" I guess, what does that micro-service do? It's just applying to risk score or it's determining to take an action on the user or something? What exactly is that?

[0:45:13.3] SR: It's just giving a risk score.

[0:45:14.8] JM: Giving a new risk score.

[0:45:15.6] SR: Yeah, because the risk score is essentially — Think of it is as to — It's a vector. I'm sending you a vector, just a feature vector, which is "Jeff lives in Japantown." So city called Japantown is one. Other things like which bank account did you sign up, bank account Wells Fargo, that feature value is one. The model configuration file, what it does is it says, "Okay, bank**:** Wells Fargo, that has a coefficient of .5. City**:** Japantown has a coefficient of -.5. We then essentially are doing a .product. Then the result is essentially is a score between 0 and 1.

[0:45:59.4] JM: Okay. Once the new risk score is generated — I guess we already discussed what is being done with the risk score. I think we've kind of talked end-to-end about what are the different aspects that go into a risk score creation and what is actually being done. Do we miss anything about the overview — What haven't we covered in terms of the data infrastructure and the risk score creation and kind of the end-to-end flow the a user is going to go through to be determined as being liable, likely to be fraudulent or not.

[0:46:38.0] SR: I think we've it covered when it comes to supervised machine learning. The other big piece that we haven't covered is the fact that supervised machine learning and human analysts are not just two legs of a three-legged stool.

The third important component here is unsupervised learning. Again, unsupervised learning is essentially solving the same problem that label data, like which user is fraudulent or not can take a long time to come back to us. We've unsupervised learning to sort of latch on to and extrapolate any new pattern that we are seeing.

There are three pieces over here, the first one is anomaly detection, which is basically we have baselines, monthly baselines for what fractional of a uses base is signing up with, let's say, Wells Fargo bank accounts. What fraction of a user base is signing up with JP Morgan Chase credit cards, or what fraction of user base is using a phone provider which is Verizon. Then any given week, we also look at, okay, fraction of users are signing up with these things.

If an any given week we see an anomalous behavior compared to the monthly baselines, then we know that there's something odd. Maybe scammers have purchased a stolen credential database of JP Morgan Chase cards or Bank of America or stolen username passwords of Verizon accounts.

Then we immediately present those users who have signed up in that week to our analysts who then then quickly go in and take a look at them, okay, good, bad, good, bad, etc. They label them.

This unsupervised approach allows us to extrapolate any new pattern we are seeing, not knowing whether that pattern is good or bad, but then we can present it to our analysts to label it further.

The second piece of the puzzle here is related user detection. There's, again, two pieces to that. The first one is deterministic the second one is probabilistic. The deterministic one is basically if our analysts, they use their sixth sense and they say this account is bad and they ban it, then we quickly find other related accounts which are very strongly related, like two accounts share the same SSN or the same encrypted bank account number. It's very likely that these two bank accounts are controlled by the same individual and therefore we'll immediately apply the ban label to the other related accounts. Therefore, an analyst has to only act on one and then the system would kickoff and take care of the others who are related.

There's a probabilistic one where they're not going to take an action automatically, but when an analyst bans an account, then we present a list of accounts which are highly related in the probabilistic sense to this account so that they can look at them and take an action.

The probabilistic one is essentially using call sign similarity. We have —Because of the supervised machine learning approach that we take, we have essentially an n-dimensional feature space, city could be one, like bank account and phone provider could be the other two, and there's so many of them. This n-dimensional space, every user is presented as a point and then you get a look, "Okay, all the other uses vary this compared to this point, and was is essentially the call sign similarity between those two points." Therefore, we'll present those other related accounts to the analysts.

The third one is rules-based system. The rules, at least, at the moment we're not discovering them automatically, but we want to. At the moment, our analysts, they take a look at accounts and if they use their sixth sense and they figured out, "Yeah, I see a pattern. I see a new fraud ring." Yeah, there's a fraud ring which is using JP Morgan Chase cars and they're all signing up with Verizon phone numbers."

Then they'll go, "Yeah, let me make a rule. All these accounts which match this pattern, I'm going to say lock their risk score at a value of 80." Therefore, their purchase power would be, let's say, like $100 a week." They'll say, "These accounts which match this pattern, all the accounts which are coming from 3064 x 768 screen resolution, I'm going to throw some friction at them. Let me just apply or require an ID verification where they have to not only take a picture of their ID using the webcam, but they have to also take a picture of themselves, as in the selfie holding the ID."

A lot of these things then are like how much friction can we throw at the fraudsters such that their ROI goes down and they go away essentially? Yeah, these three approaches essentially allow us to not have to wait for chargebacks to appear before we can catch a new pattern.

[0:51:20.6] JM: Yeah. Okay, let's zoom out a bit to close off. You've worked in ad fraud prevention as well, advertising fraud where we've done a lot of shows about advertising fraud. I think you've worked in some other security areas. Describe how Coinbase working in data science and fraud detection, a Coinbase compares to the other fields that you've worked in.

[0:51:41.7] SR: Yeah. I would say this is definitely one of the most intellectually satisfying jobs I've had in my career. I've been here at Coinbase for two years know and my work career has been — I moved to the Bay Area 2005. Yeah, 12 odd years. Yeah. 12 years before that I was in grad school completing my PhD. at Rise.

Yeah. These 12 years, I've worked on a variety of problems, cyber security first five years of my career, where essentially I built a system which could deduct cyber threats, like denial of service attacks or track botnets or detect application layered DDoS attacks which could affect protocol flooding at the voice over IP layer, let's say, or HTTP layer, etc.

That was great. That was the first five year of my life, or my work life, and then the next five around advertising ad tech which was detect click fraud in one phase of that second five year. Then another one was — I worked on real time bidding at Yelp and Flurry. Yeah, now, there was always something which was missing, which is what I can say Conbase has provided me, which is Coinbase has some very sophisticated and intelligent adversaries. The fraudsters that we see are — We didn't get to talk about account takeovers. The scammers were trying to —

[0:53:17.6] JM: I'm planning to talk about that with Philip.

[0:53:20.3] SR: Yup. Okay. The account takeovers that we're seeing, etc. They are essentially using tricks that I would say we get to see before the rest of the industry gets to see.

[0:53:36.6] JM: Like cellphone takeovers.

[0:53:36.9] SR: Yeah, like cellphone. Like phone number porting or sim swap attacks. Therefore, we are able to, A; see things before people read about them. That makes it really fascinating. The intellectually satisfying part for me is that we are able to stay ahead of them because of the team we have over here and because of the way that we can actually move very quickly over here and build tools and techniques to proactively prevent things from blowing up.

[0:54:08.2] JM: Yeah. Cool. Soups, thanks for coming on Software Engineering Daily.

[0:54:11.8] SR: Thanks, Jeff. Thanks for having me here.

[0:54:13.5] JM: Great.

[0:54:13.8] SR: Cheers.

[0:54:14.2] JM: All right.

[END OF INTERVIEW]

[0:54:18.5] JM: Your application sits on layers of dynamic infrastructure and supporting services. Datadog brings you visibility into every part of your infrastructure, plus, APM for monitoring your application's performance. Dashboarding, collaboration tools, and alerts let you develop your own workflow for observability and incident response. Datadog integrates seamlessly with all of your apps and systems; from Slack, to Amazon web services, so you can get visibility in minutes.

Go to softwareengineeringdaily.com/datadog to get started with Datadog and get a free t-shirt. With observability, distributed tracing, and customizable visualizations, Datadog is loved and trusted by thousands of enterprises including Salesforce, PagerDuty, and Zendesk. If you haven't tried Datadog at your company or on your side project, go to softwareengineeringdaily.com/datadog to support Software Engineering Daily and get a free t-shirt.

Our deepest thanks to Datadog for being a new sponsor of Software Engineering Daily, it is only with the help of sponsors like you that this show is successful. Thanks again.

[END]