SDS PODCAST EPISODE 717: OVERCOMING ADVERSARIES WITH A.I. FOR CYBERSECURITY, WITH DR. DAN SHIEBLER

Jon Krohn: 00:00:00 This is episode number 717 with Dr. Dan Shiebler, Head of Machine Learning and AI at Abnormal Security. Today's episode is brought to you by Grafbase, the unified data layer, by ODSC, the Open Data Science Conference, and by Modelbit for deploying models in seconds.

00:00:21 Welcome to the Super Data Science podcast, the most listened-to podcast in the data science industry. Each week we bring you inspiring people and ideas to help you build a successful career in data science. I'm your host, Jon Krohn. Thanks for joining me today. And now, let's make the complex simple.

00:00:52 Welcome back to the Super Data Science podcast. Today, the wildly intelligent and clear-speaking Dr. Dan Shiebler returns to the show for his fifth visit. Dan is Head of Machine Learning at Abnormal Security, a cybercrime detection firm that has grown to over $100 million in annually recurring revenue in just four years. And there he manages a team of over 50 engineers. Previously he worked at Twitter, first as a staff machine learning engineer, and then as an ML engineering manager. He holds a PhD in AI theory from the University of Oxford and obtained a perfect 4.0 GPA in his Computer Science and Neuroscience joint Bachelor's from Brown University.

00:01:30 Today's episode is on the technical side, so it might appeal most to hands-on practitioners like data scientists and ML engineers, but anyone who'd like to understand the state-of-the-art in cybersecurity should give it a listen. In this episode, Dan details the machine learning approaches needed to tackle the uniquely adversarial application of cybercrime detection. He talks about how to carry out real-time ML modeling, what his PhD research on Category Theory entailed and how it applies to the real world, and he opined on the major problems facing humanity in the coming decades that he thinks AI will be

able to help with and those that he thinks AI won't. All right, you ready for this absorbing episode? Let's go.

00:02:15 Dan, welcome back yet again to the Super Data Science podcast. I guess you're in New York as usual.

Dan Shiebler: 00:02:22 That's right. Thanks, Jon. Happy to be back.

Jon Krohn: 00:02:25 Yes, we've got an exciting episode today. Previously you've been on the show going all the way back to episode number 59, and then you came back while Kirill was still hosting. That was episode number 345. My first time hosting you was episode 451. And then episode number 630, you did a Five-Minute Friday-style episode where you answered a question specifically about resilient machine learning, which actually will build upon a bit more in today's episode. Something kind of cool for our listeners to check out, if you don't watch the video version, today I'm filming from Detroit and this hotel that I'm in, the Foundation Hotel, Detroit, it's wild. I just was expecting to record in my hotel room, but I was leafing through the hotel booklet and they have a dedicated podcast studio, so actually I've got this suite on air sign behind me. Other than that, it's just a quiet room. There's lots of curtains and stuff. I'm using all my own equipment. But yeah, it's kind of a cool look for the video today.

00:03:40 So we've got tons of content for you, Dan, building on the resilient ML stuff a bit and focusing on what you've been doing since your last episode. So it's been several years now since your full-length episode with me. And in that time there's been a lot of changes. Most notably, you're working at a firm called Abnormal Security, and so you're addressing the high-stakes challenges of cybercrime over there with machine learning. So what makes this particular adversarial machine learning challenge where you're not just building a machine learning model that is

acting in a vacuum, it's very much the opposite. The models that you build, people are trying to reverse engineer them on a regular basis to be able to overcome the kind of security that you're developing with your ML models. So this kind of adversarial scenario, this adversarial machine learning challenge, how is that unique relative to the other kinds of machine learning models that you've built historically?

Dan Shiebler: 00:04:42 Totally. So Abnormal Security is a company that builds detection systems for identifying cyberattacks that are coming in through email and through accounts that people have on various SaaS platforms including email, but things like your Slack, your Okta, your Zoom. So there's really two kinds of attacks that we're concerned about. One is an account has been compromised and we're trying to identify that this account has been taken over, the attacker has gotten the credentials, and now the person who's operating this account is no longer the account owner, it's the compromised attacker. And the other option is inbound attacks. There's an email message generally or sometimes other types of messages where an attacker is transmitting a payload, which could be a phishing link or a message that's eliciting the recipients to update their bank account information or perhaps malware or anything else that's the initial vector to begin a cyberattack.

00:05:42 And so the machine learning models that we build operates at the level of individual events, which are the messages that are being sent, the sign-in events that we're observing for these accounts and a number of other kinds of events. And at each of these events we're trying to identify is this an attack, is this malicious or is this normal behavior? And this is a very adversarial situation because the person on the other end, the attacker is going out of their way to try to cloak their actions. They're trying to make the messages that they're sending look as

similar as possible to safe business messages, they're trying to sign in utilizing infrastructure and technology that allows them to cloak the fact that they're an attacker, hide their identity and obfuscates anything that they're doing so that it looks like a normal individual.

00:06:33 And so our machine learning models that we're utilizing need to take advantage of what are the things that the attacker might not know that we know, for instance, or how do we try to build something that is resilient to different kinds of modifications that the attacker might utilize and can really get at the heart of what separates the normal business traffic and communications from what the attacker is attempting to do.

Jon Krohn: 00:07:05 Yeah. So I imagine this involves a broad range of different kinds of models. I know you've mentioned online in some of our research we dug up that some of these models involve probabilistic models that are relatively straightforward, I imagine relatively efficient, all the way up to large language models, which presumably they're a lot more expensive to run. They aren't necessarily as fast at inference time. So given the kinds of attacks that you're trying to identify, how do you decide what kind of model you're going to be using for a particular type of threat?

Dan Shiebler: 00:07:47 There's really three kinds of models that we utilize, which each of them try to capture something a little bit different and have different trade-offs in terms of what they have access to, the cost that they require in order to utilize them, their speed at which they can be invoked, and their efficacy, the range of different kinds of attacks they can effectively catch. So everything that we build, all of the models we build are powered by aggregate signals, which are the most important component of our approach towards cybersecurity. So basically this is a special type of feature that we build over raw data that then powers all

of the different kinds of our models. And so this is the sort of foundation of our detection strategy.

00:08:33 And so these are aggregates over raw email sign-ins and other kinds of raw events at individual entity levels. So, for example, we would aggregate for a particular person all of the emails that that person has received and be able to say things like, how many times has they received an email that has this header in it or this kind of phrase in it, or this kind of attachments that's routed through this IP, utilizes this infrastructure, has this HTML tag? Each of these different kinds of little individual signals that could lead to identifying some information about this email, about this sign-in events, we aggregate at the level of each person who's receiving each piece of infrastructure that's sending, each piece of infrastructure within IPs and domains that messages and sign-ins get routed through.

00:09:25 And through this, we build this historical picture, basically a summarization of everything that's happened up until this point that serves as our foundational feature infrastructure. So this is a very structured way of building representations of features. And so it means that there's now a number of different ways that we can utilize these derived signals in models effectively. And so the simplest thing for us to do is heuristics and rules built on top of these signals. This is already a very heavily data-driven approach. Fundamentally, these aggregate signals themselves are basically simple models. They are basically probabilistic models that demonstrate what's the percentage of time given some condition that X is true, that you could construct these heuristics and rules to look very similar to a Bayesian network on top of these individual aggregate signals with different sorts of conditionals that you're applying and different kinds of derived probabilities that you're building on top of it. The next level of sophistication is basic trained models. So

this would be things like logistic regressions, XGBoost, and we like Deep & Cross networks is our neural network architecture of choice for on this kind of multi-

Jon Krohn: 00:10:37 Deep & Cross?

Dan Shiebler: 00:10:39 Deep & Cross networks, yes.

Jon Krohn: 00:10:41 I've never heard that before.

Dan Shiebler: 00:10:43 It's a network architecture. It's very popular in ad tech. We have a number of people at Abnormal who have previously worked in ad tech. Basically it's a type of neural network where you consume both raw signals utilizing a deep layer as well as cross signals where basically you build derived signals from your individual raw features and then have those derived signals so you learn the derivation of your cross signals and then feed that into a deep network. And so the cross layer functions in a way where it can take things like here's a frequency that some attribute is true, and here's a Boolean signal that says, yes, true or not true. And then you can do a multiplication of these in order to build a derived signal, for instance.

00:11:35 And so these are sorts of cross features that the space of potential cross features that you could build is very, very large. And utilizing this network architecture allows just to tend to the specific cross features that are most valuable. So it allows you to sort of remove a little bit of the work required to build sophisticated cross features without having a giant parameter space. And so it's nice for cases where you have both deep embeddings and a lot of Boolean and continuous features that you're consuming at the same time and you're trying to... You want to do something a little bit different with the dense continuous signals within an embedding and the individual Boolean and continuous signals that

represents more sparse information. And so the Deep & Cross network enables you to, it's like an inductive bias that's built onto that kind of architecture. We utilize normal feedforward networks as well in this intermediate category of models that we train, but we like the Deep & Cross because we've seen good performance with it and there's nice implementations online.

Jon Krohn: 00:12:42 Yeah, that's very cool. I hadn't heard of this kind of cross layer specifically in a deep neural network before. In my mind, I imagine the workhorse layer, a dense layer as being capable of doing some of the things that you're describing. So a dense layer should be able to, in many circumstances, identify whether a cross of two input features are together, creating a lot of signal because that dense layer, what that means, that denseness is that it recombines any possible inputs from the preceding layer. So it sounds like this kind of cross layer as opposed to being a general purpose dense layer that happens to be able to do those kinds of multiterm interactions, this cross layer is explicitly designed to do that.

Dan Shiebler: 00:13:39 Yeah. So it's basically, it's similar to how a convolutional neural network is inherently less expressive than a feedforward neural network, but still more performance on image tasks than a raw feedforward network is. It's embedding in the inductive bias that these particular kinds of multiplications between your features to cross them is a useful thing that you want to do for that category of feature. We utilize it when we have these signals where there's one signal that tracks the frequency of an event, and then another signal tracks the presence of that event, where these are two features that really only make sense when they're combined together and are very difficult to cross with other kinds of signals that their poignancy relies on their combination. And so building those crosses explicitly through the cross network is useful for that kind of application.

Jon Krohn: 00:14:39 Very cool.

00:14:41 This episode is brought to you by Grafbase. Grafbase is the easiest way to unify, extend and cache all your data sources via a single GraphQL API deployed to the edge closest to your web and mobile users. Grafbase also makes it effortless to turn OpenAPI or MongoDB sources into GraphQL APIs. Not only that but the Grafbase command-line interface lets you build locally, and when deployed, each Git branch automatically creates a preview deployment API for easy testing and collaboration. That sure sounds great to me. Check Grafbase out yourself by signing up for a free account at grafbase.com, that's G-R-A-F-B-A-S-E.com.

00:15:23 Well, so I digress a bit into this. So you were saying that there's three kinds of model that you use. So you were describing the first was heuristic models, rules-based ones, and then you were kind of talking about intermediate complexity machine learning models, so things like random forests, logistic regression models, these deep & cross deep neural networks. So yeah, I don't know if I missed any there, if you wanted to go any deeper on that one or if you wanted to jump now to model type number three.

Dan Shiebler: 00:15:53 Yeah, so model type number three is large language models. We utilize both the out-of-the-box OpenAI APIs for certain tasks as well as building our own fine-tuned variance of, we've utilized Falcon and LLaMA and fine-tuned those to a few different tasks. When you think about these three different categories, they kind of grow in crescending amount of costs required to run it, increased latency and decreased speed, and different characteristics in terms of their ease of use. The first category and third category are perhaps the easiest to use and modify because large language models you can repurpose with prompt engineering and rules, you can

repurpose by tweaking things. Whereas the intermediate category of deep neural networks and such really requires retraining in order to incorporate new information. And so all three have pros and cons and can be applied to different types of use cases and challenges within the ecosystem of different kinds of attacks we're trying to catch for different customers.

Jon Krohn: 00:17:05 Nice. That's a really good high-level summary of the kinds of models that you work with. It's interesting to think about how that third tier, those large language models, that they've become so complex now that they're actually, as you say, I hadn't thought of it this way before, but they become as easy to use as a simple heuristic model because you change your prompt and they're so flexible, you don't need to be retraining the entire model. Maybe you could potentially in that third category, you could also be inserting in some PEFT layers and those are then very fast to fine-tune. So you could have this huge architecture, like you mentioned Falcon, it's a 40 billion parameter model, but you could use parameter-efficient fine-tuning, PEFT, to fine-tune to some specific task of yours, maybe just have a few hundred or a few thousand examples of some tasks that you'd like it to be able to specialize in. And you can train that in minutes or hours even though the architecture is so gigantic because there might only be a million or so or 100 million parameters that you're training in this parameter-efficient fine-tuning technique as opposed to trying to do the whole 40 billion.

Dan Shiebler: 00:18:24 It is definitely the case, we've observed at least, that once you go down the route of fine-tuning these models, you lose some of their generalizability and ability to adapt them to different tasks. We've fine-tuned these models as sort of uber classifiers that can be applied to classification tasks by taking them and utilizing their size and their really deep understanding of raw fundamental concepts and ability to reason as basis for being able to

be applied to representations of our data in a text form that they can understand that then they're fine-tuned to understand better.

00:19:09 We have a couple of different kinds of message classification tasks that we operate, both just identify whether or not something is an attack as well as identifying attacker objectives and triaging messages that are submitted to a phishing mailbox product that we operate as well. And each of these are slightly different kinds of tasks that require slightly different kinds of behavior that involve some amount of human interface that we've seen in the past. And that's where we're trying to incorporate large language models to reduce the human burden on the areas that involve that because the cost characteristics of these models make them very, very difficult for us to utilize them in an application like scanning every sign-in or every email that we process. It is really cost prohibitive to do something like that with models of this size, but something that already involves some human interaction is much more manageable to incorporate these models in.

Jon Krohn: 00:20:10 Nice. Yeah, that makes a lot of sense. Yeah, large language models, being able to augment or automate something where a human would be required is probably going to be more cost-effective, whereas trying to have huge volumes of emails be processed by an LLM would be crazy, crazy expensive. One of the big things about training any machine learning model, particularly when we're talking about that intermediate tier, your second tier, so random forest, logistic regression, now classical 10-year-old deep learning architectures, one of the big things is looking at in your kind of scenario, you'll have some true state of the world that you're trying to model. You have correct labels that you're trying to guess with your machine learning model. And anytime we're trying to do that classification, we end up in machine learning with

some false positives and some false negatives. And obviously we want to try to minimize both. But in your context in cybersecurity is one of those false positives or false negatives kind of worse than the other? And do you try to minimize one in particular?

Dan Shiebler: 00:21:28 It's really a balance to be fair. I mean, I think the worst thing that can happen is that you miss a really serious attack and it causes a lot of damage to customers. So in that sense, false negatives are more of a larger existential problem. The worst kind of false negative is the worst of all, but a false positive problem and a high rate of false positives is equally bad because it incentivizes... Businesses can't operate if people are being stopped by their security solutions from engaging in normal business. And so customers will end up putting overrides and ignoring remediation criteria and then they'll expose themselves to exactly those kinds of really bad false negatives and will have no ability to control for it at all.

Jon Krohn: 00:22:20 It's the boy who called wolf kind of scenario.

Dan Shiebler: 00:22:24 Absolutely.

Jon Krohn: 00:22:25 Cool. Yeah, that's interesting. In my head I was expecting you to answer that question and just say that false negatives are the worst. We got to make sure we avoid those. But, of course, if your clients are getting false positives all the time, then they're just going to ignore your tool and then they're going to miss the real deal. So in the last few years, I understand that the threat landscape has changed a fair bit. So how have you had to adopt your models out of normal security to handle those new challenges?

Dan Shiebler: 00:22:54 Traditionally, cybersecurity solutions function by identifying indicators of compromise and stopping threats based on matching indicators of compromise between a

particular threat and a new thing that may or may not be a threat, like a new message or a new kind of sign-in. And an indicator of compromise in this case, it's a smoking gun, a link that is known to be bad, a domain that is known to be bad, an IP that has poor reputation and attachments that hash matches some known malware. There's many different kinds of indicators of compromise. But what has happened is that the costs and ease with which attackers can switch the root tools that they're utilizing has simply gone down. Attackers have had better and better access to systems that have allowed them to evade the types of recognition of indicators of compromise and send out attacks that don't match the patterns of any previous attacks with much, much larger scale and much, much higher degree of ability to avoid detection.

00:24:18 And this has certainly gotten substantially worse with the introduction of generative AI tooling. Generative AI tooling in particular enables the personalization of attacks to a particular recipient by combining something like somebody's LinkedIn profile and integrating that seamlessly and entirely automated into social engineering scams that are highly targeted for that person. And this avoids both the indicator of compromise style checks for the templates that phishing emails would normally match, as well as just increases the degree to which these kinds of messages and attacks look to the recipients to be malicious. So our strategy at Abnormal is to avoid an overfocus on indicators of compromise as the core tenet of our strategy. Our strategy is instead to focus on identifying abnormalities and individual pieces of communication and emails and sign-ins that make them different from the types of normal business communication. And rather than try to root cause an attack, instead try to spot things that don't look like the normal safe communication. So rather than "is attack" we'd use "not safe" as our core strategy and core objective.

00:25:53 And so this enables us to be much more resilient to changes that attackers could make to their attacks to try to avoid indicators of compromise and also enables us to play to the greatest advantages that security defenders have over security attackers, which is knowledge of the targets. That attacker whose attacking somebody doesn't know what's in that person's inbox, they don't know what emails that person received yesterday. They may know a little bit about their target if they've utilized open-source intelligence, but they are unlikely to know nearly as much as a security solution that's plugged into that person's accounts, has access to that company's data and information. And by leveraging this advantage, this information asymmetry that defenders have access to, we're able to most effectively fight back against attackers. This expands very, very naturally to the growing threat posed by generative AI tech.

Jon Krohn: 00:26:55 Fascinating and very well said. You have such a crisp way of speaking, it's so easy to understand you. Thanks. I mean this is now getting a little bit into the future maybe, although maybe not that far in the future. Do you ever worry about how generative AI, I don't know, some kind of open-source alternate of like GPT-5 or GPT-6, something of that kind of capability that might be here in a few years, that's open-sourced and so can be used for malicious purposes, do you ever worry about LLMs being able to go beyond the kinds of attacks that you're describing here, like this personalization which allows for the automation of, say, phishing attacks where you can instead of needing to have a human be researching somebody and coming up with points for a phishing email that might make them feel like this is a trusted entity?

00:27:57 The LLM can now do that automatically, but in the future and with some kind of open-source maliciously usable GPT-5 or GPT-6 variant, this might be able to do much more. This might be able to plan attacks. It's not just

generating the text, but actually, in some ways it's like an independent malicious actor that some malicious human can kind of just set in motion and say, here's some money, get as much possible money back. Is that something you ever spend time thinking about or is that just too far out?

Dan Shiebler: 00:28:37 I think multistage planning with deep reasoning is very, very difficult. I think it's substantially more difficult than solving a range of different problems. So I am less concerned about this from a sort of general existential threat perspective. But that said, I think in cybersecurity there's a heuristic that you could utilize for identifying what kinds of attacks you'll see in the future that has proven to have been pretty effective. And this falls closely within that, which is that cybercriminals are financially motivated by and large. Not every cybercriminal is financially motivated. There are state actors that exist as well, but there are a tiny percentage of the overall set of cyberattacks.

00:29:24 The vast majority of cyberattacks are sent by people who are trying to receive a return on an investment. They've spent some money to invest in technology to cloak their identity, technology to acquire internet assets that they'll utilize to send out attacks. These are domains and IPs and types of internet connection ISP variants and they will try to get a return on the money that they're spending. And the attack strategies that enable them to get a return on the money that they're spending have become more and more sophisticated. In the past, if you were going to do something like a spear phishing email, you needed to spend a great deal of time investigating your targets and that time is money basically because you are assuming you're getting paid on some hourly basis.

00:30:14 So think of yourself as a cybercriminal comparing what you get paid at McDonald's to looking up someone's

LinkedIn and utilizing it to generate spear phishing emails. If a tool lets you send out 10 spear phishing emails in the time it previously would've taken you to send one, now you're going to be able to start sending more of these. And there's certain kinds of attacks that are very sophisticated that exist already. We see these types of vendor fraud attacks where a attacker will compromise a legitimate vendor's account, which is a very expensive thing to do. Purchasing an account of an email address of someone in billing at a Fortune 500 company on the dark web, that's a very expensive asset that you're unlikely to have for very long because the company likely has a security team that's going to find you. And so it's a short-lived, expensive asset that an attacker is acquiring and then attempting to get as much money out of it as possible before they lose access to that asset.

00:31:16 So these kinds of attacks are very sophisticated and very difficult to detect, but we do see some of them. We did build models and systems to detect them and it's a reasonable heuristic that things that we see a small amount of now because of their sophistication and because of the amount of money that attackers need to spend in order to generate them will become cheaper for attackers to send in the future. As technology advances, as AI advances, as cybercrime builds a larger ecosystem of tooling and systems, attackers will be able to send more and more sophisticated attacks at a lower and lower price point, which will mean that the things that we see at a maybe once a week basis will become things we see every day or things that we see 10 or 20 a day as this thing moves closer.

00:32:08 I think that this case that you're describing right now with a agent that is operating the planning and prospecting of attacks at a multistage basis where first they send a series of attacks to gain phishing emails at that person who's in billing at some vendor in order to get

access to their account, then they have access to that account, and then sending messages from that account to the various customers of that product to tell them to update their bank account info. There's this kind of complex, sophisticated, multiple stage attack that having that reaching lower price points, I think that it's feasible to imagine that that could happen. And the best way to protect against it is to take seriously the attacks that we see rarely today with the expectation that they will become more and more common in the future.

Jon Krohn: 00:33:03 Be where our data-centric future comes to life, at ODSC West 2023 from October 30th to November 2nd. Join thousands of experts and professionals, in-person or virtually, as they all converge and learn the latest in Deep Learning, Large Language Models, Natural Language Processing, Generative AI, and other topics driving our dynamic field. Network with fellow AI pros, invest in yourself in their wide range of training, talks, and workshops, and unleash your potential at the leading machine learning conference. Open Data Science Conferences are often the highlight of my year. I always have an incredible time, we've filmed many SuperDataScience episodes there and now you can use the code SUPER at check out and you'll get an additional 15% off your pass at O-D-S-C.com.

00:33:51 Nice. Yeah, that is a very sensible heuristic that as soon as you started to explain it I was like, yeah, that makes a lot of sense. So that certainly is something to keep an eye on. I guess, we don't know how quickly, if ever, machines are going to have that multistage planning capability, but I don't know with how blown away I was in the jump from GPT-3.5 to GPT-4. I'm like being surprised should be unsurprising.

Dan Shiebler: 00:34:24 Certainly. Certainly.

Jon Krohn: 00:34:26 Yeah. Okay. So clearly that kind of heuristic is something that's useful for helping you figure out what kinds of models you might need to start prototyping now. I understand that you also do head-to-head competitor comparisons on a weekly basis. So how does that help you with refining your models as well?

Dan Shiebler: 00:34:45 Totally. So I'll just talk a little bit about the process. So most companies of decent size need to spend a decent amount of money on email security. Emails are the primary vector by which large businesses get attacked with malware, phishing, invoice fraud, et cetera. And there's a number of different ways that businesses can try to protect themselves. The most common way is purchasing solutions like Abnormal Security, and we have many competitors that offer similar products that try to protect customers from these kinds of attacks. And because of the sheer volume of these attacks and the length of time that email security has been around, this has been a product category for one of the longest time periods within the SaaS types of products, is email security measured in the timeframe of decades rather than years like most SaaS products.

00:35:48 We have a pretty easy to understand way to compare two products. You simply install both and you see which one catches more attacks and which one generates less false positives. Very simple to see, very simple to evaluate. And every week our sales team works with customers to install Abnormal Security in their environments and compare us against either the customer's current email security solution or competitor email security solutions the customer is also considering. Normally customers will consider a number of different solutions at different price points, observe which ones require the most effort for them to manage, which is basically the same thing as false positives, less false positives means less effort to manage, and which ones protect their customers the best,

which is the same thing as false negative rate. Lower false negative rate, you're better protecting the employees at the business. And if we are able to find a tax that no other solution finds and generate fewer false positives than other solutions, then we'll win the deal and our revenue will increase. And if we're not, then we won't win the deal.

00:36:54 And so this is a very simple and exciting space to be in as a machine learning engineer because it's relatively rare that you get to build technology that is placed immediately into such a clear-cut competitive environment where you are immediately tested not only against adversaries, but also against other solutions attempting to do the exact same thing that you are doing. You see very, very quickly how good your system is and how you can measure that immediately in terms of the dollar value that businesses will pay to remove their current solution and replace you with Abnormal Security. So this serves as a strong rallying point and motivation function for the detection team and for Abnormal Security as a whole.

Jon Krohn: 00:37:45 Nice. That is a really cool process and probably a kind of process that, not probably, definitely the kind of process that you described there is something that's easy for me to imagine for my business and probably a lot of other people could imagine for theirs. Comparing false positive, false negative rates against your competitors and probably a lot of clients or prospective clients would be able to estimate how much each false positive costs them, how much each false negative costs them and just be able to determine, okay, so going with this product at this price point, I'm going to save this much overall and that's the best one to go with. Nice. Tying back to your previous, your most recent Super Data Science episode number 630 where we were talking about resilient machine learning. So maybe you could quickly recap what that

resilience means. Basically it's this idea of having a robust machine learning system. How is that particularly important in cybersecurity?

Dan Shiebler: 00:38:48 So resilient machine learning means, as you say, a robust machine learning system and specifically building your engine so that it is unlikely to fail catastrophically. There will always be problems that you face. Sometimes these problems are acute problems where a single system goes down, perhaps there's an outage in a service, someone pushes bad code, some type of data gets deleted accidentally. Sometimes it's changes in the underlying data distribution on your side. Perhaps you onboard a new customer that's in a new industry that you've never seen before or you have some kind of change in the way that you are categorizing the events that you're seeing such that it changes the underlying data that powers your features, your aggregates, for instance. And sometimes it's adversarial in cybersecurity. This third category is constant. Attackers are changing what they're doing every week and every day in order explicitly to fight against the system that you're building.

00:39:58 So there's a lot of strategies that you can apply to build this kind of failure resilience into your machine learning systems to make it so that when things change, your system doesn't change with it. And so this includes the data distribution shifts that is normally thought of as a core problem within all machine learning systems. You train on one set of data, you launch, now there's a new set of data and you have to deal with that. And so that's one part of it, but it also incorporates things like feature dropouts where you have certain areas or signals that you rely on that are not available in certain environments or in certain circumstances and it needs to be able to operate even when you have these kinds of outages that occur and you still need to be able to provide protection for your customers.

Jon Krohn: 00:40:49 Nice. So that makes a lot of sense in cybersecurity, but then outside of cybersecurity, why might our listeners be interested in the concept of resilient machine learning related to whatever their kind of data science modeling is or the kind of software engineering that they do or the kinds of systems that they architect?

Dan Shiebler: 00:41:06 So building systems that are resilient to changes in your customer distribution is a constant issue that every data scientist faces, especially at a growing business. When you have your initial set of customers and there's an initial behavior that's present in the kinds of data that you're seeing, you want to be able to build your systems so that when that distribution changes, when new customers are onboarded, you are able to quickly adapt to these new distributions.

00:41:42 And so there's two main principles that you could apply towards having this kind of quick adaptation to new customers. One is fast retraining. Maybe you build a machine learning model, you train it on your data, then you have new data that's coming in. If you have assembled a concrete coherent data pipeline and data labeling pipeline, then you'll be able to retrain your model. And sometimes you could even automate the retraining process depending on the nature of your data and the environment that you're operating in.

00:42:14 Another approach that we lean into, we lean into both of these approaches at Abnormal Security, but one more approach that I think is very under-discussed for quick adaptation but has a lot of usability in this kind of case is to utilize features that represents the data distribution itself. And so to make this clear perhaps rather than have a categorical feature to represent something like a user that's like, this is this user's ID, we're going to represent them with a single value that's going to go into a one hot lookup, then you're sort of expecting the model to

memorize this user. And if this user changes their behavior in the future, you need to retrain the model to update it.

00:43:01 An alternative is to utilize features like, what are the number of accounts that this user has followed in the last 10 days? I'm giving a Twitter example. What are the topics that this user has liked on tweets in the last seven days? So these are features, but they're features that represent current data. They represent the past. At Abnormal, this is the aggregate features that we were talking about earlier, how many emails has this person received from this kind of account at this point of time and day in the past? This is a feature representation of the current information.

00:43:41 And so in the case where you have one customer that you're building a model on and then another customer that gets onboarded, even if that second customer is a very different distribution, maybe that first customer only had a 10-person customer service team, this new customer has a 500-person customer service team. If you've represented what it means to be a customer service agent in terms of these kinds of signals, like how frequently does this person receive emails from the outside as opposed to things like memorizing who these individual people are, even memorizing a categorical signal on his customer service, then you'll be able to better adapt to these kinds of new circumstances because your features themselves will be modified and will adapt to this new distribution.

Jon Krohn: 00:44:33 Deploying machine learning models into production doesn't need to require hours of engineering effort or complex home-grown solutions. In fact, data scientists may now not need engineering help at all. With Modelbit, you deploy ML models into production with one line of code. Simply call modelbit.deploy() in your notebook and

Modelbit will deploy your model, with all its dependencies, to production in as little as 10 seconds. Models can then be called as a REST endpoint in your product, or from your warehouse as a SQL function. Very cool. Try it for free today at modelbit.com. That's M-O-D-E-L-B-I-T.com.

00:45:11 Nice, nice, nice. Nicely said. Lots of practical tips there for any of our listeners on resilient ML. When did you start getting into this? Was this something that you started getting into back at Twitter? It's not related to your PhD stuff directly, is it?

Dan Shiebler: 00:45:27 It was pretty important at Twitter. At Twitter, one of the core issues that I faced within the revenue science organization, which was the organization that operates the machine learning models for ad serving, was fast performance for a new ad campaign. So customers would launch ad campaigns that have a number of different creatives, line items that combine a number of different creatives and want to target a particular audience. What we need to do is very quickly identify what are the types of users for whom these ads will be most poignant. And we don't have substantial categorical information about the ads themselves and even the users themselves, their ad interaction behavior can change quite quickly. If they previously were in a situation where they weren't getting any ads they were interested in, now suddenly where they're really into sports and sports betting advertising has suddenly been legalized, and now we could show sports betting ads, for instance, that changes their behavior.

00:46:29 And so being able to represent the most recent picture of behavior at each of these categorical signals, very, very critical towards out-of-the box performance of being able to give that kind of quick turnover. Advertisers would generally tolerate worse performance for a couple of days,

but not for a couple of weeks after beginning a campaign. And so having that fast adaptation, you can't really rely on training a model that's going to be able to have the capacity to capture that at that kind of scale that quickly.

Jon Krohn: 00:47:06 Nice. Yeah, that makes a lot of sense. So on the note of reaction times and speed, another obviously super critical thing, whether it was Twitter before or Abnormal Security now is the real-time nature of processing. I imagine, I mean it's super critical in both situations. It's hard to say it's more important than one or the other. Obviously in a social media platform, people are expecting news in real time, for example, they're expecting updates from people that they're following in real time. But with cybersecurity, arguably there's a bigger danger to not being real time. Obviously it's super important in cybersecurity to have real-time processing as well. Are you able to go into any particular kinds of infrastructure or technologies or techniques that you employ to handle massive traffic in real time?

Dan Shiebler: 00:48:08 Our most direct approach towards real-time information is aggregates. We utilize airflow to instrument model retraining on a weekly basis because we're trying to take advantage more of customer shifts than attacker shifts. Attacker shifts can happen much faster than that. And so we utilize our aggregate engine for identifying and adapting to attacker shifts. So this is both at the IOC level of trying to, when we miss an attack or if we see a particular IOC within a new attack that we've caught, now being able to ensure that we catch everything else that has that same IOC. So basically utilizing a combination of abnormality to catch the first attack and then IOC to catch everything else that looks similar to it, we need to very quickly identify, okay, this signal is now something that we've seen in a malicious message, we need to distribute this out to somewhere else.

00:49:12 To make this very concrete, just to give the situation, then I'll talk about the technology. So if the attacker has purchased a domain and they're utilizing that domain to send out messages that include a malicious link with that domain, maybe they send out 100 messages that all include this domain and maybe we were able to identify that some set of these messages are malicious and we're able to identify this by looking at the differences between the way that this message was sent and the kinds of messages that the person who's receiving this normally receives.

00:49:51 But maybe we don't do that for every one of these 100 messages. Maybe 10 of these messages hit people who receive a lot of messages that look really sketchy but are totally normal. And because of that, we're not able to spot on those 10 people that this message was bad. But we have seen on our other 90 that it was bad because those were sent to people who receive mainly normal messages. And so now we have this new piece of information, which is that this domain is bad and we have this message that we wouldn't be able to identify as bad without this piece of information, now at risk of hitting this user, this individual. So this is a case where we need to react very, very quickly to pull this message and stop it from doing damage because we've identified that this indicator of compromise is bad by leveraging this information, and now we need to act on it.

00:50:47 So we utilize a Redis-based key-value store to track these types of indicators of compromise. So we stratify based on every kind of decision that our system makes and track each of the different types of indicators of compromise you could extract from messages or sign-ins in this system and utilize a triggering replay system to identify based on a last N aggregate within Redis. When any of these individual counts gets triggered, we then submit

from the last N Redis aggregate back to our core reprocessing engine.

Jon Krohn: 00:51:33 Very cool. Very cool example. You said a term in there which maybe you did define and I just missed it, but IOC.

Dan Shiebler: 00:51:42 Yeah. So indicator of compromise.

Jon Krohn: 00:51:44 Oh, yeah. So you talked about that earlier in the episode, but I wasn't used to it as an acronym yet. Nice.

Dan Shiebler: 00:51:48 Yes, it's an acronym that I'd never heard of before going into the security world, but it's constantly bandied about. It really just means anything that could indicate that something is bad. Generally it's referring to IP addresses and domains and email addresses and file hashes and things like that, but there's a lot of other things that it could refer to as well.

Jon Krohn: 00:52:14 Nice. So you talked about this a little bit earlier, but maybe we can dig into it a bit more. When we talk about real-time processing, you've kind of now covered that, things like this Redis key-value store allow you to do that efficiently. In previous answers, you talked about resilient machine learning being adaptable. In practice, how does that mean that you need to be updating your models? Is there a routine to updating machine learning models or is it event driven? How does that work?

Dan Shiebler: 00:52:51 We've built an auto-retraining framework that enables us to retrain our models on a regular cadence. We maintain a large number of different machine learning models, which we retrain on different cadences. Our auto-retraining pipeline covers our core models, our most important models that we hook up into it. And it's a series of different steps to do a auto-retraining. First, it's to collect all of the data that we're going to utilize, to

process it and extract features from that data. We need to actually run the training process.

00:53:28 And then the most important stage is the valuation. We need to identify that if we take the model that's currently deployed and turn it off and turn this new one on, we're not going to suddenly flood a bunch of customers with false positives. We're not going to stop catching attacks that we're currently catching. We're not going to dramatically increase your cost or latency or anything else. And so we have a large suite of tests that run simulations with this new model in place of the old model. And so this is a pretty heavy, expensive process, which is why we don't set this up for every single model we deploy, only our most important critical models. For our faster adaptation, we primarily rely on aggregates for capturing changes in data distribution, and we utilize auto-retraining as a way to readapt as customer distributions shift over time and take on new signals.

00:54:21 One thing that's relatively interesting about our normal process is we are constantly adding new signals. We're constantly identifying what's a new kind of aggregate to build, what's a new kind of data source to subscribe to, to be able to understand more about the indicators of compromise within emails or sign-in events. What are new ways that we can transform, apply natural language processing, apply clustering techniques to better understand each piece of data that we process. And each one of these signals is something that could be useful in a model retraining. And we set up our auto-retraining process so that it automatically consumes certain kinds of signals that the team adds.

00:55:06 We're able to operate in a mechanism where one group is building new signals and then immediately setting up heuristics around those signals to utilize these heuristic kind of models. And the auto-retraining process picks up

these signals automatically into the models that regularly retrain. And so in this way, we are able to most efficiently have this feedback loop between the very hands-on work to optimize a signal so that the signal is powerful enough to work in a heuristic and that signal then being incorporated into our next automated retraining of our core machine learning models.

Jon Krohn: 00:55:41 Nice. So in the software engineering world, there is a term CI/CD, continuous integration, continuous deployment, that is a very common practice these days. So the analog for what you're describing, could we call that CI/CT, continuous integration, continuous training, for a lot of these core models that are in your auto-retraining framework?

Dan Shiebler: 00:56:08 To be honest, I would say no. I generally think of continuous training as being a somewhat separate thing, where you're really looking at less than a 10 to 20-minute difference between when a sample shows up and when the weight update has been applied to the model that's deployed in production. At Twitter, we had several systems that utilized this framework where we did have what you would call CI/CT, where we had models that were deployed, and the time between when a person clicked on an ad or chose not to click on an ad, and when that fact had been propagated into a feature update or a back propagation gradient step for the model that serves ads was less than 20 minutes.

00:56:57 At Abnormal, it's going to be a substantially longer period of time because of our auto-retraining. But there is a very fast turnaround time towards that information being incorporated, but it goes through the aggregate signals. It goes through the fact like, after this message is sent, we'll extract all these signals and update the aggregates. The features and the next prediction are different. So you can think of it as like... It's all sort of the same thing. When

you blur your eyes, it's take a step back whether you're applying this update to the features or applying this update to the weights of the model, but at Abnormal, only real-time updates are being applied to features. I think of continuous training as referring to the real-time updates being applied to the weights.

Jon Krohn: 00:57:41 Yeah. So in CI/CT, like you were doing at Twitter, you're talking about some actual training of the model weights like a back propagation step, whereas the kind of retraining that you're doing with your auto-retraining framework, this is more holistic. So it's kind of like, it's going all the way back to feature creation, aggregation, so you could take advantage of the kinds of cross terms that you were describing way back earlier in the episode, being able to be recreated afresh. So it's a more comprehensive retraining. It's not just one step of back prop.

Dan Shiebler: 00:58:23 Yeah, that's right.

Jon Krohn: 00:58:24 Cool. I don't know how much you can get into this kind of thing, but I can at least ask, but are you able to give examples of instances where a cybersecurity system would miss a threat or identify a false positive and then requires you as a human or your team to come in and make some changes to address that kind of miss?

Dan Shiebler: 00:58:52 Yeah, I can give one example. I'll talk about it a little bit more vague. There's a type of pattern that we observe in cybersecurity where there's things that are sometimes referred to as the Nigerian Prince Scam, which is essentially a type of scam where you begin the scam by saying, "I want to give you money," in some way or another, and the person then engages and they trick them into giving bank account details. And so sometimes this is considered to be a less harmful scam because you're just trying to steal money. You're not necessarily trying to steal credentials that would allow you to

advance into the business. But many things that begin with "I want to give you money" may end up with malware credentials, bank account information, many very valuable things being stolen. So this is a very important type of attack that we need to defend against.

00:59:49 However, we have seen, there was one case where we integrated with a church and this church received a lot of messages from people in other countries saying something along the lines of, "Hi, here's the donation for $10,000, for $5,000. I want to give this money to you." And these kinds of messages had many of the similar attributes to what you would see in the Nigerian prince scams. They were sent from previously unknown senders, from shady parts of the world, a shady infrastructure offering money to the recipient. And so this is a clear-cut case of false positives going crazy and being totally unmanageable from the perspective of this customer security team.

01:00:38 And so the strategy that we apply to this is relatively... We have a few different types of approaches, but the most scalable best approach is to start by trying to figure out what is it about these messages that makes our models flag it, extract that as a signal, build an aggregate for that signal keyed on the user or keyed on the recipient. So how frequently does this recipient receive messages that have this signal in it? And then retrain the models with that signal. And going through this process enables the models to stop flagging these kinds of messages because now you extracted away what is suspicious and taught the model that this type of suspiciousness is not something to block this message for for this user. And so in this case, it's this multistage process, extract signal, build aggregate, retrain model with aggregate. That's the recipe that we utilize when handling issues like this one.

Jon Krohn: 01:01:44 Sweet. Thank you for being able to get into that example. And maybe I'm just going off on a tangent here, but one of the things that I remember about the Nigerian prince scams is that those scams, which I don't see as much anymore, but-

Dan Shiebler: 01:01:59 Email security solutions have gotten much better at blocking them.

Jon Krohn: 01:02:03 But yeah, back when I did used to get them, my understanding is that they were... Because when I get them, there were lots of spelling mistakes. The grammar was poor. And so I was always like, man, these are so bad. How do these ever work? But then I later learned that them being bad is a feature, not a bug, because actually you're trying to find the most gullible people and the most gullible people will fall for an email that looks terrible.

Dan Shiebler: 01:02:34 There's a lot of different attacker philosophies on how to approach this, and certainly the scam emails that are structured in a way such that they filter out people who won't end up falling for it and therefore save the attacker time on the escalation stages. Because what will happen is you'll have to talk to the attacker more, they'll need to spend time and effort when everything kind of goes back... Most attacker behaviors can be explained by thinking about this from this simple cost benefit analysis from the attacker's perspective, they want to maximize the number of dollars that they get for every minute they spend operating something and time they need to spend on the phone with you is time they only want to spend if they think they have a decent chance of convincing you to give them your money.

Jon Krohn: 01:03:25 Cool that we can go into a specific example like that in a bit more detail. So one big thing that's changed for you since we did a full length episode is you are now Dr. Dan

Shiebler. So you finished your PhD work at Oxford University, and during that time you were looking into applications of category theory to machine learning. And you did define that for us back in episode 451 when we had that most recent full length episode several years ago now. My memory from then is that category theory, it had a lot of applications to clustering in particular.

01:04:05 And from everything that you've been saying so far, it seems to me like clustering could be something that's very useful for identifying cybersecurity risks because there's going to be particular kinds of features that like in the Nigerian prince scam where poor, poor grammar and spelling could be this feature that could help a clustering model identify Nigerian prince emails as opposed to emails that are not Nigerian prince emails. And maybe you could even be using clustering to identify new kinds of threats that you haven't before. Just like, oh, this is an interesting cluster over here. It seems to correlate with this kind of attack. I don't know. So where I'm getting at with my question is, first of all, congrats on the PhD and yeah, is there any way that category theory applies to the kind of work that you're doing now at Abnormal Security?

Dan Shiebler: 01:05:03 On a high level, yes, in that there's a great benefit towards being able to look at the kinds of problems that we face through a template of how they fit into general categories of problems and then identify what's worked for other types of problems that share these characteristics with cybersecurity. A lot of the challenge that we face, as you described, is in the realm of clustering. And one thing that I studied a great deal in my PhD was the relationship between clustering and manifold learning. So manifold learning is embeddings and vectors and vector databases and embeddings, and the types of ways that you can represent some kind of entity as a dense vector related to other entities, query for them and group them together and understand their

behavior and characteristics in this lower dimensional form are all general characteristics that apply to a number of different applications.

01:06:18 And the relationships between building a manifold, which you would project your entities basically means building embeddings for the data that you're working with, which in the case that we're operating with is things like employees, IPs, domains, links, attachments, devices, vendors, companies. These are the core nouns that we reason about, each of which are things where we derive embeddings and we group them together. When you identify a new domain, you want to understand what are the other domains that have similar characteristics to this one. So this is something that can go through the process of derive and embedding for this and then feed it into a model that knows how to process some embeddings of domains or identify how the structure of this domain enables us to cluster it in a group with other domains. And so the derivation of these kinds of strategies and how you would utilize this to build this kind of approach is something I would say that's benefited me a great deal as we set out our strategy.

Jon Krohn: 01:07:29 Very cool. It's nice that that academic stuff can actually be useful in practice and amazing to me that you did a PhD while working full-time in really challenging roles, first at Twitter and then at Abnormal Security. It's an amazing accomplishment. I really felt like I had my plate full doing my Oxford PhD all on its own. And to some extent, now having been in industry for over a decade post PhD, I'd love to be able to have the space to do a PhD full-time. I think I'd really relish that a lot more than I did when I was much younger, because you see all these real-world applications now, and there's so many questions that I have that I'd love to have an infinite amount of time to dig into.

01:08:26 So we dug up in our research on you that you do still manage to find some time for some other things. So, for example, from your About Me page, it looks like you have an interest in both math and history podcasts. So that's kind interesting, but specifically it kind of leads me to some more open-ended questions because also something that I know about you, Dan, and that you're really big into fitness, and even just before we started recording, so if you're watching the video version of this, there's almost no way to tell. The only thing that kind of gives it away is it seems like the camera's kind of on a shaky surface. And the reason why that is is because Dan is on a treadmill right now. So before we hit the record button, Dan was actually walking as he was talking to me, as we were catching up and kind of getting set up here.

01:09:25 So you walk five or six miles a day it sounds like during a full workday. That's a cool hack there. But I also know that you live in New York like I do. You like riding around on your bike. This is all related to the math and history podcast still. So that kind of reflection that you do, and thinking about time passing, particularly with the history podcasts, what do you think about how AI might impact things like urban planning or transportation, particularly maybe in the context of climate change? I'm curious if you have any interesting thoughts on how AI, machine learning might transform our urban world over the coming decades.

Dan Shiebler: 01:10:21 It's a good question, and to be honest, I'm not tremendously optimistic. I'm very optimistic about a lot of things, but our urban world, I think, is something that in many times the largest problems that we face are due to people problems rather than technical problems. I think that AI is a powerful tool for solving people problems, I'm sorry, for solving technical problems and a less powerful tool for solving people problems.

01:10:58 One example is I had a professor back when I was at Brown. He was an incredibly brilliant guy and one problem that I remember he was working on, his name was Philip Klein, a professor at Brown. One problem I remember he was working on was applying graph coloring algorithms to gerrymandering problems. So identifying how would you most equitably assign voting districts to a particular region based on where people live and populations. And I remember thinking, this will never happen or ever come into play because nobody has an incentive to make things the most equitable way. The incentives are to try to benefits whichever policy you're trying to get past or person you're trying to put into power. And those are always how the decisions will be made for these kinds of things. And perhaps that's a very cynical perspective on this particular area, but I think the human angle on things like city construction is perhaps too dominant for technical approaches to really have the same kind of transformative power that they'll have in other areas, at least in the immediate future.

Jon Krohn: 01:12:20 Okay. Yeah, that's a good answer. Maybe there's tangential ways. I guess things like to the extent that AI could be helpful in keeping the plasma contained in a nuclear fusion reaction, I guess we could have a lot more abundant energy, but in terms of actually urban planning-

Dan Shiebler: 01:12:46 There's tons of applications in energy. In even global warming, things like simulations are possible. The chemical development is something that's tremendously enabled by simulations as well as all sorts of different areas than engineering and manufacturing. There's many things in the world of atoms that advances in machine learning technology and AI technology have already shown tremendous advances on and will continue to do so, but there's always a tension between what is possible from what technology makes cheap and efficient and

effective, and what are the incentives and structures of our society that we need to operate within.

Jon Krohn: 01:13:31 Great answer. As we start to wrap up a little bit here, it's clear to me and probably to a lot of our listeners that you have a tremendous breadth of knowledge. Do you have any particular tips for us on how we can stay updated and ensure we're continuously learning, I guess both inside of our field in data science, in AI, but maybe outside of it as well?

Dan Shiebler: 01:13:57 I think trying things out and exploring new technology when it becomes available, just opening up some little projects and trying to challenge yourself to build something. There's really nothing that lets you learn about something better than trying to build. There's something about putting yourself in a situation where you need to demonstrate the knowledge that you've acquired that lets you understand it at a really deeper level. I like looking at various GitHub repos and cloning them and making small changes and building little toys for different applications, I want to explore new kind of technology, and I find that that's really the best way to challenge yourself and to grow in new areas technically.

Jon Krohn: 01:14:51 Yeah, great answer and definitely one that I agree with as well. I mean, that's always the thing is it's like, I don't know, just reading a book for me, especially a technical one. Reading for pleasure, okay, that's one thing, but when it's about learning some new machine learning approach, I definitely just prefer being like, okay, I want to learn this cool thing. What's something I can do with it? And it could be even as simple as finding someone else's Jupyter Notebook where I can just use that notebook and, like you're saying, make small changes, upload my own dataset or something and just see how things go. All right. Dan, you probably remember from your previous appearances on this show that before I let

guests go, we like to ask for a book recommendation. You got anything new for us?

Dan Shiebler: 01:15:53 Recently I've read a few history books called Barbarians, Marauders, and Infidels, I want to say is the name of the book. It's really, yeah, Barbarians, Marauders, and Infidels, incredible book. This book on medieval warfare, it just covers a number of different types of battles and locations of battles happening and covers the broad themes of what was the way that from the fall of the Roman Empire to the fall of Constantinople, that warfare changed, the introduction of the Magyars and the Vikings and the Arabs as three different groups that dramatically changed the landscape of the areas that they operated in, how the different weaponry, the rise of artillery, the rises and falls of different kinds of projectile weaponry, the different roles of the horse and the boats. Really just a fascinating survey of a really fascinating and complex time in history and what it sort of says about the people who lived then and how their lives are similar and different from people today.

Jon Krohn: 01:17:08 And it ties together a lot of your interest there. You got security, you got history. Nice. That sounds like a great recommendation and amazing that you can offer such a detailed account of what's covered in the book on a whim. Thank you very much for that suggestion, Dan, and thank you very much for a wonderful episode. Maybe we can check in again in a few years once more on how things are going with your very articulate way of speaking on such technical concepts. No doubt our audience will be craving that again.

Dan Shiebler: 01:17:43 Sounds good. Thanks, Jon. Really glad to be here today.

Jon Krohn: 01:17:47 Oh, and I also need to, before you leave, in the meantime, between now and that inevitable Super Data Science episode, maybe like Super Data Science 1000 or 900 and

something, before that episode, if people want to be following your thoughts, what's the best way for them to do that?

Dan Shiebler: 01:18:05 Probably my Twitter or my LinkedIn, I would say. So I'm dshieble on Twitter. It's just D and then my last name without an R. Eight character UX code.

Jon Krohn: 01:18:17 Nice. We'll be sure to include that in the show notes and yes, now I really will let you go. So thank you very much for being on the show and we'll catch up with you again soon.

Dan Shiebler: 01:18:27 Thanks, Jon.

Jon Krohn: 01:18:33 What an impressive, confident speaker. Always awesome to catch up with, Dan. I hope you enjoyed the conversation. In today's episode, Dan filled us in on the heuristic intermediate ML models as well as the large language models that they develop at Abnormal Security to identify cybersecurity risks in messages. He talked about how false negatives are individually the biggest classification error to avoid in cybersecurity, but false positives accumulate to create a dangerous boy who cried wolf situation as well. He talked about how Redis key-value stores and an auto-retraining framework allow for efficient on-the-fly model updates, how the clustering associated with category theory is useful in real-world applications, and how AI is great at solving tech problems, but not always human problems like those associated with urban planning and politics.

01:19:20 As always, you can get all the show notes including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Dan's social media profiles, as well as my own at superdatascience.com/717.

01:19:33 Beyond social media, another way we can interact is coming up on November 8th when I'll be hosting a virtual half-day conference on building commercially successful LLM applications. It'll be interactive, practical, and it'll feature some of the most influential people in the large natural language model spaces, speakers including some that have been on the show. It'll be live in the O'Reilly platform, which many employers and universities provide free access to. Otherwise, you can grab a free 30-day trial of O'Reilly using our special code SDSPOD23. We've got a link to that code ready for you in the show notes.

01:20:08 All right, thanks to my colleagues at Nebula for supporting me while I create content like this Super Data Science episode for you. And thanks of course to Ivana, Mario, Natalie, Serg, Sylvia, Zara, and Kirill on the Super Data Science team for producing another absorbing episode for us today. You can support this show by checking out our sponsor's links, by sharing, by reviewing, by subscribing, but most of all, just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until next time, my friend, keep on rocking it out there and I'm looking forward to enjoying another round of the Super Data Science podcast with you very soon.