EPISODE 786

[INTRODUCTION]

[00:00:00] JM: The demand for electricity is based on the consumption of the electrical grid at any given time. The supply of electricity is based on how much energy is being produced or stored on the grid at a given time. Because these sources of supply and demand fluctuate rapidly, but somewhat predictably, energy markets present profit opportunities for financial market traders.

Minh Dang and Corey Noone are engineers with Advanced Microgrid Solutions, a company that builds software to help traders captures better opportunities in the energy markets. Minh and Corey join the show to talk about how their company builds and deploys machine learning models for market prediction. We discussed data infrastructure, machine learning model deployments and the dynamics of the energy markets.

[SPONSOR MESSAGE]

[00:00:56] JM: DigitalOcean is a reliable, easy to use cloud provider. I've used DigitalOcean for years whenever I want to get an application off the ground quickly, and I've always loved the focus on user experience, the great documentation and the simple user interface. More and more people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A $15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU optimized droplets, perfect for highly active frontend servers or CICD workloads, and running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily, and as a bonus to our listeners, you will get $100 in credit to use over 60 days. That's a lot of money to experiment with. You can make a hundred dollars go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure, and that includes load balancers, object storage.

DigitalOcean Spaces is a great new product that provides object storage, of course, computation.

Get your free $100 credit at do.co/sedaily, and thanks to DigitalOcean for being a sponsor. The cofounder of DigitalOcean, Moisey Uretsky, was one of the first people I interviewed, and his interview was really inspirational for me. So I've always thought of DigitalOcean as a pretty inspirational company. So thank you, DigitalOcean.

[INTERVIEW]

[00:03:04] JM: Minh Dang and Corey Noone. Guys, welcome to Software Engineering Daily.

[00:03:06] CN: Thank you, Jeff.

[00:03:07] MD: Hey, Jeff.

[00:03:08] JM: So you both work in engineering at AMS, and AMS is a company that builds technologies around trading energy assets. So I'd like to start at just at a high-level explaining why there is a market for trading energy. Corey, can you explain why there is an existing energy market?

[00:03:27] CN: Absolutely. In California and I guess across the United States, energy is traded and served by an independent system operator or ISO, and their responsibility as an independent operator of the system is to forecast and schedule a generation. So in order to do that, they basically take in bids both from consumers and generators of electricity and optimize a dispatch schedule accordingly such that the grid is stable. What that essentially means is that supply always equals demand.

The way in which they do that is they collect bids that represent the willingness to sell energy into the market at various horizons. There's a day ahead energy market, there's a real-time energy market and a host of ancillary services that essentially keep the energy flowing at all times 24 hours a day.

[00:04:17] JM: Can we think of the energy market as something similar to the stock market? I think most of the listeners know that the stock market has lots of trades and exotic derivatives and all kinds of things like that. Is it that kind of pacing?

[00:04:31] CN: Oh, absolutely. The only difference is that the financial traders get to go home and sleep at night. Energy trades 24 hours a day.

[00:04:39] JM: Can anybody trade energy or do you need to be an accredited investor?

[00:04:42] JM: So anyone has an energy asset that is valued by the market. So whether you're a commercial or an industrial consumer of electricity and you want to participate in a demand response program that is a program run by a load serving energy that values your ability to reduce consumption over a period of time, or you're a large project capital financer and you're looking to get into renewables or storage at the grid level. That is all valued by the market.

[00:05:13] MD: Yes. So to give more context there, you can participate either on the generation or consumption side. So if you have an energy generating asset, you can participate on that side, or if you're an energy consumer, you can participate by lowering your energy needs during certain times.

[00:05:30] JM: Describe some of the tools that are used by people trading in the energy markets.

[00:05:35] CN: So forecasting is a common thing that occurs in both energy and financial markets, is the ability to quantify uncertainty around future events. So when we talk about trading, we typically talk about energy traded intervals. So it could be for everything down to a 5-minute trading interval up to potentially an hour, energy generators who are selling blocks on the market. So forecasting and being able to assess the volatility of the market is extremely important.

[00:06:06] JM: And what are you guys working on at AMS? What is the product and who is it for?

[00:06:10] CN: So we have a couple of different products I think ranging from the scale of which I was speaking initially, the commercial and industrial load scale. So everything greater than typical residential load. So a lot of your listeners might only have exposure in energy markets, like paying their electricity bill. Typically, those rates are based on dollars per kilowatt hour, so like a fixed energy charge.

A lot of commercial and industrial consumers of load pay not only based on energy charge, but based on a demand charge, and that demand charge is proportional to the maximum power that they use during their billing month. So we have products that manage that demand charge throughout their billing month as well as products that can trade directly at the grid scale into energy markets, so via the ISO.

[00:06:59] JM: Can you unpack that in a little more detail? Paint more of a picture for who is the end user of your technology, and I guess how they are using it.

[00:07:10] MD: So we have a few different products right now. Two of our main users are commercial and industrial companies, then also energy generators. On the commercial and industrial side, we do what Corey was saying, demand charge management for that customer. So AMS started off building huge storage systems, and with that we found out with the byproduct of actually needing to solve big storage systems for commercial and industrial customers such as a Keizer or something local, CalState schools, we need to build a huge software platform to manage the operations of that storage system too.

So what we would do is we gather a bunch of data from that specific site, run a bunch of forecast on how we expect a little profile to look for that building, pump that through a huge optimization and figure out what's the best way to use the battery to charge or discharge pretty much [inaudible 00:08:02].

Our other customers on a generation side, we provide a trading platform to help energy generators bid energy into the market. We do that again by data science [inaudible 00:08:16] forecasting market prices. Also, again, running that through a huge optimization and then making recommendations on what's the best way to bid energy into a market for the best financial outcome.

[00:08:27] JM: So these energy traders that you're catering to with one of your products, these are traders that work at energy generation companies or do they work at trading companies, like Goldman Sachs?

[00:08:41] MD: Energy companies, as right now a lot of them are renewable companies. They are also some utility companies of which we can't name at this time, but we also are working with some utilities trying to apply our technology to overall optimization and trading on their energy assets.

[00:08:58] JM: Okay, cool. So before we get into the technology, can you just take me inside the life of an energy trader? What do I want out of my technology stack? Am I spending my day looking at charts and planning how I want to bid on various types of electricity, or am I doing like fundamentals research? What's the mix between algorithmic trading and manual trading? Take me inside the life of an energy trader.

[00:09:28] CN: Yeah. So that's very much a function of the technology that they're trading, so the underlying generation source as well as the market that they're in. So if you're a traditional energy trader that is selling fossil fuel based electricity generation on the market, the algorithm by which you value your product is going to be with long ramp up times for your generation source.

You typically value it slightly higher than your marginal cost to produce that electricity, and then when the market clears above that level, you make money, and when the market clears below that level, you just don't generate electricity. It's a little bit more sophisticated when you're starting to talk about intermittent generation, such as wind and solar. Then it gets even more sophisticated when you're talking about battery storage, which is our expertise here at AMS. So batteries can represent both the generation source as well as the load and it can turn on and off very quickly.

[00:10:27] JM: We're introducing a lot of domain specific discussion here. So I would like to transmute what we're talking about into terms that just regular software engineers and data scientists can understand. Can we discuss a certain dataset that is flowing through this system

that you are building machine learning models on top of? Let's focus on a specific dataset. What would be a good exemplary dataset that we could talk about?

[00:10:54] CN: Yes. So when we typically train our machine learning models, we're feeding in historical electricity prices as well as the state of the grid at that time. So for example, we might want to know what the inner connector limits are. So how much energy can we flow across certain portions of the grid as well as what the flows were at that time. Maybe there are some other exogenous factors such as weather. Was it a predictably warm day? That could have results at or explain the price spike in electricity.

[00:11:25] JM: Where do those datasets came from?

[00:11:27] CN: So there are open data sources, but it also varies significantly based on the market that we're interested in. For example, in Australia, they make all of their market trading data public. So it's very easily accessible. Whereas for most markets in the United States, that data is proprietary.

[00:11:45] JM: So describe how these different datasets make their way into your infrastructure. How often are you pulling them – Whether it's a REST API or some antiquated FTP server that you're getting the energy data from. Give me an overview of the process of pulling in that data into your infrastructure.

[00:12:07] CN: Sure. We like using a data pipeline management tool called Airflow, and we have a direct database connection with a replica that sits in Australia that has all of the real-time market data provided by the market itself.

[00:12:24] JM: I see. So you're querying their database. Are we mostly focusing on the Australian energy market for the purposes of this conversation?

[00:12:32] CN: Sure, we can do that. I think the Australian market presents some unique challenges, one of which is that they clear on a five-minute basis, and that is the only time horizon in which they're interested in. So as I spoke to you earlier, in California, there are multiple different markets at different time horizons. So the ISO will schedule or dispatch energy

on a day ahead market based on what they anticipate load will be and then they fine tune adjustments real-time on an ancillary services market.

Whereas in Australia, every five minutes they're looking at state of the market and then procuring energy based on that. Yeah, I'm happy to talk about Australia. I think that presents a unique use case.

[00:13:14] JM: Yeah, sure, and let's continue to just talk a little bit about the top level just getting the data from that database connection. So are you just relying on using their database and just making queries to that database all the time or are you tailing the logs of that database and replicating it over to your data store?

[00:13:36] MD: So for Australia, if the data provided by the market operator, there is a replicated database that we currently pull data from and then replicate it into our system in AWS. We also gather some data specifically from the customer from the generator. What we do there is we'll put a gateway on their site that say the customer has a bunch of wind turbines. With that gateway on site, we'll gather telemetry data from that site as we do and then we'll push that to our system in AWS.

[00:14:08] CN: Then regarding the mechanics of pulling that data, like I said, we do enjoy using Airflow, but in terms of the data storage internally, we also use technologies like Cassandra and Scylla to manage our own data source.

[00:14:20] JM: Cool! So you use both Cassandra and Scylla.

[00:14:24] CN: Well, in production we use Scylla, but we do local development using Cassandra.

[00:14:28] JM: Got it. So if I understand, Scylla is like a faster version, it's a more performant version of Cassandra, and Cassandra is a masterless key value storage system. So why is Cassandra a good fit – Well, I guess reiterate what this use case for Cassandra and why Cassandra is a good fit.

[00:14:49] MD: We store all of our time series data in Cassandra or Scylla. Scylla is basically just an open source Cassandra-like implementation that's very performant compared to Cassandra. Since we store a lot of time series data, going with Scylla has increased our performance, and for our use case, since we're pulling that data for forecast fairly frequently, in our case, milliseconds count. We had to go with Scylla just to get the performance levels that we needed to be able to generate forecast quick enough to make sure that we can [inaudible 00:15:22] in time.

[SPONSOR MESSAGE]

[00:15:32] JM: This podcast is brought to you by wix.com. Build your website quickly with Wix. Wix code unites design features with advanced code capabilities, so you can build data-driven websites and professional web apps very quickly. You can store and manage unlimited data, you can create hundreds of dynamic pages, you can add repeating layouts, make custom forms, call external APIs and take full control of your sites functionality using Wix Code APIs and your own JavaScript. You don't need HTML or CSS.

With Wix codes, built-in database and IDE, you've got one click deployment that instantly updates all the content on your site and everything is SEO friendly. What about security and hosting and maintenance? Wix has you covered, so you can spend more time focusing on yourself and your clients.

If you're not a developer, it's not a problem. There's plenty that you can do without writing a lot of code, although of course if you are a developer, then you can do much more. You can explore all the resources on the Wix Code's site to learn more about web development wherever you are in your developer career. You can discover video tutorials, articles, code snippets, API references and a lively forum where you can get advanced tips from Wix Code experts.

Check it out for yourself at wicks.com/sed. That's wix.com/sed. You can get 10% off your premium plan while developing a website quickly for the web. To get that 10% off the premium plan and support Software Engineering Daily, go to wix.com/sed and see what you can do with Wix Code today.

[INTERVIEW CONTINUED]

[00:17:30] JM: Why is Scylla a good fit over – I mean, there are these custom time series databases. You have InfluxDB, you have TimescaleDB, etc., etc. Why is Cassandra or Scylla a good fit over one of these custom time series databases?

[00:17:46] CN: I'll have to admit I can't speak very well to all the options on the market, but what we have found internally is that our application is unique in the ability that we're able to segment a lot of our data services, which presents a nice opportunity to architect around that natural segmentation based on, say, assets or when the intervals were created.

We also generate a lot of data in batches. So we're able to – When we create a forecast for what we believe the future of the market looks like in terms of prices, we generate multiple percentile streams. So in every forecast update since we're talking about Australian market, that occurs every five minutes, we're generating a ton of data on a five-minute basis, but all of that we want to consider as a single batch process result and that fits very nicely in this key value store.

[00:18:38] JM: So the data from the Australian energy markets that you're getting from I guess some third-party data provider in Australia, you're storing that in the same database that you're also storing this data that's getting ingressed from wind turbines in Australia. Is that what you said, Minh?

[00:18:57] MD: Actually it's stored in – So we not only use Scylla. We also have PostgreS also. So depending on the type of data [inaudible 00:19:07] data, we'll put it in a standard SQL data store. For the time series data, we will put it in Scylla.

[00:19:12] JM: And just to get a little more color on that ingress process, you said you put some kind of gateway on the physical wind turbine system so that you can get the data from the wind turbines. What's the process of ingressing that data? Because I mean if you have these time series from the wind turbines that are getting generated where the wind turbines are, that data is getting fed into that gateway that you have on the premises of the wind turbines at some interval and then you have to pull that gateway at some interval. So I imagine there are these micro-

batches, or maybe you could stream the data or you could take really big batches. What's your process for ingressing that data?

[00:19:54] MD: So we put the gateway on site and usually if it's a wind farm, they would have some sort of other data aggregation device. So with that device, that would communicate down to the individual turbines or whatnot. There's usually integration between that data aggregation device and our gateway. Once the data gets to our gateway, we send it up to the cloud in micro-batches. We want the data fairly frequent. We send it about every five minutes or so.

[00:20:23] JM: Great. So we covered the ingress process a bit, and now we can imagine that on your AWS instances you've got – Or on your AWS database, managed database situation, you've got Scylla with some data in it. You've got some PostgreS with some data in it, and we can start to do data engineering on top of these things. Know that we want to build some machine learning models with these data sources. So what kinds of machine learning models do we want to build?

[00:20:55] MD: So as I said, the primary use case for machine learning here at AMS is in time series forecasting. So we're forecasting namely electricity prices, if we're talking about grid scale assets, but if we're also talking about demand management at a customer's facility, then we're also forecasting potentially not only their building load. So the electricity that we anticipate they'll be using for the upcoming, say, week. In some cases we also forecast onsite generation. So if there's a collocated solar farm, for example, we'll want to also forecast the output of that solar farm.

[00:21:33] JM: In this kind of application, it seems like time horizon is a really big deal, because I guess you want to make forecasts based off of some specific time horizon, like maybe you want to make a forecast for how things are going to change in the next hour, or some ceiling that something might hit in the next year, for example. So how does time horizon fit into the way that you build models?

[00:21:58] CN: Yeah, definitely. There are a couple of scales of interests here at AMS, because we're focused primarily on energy storage, battery systems. That's a finite capacity of resource. So we would need to schedule. We have to have an estimate for what we believe the market

will go for the upcoming, say, 24 hours and plan a charging and discharging schedule around those market prices.

But then for the purposes of bill savings for our host customers that like to have these batteries installed on their property, they're paying their electricity bill based on a monthly peak demand. So the time horizon of interest for us are typically in the 24 hours to one month range.

[00:22:41] JM: Take me through the process of training a machine learning model for predicting, for example, the price for – I don't know, a specific type of battery storage. What would be a good prototypical use case, a prototypical machine learning model that you built?

[00:22:59] CN: Sure. If I were to take battery storage an South Australia, batteries as they said can act as both a generator and a load for the market and it can participate in up to nine different tradable instruments, if we want to take the finance analogy, representing both energy as well as ancillary services that keep the grid resilient at all times. So typically, use case, what we'll want to do is forecast a distribution, and this is kind of a critical point, is that we're not just forecasting point estimates, but we want to have an understanding for what the volatility of the market looks like with upcoming, say, a day for each of the nine products. So that when we take those probabilistic forecast and we feed them into a stochastic optimization framework for determining when to charge and when to discharge that hedges against that volatility, we can do that and make the optimal decisions to which products to bid into and at what levels.

So a typical bid for an individual product, it represents 10 price quantity pairs in Australia. So you might say, "I'm willing to sell you 50 megawatts of power during this 5-minute interval and I'm willing to do it at $5." Then I may also say, "At $10, I'm willing to sell an additional 10 megawatts of power." So a bid file for that 5-minute interval represents 10 of those price quantity pairs.

We have an optimization engine that takes in what we believe the forecast for all the eligible products that asset can bid into and then determine how to allocate its resources accordingly for the upcoming 24 hours.

[00:24:42] JM: What tools are you using to train that model?

[00:24:45] CN: So we're currently using an open source library called Tensorflow.

[00:24:50] JM: And describe how data makes it from your [inaudible 00:24:54] DB instance and your PostgreS instance into Tensorflow and like how you're using that and are you training it offline or are you continuously training it and updating the model. Give me some overview for the construction and the ongoing updating of the model.

[00:25:11] CN: Good question. So we currently train the model offline. We do continuously retrain it roughly, say, anywhere from a weekly to monthly cadence. We extract the data from Scylla. We construct features that's from that raw data. We then normalize the data. Then that represents a single process, which we then persist to that data source so that we can always rerun a training model against that exact same data source.

We then leverage an AWS service called SageMaker for spinning up instances that are capable of training on a GPU and can spin up many parallel instances for the purposes of tuning these neural networks. Then we deploy the trained models to endpoints via the same services capabilities.

[00:26:04] JM: Is SageMaker – So I have heard of AWS Sagemaker. I know it is an AWS tool that helps you with machine learning models. What exactly is it doing? Does it help you with allocating resources to those machine learning models or what else does AWS SageMaker help you with?

[00:26:22] MD: It's primarily a flexible resource that allows us to spin up GPU backed EC2 instances on demand.

[00:26:28] JM: So it's like a scheduler tool is mostly what it's useful for.

[00:26:31] MD: Yes. There are some type of parameter tuning tools that are based into it that allow for very efficient search over a hyper-parameter set that allows you to refine your models. But for the most part, it allows us to circumvent having on-premise resources invested in capital-heavy GPU hardware and then just push that all to the cloud and do all of our training all day.

[00:26:55] JM: We've done some coverage of a tool called CubeFlow, which is a Kubernetes Tensorflow integration system that came out of Google. Have you looked at CubeFlow at all?

[00:27:06] MD: No, we haven't yet. We are interested in distributed training as well as those set of resource assignment tools, but we haven't looked into that yet. We haven't found it useful.

[00:27:18] JM: Right. So I guess the use case for SageMaker, like you want to do this Tensorflow training on a periodic basis. Maybe you want to do it once a month. Maybe you want to do it once a week, and whenever you do that, it requires spinning up a large number of GPU-backed EC2 servers. So you use SageMaker to be kind of your proxy into the scheduling resources side of things.

[00:27:46] CN: Yes. It just represents a very flexible resource. For example, if you have a training job that takes four hours and you deploy that training job to SageMaker, it will allocate that resource for you and then automatically tear it down after the training is complete. Now, if you want to run multiple jobs in parallel, if you were to invest in the hardware on-premise, you would have to buy multiple GPUs. Whereas with SageMaker, it will happily allocate more GPUs for your distributed job. So it's extracted a lot of the work of deep learning and a lot of us to focus on the model development.

[00:28:24] JM: What are the other AWS services that you use?

[00:28:27] MD: A little bit of everything. So on top of all these data forecasting and optimization, we build web apps on top of this to expose all these information through fancy GUIs to our customers. So everything from standard EC2 [inaudible 00:28:42], standard BBCs, all the networking – A little bit of everything actually.

[00:28:47] CN: Elastic Beanstalk, S3, you name it. We probably – We touched it.

[00:28:52] JM: I was watching your presentation that you gave at Reinvent. There were some services that I was not familiar with. There's something called AWS Chalice. What is AWS Chalice?

[00:29:02] CN: So Chalice essentially allows us to deploy a flask-like endpoint that was initially serving as a way in which we were able to preprocess samples for inference deployed neural network model. So what that means is that basically we could hit an endpoint and say, "Give me the latest forecast for a region in Australia," and it would go collect the necessary information to construct a sample. So what was their recent price activity? What is in their connector flow? Everything that we use as explanatory features in our forecasting model, construct that, do the normalization and then hit the endpoint that is assuming is you've already constructed that sample in the same format you provided the training data and then return the forecast –

[00:29:49] JM: Wow!

[00:29:51] CN: Exactly. It's pretty exciting.

[00:29:53] JM: So that's like a tool for grabbing a sample set of data points so that you can have an input into the model, because you can train this model, but in order to have it do something on a given day, you're going to need to dip your chalice into the lake of data and pour it into your model.

[00:30:16] MD: I would characterize that as a little more general purpose. It is comparable to a flask sort of endpoint, but AWS has made it easy to deploy that server and access it remotely on AWS.

[00:30:28] JM: I see. Can you explain it in a little more detail? So you deploy Chalice to where exactly and how do you – Just walk me through that use case in a little more detail.

[00:30:38] CN: Sure. What AWS Chalice, and actually we currently don't use this, but it's really good for prototyping purposes. Essentially define an API in a set of routes that you can then deploy in a single line, a single command line, and it will automatically deploy that endpoint to Amazon API gateway and AWS Lambda. So all of your requests are essentially now Lambda functions that you can access externally.

[00:31:06] MD: Yeah. So just a high-level overview, it's a quick and easy way to deploy an application that uses their API gateway and Lambda. So you can quickly bring up endpoints that need to pull data from a backend data source. Further quickly, they're internally and externally accessible.

[00:31:23] JM: And you mentioned Lambda. What are some ways that you're using AWS Lambda?

[00:31:28] CN: So unfortunately we currently aren't using a whole lot of Lambda at this time.

[00:31:31] MD: Yeah. We would use it a lot more for prototyping, not much in production.

[00:31:36] JM: Why is that?

[00:31:37] CN: There are some resource limitations around both the Chalice and Lambda. So your deployment package on Chalices is limited to something like 50 megabytes. So when you start to wrap up all of your Python dependencies, it's very easy. You pull I a SciPy and a NumPy, you're at your 50 megabytes already.

But then there are also resource limitations on Lambda. So I know they've recently increased request time outs. I believe it was at the time I was looking at it 5 minutes. So for tasks, that just does not fit within our use case.

[00:32:10] MD: Yeah. I think Lambda is great for some use cases, but for this specific one that we need, we use it maybe for prototyping and whatnot, but in production, those APIs that we build, they're a little bit bigger, a little bit heavier. Also, they're frequently used too. So we found that if we just build the APIs and deploy them to container on Kubernetes, it works just as well.

[SPONSOR MESSAGE]

[00:32:37] JM: The blockchain is a new computer science primitive. It allows us to build applications that we could not have built before, and we're in the early days of blockchain applications. It's a great time to get started.

Blockstack is an open computing protocol for building applications where users truly own their data. They own their identity and even their content and connections. With a Blockstack ID, users can have a more transparent identity system rather than the modern internet identity systems that are closely tied to advertising. At blockstack.org/sedaily, you can learn about how to build decentralized applications easily. Blockstack is open source, it's free and it's an application stack that won't serve you ads or demonetize online media personalities or be subject to the whims of an individual CEO.

Developers who build on Blockstack can even get paid to build better applications using Blockstack via the app mining program. To find out about Blockstack including these programs, you can go to blockstack.org/sedaily. You can learn how to build decentralized applications that are private and secure and easy to build, thanks to Blockstack.

Cryptocurrencies are a huge unexplored space. If you're a developer, there's no better time to get started. Just so you know, it's not easy to build decentralized applications today, much like it was not easy to build internet applications in 1994, but we know that things get easier overtime, and Blockstack is one of the easier ways to develop on the decentralized internet today.

So if you're getting started, it's a great place to go. Go to blockstack.org/sedaily and learn more about how to build decentralized applications.

[INTERVIEW CONTINUED]

[00:34:45] JM: Just to return to an example that grounds what we're talking about here, you might be trying to build an model that predicts price of a certain type of energy, let's say a kind of battery storage, like I need. Some battery storage to story energy or to get energy from the battery and you're trying to predict price on that so that people can trade. They can buy and sell energy assets based off of those predictions of price, because if the price is going to go up, maybe I want to buy some energy in advance of that price increase, and if the price is going to go down, maybe I want to sell. Do I have the model and the use case right?

[00:35:26] CN: Yes. Like I said, it's not only the point forecast that's of interest. There are some unique challenges and energy markets, namely the energy resources are becoming increasingly flexible and their shorter trading intervals, there's more intermittent resources on the grid, such as solar and wind that necessitate not just that point forecast, but a good understanding of where the distribution is. So we can develop strategies that hedge on potential price spikes or dips.

So the output of a model just to ground this would be a multi period ahead forecast. So if we're talking about a 5-minute interval, maybe you want to forecast 24 hours into the future, we're generating 288 points. Then not only are generating 288 points for a single point forecast, but we're also generating a forecast for each quantile. So what that represents is we have 280 lines that represent what's the most likely outcome that's kind of 50th percentile.

We also generate, say, 60% and 70%, all up to 99%. So we've got this distribution captured by all of these quantiles ranging from 1% to 99% and then we can develop change strategies that hedge against that whole range of scenarios.

[00:36:44] JM: Was it tricky to get these models trained in a way that avoided over-fitting, maybe over- fitting to certain section of time that you were sampling from? What was the process of training that first model and getting a sense of confidence that the model was not over-fitting or under-fitting?

[00:37:02] CN: So in time series forecast, it's very common to split your historical data into three datasets, a training dataset and then what chronologically follows would be a validation dataset, and then after that, a test dataset. So the training dataset represent the largest amount of samples that you're exposing to this [inaudible 00:37:22] training process. Then when you want to say optimize the design of your neural network, you may iterate over a few different architectures and then evaluate the accuracy of each architecture on this withheld validation dataset. Once you've arrived on what you believe to be the best model architecture, then you can finally test it on the third dataset.

So you're withholding data at each one of these stages. So you're fairly confident that by the time you've tested on your most recent test dataset, that should be the accuracy that you should expect moving forward once you deploy this in operations.

[00:38:03] JM: This is an application of deep learning, and I'd like to disambiguate what makes this use case deep. Can you describe the model in a little more detail that will illustrate why this falls under the rubric of deep learning?

[00:38:19] CN: Sure. So we have a number of different neural network architectures, but in each we've highly parameterized how the architecture is set up. So part of the hyper parameter tuning process is determining, for example, how many layers, how many stacks do you want to include from input to output. So one of our best performing networks can be characterized as the temporal convolutional neural network, and in a convolutional neural network, you tend to stack convolution and activation layers. So what we do is when we do the hyper parameter tuning of this neural network, we're finding the optimal number of layers such that we're neither under-fitting, nor over-fitting the training data.

[00:39:02] JM: What's the process of training – What was it called? Temporal –

[00:39:07] JM: That's right.

[00:39:09] JM: Temporal convolutional neural network. So we've done some shows about like the process of training image recognition neural nets, and there's the convolutional step where you're creating these different rotations over an image to understand the image kind of different angles, I guess, is a way of looking at it. You're rotating this metrics. So when you're doing a temporal convolutional neural network, is that like looking at different time windows?

[00:39:41] CN: Yeah. Sorry for the jargon. All that really means is – The only difference between image recognition and what we use it for in time series forecasting is that an image recognition, you're defining a two-dimensional kernel that you're sliding over the pixels of your image. You may define that kernel to be maybe 5 pixels by 5 pixels. Then the way corresponding to each one of those pixel values is consistent as you slid it across the image.

In time series forecasting, the only difference is rather than being a two-dimensional kernel, you're talking about a one-dimensional kernel. What we're saying is that this is one of the mechanisms that prevents over-fitting to the training data is that because the weights have to be consistent within a layer, within a filter, that when you find the optimal ways, let's say, for example interconnector flow is an important feature at a given time, you should anticipate that same explanatory feature will be valuable the next interval. Does that make sense?

[00:40:41] JM: Okay. To a very limited degree for me, I mean, I've had a lot of trouble covering deep learning algorithms on this show. I think there are other podcasts that are actually much better at this than I am. I've just found that it's not a domain that I can explain well over a podcast, or maybe it's just because I don't understand it to be honest. But I do understand data infrastructure to some degree.

Are you using any streaming systems? Are you using Apache Spark or Flink? Are you using streaming systems for anything?

[00:41:11] MD: No. Not at this time.

[00:41:12] JM: So basically the UI, does it read directly from a Tensorflow model?

[00:41:18] CN: So we do have ETL processes that kickoff as new market data becomes available. Then our forecast are persisted in our Excel database. That data is then easily served by our API, which serves many of our applications internally.

[00:41:33] JM: Okay. So what's the interface like for a trader that is working with these model?

[00:41:39] CN: So I think the most important view for a trader is the resulting bid. So that is how much energy are we allocating at each price bend for the upcoming day. Typically looking at what you might imagine to be like an Excel table of prices and quantities, but they're also looking at forecast view. What is the probability of prices spiking above a particular threshold, as well as insuring that the price forecast are greater than their marginal cost to produce electricity.

[00:42:12] JM: Now what if you had a new dataset that you got your hands on and you wanted to integrate the new dataset into your models? How would you integrate a new dataset into an existing model?

[00:42:24] CN: So we typically wouldn't integrate it into an existing model, but we might kick off a new training job that – We do have a fairly accessible framework for defining named feature sets that the model can then access. So it would essentially building a new data stream, incorporating it in the data preprocessing step and then referencing it in the neural network construction.

[00:42:48] JM: And that's your process for testing out new datasets? I mean, there are so many datasets you could have available to you, and some of them are just going to add a necessary noise. They aren't going to really get you any mileage, but other things probably would be extremely valuable. What's your system for testing new datasets?

[00:43:04] CN: Yeah. I guess an additional point is that for each market that we operate in, the nature of the data that is available to us also varies. So we have a very rich dataset in Australia, because they make available most of their market data, including their own forecast for electricity prices. That's fairly uncommon in the United States. So it is important that we have robust test framework that allows data scientists internally to leverage existing architectures and deployment processes and quickly experiment on a new feature.

So what that might look like is a data scientist identifies feature in our feature store and then can quickly point to that feature as part of the data preprocessing step to include it in this aggregated dataset that we then feed to the model.

[00:43:54] JM: So are you saying there really is not that much data out there. There's not very many data sources that are at least publicly available that would be potentially useful for you?

[00:44:03] CN: It depends on the market, but we do have an experimentation framework that allows fairly rapid iterations. So the majority of the time is going to be in training the neural network and not so much the preprocessing of a new data stream. It's then fairly easy to say,

“Okay. Based on the validation area that we calculate, did we see an improvement in accuracy? If not, we’ll disregard that feature in future training.”

[00:44:28] JM: Let’s take a bit of a step back here and think about the broader energy market. I think you’ve alluded to the fact that the market for renewables has changed the world of energy trading. Can you just give a little bit more context for what’s going on in the broader market and how that’s impacting what you’re building at AMS?

[00:44:51] CN: Absolutely. So in California particularly, and this is true in other parts of the markets in the world as well, the amount of intermittent penetration has dramatically changed the supply stack, and this has shifted where energy and when energy is needed. So we commonly refer to the duck curve in California, the resulting curve. You can imagine peak consumption prior to solar energies, solar generations penetration as having – A peak occurring in the afternoon. Not that there’s so much solar on the grid. Actual peak consumption tends to occur at 2 points when the sun is first coming up and when it’s coming down.

So in the middle of the afternoon when solar generation is it at its highest, the grid actually has surplus energy sometimes. So what we’re finding is that there’s an increased need for energy storage to balance out both over-generation and under-generation to get a more consistent load throughout the day.

[00:45:50] JM: Does this have any meaningful impact on climate change, like reducing energy usage such that climate change will be improved?

[00:45:59] MD: We sure hope so. I mean, I guess I think that’s what every environmentalist is hoping that renewable energy sources will help that. Only time will tell though.

[00:46:09] CN: Energy storage is synergistic with green energy in generation as well. So it makes the generation storage projects more economical when you have the ability to store, say, peak spikes in wind generation as well as over generation by solar.

[00:46:27] JM: Tell me a little bit about the business. When you’re trying to find a new customer for AMS or convince a new customer, describe your go-to-market strategy.

[00:46:37] MD: On the energy trading side, we look for customers with various energy assets that can pretty much benefit from better price [inaudible 00:46:47], and that's pretty much everyone. With that, we offer – It's pretty simple pricing model where there is a standard licensing fee and there's also performance incentive for us if we perform above a certain level.

So it's almost a win-win for the customer where you have this asset, I'm using some more archaic ways of predicting prices and spreadsheets and doing some complex populations that I have to do manually. If I can get a software application to do that for me and get better bids on, make more money, why wouldn't I do so?

The recurring fee is fairly low. We actually make a lot of money off of performance incentives above [inaudible 00:47:28]. So it just shows that we're confident that our forecasting and optimization are pretty good.

[00:47:34] JM: Can you use the platform to make your own trades?

[00:47:38] CN: No, we can't and we don't. We actually don't want to. In some cases, we think if we do, then it becomes a conflict of interest of what we're providing to our customers.

[00:47:48] JM: Yeah. Is that consistent across the industry? Like companies that make software for traders. They can't also use it to trade themselves?

[00:47:58] MD: That's a good question. Do you know, Corey?

[00:47:59] CN: I believe that's a least valid in the financial industry.

[00:48:03] JM: Well, let's talk a little bit about the future. What are you working on at AMS right now? What are the projections for how the software infrastructure is going to change in the near future?

[00:48:11] CN: We're really just focused on the resiliency of our platform. These are markets that operate 24 hours a day. So ensuring uptime for our customers is critical. In the algorithmic

front, we're starting to look into new techniques for agent-based learning. So a typical workflow might include or a process flow might include forecasting and then an optimization, a stochastic optimization that takes into account market volatility. I think what's really interesting in an avenue that we're currently pursuing is in reinforcement learning, so teaching an agent to respond to a stochastic environment and choose a set of actions that maximize expected revenue for that asset owner.

[00:48:51] JM: Very cool. Minh, anything you want to add?

[00:48:52] MD: In addition to that, we're also expanding our trading platform to additional markets. We're working on Australia and California right now, and then we have U.K. roadmap for later this year. We're also heavily looking into which other markets should we get into next.

[00:49:09] JM: Okay, guys. Well, thank you for coming on Software Engineering Daily. It's been great talking to you.

[00:49:12] MD: Thank you, Jeff.

[00:49:13] CN: Thanks, Jeff.

[END OF INTERVIEW]

[00:49:17] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on-the-fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on

GoCD with the new Kubernetes integrations. You can check it out for yourself at gocd.org/sedaily.

Thank you so much to ThoughtWorks for being a longtime sponsor of Software Engineering Daily. We are proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]