SEI Cyber Talk (Episode 9)

Efficient NetFlow Partitioning via Minimum Cuts
by Ritwik Gupta and Anusha Sinha

Ritwik Gupta: Welcome to another series of the SEI Cyber Talk. I'm your host Ritwik Gupta. I'm a machine learning researcher at the Emerging Technologies Center. And today with me I have Anusha Sinha. Anusha.

Anusha Sinha: Nice to meet everyone. My name is Anusha. I work at CERT in the Security Automation Directorate, and I'm really excited to be here today.

Ritwik Gupta: We're excited to have you. So, I heard that, from a lot of people, and that's why I have you on the show today, is that you're doing some really cool stuff with NetFlow clustering of flow traffic out whether it came from a human or an autonomous agent. And I know absolutely nothing about NetFlow, and I'm sure a lot of people watching know nothing about NetFlow. Why don't you tell us a little bit about it?

Anusha Sinha: Sure, so the idea behind this project was just to be able to attach a label to flows and basically just say whether they were generated by a human or a machine. And one of the primary reasons I started thinking about this is because I went to FlowCon back in January, and I heard a presentation by Jeff Dean where he talked about how human generated flows typically tended to be like Gaussian, bursty, in nature. And so, for his project, for his PhD actually, he'd gone through and used traffic on a network where there just weren't very many humans active at the same time. And so, he was able to go in and find these Gaussian bursts quite easily because there was no overlapping activity. But I wanted to see if it would be possible to do this in more of a real-world setting where there is many humans that are active on the network at the same time.

Ritwik Gupta: So, why is this such a big problem? Are machines that good at generating realistic NetFlow?

Anusha Sinha: So, it's not about necessarily the machines generating realistic NetFlow. It's just that there are a lot of automated processes that are going on on your computer at any given time, like there's all sorts of updates. If you go to cnn.com, it will automatically reload a page every thirty minutes. So, there's a lot of stuff going on behind the scenes. And I guess one of the motivations for having this labeling system would be that it would be a little bit easier for people to go in and identify potential threats to the network because human users tend to have internal access to the network. And it would be good for analysts to be able to look at a set of flows and decide that they're human or auto (inaudible) generated.

Ritwik Gupta: Got you, so Anusha, that sounds really hard because, as far as I know, NetFlow is just a whole bunch of basically network capture, packet capture data or whatever.

Anusha Sinha: Yeah.

Ritwik Gupta: Or maybe it's different.

Anusha Sinha: Yes, so it is packet capture data. And one of the big issues that we've been having is that there is just so much of it. We get millions and millions of flows a day.

Ritwik Gupta: Okay.

Anusha Sinha: And when you want to cluster millions of flows a day, you need to find a way to do it that's pretty efficient so that ideally, at the end of the day, we would want to be able to add these labels sort of in real-time, not theoretically real-time.

Ritwik Gupta: Right.

Anusha Sinha: But as close to real-time as possible just so that we could kind of get the information out to analysts as quickly as possible, and they're not waiting for days or weeks or months to find out whether some set of flows is produced by humans or auto (inaudible).

Ritwik Gupta: That makes sense. And so, a lot of cluster methods that I'm familiar with that are like K-means or some sort of hierarchal clustering methods. What are you proposing that's more efficient or faster or better for this type of data?

Anusha Sinha: Okay so, I experimented a little bit with K-means and DBSCAN clustering. And I found that they don't tend to give us great results. It's hard to interpret what the clusters are because the algorithms will go in and tell you that you have clusters in all these areas of your data, but it's hard to actually know what any of that stands for.

Ritwik Gupta: Sure.

Anusha Sinha: So, in order to get clusters that kind of correspond to properties that we're interested in, I thought it would be interesting to formulate this as a max-flow problem.

Ritwik Gupta: Okay.

Anusha Sinha: Because so, the max-flow problem is used kind of along with its duo which is the min-cut problem in a lot of partitioning problems. And so, the idea was basically we create a virtual flow network which is different from NetFlow.

Ritwik Gupta: Okay.

Anusha Sinha: And then we use that to basically partition the data as being either human-generated or machine-generated.

Ritwik Gupta: So, how does that work? So, going back to my algorithms class back in undergrad, shout out Dr. Vernon, max-flow was basically if I have a source and sink, and I basically need to figure out how to maximize the flow in any network, right? I want to make sure that all edges are being used to their max potential. So, how can I use that to basically cluster a graph that is composed of human and machine traffic?

Anusha Sinha: Okay so, just to give you-- to start constructing this graph, the first thing that we want to do based on prior research by Jeff Dean, the guy who gave the Flow Con talk, he said we should look at server IPs because the activity over server IPs tends to be-- it's like Gaussian kind of shaped when it's produced by humans or human-originated. So, we basically started by making a bunch of nodes in a graph where each node corresponds to a server IP. And then we make two super nodes. So, we have a human super node and an autonomous super node. And they're called super nodes because we have an edge that goes from the human node to every single other server IP node and the same thing with the autonomous node. And then the idea is that we just want to find ways to attach weights these edges in a way that makes sense such that we can find a minimum cut or a partitioning through the network that results in a split that we're interested in, that would kind of correspond to human and autonomous activity.

Ritwik Gupta: Got you. So, basically because of these super nodes, I can get from any point in the network to any other point via the super node, right?

Anusha Sinha: So, actually, I guess I should clarify that the super nodes are kind of the source in the sink.

Ritwik Gupta: Okay.

Anusha Sinha: Yeah.

Ritwik Gupta: I see, okay.

Anusha Sinha: Yeah, and then, so once we have these super nodes set up, we also set up an edge between every pair of internal nodes, like all those server IP nodes. So, now we have these two sets of super nodes with edges going to them on either side of the network, so there's a source and a sink, and then we have all the internal edges within the network as well. So, then-- yeah, so then now the problem is just how do we assign weights to these edges in a way that's meaningful, right?

Ritwik Gupta: Right, yeah.

Anusha Sinha: And so, to start doing that, we want to take Jeff Dean's idea that these bursts of human activity are Gaussian in nature. So, I started by building a Gaussian mixture model for each server IP to basically assign a probability of whether it belongs to some combination of human-generated bursts, or if it is very unlikely to belong to any of those bursts.

Ritwik Gupta: Sure.

Anusha Sinha: So, then for each server IP, you know it has one edge that goes to the human super node and one edge that goes to the autonomous super node. So, we assign the edge that's going to the autonomous super node with the probability that it's going to be autonomous, based on the GMM, and the same thing for the edge that's going to the human super node. So, we basically do that for all of the server IP nodes.

Ritwik Gupta: Got you. And so-- I mean, so this sounds to me like it's probably a more efficient method than K-means for large graphs because K-means is probably terrible the larger and larger and larger my input size gets, right? And but we have-- I assume-- again, I don't know the research in this role. I assume that our max-flow methods or graph methods are a lot more well developed for large graphs than K-means is?

Anusha Sinha: So, it's actually not, unfortunately. The max-flow algorithms tend to be polynomial in the number of edges. And so, as I mentioned before, we have all these internal edges between the server IP nodes. And so, those edges-- also, just to go back real quick, we assign weights to those edges based on the similarity of two server IPs. So, you could use all sorts of metrics. You could use whether they come from the same subnet or whether they're active around the same times to basically assign those weights. But then now we have this problem where if we have thousands or millions of IPs that we're interested in, now we have that number squared internal edges. And then the max-flow algorithm is going to take some multiple of that number of edges in order to run. I think it's like the number of edges squared typically is the time you can expect.

Ritwik Gupta: Got you. So, what can you do when you have huge networks like this?

Anusha Sinha: So, one of the ideas that I'm really interested in pursuing going forward is looking into spectral sparsification of the graph. So, the idea is just that we have a lot more edges than we probably actually need in our network. And so, it would be nice if we could potentially combine some of the edges by basically- if we have three edges between a set of nodes, we could kind of maybe combine them in some logical way along with their weights to create a single edge between two, I could say, clusters of nodes.

And so, one of the interesting parts of doing this is that it vastly reduces the number of edges in our graph. And because all of the max-flow and min-cut algorithms tend to be polynomial in the number of edges, this would be like a huge speed up for us and allow us to actually tackle the challenge of millions of flows a day. And then so, when we're going about actually doing the sparsification, one of the interesting abstractions that I've found is that you can kind of consider the graphing the resistive model of the graph.

Ritwik Gupta: Okay.

Anusha Sinha: So, basically all that means is that-- so, each edge has a weight, and so all we want to do is say that the weight corresponds to the conductance of the edge. So, the higher the weight, the more conductive it is. And then there's a paper by Spielman and Teng, which we will link below. And they basically go through and say that if you sample the edges in the original graph with the probability proportional to the effective resistance of that edge in the resistive model of the graph you get a special sparsifier. And so, basically, I just want to go through and actually implement that probably using the Johnson-Lindenstrauss embedding to actually calculate the effective resistance for each edge.

Ritwik Gupta: I wish I knew as much about graphs as you do. What I do know about it is machine learning. So, let me hop back a little bit.

Anusha Sinha: Okay.

Ritwik Gupta: And I apologize to the audience. You're probably like trying to build a graph of where we jump back and forth in the conversation. So, what did you expect coming to a graph talk? So, you're trying to fit this Gaussian mixture model onto this data to try and figure out which cluster it belongs to. Could you do something more interesting like build a generative model instead or maybe make the GMM in a generative fashion such that we can actually generate flow from one class or another?

Anusha Sinha: Yeah, I would actually be really interested in doing that because right now, in terms of kind of validating my results, I'm mostly just going by well do we see spikes of human activity during the day when we would expect to and do we also see spikes of-- really regular spikes of automated activity overnight when there shouldn't be any humans active on the network. But it's not really a great metric to use because it's very hand wavy and wishy-washy. So, it would be nice to have some sort of data that we could generate in a specific class to kind of help validate our results.

Ritwik Gupta: Well, that's honestly fascinating. I don't think I know as much about graphs as I thought I did before. I feel like I left a bit dumber but probably for the best reasons.

Anusha Sinha: Hopefully, you guys don't feel the same way.

Ritwik Gupta: I'll definitely have to go up and look at some of these research that you linked to. If there's probably one or two things you would recommend the audience to read, what would they be?

Anusha Sinha: So, it would be the dissertation by Jeff Dean. It's called "A Systematic Assessment of Network Flows," I believe. We'll link it below. And he just does a great job of going into the background of specifically why we're interested in detecting these human users on the network, the work that he's done so far, why he decided that human activity is kind of Gaussian in nature. And he also gave a really good talk at FloCon 2019. So, if you prefer to see it in kind of a visual presentation, you can go ahead and look that up as well. And then there's two papers, one by Spielman and Teng, and one by Spielman and Srivastava, which talk about spectral sparsification using effective resistance. So, I'd also check those out if you want a better idea of how that would work.

Ritwik Gupta: Awesome. Well, thank you so much. That is a lot for me right now. I'm sure it's a lot for the audience. But as you guys know already, and if you don't, if you guys want more information, feel free to shoot me or Anusha an email, or better yet just email info@sei.cmu.edu. And we hope to see you guys next time.

Related Resources

Jeff Dean's (FloCon 2019 Speaker) dissertation
Spielman & Teng
Spielman & Srivastava
FloCon 2020