Fall Alumni Networking Lecture & Reception 2019: Professor Nicholas Papernot

Deepa Kundur: I wanted to take this opportunity to welcome you to the Fall Alumni Networking Reception & Lecture. And also welcome you to U of T Engineering's newest building, the Myhal Centre.

So I may be new to my role as Chair, but I'm definitely not new to ECE. As many of you are aware, I'm also an alumna of the program, in fact I'm a triple alumna! I did my Bachelors Degree in 1993 and my Masters and PhD in 1995 and 1999. And what's been wonderful is that I've had the opportunity to see the strength of this department from many different perspectives over the last thirty years. From a fresh undergraduate student all the way to finally becoming Chair. And even for a ten year period just being an alumnus, not working within the department.

So it's been wonderful to see the excellence of this department from many different perspectives. As many of you know, this is the top ranked ECE Department in Canada, and one of the best in the world. And I would say the reason is because of the strength we gain from our people. We have the very best professors, who do cutting-edge research in areas such as 5G, cybersecurity, healthcare, quantum technologies. We have some of the brightest and best undergraduate and graduate students. The demand for our program is exceptionally high and as you can also attest to, our students are not just technically strong but they're also well-rounded. And our strength critically comes from our alumni. Now there's a beautiful quote I wanted to read you that relates to this. It's by John F. Kennedy and he says "I think the success of any school can be measured by the contribution the alumni make to our national life." Now I would go one step further and I would say the success of our Department can be measured by the contributions our alumni make to the world. So I want to take this opportunity to thank you for your contributions to the excellence of the ECE community. You are leaders in your fields. You hire and mentor our students. You form critical partnerships--industrial partnerships-- with our professors. You are dedicated volunteers and donors to the Department, the University and the community at large. And so thank you for what you bring to the ECE community and I look forward, through my new role, to meet each and every one of you.

So now I'm happy to start the fun part of the event and the real reason you're here, our talk. So as the new chair, standing in the new building, it's my great pleasure to introduce our newest professor in ECE, Nicholas Papernot. Nicolas is an Assistant Professor in ECE and a Faculty Member at the Vector Institute where he holds a Canada CFAR AI Chair. His research interests are at the intersection of computer security, privacy and machine learning. He earned his PhD in computer science and engineering at Penn State University, supported by a Google PhD Fellow in security and privacy and his PhD research focused on characterizing the attack surface of machine learning systems and inventing defense mechanisms to improve their security and privacy. Prior to joining U of T he was a Fellow at Google Brain and his work has been applied to industry and academia to evaluate and improve the robustness of machine learning models, to input perturbations known as adversarial examples, as well as to deploy machine learning with privacy guarantees for training data at industry scale. Not only does he do all of these, I can attest to the fact that he's a very nice guy. So please join me in welcoming Nicholas.

[Applause]

Nicholas Papernot: Thank you Deepa. So tonight I'd like to tell you a little bit about some of the work we've been doing at the intersection of security, privacy and machine learning.

A new computing paradigm brings social disruption at scale slide

So to start I think it's nice to take a step back and look at what machine learning is doing. So as you know, machine learning is bringing this new paradigm to create software where instead of writing programs and giving the computer instructions on how to solve the problem we just give it inputs and outputs and let it figure out what the program is on its own. So this has allowed us to address problems like healthcare, energy, transportation or even education in ways that we have not been able to with traditional software.

Machine learning is not magic (training time) slide

But machine learning is obviously not magic, right? So to take a step back I like to give this very simple example of classifying cats versus dogs to understand a little bit where the vulnerabilities in machine learning may come from. So obviously this is why we're doing research in machine learning, to classify cats vs dogs. This is an example of a training set that you would use where you would give the algorithm each of these images along with the expected output of what you would expect the model to produce. And so the algorithm will analyze all these images and extract features from the data. So this may be the shape of the ears or the length of the tail for instance. And so this is a very simple picture that illustrates what a model might look like and you can see that now I can project all of my training points based on these two features that I've extracted.

Machine learning is not magic (inference time) slide

So now at test time with this model I can present it with images like this one and it's going to be able to correctly tell me that this is a cat. However if I present images that are completely out of distribution like this octopus, because the model has not learned any features that are relevant to classifying this class it's not going to be able to predict correctly, obviously. But you can see that there's this very large space in the input domain where we really don't know how the model should respond. So this exposes the model to attacks if the adversaries are able to systematically find these types of inputs.

Machine learning is deployed in adversarial settings slide

And unfortunately the answer is that they can find these inputs systematically. So we've already seen attacks against machine learning in the wild. So if you look at this example here what happened is that a couple of years ago Microsoft released a chatbot that interacted with users on Twitter. So the users could submit some text and the chatbot was basically a language model. It was learning from these interactions basically how to generate human language. And very quickly, I think within the first 24 hours that the bot was online, people realized that they could teach it how to be racist. This has implications that are funny in this case but if you take this out of context this can have very dramatic implications but for instance this is an example of a tweet that the bot produced and trust me this was

one of the only that I was comfortable putting on the slides. [Laughter] There are much worse ones. And this shows you what happens when basically the adversaries are able to manipulate the training data that you're learning from. Once you've trained and deployed the model there's another type of attack where the adversary may find inputs that will be directly misclassified by your model. So for instance Google uses machine learning to filter out videos that are not appropriate for children and remove those videos from the YouTube Kids app. And a few years ago some parents noticed a few videos like this one which is definitely not appropriate for children, but that made it through the filter. And this shows you that once you are trying to do machine learning on very complex things like the space of videos, where you have very high dimensionality, then it's hard to learn a correct model from a small number of training points.

Machine learning does not always generalize well slide

And so the question that we asked ourselves is 'why this happens'? And so if you look at the example that I give here with my very simple training set what happens is that there are some biases that we had not expected in the data. If you look at all the images of the dogs, they're all black dogs, and all of the cats are white. So what this means is that if you have a very large model that has a lot of capacity it's going to memorize the fact that animals that are white are cats and animals that are black are dogs even though you would not use this as a heuristic as a human to classify cats versus dogs. And so this means that the machine learning has learned logic that is different. So if you show it pictures like this one here, it's going to predict that this is a cat because the animal is white. Or here because it has grass in the back, the only image was the one of a dog, so it's going to misclassify that.

What if the adversary systematically found these inputs slide

And the unfortunate news is that the adversaries can find these inputs systematically. It's not just finding those images by chance. And so this is an example that some of my collaborators put out and we've generated a lot of these images that show how you can manipulate machine learning models. So this here on the left is an image of a panda. If you show it to a machine learning model it's going to classify it with 60% confidence as the image of a panda. What our algorithm computes is this perturbation here. So here I've magnified the perturbation so that everyone in the room can see it but in practice we take this matrix and we multiply it by this very small number and we obtain this image here on the right. So this image here is the sum of this multiplied by this that you add to this. And so now if you take this image and you show it to the same machine learning model it's going to say gibbon, so monkey, with almost 100% confidence. So this shows you that not only the model is making the wrong mistake, the wrong prediction, but also with more confidence than before. And so if you take this out of this benign example you can see that basically this gives the ability to adversaries to control what the model will predict and the way that we find these perturbations is by taking the model-- and the reason why the models are easy to train is the reason why they're easy to attack. Because if we can basically compute derivatives of the model to train them to update their parameters and make them more accurate then we can use exactly the same algorithms and compute the derivatives and estimate their sensitivity to their inputs and use that as a heuristic to modify the inputs and force the model to make the wrong prediction.

The threat model slide

So when we did this initial work people said well you need access to the model because you're computing these gradients so this won't work if you don't have access to the model. So we went out and asked ourselves can we do exactly the same thing, but in the black box setting where the only thing we can do is take an input, present it to the model and see what label the model predicts?

Attacking remotely hosted black-box models slide

And so the strategy that we used is to basically extract a copy of the model. So what happens is that we take these synthetic inputs and we submit them to the model that we're attacking. And so the model that we're attacking is going to respond with labels. What this means is we now have inputs and labels for them so we can re-train a different model on our machine. And so we can basically gradually extract the model that the defender is using and create a copy of it on our local machine. And because the remote system is giving us free labels this is really easy for us to do. All we have to do is carefully select the inputs in a way that maximizes the amount of information that we are receiving from the victim model. And so by doing this once we've completed this process, and just to be clear, this process does not require that we have access to real training data. We can completely synthesize the queries, starting from random inputs. And so once we've done that we can use the copy of the model to find perturbations of inputs that will force our copy of the model to make wrong predictions and submit those to the victim model that we don't have access to. And so this victim model also misclassifies the images.

Adversarial example transferability slide

And so this is a very surprising property that we spent a lot of time looking at this and it turns out that across the space of machine learning techniques, this is pervasive. It doesn't matter if the adversary knows what technique you're using, they can just learn a copy of the model, even if it's a different type of model and use that to find inputs that are misclassified that you will also misclassify with your own model. Just to give you an idea, here we considered 5 types of technique; deep neural networks which are sort of the thing of the day, logistic regressions, support vector machines, decision trees and nearest neighbours. For each of those we found inputs that they misclassify and then we sent them to the other 5 types of models to see how likely they were to misclassify these same inputs. And so the numbers here in the matrix are the error rates. So how much input is computed here transfer to models that you evaluate here. And you can see that a lot of these numbers are very high. Which means that the adversary does not need to know what kind of model you trained. And so if you make this very concrete what this means is that if you have access for instance to a Tesla, you don't really need to know what model they used to learn how to drive, all you have to do is to be able to submit an input and observe how the car reacts and then you can learn enough about the system that you can attack it afterwards.

Properly-blinded attacks on real-world remote systems slide

So in practice we mounted these attacks. So for instance there are cloud companies that allow you to basically train models and you don't have access to the algorithm that they use to train the models so

this is for the perfect black box. And so what we found is that we can take our attack and submit a small number of queries, so this is the number of interactions we've had with the cloud company, and then we have an efficient copy of the model that we can then use to find inputs that are misclassified. So this is, here, the error rate of the model that the cloud companies have created. So you can see between 84% and 97% success rate for us as adversaries. And the number of queries that we have to make before we can compute as many misclassified inputs as we want is an order of magnitude smaller than the training set that they use. So this is a lot easier for the attacker than the defender.

Just to drive the point home, here are other examples of correctly classified images along with the model prediction here and these are adversarial images along with the model prediction. So you can see when the image is... it's really hard to notice, especially on a projector. Even if you look on a computer screen it's very hard.

Learning models robust to adversarial examples is hard slide

So this is sort of the sad part of my presentation. Where we actually don't have good ways to defend against this and the reason is that, as I mentioned at the beginning of my presentation, the error space is very large. So we don't have very good training algorithms to cope with that and we also have a very hard time to defend in a way that is robust to new types of attacks. So we can very easily generate these attacks when we create our models and teach our models how to defend against these specific attacks that we're aware of. But when adversaries come and design new forms of attacks we have to continuously retrain our models to learn about these new forms of attacks.

Admission control at test time slide

So what I've been interested in recently, and this is something that we're working on right now, is to basically open up the box and try to understand how models represent data and leverage this to figure out when they are making predictions that they should not be making because they don't have relevant evidence in the training set. So what we're doing is at test time when we pass inputs through our model architectures, we're looking for evidence in the training data of points that have very similar representations to the representations that we're outputting on our test input. And if we look at the labels of these training points, when the model is making the correct prediction, these labels are homogenous. So they all agree with the prediction that the model is going to make. For instance if you classify this image of a panda, here these are visualizations of the training points and the different representation spaces of the model, you can see the panda is always surrounded by lots of training examples from the panda class. Whereas the adversarial example is first surrounded by pandas but then as the model goes up its logic towards making a prediction, the image is now projected into a wrong class, for instance a school bus. In between it's sort of ambiguous. So we're using this analysis to figure out when the model is uncertain.

And this has a lot of advantages. One of them is that it allows us to interpret the model's predictions. So here this is an example where I presented this image to a classifier and I was surprised that it predicted basketball. Maybe this classifier is racist. It has learned some correlation between skin colour and the class basketball. But then I looked at the nearest neighbours, so these are the images that have the

closest representations internally in the model. And there actually not that many black people there, a lot of white people in the training images. And you can see that one of the correct heuristics is that a lot of them are dressed in white and the basketball is always high up in the air. And so what I did is I then cut the ball from the image and showed it to the same classifier and repeated the same analysis where I looked for training images that have the closest representation. And you can see now the classifier is predicting racket like in tennis racket. And that the images are all again people wearing white, so this is probably Wimbledon, and the background is also green and so on. So this allows us to not only figure out when the classifier is not making the right prediction, but why and what is the sort of quantified uncertainty in the prediction.

Is ML security any different from real world computer security? slide

But in the end this does not give us a provable guarantee that the model will make the correct prediction. And this is sort of a traditional problem that we have in computer security where we have to balance the cost of the protection with the risk of the loss. This is a very well-known principle and it's the same in real world security. If you have a house you put a lock on it, but it's not going to prevent bears from entering your house. It's the same thing for computer security. But the nice thing about machine learning is that a lot of its components are expressible using mathematics. So you can very easily describe a machine learning pipeline with mathematics. And so I hypothesized that this makes it much more amenable to more principled approaches to achieving security and privacy. And one of the examples that I give is that this makes it very similar to what happened in crypto. In the real crypto -- cryptography, not cryptocurrency. If you look a couple of years ago, the community was in a dead end before it formalized very precisely what is the game that adversaries and defenders are playing. And once that happened there was a lot of progress. And now we have crypto that we use everyday that is robust to some extent to attacks.

Machine learning does not always generalize well slide

One of the things that I've been working on is privacy. Privacy in machine learning shows that we can actually achieve provable guarantees and have a principled approach to securing machine learning models, very similarly to crypto. So I'm going to give you, in the rest of the presentation, examples of how we do this in practice. So if you remember this very simple example I gave earlier. Here, if you notice, I added one image of a dalmation. So this is what we would call in machine learning, an outlier. It's a very different input from the rest of the distribution. And what current machine learning models do when they see these types of training inputs is they just memorize them. Because that's the cheapest way for them to have a small error on these inputs and to maximize their training objective. What this means though is at test time if you present this image the model is going to say I'm 100% confident this is a dog. But then if you present a very similar dog that is not exactly the training point it's going to have very high ambiguity. And what this means is if you're an adversary you can look at the confidence values and tell which ones were used to train the model. So now this is fine for a cats versus dogs model but what if you're training a medical model with healthcare data that is very sensitive. Or if you're learning a language model from people's emails. Then you don't want people to be able to tell when the specific input was part of the training set.

PATE: Private Aggregation of Teacher Ensembles slides

And so we have ways to train models that are robust against these attacks and it's interesting to look at those because they point at some very subtle but interesting synergies between security and generalization, which is sort of the goal of machine learning. So let me just walk you through this approach very quickly. And the name of the approach is PATE. So even though I am French, I'll ask you to believe that I didn't come up with the name. My collaborators did. It stands for Private Aggregation of Teacher Ensembles. The idea is quite simple, you have this sensitive dataset that you want to learn from while respecting the privacy of individuals that contributed to this data. What you're going to do is you're going to partition this data in n subsets. And from each of these subsets you're going to train one machine learning model. So what this means is that you have n models that were trained using the same algorithm but independently because they learned from different training sets. And so now you can ask these models to make predictions jointly.

How you do this is you ask them to vote for predictions and very simply you build a histogram of the votes and you return the prediction that received the most number of votes. So this is very intuitive. You can see why if all of the models agree on the prediction then it means that the prediction doesn't depend on one of the individual partitions. So it doesn't depend on individual training points that you had in your originals--original data set. So you have this intuitive notion of privacy. However if you have cases like these where the histogram is such that multiple classes receive about the same number of votes then it's possible for one model to affect the class that receives the most number of votes by flipping its prediction. And so in these cases you're basically able to leak private info because changing one point can change the class that receives the most number of votes in this histogram.

So how do we deal with this? We add noise. If we add noise before we take the max then we can prove theoretically that this mechanism achieves a notion of privacy which is called differential privacy, which you may have heard of in the news. This is a very well studied mechanism in differential privacy.

But the reason that I wanted to mention this approach is that here you can see how privacy and machine learning are very well aligned. Because if you have a lot of agreement between the teachers then the prediction is very likely to be correct because you are making this prediction n times from different training sets. So this is a prediction that is very likely to be correct. It's made independently n times. If you have, again, a lot of consensus among the teachers the prediction is also very private because it's very hard for one model to change the outcome of the prediction. And so you have this really rare synergy between the security property here, which is privacy, and the performance of the system, which is machine learning. Which is machine learning generalizing well.

Privacy is aligned with generalization slide

So this is very rare and we've actually made a lot of progress recently this summer. Just exploding this observation we can design models that perform a lot better when they're trained with privacy if we incorporate the fact that we're going to train them of a privacy preserving optimizer in the design phase of the machine learning algorithm. And one interesting observation is that by changing one of the activation functions in the model -- so the models are basically just matrix multiplications that are

interleaved with non-linearities. And so we basically found that the non-linearities that are very popular today because they allow very efficient computations are actually much worse than the non-linearities that we used in the 90s for training with privacy. So it's better if we go back to the 90s if we want to train with privacy. And so this single change basically gave us the same amount of boost in performance than three or four years of theoretical analysis and it takes literally 15 seconds to change that in the models. So these are the types of insights that are very exciting. And that's basically the reason why I'm working on privacy for machine learning right now because it's very rare in security that you come in the room that you say I'm going to help you build a better system. It's usually 'Hi, can you give me 3% of your performance and I will give you this shiny security property'. So here we are able to work together with machine learning researchers and basically help them achieve properties like generalization that they were not able to achieve by trying to directly optimize for generalization. So this is really exciting. If you have any questions I will be around for a couple of minutes, otherwise you can shoot me an email anytime. Thank you.

[Applause]

Question 1: Great talk, thank you very much. I come from a very traditional security background so there used to be an 'Alice and Bob' kind of thing. So now I'm trying to understand this concept of teachers over your partitions. Explain that, but my understanding is that each teacher gives a score on one part of the ...one partition. So how are the partitions selected?

NP: If you look back here what happens is that here what you would do is typically you would train one single model from this entire dataset. So one big machine learning model. Instead what we do is partition the data and then we repeat the training process on each partition. The only requirement for these partitions is that if you have one point here then it should only be in only one of the partitions. So it's a real partition. And what this allows us to say is that when we change a point in the data here it's only going to affect one of the partitions and so we know it's going to affect, in the worst case, the predictions of only one of the teachers. So what this means is that here when we have these histograms we know that, be it in the worst case, if we change one data point one of these classes is going to go down by one and the other is going to go up by one. This is sort of the property. Basically when you want to obtain privacy you need to have a way to be able to estimate how sensitive the algorithm is to changes to its input. By doing this partitioning we're basically able to measure precisely what the sensitivity is.

Question 2: Today Google announced something called Explainable AI and Explainable AI is the contribution of each of the features to the outcome to the algorithm. How does that compare to the work that you're doing?

NP: So this is very related. Basically if you remember the very first picture which is sort of the reason for the title of the talk. This here is an explanation for why the model is making the prediction that it was making. So this perturbation here is basically computed by looking at derivatives of output of the model with respect to its inputs. And so what these tell us is that if we change any of the input features how is it going to affect each of the outputs that the model has? And so what we do here is we say let's find

features that when we increase them they're going to decrease the original class and increase the wrong class that we want to predict and if you want to explain what the model is doing you want to understand what makes the correct class increase, like what features have the highest gradients with respect to the correct class. And that tells you ...gives you some sort of saliency map that you can figure out which of the features contribute most to the prediction of the model. The one catch with these types of explanations is that they're very local. So if you have a slightly different input you might get a very different explanation for the prediction that the model is making. Another relationship is in this type of analysis here because again these types of ... the fact that we can find the training images with the closest representations is one way to explain why the model is making a prediction. We're basically trying to estimate the support in the training set.

Question 3: Thanks for the talk. A quick question on the ... before the aggregation you have a noisy aggregation rate. So when you have an ambiguous decision before that aggregation how sensitive is that to the final decision made after the aggregation?

NP: If you have an ambiguous input? So if the dataset is not well separated initially? So in that case it's very difficult to learn with privacy. So actually it's very interesting if you look at what is difficult... what kind of training points are difficult to learn with privacy they're basically the outliers or points that are sort of ambiguous between different classes. And that is exactly what privacy is trying to protect against. You want to learn points that are prototypical examples of each class but you don't want to learn things for which you have very few examples, or things where you can't really tell if they're from one class or the other.

And so this is explicitly why this mechanism here will respond something random if the bars are close for two classes. It's designed explicitly for that reason. Because this is where you would leak the most of the private information.

Question 4: One follow up to that. Then you would just transfer the teachers to a new…

NP: So I didn't describe this but here every time the teachers make a prediction, even though it's private it leaks a very small amount of private information so there is what we call the privacy budget which bounds how much private information is leaking. So in practice what this means is we can only ask a fixed number of questions to the teachers before we exceed our privacy budget. And so what we do is we ask them to label some unlabelled data for which we don't need to protect the privacy of and then we use the unlabelled inputs with the privacy preserving labels that the teachers produced to train another model which is called the student model. And that student model can respond to as many predictions as we like because the privacy cost is fixed at the end of the training of the student. So once the student has asked enough questions to learn, the privacy cost is fixed. In the worst case if the adversary has access to the student model they will, in the worst case, be able to recover the labels that were used to train it but these labels were produced with privacy. So we can release that model and put it ... we can give it to the adversary and still have the same strength of privacy guarantee.

Question 5: Thank you for your talk. In your talk you elaborated various ways humans have come up with to fool machine learning systems. I'm curious about the inverse process as to how machine learning systems can be used to fool humans in the form of optical illusion or deep fakes and how … and whether there are defenses or attacks in this domain. How your research informs this area.

NP: So these types of images, when you show them to humans of course we can correctly classify this image but what we found is we actually conducted a study where we controlled how long humans are allowed to look at the image. And if you allow the human to only look at the image for a very brief amount of time they will more likely misclassify these images as well. So there is one relationship here. So it suggests --and again I'm saying it suggests-- that we are able to use higher abstractions to correct some of these perturbations. As far as deep fakes are concerned, this is a completely different area where it is true that progress in models that generate data is allowing us to generate content that is very hard for humans to distinguish as being generated by machine from content that another human would have generated. This raises a very fundamental question for a society: how to deal with this.

The unfortunate piece of news is … I have lots of unfortunate pieces of news today! [Laughter]

So the unfortunate piece of news is that you can't really use technology to fight deep fakes because whatever detector you come up with for a deep fake will basically allow you to train a better generator for deep fakes because the algorithms are designed to learn from this interaction. So probably the best response is through education and policy and other responses than technology.

I guess privacy is working well, that's one of the positive messages.

DK: Are there any final questions? Well if not I would like to thank Professor Papernot for his fascinating talk. Thank you Nicholas!

NP: Thank you Deepa.

DK: And I would like to encourage you all now to enjoy the networking part of the event. Speaking of networking as some of you may know, I would encourage you all to join our online community for alumni, U of T engineering connect. It's a great way, if you're interested, to mentor people. And if you are someone who would like to be mentored, a more junior alumnus, there are opportunities to find mentors as well. And it's a very nice, strong community and a good way to actually communicate and connect with the department as well. You have to go, if you're interested it only takes seconds to join. You can join through a LinkedIn account. Or if you go to www.uoftececonnect.ca

Thank you, enjoy!