**Interviewer 1:** [00:00:00] So first, just so that we have it on record that you allowed us to tape the interview and share it.

**Interviewee:** [00:00:04] Yes, I agree.

**Interviewer 1:** [00:00:05] Ok, great, thank you! So, I think we'll start with some general questions and background questions. What is your current position at your job?

**Interviewee:** [00:00:15] It's fine. I am a Research Associate at [University: Removed for Anonimity], which is the equivalent of a postdoc.

**Interviewer 1:** [00:00:20] Ok. So could you tell us about your overall work experience, like how many years of experience do you have and specifically in machine learning systems?

**Interviewee:** [00:00:30] So I started working on machine learning in 2010. That was actually the main topic of my master's degree. My master was actually in machine learning, my master's thesis was in machine learning. And then I did a PhD in machine learning that started in 2012 any I finished December 2014, then I moved to cybersecurity, but I was applying machine learning models to detect malware, adversarial machine learning models to find ways to evade classifiers and then I moved to testing when I was applying different methods of information theory that is actually close to machine learning in terms of probability distribution and etc., but I was not directly applying machine learning, but at the same time I was also using some models of machine learning for malware stuff. So I think my experience is like 9 years of machine learning.

**Interviewer 1:** [00:01:24] Okay. So what type of deep learning/machine learning networks have you developed/implemented and by that I mean supervised/unsupervised/reinforcement learning?

**Interviewee:** [00:01:36] So I have developed a lot of algorithms for clustering, but they were not deep learning. They were actually based on genetic algorithms. So, I don't know if you're familiar with the spectral clustering?

**Interviewer 1:** [00:01:48] No.

**Interviewee:** [00:01:48] So, basically what they are doing is graph analysis. So they create a similarity graph, they analyze the spectrum of the graph and then they apply a clustering algorithm. In this case, k-means in the inner space of the spectrum. I have used a lot of classifiers, some of them were neural networks. I have used deep learning recently for work that was in identifying different ways to imitate the behaviour of an antivirus.

**Interviewer 1:** [00:02:20] Okay.

**Interviewee:** [00:02:21] But in my experience, deep learning is not always the most accurate classifier. I mean deep learning is very good for translations, is very good for specific things with images and etc. But for the basic classification, a boosting system or a random forest might be good enough, fast enough, doesn't require as many resources as a deep learning algorithm. Actually in my experience, I mean in the last paper I wrote, I was comparing with 20 different classification systems, including deep learning, and deep learning was the 4th/5h, I think, in the ranking.

**Interviewer 1:** [00:03:00] Okay. That is useful information. So which programming languages and frameworks have you been using?

**Interviewee:** [00:03:09] R. That is my favourite. Java. And Python for machine learning.

**Interviewer 1:** [00:03:14] Did you use specific frameworks like Tensorflow, Keras, Pytorch?

**Interviewee:** [00:03:18] Yes. Okay, for Java I have been using, when I started I was using WEKA.

**Interviewer 1:** [00:03:32] Okay.

**Interviewee:** [00:03:33] And there is an extension of WEKA for big data of online analysis, that is MOA - Massive Online Analysis.

**Interviewer 1:** [00:03:44] Okay.

**Interviewee:** [00:03:46] I also use for Python is Skipler (?). In R, for deep learning I use H2OYes.

**Interviewer 1:** [00:03:59] Okay.

**Interviewee:** [00:03:59] But I haven't used Keras. I have used Spark (?) for deep learning libraries. But neither Keras or Tensorflow.

**Interviewer 1:** [00:04:10] Okay, there were just examples of frameworks.

**Interviewee:** [00:04:11] There a very nice framework in R that I am using at the moment that is called MLR, that is for machine learning in general.

**Interviewer 1:** [00:04:23] Ok. Thank you. Here our background questions end. And as I told we have one general question, which is what are the problems/bugs that you have faced while developing these algorithms. So if you just start from there and tell us about the things that you have seen so far, that would be very nice.

**Interviewer 2:** [00:04:43] Like big problems, tiny technical problems.

**Interviewer 1:** [00:04:46] Everything.

**Interviewee:** [00:04:46] So the major problem is always to tune up the algorithms, it is the most basic problem. So when I was creating algorithms, one of the problems was possibly in the clustering algorithm, one of the problems was how to make sure that... You know that the algorithm is converging because, in my case, I had a genetic algorithm and had the fitness function, and the fitness function was saying that the solution was the maximum value compared with others.

**Interviewer 1:** [00:05:17] Okay.

**Interviewee:** [00:05:18] The problem is that the iterative process of the algorithm was not always able to converge. Okay. So my problem was always trying to find how to improve the iterative process.

**Interviewer 1:** [00:05:29] Okay.

**Interviewee:** [00:05:29] And that moved me in my case to find a lot of different ways of finding proper fitnesses to lead the algorithm to converge. So that was one of my problems. Another problem is when you have a lot of data, explosion of memory, so the computer is not able to compute it anymore. Actually, that made me to find representations of data that not were only doing a feature selection process, but also for example when I was working with online machine learning. I needed to find first an algorithm that... Do you know k-means?

**Interviewer 1:** [00:06:10] Yes.

**Interviewee:** [00:06:11] You know that there is an online version of k-means that just move the centroids. So I created one of the algorithms that actually not only moving the centroids, but it does define a vector called tendency of the center. So, if the centroid will be moving in an opposite direction of the tendency, the tendency was reduced, but the centroid didn't move.

**Interviewer 1:** [00:06:33] Okay. And what advantage did that give to you?

**Interviewee:** [00:06:36] It was a bit more conservative, it was using the momentum of the previous data to make sure that you are not affected by outliers.

**Interviewer 1:** [00:06:46] Okay.

**Interviewee:** [00:06:46] So that's another problem - outliers. In feature selection, the problem is to always to choose the proper feature, because if you are using wrapping , for example, another problem is that you might be overfitting, especially with classifier, but you might not be actually selecting the proper feature, because you have a lot of features that are correlated between them. They will force a classifier to think that those features are the relevant set and they will ignore other features that might be complementary. So in terms of bugs, I found a few bugs, real bugs in algorithms. Like, especially when they have automatic tuning of a specific parameter, they sometimes do the tuning only and you need to and you need to have a check. Because sometimes using a different tuning process that is not part of the algorithm, it finds like, in orders of magnitude, like maybe 10 points better than the automatic one that is implemented in the algorithm. So yeah.

**Interviewer 1:** [00:08:03] So I have a question. When you have this automatic tuning of parameters, how did you discover that what they have? That what has the algorithm provided is not the optimum?

**Interviewee:** [00:08:15] For example, in this particular clustering.

**Interviewer 1:** [00:08:18] Okay.

**Interviewee:** [00:08:19] So as I was working with that algorithm like crazy, because it was the main competitor to my thesis, I was able to discover after several iterations that the parameter was actually not properly optimized and there were better tools to optimize, for automatic tuning of parameters for machine learning in general, it is part of the MLR. And with that tool it was able to discover a better parameter whose orders of magnitude more stable, in a more stable way than the automatic system. That's amazing implementation. So I think more or less that's it.

**Interviewer 1:** [00:09:00] Okay. So did you use existing datasets to train your network?

**Interviewee:** [00:09:10] Yes, so every time that I create an algorithm I always compare with the USI (?) machine learning repository.

**Interviewer 1:** [00:09:23] Okay.

**Interviewee:** [00:09:24] My algorithm, for example, for spectral clustering which is focused on separating shapes. So, imagine that you have two spirals and you want to separate it automatically. So the points are defined in the spirals and if you want to separate them automatically. So there is a dataset for that and you can add noise to the spirals and etc. So these are synthetic data that you can just use for your algorithms. So, I normally find them online or I find different references to the papers.

**Interviewer 1:** [00:09:56] Okay. So do you pre-process this? Have you ever pre-process the training data and, if yes, so what kind of pre-processing steps you do?

**Interviewee:** [00:10:08] Normally the first thing I do, well when there are a lot of features. The first thing I do is to eliminate correlations.

**Interviewer 1:** [00:10:16] Okay.

**Interviewee:** [00:10:17] So everything that the correlations are higher than 85 or 90%, I just remove them, I have used wrapping. What else? Let me remember something.

**Interviewer 1:** [00:10:31] Did you ever have a case when you did not do some pre-processing?

**Interviewee:** [00:10:36] Yes, normally...

**Interviewer 1:** [00:10:37] But then you discovered that it causes you a problem and you have to go back and pre-process your data in some way.

**Interviewee:** [00:10:44] Yes. Yes. I remember once I was using the MNIST dataset. That is the one with the numbers from 0 to 9.

**Interviewer 1:** [00:10:55] Okay.

**Interviewee:** [00:10:56] It's a pixel-based dataset. I wasn't doing any pre-processing, just for testing and the results were around 70-something percent.

**Interviewer 1:** [00:11:06] Okay.

**Interviewee:** [00:11:06] And after the pre-processing the results went to a 98%.

**Interviewer 1:** [00:11:16] What kind of pre-processing step did you apply? If you remember.

**Interviewee:** [00:11:18] A very simple feature selection. First, correlations, then normalization of data, min-max normalization, a wrapping (?) feature selection, so might be, no it was not wrapping, it was a PCA project, that improved a lot from the previous one.

**Interviewer 1:** [00:11:44] Do you remember any other kind of problems related to training data that you've ever had?

**Interviewee:** [00:11:54] Any example that you can give? So what do you mean by...

**Interviewer 1:** [00:11:59] Like if you did not have enough data or had data that was wrong in some way.

**Interviewee:** [00:12:04] Okay. So, for example, yes, I remember, I was very annoyed with online neural networks. I was training the algorithm and the algorithm was not precise. It was not able to classify it, but the algorithm was actually properly implemented. The problem is that because I was training a non-linear neural network, I needed to do more passes through the training data. Because normally you only get one pass, but with online training it is so that you need to give it more passes depending on the quality of the algorithm, to make the algorithm converge.

**Interviewer 1:** [00:12:36] Okay.

**Interviewee:** [00:12:37] One of the biggest problems that I have...

**Interviewer 1:** [00:12:39] By more passes you mean more epochs or?

**Interviewee:** [00:12:40] Passing the same data twice or three times.

**Interviewer 1:** [00:12:47] Ok.

**Interviewee:** [00:12:48] One of the biggest problems I have, that I always have is balancing of the classes. When the classes are unbalanced, one of the classes, especially if you have a lot of it, one of the classes will have no representation and the algorithm will not learn. Okay, so that's one of the reasons I'm going to try to balance the classes in the pre-processing.

**Interviewer 1:** [00:13:08] Okay. So you always use existing datasets? You never had to collect your own data?

**Interviewee:** [00:13:12] No, I have collected my own data.

**Interviewer 1:** [00:13:14] Okay. So how did you collect your own data?

**Interviewee:** [00:13:17] Depends on the problem. So when I was working with Twitter, for example, I collected the data from Twitter and then I apply a pre-processing based on text mining. So in this case, it is like eliminating stopping words, clean the data, you know doing all the tf-idf stuff and then creating the vectors and send it to the algorithm. For malware, for example, we run the malware or we are doing static analysis of the malware and we put a feature related to in Android: permissions, interruptions, flows between different sources of the program and etc. Or even when I was working with Java malware we were collecting imports, just to check if we can find a function items that were related and then we just created the dataset based on that and applied machine learning. So that's more or less...

**Interviewer 1:** [00:14:10] Did you always work with unsupervised learning? Like did you ever had to do manual labeling for this data that you're collecting?

**Interviewee:** [00:14:16] Yes, when I was working with Twitter, hahaha.

**Interviewer 1:** [00:14:20] What kind of labels?

**Interviewee:** [00:14:21] I remember that in one of them… So when I was working with Twitter we were working on marketing campaigns, that is called "Social Corporate Responsibility" and there are different rates of corporate responsibility. This was with one of my collaborators from marketing. So, he knew about great sources of corporate responsibility and he was just having a look at Twitter. And he and one of his colleagues were doing manual labeling at the same time independently. So then they apply these Kappa and the value was higher than 95%.

**Interviewer 1:** [00:14:53] Okay. The value was higher than 95% when they agree that labelling was correct. Okay. Was it the initial Kappa value or...?

**Interviewee:** [00:15:05] It was the initial.

**Interviewer 1:** [00:15:06] So, they agreed most of the time.

**Interviewee:** [00:15:11] Yes.

**Interviewer 1:** [00:15:13] Okay, so let's talk about the model structure. So you said that you use deep learning algorithms.

**Interviewee:** [00:15:21] Yes, basic ones.

**Interviewer 1:** [00:15:24] So did you ever have to decide on the model structure? By that we mean the layers, the type of layer, the number of layers.

**Interviewee:** [00:15:32] Let me think about it, because I was using an automatic parameter tuning system. I think, I didn't change the number of nodes per layer, but it was changing the number of layers and the number of nodes. Normally, I leave the number of nodes per layer by default.

**Interviewer 1:** [00:15:53] So it was all automatic, you did not do it yourself.

**Interviewee:** [00:15:57] But when I was using neural network, not deep learning, but neural networks, I was choosing them. And normally my network will not be deeper than for 4-5 layers.

**Interviewer 1:** [00:16:09] You ever had problems because of wrong model structure? Where like it wasn't working, you added a layer/you removed a layer, you change something and it became better just by changing the model structure?

**Interviewee:** [00:16:22] With synthetic data, yes. With real data, I am not completely sure, because the evaluation is a bit... You have more dimensions and etc.

**Interviewer 1:** [00:16:31] So what do you mean by synthetic data?

**Interviewee:** [00:16:34] You know the xor(?) function?

**Interviewer 1:** [00:16:36] Okay.

**Interviewee:** [00:16:36] So with that example it was when I was testing my neural network algorithms. You knew that when you add more layers, sometimes it's easier to separate. It is how I was experiencing this layer problem.

**Interviewer 1:** [00:16:58] So do you remember any improvement steps that you have taken? Did you add a layer? Did you remove a layer?

**Interviewee:** [00:17:05] Yeah, of course when you add layers, the non-linearity improves.

**Interviewer 1:** [00:17:11] So in your experience adding more layers helps or? It is a very general statement.

**Interviewee:** [00:17:17] Depends. Adding more layers overfits, hahaha. So you need to overfit a little.

**Interviewer 1:** [00:17:26] So did you ever have to change dimensions of the layer or ...?

**Interviewee:** [00:17:31] I never worked with the dimension of the layer [inaudible]

**Interviewer 1:** [00:17:38] Ok, about hyperparameters. You told us a bit about it before, but did you like ever had to change hyperparameters like batch size/activation function and so on.

**Interviewee:** [00:17:48] In neural networks you mean?

**Interviewer 1:** [00:17:50] Well, yeah.

**Interviewee:** [00:17:52] I normally leave it to the automatic tuning. I leave like a lot of parameters by default and only a few of them are defined automatically. I don't remember which ones are the recommended in H2O. I remember that there were two of them recommended and I just left them to the automation.

**Interviewer 1:** [00:18:23] Did you ever use loss functions?

**Interviewee:** [00:18:27] Yes, for evaluation, you mean?

**Interviewer 1:** [00:18:31] Yes, for evaluation and training, both.

**Interviewee:** [00:18:32] Yes, for evaluation, I did.

**Interviewer 1:** [00:18:33] Only for evaluation? And did use the predefined loss function or did you ever use the custom-written one?

**Interviewee:** [00:18:37] When I was creating clustering algorithm, I used loss functions as an optimization. Normally you always start with Euclidean distance, which is the most common one and I remember that I was trying to improve it. I was trying to find all the kinds of distance, like the rather basis functions and etc. It's always difficult to get one from the top or the bottom of your head than this one. Because you don't really know in the loss function, you don't really know what is going to work in a general environment. So, I just use the common ones, at the end.

**Interviewer 1:** [00:19:33] Okay. Where do you train your models? Like do you use specific hardware? And have you ever encountered problems related to hardware?

**Interviewee:** [00:19:46] Of course, memory is full.

**Interviewer 1:** [00:19:48] So you had memory problems because your machine just was very weak or that your data was very big?

**Interviewee:** [00:19:57] Or maybe both? Haha. Whatever work I do, I always try to publicize everything. Okay, if I'm going to do 20 iterations of the algorithm, I want to run the 20 iterations at the same time in the same machine. So it must have enough memory.

**Interviewer 1:** [00:20:13] Okay.

**Interviewee:** [00:20:13] And when I was doing my thesis everything was in my laptop.

**Interviewer 1:** [00:20:17] Okay.

**Interviewee:** [00:20:19] So, as you can imagine, the memory, every time that they went to data sets with more than a hundred thousand data instances the laptop said "Okay, I had enough. I can't do this anymore."

**Interviewer 1:** [00:20:32] How did you deal with that?

**Interviewee:** [00:20:34] Yes, absolutely. That's the reason why the last chapter of my thesis is about online clustering.

**Interviewer 1:** [00:20:39] I see. Okay. So you used online clustering.

**Interviewee:** [00:20:41] Yes. I started working with online methods.

**Interviewer 1:** [00:20:44] Okay, so any other bugs/problems? Things that you did for the improvement of your model qualities that come to your mind at this point.

**Interviewer 2:** [00:20:55] Like technical stuff, like just bugs that occasionally pop up.

**Interviewee:** [00:21:00] Actually, there's something that is very useful when you're programming these algorithms: to use a profiler, because the profiler will tell you this specific function. For example, if you're doing a pairwise comparison in your algorithm or something like that, it is consuming too much time. One of the things that I used to do is to go to the function, look for alternatives to that function for example in R and find functions that are faster and can do exactly the same. Or if you are doing, for example, one of the common mistakes, when you are doing a symmetric matrix, is that you create all the matrix when you actually need to create just one of the triangles of the matrix. So, although this is not a real bug, but this is a misperception of what you should really do and one of the things that I found very useful is matrix factorization methods or sparsity methods that can actually optimize a lot, when you are working with big data, all the performance of what you are doing. So, normally when I have the project that when you are using a predefined algorithm, you are not really a hundred percent sure about the quality of algorithm. I remember one of the problems with Mahood(?) that was famous framework in Java in 2013-2014 is that it was very difficult to find how the algorithms were implemented and some of them were working very badly, especially the recommender systems. So there was a moment when people have stopped using it and moved to somewhere else, because it was difficult to change it.

**Interviewer 1:** [00:22:45] Okay. Do you remember any kind of bugs or errors that you got like, the most interesting one as they always ask that you have ever faced in your experience?

**Interviewee:** [00:22:58] You mean not mine, but from the algorithm that I was using.

**Interviewer 1:** [00:23:01] Yes. No, also some bugs that you have introduced when writing the code.

**Interviewee:** [00:23:06] Well, I remember from statistical tests.

**Interviewer 1:** [00:23:13] Aha.

**Interviewee:** [00:23:13] Now there was some sort of bugs with respect... For example, when you pulled a picture that was completely zero, the statistical test just... I can not remember which statistical test was this. It was one of the uniformity test, I was using with [people names: removed for anonymity].

**Interviewer 1:** [00:23:45] Okay, so this uniformity test was part of the machine learning algorithm that you were ...?

**Interviewee:** [00:23:57] It was part of the evaluation, but not the algorithm. [Pause]. I remember with specific characters, if you are using a text mining algorithm and you are using some characters that are extremely weird, you might have problems with the training.

**Interviewer 1:** [00:24:21] What kind of problems with training?

**Interviewee:** [00:24:23] Segmentation fault. Because, you remember this issue with the character that was blowing up [inaudible]. Okay, so it was something very similar. I don't know why, especially one of the problems is always there when you're working with Unicode or you're working with the ISO 15, or whatever it is called. So I think that Unicode is normally creating problems with UTF-8?

**Interviewer 1:** [00:24:48] Okay.

**Interviewee:** [00:24:49] I don't know, you always need to move to ISO or the opposite. I can't remember, one of them is generating problems.

**Interviewer 1:** [00:24:59] Okay. So this is when you're using textual data, like the one you were mentioning you were collecting from Twitter.

**Interviewee:** [00:25:04] Yes, when, for example, in the tweet you may have emoticons, you might have a special character that are from operating system like the "/" part that is from Windows.

**Interviewer 1:** [00:25:15] And what is the way to deal with this special characters and unicode?

**Interviewee:** [00:25:27] A lot of data pre-processing.

**Interviewer 1:** [00:25:28] Okay.

**Interviewee:** [00:25:28] There is a name for this kind of functionality on everything that is not a proper ASCII.

**Interviewer 1:** [00:25:34] Okay.

**Interviewee:** [00:25:34] But this is the perfect way, just to deal with that, otherwise it will not let you to proceed with your algorithm.

**Interviewer 1:** [00:25:42] Okay.

**Interviewee:** [00:25:43] But I haven't found any big problem like a real bug in an algorithm.

**Interviewer 1:** [00:25:51] We're also interested in cases when let's say you had a new your initial model, it had some accuracy and then you have taken some steps and the accuracy has increased.

**Interviewee:** [00:26:04] Increased or decreased?

**Interviewer 1:** [00:26:05] Increased. No, we don't want the accuracy to decrease.

**Interviewee:** [00:26:07] After re-training or something like that .

**Interviewer 1:** [00:26:09] So yeah, yeah. Do you remember any specific cases that you have done to improve the accuracy of your model?

**Interviewee:** [00:26:18] Yeah, I cleaning or feature selection of things that are normally... In the last paper I have this, where I was comparing different classifiers for mimicking antiviruses. And one thing that was very interesting is that one of the classifiers was always better than the rest. In every single case.

**Interviewer 1:** [00:26:36] Okay.

**Interviewee:** [00:26:36] So it was trying to learn 56 different antiviruses. And for each of them it was always better than the rest of classifiers. And this method is just very very simple. It has like, it's a voting system between very simple classifiers and it was competing with neural networks, random forest, and it was always at the top. Yeah, I think I would say choosing different classifiers always helps, improving parameters with automatic tuning and feature selection. I think that's especially the most relevant one, features that are not correlated.

**Interviewer 1:** [00:27:19] So you were mentioning about your work to detect malware in Java programs. And you said that you were analyzing the source code, imports and etc. So did you have like an initial idea of what you should analyze in the source code?

**Interviewee:** [00:27:36] No.

**Interviewer 1:** [00:27:37] And so how did that go? How you decided what to add, what to discard?

**Interviewee:** [00:27:40] It was very simple. We were looking for behaviours in the malware, but not running the malware, but in the static analysis. And the idea was, ok, normally you can obfuscate everything in malware, but the only thing that you cannot obfuscate is the third party called libraries, calls for those libraries. So we tried to find behaviors between those libraries, so let's try to see which libraries the program is calling to.

**Interviewer 1:** [00:28:08] Okay.

**Interviewee:** [00:28:09] And let's try to see if between a corpus of malware and corpus of [inaudible], if we can find using a frequent item algorithm, if we can find differences between the frequency, that we can find combinations of libraries in different algorithms. And we actually find that there were some groups that were proper form malware, some that were more likely to be benignware and it was helping us to distinguish between the two of them.

**Interviewer 1:** [00:28:35] So the call to third libraries was your distinguishing feature?

**Interviewee:** [00:28:44] Right.

**Interviewer 1:** [00:28:45] So, how did you decide how many Java programs you are going to analyze?

**Interviewee:** [00:28:50] Normally, from the state-of-the-art you say okay normally people are analyzing like a 1000 or 2000 malware or benignware. So, I used a similar corpus to compare in scalability. So, normally the most reliable thing is to try to be close with respect to the state of the art.

**Interviewer 1:** [00:29:09] Okay, so just using the amount suggested by state of the art was enough?

**Interviewee:** [00:29:14] Yes.

**Interviewer 1:** [00:29:14] Okay, you did not have to add more?

**Interviewee:** [00:29:17] No, normally not, I mean you can do it, if you are like I want to analyse millions of malware in 10 seconds. But that was not the case. Oh, one of the biggest problems that I have with ML algorithms is the time of training, specifically with deep learning.

**Interviewer 1:** [00:29:35] Okay. So have you ever done anything to improve the time of the training of your model?

**Interviewee:** [00:29:42] Yes, forcing to have less layers.

**Interviewer 1:** [00:29:44] Aaa

**Interviewee:** [00:29:44] Limiting the maximum number of layers of the algorithm

**Interviewer 1:** [00:29:45] So you decreased the number of layers... And did it end up in decrease of accuracy or did it remain the same?

**Interviewee:** [00:30:02] I mean, even if the accuracy was decreased, after 20 rounds you can not notice the statistical significance between the original one and the decreased one.

**Interviewer 1:** [00:30:15] So what was the improvement in time after you removed... How many layers did you remove and what was the improvement in terms of accuracy?

**Interviewee:** [00:30:21] Normally, 2-3%, and the improvement was like from 10 to 5 minutes. It was reasonable.

**Interviewer 1:** [00:30:29] From 10 to 5. So, it decreased twice.

**Interviewee:** [00:30:32] Yep.

**Interviewer 1:** [00:30:32] And the difference in accuracy was not statistically significant.

**Interviewee:** [00:30:40] No, it was not statistically significant.

**Interviewer 1:** [00:30:43] Okay. Did you ever had to add a layer?

**Interviewee:** [00:30:46] Not, really. I didn't, as I was limiting the number of layers, I was not able to add. But it was the automated system that was really doing everything. I knew that it decreases, because I checked. But I never checked if it was increasing significantly the number of layers.

**Interviewer 1:** [00:31:09] Okay. So you said that you were using different languages like Java, R, Python? Have you ever noticed that when switching to a different language or framework the number of bugs that you get decreases? If yes, what type of bugs.

**Interviewee:** [00:31:25] Actually, what I notice is that they are different. For example, it is clear that implementations are different. But for example, I remember k-means in Java and k-means in R is not normally having the same process for initialization of the algorithms, because it's one of seeds for the algorithm. But they were giving me different results. And, in terms of, the square error function R was better than Java.

**Interviewer 1:** [00:32:01] You mean better in terms of performance or accuracy?

**Interviewee:** [00:32:06] Better in terms of squared error. So, it is supposed to be very optimized and that was actually one of the reasons I started to use R.

**Interviewer 1:** [00:32:19] Okay. So, do you remember any bugs that took you a lot of time to fix?

**Interviewee:** [00:32:32] Mine yes. So, normally, my biggest problem was the data processing. Not the algorithm itself.

**Interviewer 1:** [00:32:42] I see.

**Interviewee:** [00:32:43] So when I was processing data from Twitter, or... It was a nightmare when I was using a CSV files, for example, because in Twitter you have spaces, you have tabs, you have commas inside. So, the CSV files are a nightmare, that's the reason I moved to Mongo DB, that just has ISO files and everything is organized and you can just make a query and the query is normally appropriate.

**Interviewer 1:** [00:33:13] So what was the purpose of your data collection from Twitter?

**Interviewee:** [00:33:18] Marketing analysis. So we have data from campaigns, marketing campaigns, and we will try to analyze, first, if we were able to identify it using clustering the campaigns by keywords. So we take all the data..

**Interviewer 1:** [00:33:34] So you are trying to identify, sorry, if the piece of text is a marketing campaign or not?

**Interviewee:** [00:33:39] So we have a purpose of tweets.

**Interviewer 1:** [00:33:41] Aha.

**Interviewee:** [00:33:41] We group them by topic and we check the topic of each group. And we try to identify this group is related to our marketing campaign or no. In that case, we create a network based on the users and then there was a directed graph based on how these tweets has been moved, the tweets of the cluster have been moved between the different users and then we apply complex network analysis to identify the kind of networks. And one of the things that we also did is to analyze when the campaign in time, when the campaign was more explosive or had more tweets and when it was more quiet, based on that changes of messages in different movements that we have in all dataset. So that was more or less what we were doing. The machine learning part was just the topic identification that was giving us information about the campaigns.

**Interviewer 1:** [00:34:44] This is when you had two people who were labeling it manually, right?

**Interviewee:** [00:34:49] No, that's another.

**Interviewer 1:** [00:34:51] So, did you label this data?

**Interviewee:** [00:34:52] No.

**Interviewer 1:** [00:34:52] It was just clustering thing?

**Interviewee:** [00:34:52] We did clustering and [name:removed for anonymity] who was the expert, saying [00:35:02] ok this topic is related to a campaign or this topic, you know, is not related to anything at all.

**Interviewer 1:** [00:35:08] So you had a human evaluator?

**Interviewee:** [00:35:10] It is impossible, we had 7 million tweets.

**Interviewer 1:** [00:35:16] Okay, I see. So did you have any heuristic by which you judged how your ...

**Interviewee:** [00:35:23] To measure the quality of the outcome? No, we just we just believed the internal clustering, metrics for evaluation. So, we were doing, do you know there's a package in C, called C-valid.

**Interviewer 1:** [00:35:41] Okay.

**Interviewee:** [00:35:42] These package runs different clustering algorithms and is using different internal measures for the clustering, for the robustness of the clustering. One is the silhouette (?), another one is the num index(?), another one is called a connectivity(?). Based on this internal methods, it will tell you or try to tell you what is the proper number of clusters for the data that you have. Because after we run for different numbers of clustering metrics, it will give you like the most stable solution.

**Interviewer 1:** [00:36:11] So, in a way it is automatic. But now that we've talked about this project, maybe you remember some problems that you face other than pre-processing your data.

**Interviewee:** [00:36:21] Pre-processing in terms of...

**Interviewer 1:** [00:36:29] No, you said that you had pre-processing problems because Twitter has special characters. And I was asking if you remember any other problems that you face in this project?

**Interviewee:** [00:36:39] Yeah, of course, I was complaining all the time that it was not able to read the data.

**Interviewer 1:** [00:36:47] And why?

**Interviewee:** [00:36:48] Because of the special characters.

**Interviewer 1:** [00:36:50] Ahh, ok. Nothing else? As soon as you were able to read the data, it was good?

**Interviewee:** [00:36:55] Out of memories, because of the size of the data.

**Interviewer 1:** [00:36:58] Aha. And how did you deal with this out of memories?

**Interviewee:** [00:37:02] Actually the only solution we had was to choose a better computer.

**Interviewer 1:** [00:37:06] Okay.

**Interviewee:** [00:37:06] So it was not easy. But there was one, but this is not actually related to machine learning, it is more related to the network analysis part, that we were creating the graphs for the network. The graphs explode in terms of size and it was not a problem of the size of the graph, it was a problem of the library.

**Interviewer 1:** [00:37:29] Okay.

**Interviewee:** [00:37:29] We had a timer library, I can't remember which one it was and then we move to a library in R.

**Interviewer 1:** [00:37:34] So you had a problem in library and you had to change it to something else?

**Interviewee:** [00:37:38] So, but it was not the machine learning part, the machine learning part was actually very simple for us, because they were very common algorithms that C-Valid was using.

**Interviewer 1:** [00:37:48] Okay. So do you have anything else to add?

**Interviewee:** [00:37:57] No, not really. Good luck testing machine learning algorithms, hahaha.

**Interviewer 1:** [00:38:07] Thanks a lot.