**Interviewer 1:** [00:00:00] Could you confirm on recording that you agreed that this would be taped and transcribed?

**Interviewee:** [00:00:04] Okay.

**Interviewer 1:** [00:00:04] Yeah, so I will start with some background questions. The first one is what is your current position at your job?

**Interviewee:** [00:00:13] Okay. I am a PhD student at [Removed for Anonymity].

**Interviewer 1:** [00:00:20] Okay, could you tell us what is your overall work experience and specifically experience in

**Interviewee:** [00:00:26] deep learning/machine learning systems?

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

**Interviewee:** [00:00:29] I've used the machine learning and deep learning basically in my application domain. So, basically, we perform traffic classification and the monitoring, and we are trying to perform these kind of tasks using machine learning based and deep learning based classifiers.

**Interviewer 1:** [00:00:51] Okay, so did you mostly do supervised deep learning/machine learning networks?

**Interviewee:** [00:00:58] Yes. Yes, up to now we mainly used supervised approaches.

**Interviewer 1:** [00:01:03] Okay, and which problems were trying to tackle? Like image classification/speech recognition?

**Interviewee:** [00:01:08] Okay, we are trying to tackle mobile and encrypted traffic classification. So, basically, we have traffic flows and we try to associate, so to classify each traffic flow, we try to associate each traffic flow to the specific mobile application that generates this traffic flow.

**Interviewer 1:** [00:01:29] Okay, and which programming languages and frameworks have you been using?

**Interviewee:** [00:01:35] We use Python and we basically use sklearn for machine learning and keras for deep learning.

**Interviewer 1:** [00:01:44] Okay. Thank you. So, actually, in this interview we have only one general question, which is what kind of bugs/problems/challenges you have faced while implementing this systems. So if you have anything on your mind, we could start from there and you would tell us.

**Interviewee:** [00:02:01] Yeah, okay, basically a problem that I usually found in this type of frameworks is that the performance evaluation part is not uniform. So, basically, when we have to use some performance evaluation matrix, like so for example, micro or macro f-measure, micro or macro ( ? ) and so on, we have to basically only extract the predictions from Keras or sklearn and then we have to write our script to perform this type of advanced matrix. So, on the part of performance evaluation that is a lack maybe in this tools, since we have to implement these parties, but by end based only on single predictions of the classifiers. Okay?

**Interviewer 1:** [00:02:51] Okay.

**Interviewee:** [00:02:51] This is from the general part. Then when I, since we try to use also advanced classification schemes, for example, when we have to implement some ensemble methods using, for example, different machine learning algorithms and then combining them, we have also to implement these ensemble methods, specifically the ones that should be trained externally. So we have to take different predictions of the different algorithms and then implement by hand an ensembled methods to combine them.

**Interviewer 1:** [00:03:29] So you have multiple models that make predictions and then you have to put the output together.

**Interviewee:** [00:03:34] Yes. And in this case this should not be done transparently. For example, in sklearn, but they need a two-step process. So, basically, you have to perform a prediction, store them and then implement by hand, for example, a combining methods. Like for example, a weighted average or a majority voting that should be implemented in the second step, basically, but also more advanced methods.

**Interviewer 1:** [00:04:06] Could you tell us like about some specific bugs that you faced quite often when you tried to implement these things, if you have anything on your mind, of course.

**Interviewee:** [00:04:17] Sorry, could you repeat the question?

**Interviewer 1:** [00:04:18] If you could you tell us about some specific bugs that you get while doing this. You know the ones that appear too often and...?

**Interviewee:** [00:04:27] One bug that I have found is for example, this is very specific to the implementation of the staked(?) autoencoder in Keras, that if I remember well, they are not, they were implemented like only in the first versions of Keras, then they have been removed and basically when we try to implement them in, for example, using Tensorflow, I'm not very experienced in Tensorflow, but we implemented them and we have found that their performance are not so good as we expected. So maybe there is some sort of bug in implementation and, indeed, they have not implemented, for example, Keras does not provide up to now an efficient implementation of this staked(?) autoencoder.

**Interviewer 1:** [00:05:23] What about the kind of bugs which you get because you have done something wrong, rather than the framework? Like you use them in a wrong way, you get some errors.

**Interviewee:** [00:05:36] Which type of bugs are usually found in the Keras framework, for example?

**Interviewer 1:** [00:05:40] No, not in the framework itself, but when you're using the framework, what kind of bugs do you have, the ones that you produce?

**Interviewee:** [00:05:52] From my my point of view?

**Interviewer 1:** [00:05:55] Yes.

**Interviewee:** [00:05:55] Okay, usually I found a bit difficult, so I usually wrong to implement for example in Keras the callbacks mechanism, that is a bit tricky to implement. So usually I have problems in this type of, when I try to implement this kind of code.

**Interviewer 1:** [00:06:14] Okay, do you have anything else to add?

**Interviewee:** [00:06:18] Do you have, sorry?

**Interviewer 1:** [00:06:19] Do you have anything else to add?

**Interviewee:** [00:06:21] From this point of view, no. My main concerns are about this type of callbacks mechanism and about the performance evaluation framework, that I told to you in the beginning of the interview.

**Interviewer 1:** [00:06:34] So, when training your models do you use any existing data sets or do you have to collect your data yourself?

**Interviewee:** [00:06:41] We usually have to collect our data by ourselves, because we use, since we use traffic traces, we have to collect traffic traces, for example, in the classic pickup format and we have to pre-process the data. For example, to generate pandas dataframe or numpy arrays that should be fed to our deep learning or machine learning architecture.

**Interviewer 1:** [00:07:10] Can you tell us about the problems with pre-processing your data? Did you ever had any bugs or problems when you were doing that?

**Interviewee:** [00:07:20] Yes, usually the main problems that I found when I pre-process my data, maybe this is a limitation, is the use of categorical data. For example, there are no so many problems when I consider numerical data, when I consider categorical data, for example, a number of machine learning or deep learning architectures are not able to manage, for example, the one of the encoding of these data. So, and this is the most used one in the current framework, so I have to leverage other types of encoding like the target encoding, but this type of encoding usually bias the final results they have found. This is one of the main problem with large heteregenous data, like the piece of network data, that are not usually numeric. So you have to perform this type of pre-processing, in order to make this data feasible to feed the machine learning or a deep learning classifier.

**Interviewer 1:** [00:08:26] Okay. So you said that you're using supervised machine/deep learning and that you collect your own data. So I would guess that you have to do some kind of manual labeling of the data.

**Interviewee:** [00:08:36] Yes. Usually we do, yes we do or manual labeling or at least manual validation, since in our case we use a system that automatically labels the data, based on the information present on the system on which we collect data. So, basically, we generate traffic and then we associate the generated traffic with the system, that in our case is the app name that generates this traffic. So in this case, we construct the ground truth for our analysis.

**Interviewer 1:** [00:09:18] So when you label it yourself or when you validate, do you often get mislabeled data? Do you get any at all? Is there any problem related to that?

**Interviewee:** [00:09:29] When we label it? Yes, when we don't have the label for the data, if there are only few samples, we just drop them. But if there are more samples, we try to reconstruct labeling using some approaches like the most common label found in the traffic trace or try to study guides in some way, the traffic flows that we have collected.

**Interviewer 1:** [00:09:59] So did you ever have a case where you had like your training data, you train, you saw that the outcome is not, like the accuracy is not high and it was due to the presence of some wrong labels in your training data?

**Interviewee:** [00:10:13] No, in this case, I have not found some of these pathological cases, since we have this automated system that tries to be as much as accurate as possible when we label our data. So we try to avoid these errors.

**Interviewer 1:** [00:10:30] Is there more than one person who does the labeling?

**Interviewee:** [00:10:36] No, we do it in basically in an automatic fashion. So we use this systems that associates the system information with the traffic information and we construct our ground truth and then we validate them on some symbols. But now the system is accurate in this association.

**Interviewer 1:** [00:11:02] Okay, so any other problems related to training data that come to your mind? Bugs/problems/challenges, things that you had to do about training data to improve the accuracy or performance of the model?

**Interviewee:** [00:11:18] Sometimes we found that, especially with Keras, with deep learning classifiers when we have a lot of data, of course, maybe this could also be caused by the fact that we don't have so powerful machines to perform training on large amount of data, we have that training stops without generating any type of bugs or errors. Maybe, for example, since we have memory errors, because we have too much data and the we can not fit our model using Keras. But the problem is that in this case Keras does not provide any information about the feeling of these procedures. It simply generates not a number weights for our model and so your training phase fails basically.

**Interviewer 1:** [00:12:09] So, do you like perform any steps before training to try to anticipate this problem so that you don't get these memory issues?

**Interviewee:** [00:12:18] Yes. We usually to avoid this problem try to periodically check if there has been a problem and we save the model that is being trained, so the weights of the model that has been trained just before the error occurs.

**Interviewer 1:** [00:12:39] Okay, so maybe we could talk about the model structure. So like layers, dimensions of layers. Have you had problems related to wrong model structure?

**Interviewee:** [00:12:52] Yeah, one limitation that I found when I try to optimize the model structures that Keras provides some tools to optimize the single layer, but for example when we have complex structure, like for example, a convolutional neural network hybridized with the long short term memory in this case, there is no automatic tool to perform the selection of the number of layers and the type of layers. So you are able only to optimize the single layer using, for example, the setting for example, the dimensionality of the parameter of that layer, but you are not able to optimize in conjunction different layers of a more complex architecture.

**Interviewer 1:** [00:13:43] Okay. So did you ever have problems, like when you realize that you are using too many layers or too few layers, where you have to add or delete the layer?

**Interviewee:** [00:13:55] Yes, I found some problem,specifically based on the convolutional neural network, when you add too many layers, to this type of convolutional neural network. We found that using a stratified cross-validation, the performances are really different between the different folds(?) . And this is not expected to see, since we use the pre-processing step that randomize our dataset and then the cross-validation is ratified. So, it should guarantee that in every fold(?) is the same the proportional number of symbols of each class. But in this case we have that we found that four different folds(?) we have a variance in performance that is very very high, especially related to convolutional neural network with more than 5-6 convolutional layers, so more complex convolutional neural networks.

**Interviewer 1:** [00:14:55] Okay, I see. So what about hyperparameter tuning, any problems that you have faced related to that?

**Interviewee:** [00:15:04] Which kind of hyperparameters?

**Interviewer 1:** [00:15:07] Any kind.

**Interviewee:** [00:15:08] Yes, in this case one type of parameter that is most difficult to tune for me in my experience is the learning rate of our network. So, in this case this is maybe one of the parameters that the framework should provide more tools to optimize this type of parameter. And also the number, the dimension of the batch size is also another critical parameter that usually is very difficult to optimize.

**Interviewer 1:** [00:15:49] I see. So about learning rates. Do you remember any cases when like increasing or decreasing it helped the performance or accuracy of the model?

Yes,

**Interviewee:** [00:16:00] we have changed the learning rate of our optimizer. When changing it from 1 or 2 orders of magnitude, we have found that it impacts the performance of about up to 10% to 15% in terms of accuracy.

**Interviewer 1:** [00:16:27] Okay. So what about loss function? Do you use a predefined or a custom written loss functions?

**Interviewee:** [00:16:38] Yes, we usually defined a predefined loss functions.

**Interviewer 1:** [00:16:42] And did you ever have a problem with this loss functions?

**Interviewee:** [00:16:45] No. Basically, I have not explored this in depth. So, we basically use the predefined loss function. So I've not found any problem on this point of view.

**Interviewer 1:** [00:16:57] Okay, so you said like about the memory and about the machines that you are using that you don't have very powerful machines. So where do you train your models? And icould you tell us more about the problems related to hardware?

**Interviewee:** [00:17:11] Yeah, basically the problem is that now we don't have a GPU-equipped machines. So we basically perform a CPU-based the training. We perform them on cloud-based open stack.

**Interviewer 1:** [00:17:30] Okay.

**Interviewee:** [00:17:30] We have a number of virtual machines deployed on this cloud and we have to use at least 32 power and 64 gigabytes of memory.

**Interviewer 1:** [00:17:45] Okay.

**Interviewee:** [00:17:45] And up to 16 virtual CPUs. With configurations that are not powerful as this one, we are not able to perform experiments also on not too large experiments. In our case, we don't use a dataset that has more than 100,000 samples, more or less. So in this case, this is the hardware configuration that we use.

**Interviewer 1:** [00:18:18] So, do you remember any like bug messages that you got because you were using this specific hardware rather than...?

**Interviewee:** [00:18:25] In this case, usually when you use the pre-packaged version of Keras, that you find on the Python repository, the warning messages that is given by Keras that it is the compilation is not optimized for my architecture. So usually to have better performance, I have to manually compile and install Keras from the source code. I found, I've done some tries on the GPU-based architecture. In this case, I found that the performance are better. Especially in terms of the training needed to perform with the train of each epoch on my deep learning architecture. So, basically, the problem is that we need a lot of time to repeat each single experiment, also when we try to simply optimize our network.

**Interviewer 1:** [00:19:40] Okay, so any other problems that come to your mind not related to the parts that I've mentioned? Or even related to them, but the things that we haven't talked about, like are there any bugs or error messages that you think that you appear too often?

**Interviewee:** [00:19:57] No, I don't remember any pathological situation, that are not caused by my fault in the code.

**Interviewer 1:** [00:20:07] You should tell us about those cases too, that are caused by your fault. So what we would like to know is what is hard for developers?

**Interviewee:** [00:20:17] Okay, maybe the problem, the class of problems, of bugs that I usually found when I implement this type of solutions is related to the type of data that they need. So, basically, there is a types mismatch, that could be because of my fault or from the specific data type that are needed by each deep learning architecture. So they are, usually I have errors that the types are not coherent between them and so I needed to perform some casting or transformation, in order to make my data uniform and to perform and to continue.

**Interviewer 1:** [00:21:14] What kind of transformation? Do you remember any transformation that you had to do?

**Interviewee:** [00:21:21] Yes, as I feel before the main problems in my in my case is to manage strings and the categorical data that data needs to be transformed into numerical data. In this case, it is not usually simple to provide a meaningful transformation that does not introduce bias in the results.

**Interviewer 1:** [00:21:47] Okay. So about training data that you're using, did you ever have to manipulate it in any way, like any pre-processing?

**Interviewee:** [00:22:01] Yes, since we have, we use traffic-based data, we have to basically eliminate from our data all the information that are context-related, so that are related to the capture process. Like, for example, the IP addresses of the device that we use, they will be different from any other device on the network, or, for example, to Mac addresses and other information that should bias the final evaluation. So, basically, we try to instruct only the information related to the application that generates the traffic and that we will to classify. Like, for example, in this case since they generate the traffic on the network, we only expect the application-level payload, and then we use it to perform the training of our data. Okay. Moreover, we have to, for example, when we have for example string of bytes, we usually have to transform that in a numerical data, so each byte is transformed into an integer number between 0 and 255.

**Interviewer 1:** [00:23:27] I see. So how do you decide if you have enough training data? Let's say you have some amount of training data, you trained on that, you saw that the model is not performing well. Did you have this case? Did you collect more data? Or did you have enough from the beginning?

**Interviewee:** [00:23:43] Yes, we usually try to be as discreet as possible when we collect data. So, we usually try to balance it between the sample of each class. And if it is not possible, we try to use some techniques to avoid imbalancing. Like, the most used one is a cost-sensitive learning.

**Interviewer 1:** [00:24:12] Okay, so you had problems related to unbalance training data, right?

**Interviewee:** [00:24:17] Yes. Yes. Yes, especially when we use a pre-collected dataset, for example, related to attacks on the network. We have seen that it is highly imbalanced between the normal traffic, so benign, and the traffic that is related to the different attacks, that is usually less than the normal traffic. So in this case, we have to understand how to properly train our classifier.

**Interviewer 1:** [00:24:49] You also said that you have like different models that give your= predictions and you combine them? Do these models interact with each other or is it just each of them gives an output and you are just putting them together?

**Interviewee:** [00:25:04] We have different types of combiners. Basically, they are the hard to combiners, that only emits a verdict based on the prediction of each classifier, so which classifier operates separately. And then we combine the results, but we have also other types of combiners, that are based on the prediction probability, that is related by each classifier, and that could also be trained. So we basically have three different datasets, one for training the classifier, one to train the combiner and then one for testing.

**Interviewer 1:** [00:25:45] Okay, did you ever have problems related to interaction of these different models?

**Interviewee:** [00:25:53] In this case, we have some problems related to the fact that we have to implement them. For example, using also different languages. In this case, since we have a member of our team that is more expert in Matlab, we implemented these combiners in Matlab and they are fed with the predictions or the prediction probabilities, that are obtained using Python.

Okay.

**Interviewer 1:** [00:26:25] So in hyperparameters, you mentioned learning rate, right? Did you ever like had problems, not problems, but rather the effect of batch size or number of epochs on the...

**Interviewee:** [00:26:41] Yes. The one of the many hyperparameters that we try to do optimize is the number of epochs, since we were not experimenting that much with batch size. We usually use a batch size that is 32 or 50. We don't we don't change it too much, for the epochs, we try to improve the number of epochs. But we usually implement in the training procedure an early stopping procedure, trying to avoid the the overfitting of our network.

**Interviewer 1:** [00:27:23] Okay.

**Interviewee:** [00:27:23] So we try, for example, for 100 or 300 epochs, but implementing at the same time an early stop with to avoid overfitting.

**Interviewer 1:** [00:27:39] Okay. So one of the last questions, do you remember any kind of nasty bugs that you've encountered and that took a lot of your time? Yeah, if you remember it could be useful.

**Interviewee:** [00:27:56] Yes, the one of the last times, it's a, on which I spent a lot of time is about the need to, as I said before to categorize, to transform string features into numeric features, to feed my classifier. So I spent a couple of days trying different transformation schemes that could not introduce bias. But then when you have the correct pre-processed data ,the construction of the classifier is not too difficult to in my case.

**Interviewer 1:** [00:28:42] Okay. What is the output of your model in the end?

**Interviewee:** [00:28:49] Okay, in the end, we basically, in the end of our pipeline we usually use your commonly used metrics. So the accuracy, the F1 score and usually also I use the g-mean, that should be the geometric mean of the records of each class. Unfortunately, some metrics like g-mean, are not implemented in sklearn, we have to use another library, that is called InLearn(?). We also use the confusion matrices to perform, to show error patterns of our classifier, then finally, we also implement a reject option based on a classification threshold. So basically we evaluate all the performance, like this F-measure and accuracy, based on a threshold. So in this case, the only when the maximum prediction probability exceeds a defined threshold, we emit the classification. So in this case, we evaluate how the performance varies and also how much samples are dropped by the classifier for different values of these threshold from 0.0 to 0.9, of course, this is the ranges of our minimum the maximum prediction probability.

**Interviewer 1:** [00:30:22] Yeah, but when your model outputs, do I understand correctly that it says that this traffic belongs to this application? Is it how it's supposed to classify? So you train it on traffics and say that it's from this application and then if you give it...

**Interviewee:** [00:30:37] Yes.

**Interviewer 1:** [00:30:38] Okay. And like how many different applications you consider there?

**Interviewee:** [00:30:42] Okay, we usually consider, it depends, we have done some analysis in a binary case, trying to classify very similar applications like Facebook and Facebook Messenger, for example, so we try to understand if there are differences between similar applications. We perform also multi-classification tasks. In this case, we use hundreds of applications, no more than 100.

**Interviewer 1:** [00:31:15] And did you try to use more than 100?

**Interviewee:** [00:31:18] Yes, we tried also with 1.000, but usually the classifier is not able to train on this number of applications. So it is basically a compositional problem.

**Interviewer 1:** [00:31:32] Okay, because the training data becomes too big?

Yes, too much in

**Interviewee:** [00:31:37] this case.

**Interviewer 1:** [00:31:39] Okay.

**Interviewer 2:** [00:31:39] I have a question. So do you find the documentation for the libraries and frameworks you use extensive enough? Did you ever had some problems related to that?

**Interviewee:** [00:31:54] Okay, some of the recommendations especially for the Keras framework is that in the beginning to be more flexible on the possibility to implement the performance evaluation workbench. And then for more complicated tasks, like multitask or multimodal deep learning, so it is basically a multi-output or multi-input deep learning, there are the so-called functional APIs that we are trying to explore them, but they are not so complete. Like, for example, if you have deep learning network with multiple outputs. So in case of multi-task deep learning, in which you would try to train the network considering different tasks at the same time, for example, you have to define an early stopping. In this case, you have to define a custom early stopping, since it is not implemented in the framework an early stopping, in the case of multi-output DL network. So, in the cases of more complicated tasks or more complicated deep learning networks, there are some parts that should be customized, going also to modify the single libraries of Keras, for example.

**Interviewer 2:** [00:33:24] Did you ever had like some bugs related to customization?

**Interviewee:** [00:33:28] No, I'm not found some bugs because there are the classes fortunately our are prone to be expanded and changed. There is only a minimum of manual effort, but then you can more or less implement them.

**Interviewer 1:** [00:33:51] Okay. So do you have anything else to add ?

**Interviewee:** [00:33:55] No.

**Interviewer 1:** [00:33:57] She has one more question.

**Interviewer 2:** [00:33:59] Yes, I have one more question. You told us that you had some problems regarding the types of data you used. Was it something about dimensionality of your data?

**Interviewee:** [00:34:14] No, about the, no... In this case, we have, the only problems is that about the imbalance of dataset. No, we usually use a, in order to have the data of the same dimension, we use padding or truncation, but we haven't explored any specific issue about this type of problem.

**Interviewer 2:** [00:34:45] There is one little question about the libraries. Did youface the problems about versioning maybe?

**Interviewee:** [00:34:54] Yes, as I said in the beginning, there are some architecture that has been removed from Keras from one version to another. For example, the staked(?) autoencoder. So in this case when I use, I don't remember which is the version number, but when I use a past version of Keras, about two or three years ago, there was implementation of the staked(?) autoencoder, then when I update Keras, it was removed because simply maybe there is a bug. So, yes, sometimes I found some bugs, problems, but if you google it, you search for it, I found a solution in a little time.

**Interviewer 2:** [00:35:53] Thank you.

**Interviewer 1:** [00:35:54] Okay. So do you have anything else to add? Maybe while talking to us you remembered about some problems and bugs?

**Interviewee:** [00:36:00] No. As I said you, one of the main problems are about, are related to the pre-processing of data. Okay, and to the time needed. This is a problem common in deep learning to perform training, but I don't have any other specific problem to underline.

**Interviewer 1:** [00:36:27] I see. So, thanks a lot for your time.

**Interviewer 2:** [00:36:31] Thank you.

**Interviewee:** [00:36:31] Thanks to you

**Interviewer 1:** [00:36:31] It was very useful and nice to talk to you. Thank you.

**Interviewer 2:** [00:36:34] And if something comes to your mind or occasionally you have some new bug, you could let us know.

**Interviewee:** [00:36:41] I would drop you an email in case.

**Interviewer 1:** [00:36:43] Thank you.

**Interviewer 2:** [00:36:44] Thank you.

**Interviewee:** [00:36:45] Thank you. Bye. Bye. Bye. Bye.