A. RoBERTa Finetuning

Pour apprendre les protocoles encapsulés à partir des donnnées fibre, on doit envisager d'utiliser une approche hiérarchique dans laquelle on classifie d'abord le protocole de niveau supérieur (par exemple SDH), puis utiliser un modèle distinct pour classifier les protocoles encapsulés dans chaque trame SDH.

Par exemple, on implémente un classificateur binaire pour distinguer les données fibre des données non fibre. Une fois les données fibre identifiées, on utilise un autre modèle pour classer le protocole de niveau supérieur (par exemple, SDH). Enfin, on utilise un autre modèle pour classer les protocoles encapsulés dans chaque trame SDH.

Les trames SDH elles-mêmes peuvent être considérées comme une structure de données séquentielle, de sorte que les transformers sont SOTA pour modéliser les dépendances temporelles dans les données et classer les protocoles encapsulés.

Cette approche hiérarchique va nous permettre d'abord de filtrer les données non liées à la fibre, puis de nous concentrer spécifiquement sur la classification des protocoles encapsulés dans les données liées à la fibre

B. Training itself

To use RoBERTa and transformers for your telecom protocol learning task, you can follow these steps:

Prepare your data: Convert your binary data into a text format that can be fed into the RoBERTa model. One way to do this is to use a technique called pulse code modulation (PCM), which converts the binary data into a series of numerical values that can be represented as text.

Fine-tune the RoBERTa model: RoBERTa is a pre-trained model that has been trained on a large corpus of text data. To adapt the model to your specific task, you'll need to fine-tune it on your telecom protocol data. You can use the Hugging Face Transformers library to load the pre-trained RoBERTa model and fine-tune it on your data.

Train a classification model: Once you have fine-tuned the RoBERTa model on your data, you can use it as a feature extractor to extract features from your telecom protocol data. You can then train a classification model, such as a logistic regression or a support vector

machine (SVM), on the extracted features to learn the telecom protocol.

Evaluate and optimize your model: Once you have trained your model, you can evaluate its performance on a held-out test set. You can use metrics such as accuracy, precision, recall, and F1 score to evaluate the performance of your model. If your model's performance is not satisfactory, you can try optimizing the hyperparameters, such as the learning rate, batch size, and regularization strength, to improve its performance.

C. Data Visualization

1 Bar Chart of Encapsulated Protocols: You can create a bar chart to show the number of instances of each encapsulated protocol in your dataset. This will give your team an idea of the distribution of protocols in the data.

2 Distribution of SDH Frame Lengths: You can create a histogram to show the distribution of SDH frame lengths in the dataset. This will help your team understand the structure of the data.

3 Heatmap of Correlation Matrix: You can create a heatmap of the correlation matrix between the features in the dataset. This will help your team understand the relationships between the different features and how they may impact the encapsulated protocol classification.

4 PCA Scatterplot: You can perform Principal Component Analysis (PCA) on the data and create a scatterplot of the first two principal components. This will help your team visualize the distribution of the data in two dimensions and identify any patterns or clusters.

5 Confusion Matrix: After training your model, you can create a confusion matrix to visualize the performance of the model on the test set. This will help your team understand which protocols are being misclassified and where the model can be improved.