Rethinking Network Traffic Scenario Prediction with 1D Convolutional Network

Alabi Bojesomo (Username: HungryLearner)
{asbojesomo@gmail.com

I. Introduction

The undeniable growth in deep learning to everyday activities understanding and simplicity has sprouted the recent awareness of various disciplines into how to benefit from the success of this direction. Network traffic scenario is an important study on understanding the relationship between the traffic parameter and the underlying scenario. It helps in making proper planning for the network configuration including bandwidth requirement among others.

Machine learning (ML) is a common endpoint for small-size data due to the size of the data because machine learning tends to work well in such scenarios compared to deep learning. However, deep learning has been successfully applied to tabular data, commonly tackled with machine learning algorithms. Traffic scenario is a time series data thereby posing a clear direction on how to tackle it. Being represented in tabular form, ML algorithms like RandomForest can be used. However, the nature of the target value which spans multiple steps posed a serious challenge to feature engineering, a fundamental necessity for using ML algorithms. Also, the data was sampled in microseconds, leading to an otherwise huge number of rows when concatenated for ML algorithms. Such a huge number of rows requires very high computational power, otherwise limiting ML usage for this data.

II. Methods

A closer look at a sample of the provided dataset in Figure 1 shows the possibility of using one-dimensional-based deep learning algorithms that work well with time series. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are generic algorithms that work well with such time series data. However, the high number of time spanned by a single category of label (Figure 1) could lead to long-term forgetfulness of both LSTM and GRU. Apart from this, LSTM and GRU requires a substantial memory for keeping states. Another possibility is the use of Transformer which could eradicate the forgetfulness concern but this will be impossible for such a huge time span in millions like the dataset. This limitation is because of the $O(N^2)$ and $O(N^3)$ memory and computational requirements respectively, for the self-attention computation.

In this report, 1D convolutional neural network (CNN) was used. The prediction task can be viewed as a 1D segmentation problem or simple waveform segmentation. Based on this, 1D variant of popular UNet architecture was employed with some tweaks for better performance. The use of 1D CNN, which is a deep learning approach completely eradicates:

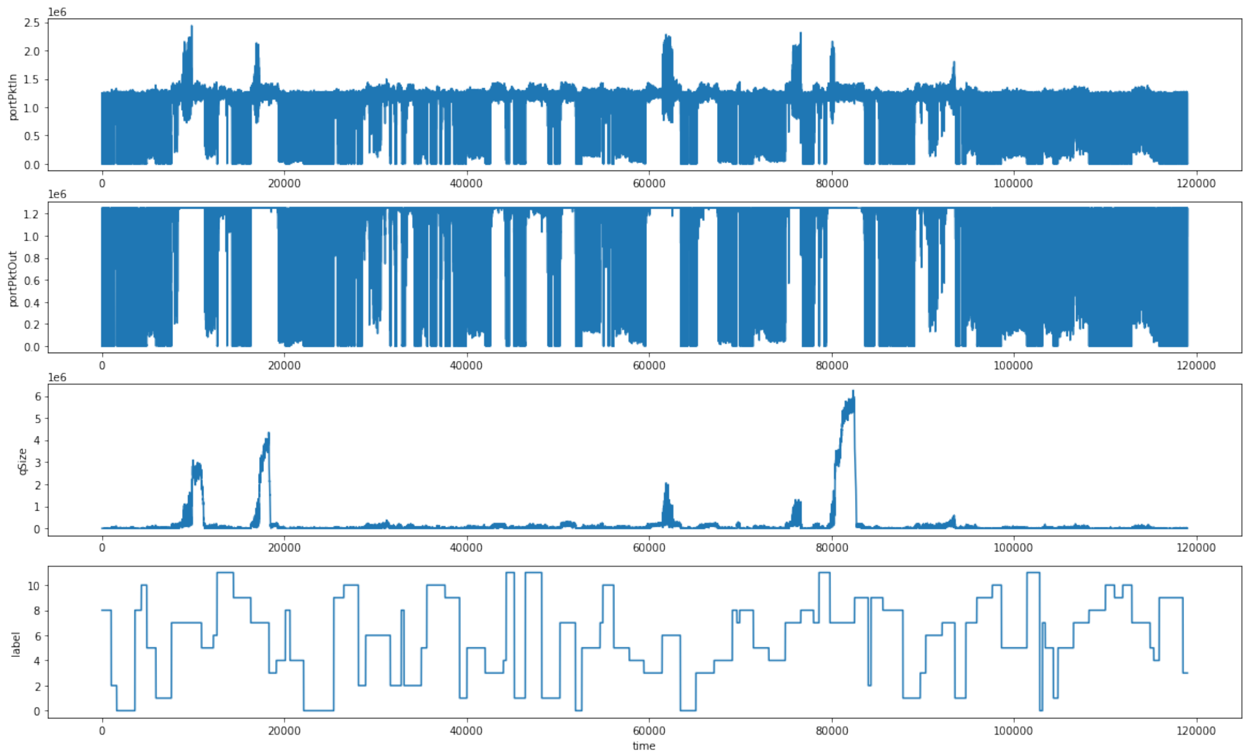

Figure 1. An Example series plotted. This shows the multiclass nature of the label. it also shows the huge number of steps (time) reflecting high computational power requirement for preprocessing for ML algorithms.

1) Feature engineering: ML algorithms require domain knowledge. There is a high relationship between the quality of features engineered in terms of relationship with the target and the overall performance. Engineering features require in-depth knowledge of the data, its collection, and many other time-consuming analyses. The use of CNN removes this requirement as the right feature will be learned during training.
2) Complex preprocessing: ML algorithms require careful preprocessing to avoid giving unnecessary importance to one feature above others. CNN on the other hand may simply require standardization only, a simpler preprocessing method. In fact, for this work, the only included preprocessing used are scaling and transformation. Since the time feature is in microseconds and other features are in order of 1e6, all the features are scaled by 1e6. Also, the resulting features are further transformed by using $\ln{(x+1)}$.
3) Computation power: The available traffic scenario prediction dataset used in this study is huge in terms of the number of rows for a single file, representing a single network. Using ML on such a number of rows will require high computation power as we have to train the ML algorithm with multiple networks at the same time for generalization to other network scenarios. The use of 1D CNN represents single network data as a single waveform, just like a single image in an image dataset. Doing so, the number of training samples is actually the number of network data provided.

Table I
EXPERIMENTAL SETUP

filters	Public Score	Private Score
64	0.7731	0.7728
96	0.7770	0.7769
128	0.7785	0.7782

The proposed model for this study is a 1D UNet architecture. This network is similar to regular UNet but composed of 1D components, including convolutional, pooling, and upsampling layers. To capture the wide span of label categories as shown in Figure 1, the network used a scaling factor of 5, filter kernel size of 5, and block depth of 5 in the encoding stage of the UNet architecture. The network also has five encoding stages to capture the coarser relationship of the data. Also, the Squeeze and Excitation (SE) block was applied to each encoding stage of the proposed model. One unique property of the proposed model is that additive filter increments were used. This is different from the traditional doubling employed in UNet. The choice of additive filter increment is one way of limiting overfitting and also helps to reduce excessive increases in the number of network parameters.

III. EXPERIMENTS

The 1D UNet architecture described in Section II was implemented in Pytorch and trained using Pytorch Lightning. The available 78 network data was randomly split into 10 folds for training the model and average predicted probabilities of the fold models were used as the final prediction of the test data. Three sets of experiments were conducted, varying the number of filters used in the first layer. Specifically, filter size 64, 96, and 128 was used and the resulting performance is shown in Table I. The final submission was a simple average ensemble leading to a score of 0.7791 and 0.7787 on the public and private leaderboard respectively.

IV. CONCLUSION

This work illustrates a novel application of 1D UNet to the prediction of network traffic scenarios. A simple UNet architecture with 1D convolution was designed with additive filter increment and SE block. This model was trained with just 78 network data with 10 kfolds leading to 2nd place solution for the Network Traffic Scenario Prediction Challenge hosted on Zindi.