“© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”
5G RAN service classification using Long Short Term Memory Neural Network

1st Mohamed Khadmaoui-Bichouna
University of The West of Scotland
Paisley, Scotland
https://orcid.org/0000-0002-3530-6419

2nd Jose M. Alcaraz-Calero
University of The West of Scotland
Paisley, Scotland
https://orcid.org/0000-0002-2654-7595

3rd Qi Wang
University of The West of Scotland
Paisley, Scotland
https://orcid.org/0000-0002-7764-9858

Abstract—5G brings many benefits such as enlarged capacity and improved connectivity. However, it also poses challenges especially due to a significant increase in the amount of traffic on the network. This creates difficulties for operators to maintain the Quality of Service (QoS) for each of the services offered. Therefore, in order to improve the performance of such capabilities and, consequently, the experience of the users, it is necessary to identify which traffic requires more prioritisation. This would help allocate more resources to those services. This concept makes the identification and classification of traffic to gain more and more relevance and importance. In this paper, we propose a Long Short-Term Memory (LSTM) model to classify 5G Radio Access Network (RAN) behaviour into four different scenarios: streaming, video conferencing, Voice over IP (VoIP) and gaming. The results obtained show a 93% accuracy.

Index Terms—5G, Artificial Intelligence, LSTM, Deep Learning, Service Classification

I. INTRODUCTION

The increase in mobile internet traffic in recent years has been remarkable and it is expected to go even higher, with a forecast of up to 403 Exabytes (EB) by 2029 [1]. One of the perpetrators of this event is the 5G. Its introduction as the new-generation wireless technology deployed for multiple use cases has been notable. In addition to its approaches, the increase in the number of simultaneously connected devices [2] results in the increase of internet traffic. Its implementation has become a standard in industry [3], [4]. The advantages it offers such as increased bandwidth, improved speeds and reduced latency make it the technology of choice [5]. However, an increase in the number of users has also introduced significant challenges in network management. The Quality of Service (QoS) offered may be affected as the resources of each network are limited [6]. Consequently, the performance of the services offered may be downgraded. One solution to this type of situation lies in ensuring that priority and thus sufficient resources should be given to the most important services. Thus, knowing how to differentiate the diverse services is crucial. With this knowledge and considering the possibility of the network slicing technology offered by 5G networks, it is pivotal to prioritise the most important services by assuring their QoS. Nevertheless, the identification of these services is difficult.

Many studies have sought to classify traffic and services using various techniques. In [7], Xiao et al. generate a dataset with traffic from various applications and create a pipeline to convert this traffic into numerical vectors. Once this is done, they look for Machine Learning (ML) models based on decision trees that best classify the traffic. Their experiments conclude that the Light Gradient Boosting Machine (LGBM) achieves 95% accuracy with a response time of 10 ms. Zhuang et al. [8] target to classify Over-The-Top (OTT) Virtual Private Network (VPN) encrypted voice traffic. They develop a framework that includes Long Short-Term Memory (LSTM) models and Convolutional Neural Network (CNN) to extract characteristics of the traffic. The conclusion they reach is that LSTM models are promising architectures that help the identification of OTT voice applications. In [9], Noor et al. aim to classify different 5G services in order to assign more concrete and optimised slices. Different ML algorithms are compared, including tree-based models, Support Vector Machines (SVM) and k-Nearest Neighbor (kNN) or neural networks. In [10], Georgios et al. seek to identify network traffic in order to detect anomalies. They develop a pipeline in which traffic characteristics are extracted and passed through an image-based AI model (ViT and CNN). This raw traffic is then transformed into black and white images. The dataset used for this research was CIC-IDS-2017 [11] and the results obtained were promising reaching 98% of accuracy. And finally, in [12], authors propose a service classifier to improve classification and to support a better distribution of resources and traffic over beyond 5G networks achieving a 97% of accuracy.

Most of the studies use machine learning or deep learning solutions to achieve this goal but, the datasets used are based on traffic information such as IP addresses, ports and so on. This involves inspecting the traffic which entails a time expense related to waiting for the traffic to reach the application layer, and expecting it to be decrypted. If the traffic is encrypted, it is possible that it would not resemble the same traffic used to train the model and would therefore cause the model to behave incorrectly. As a solution to this, in this paper, we propose a LSTM model capable of classifying services based on the behaviour of the 5G RAN in 4 different scenarios: streaming, video conferencing, VoIP and gaming. The model does not use traffic information, but RAN metrics, so it is independent of what is contained in the traffic itself, which makes it suitable for encrypted traffic. With this, the
contributions of this paper can be summarised as follows.

- Creating and labelling a dataset with data extracted from the live beyond 5G RAN.
- Presenting an LSTM model capable of classifying RAN services in 4 different scenarios.

The rest of the paper is organised as follows. Section 2 explains the architecture deployed to extract the metrics, obtain the dataset and train the model. Section 3 describes the tested and the dynamics for the creation of the LSTM model. Section 4 presents the empirical validation of the model and finally, section 5 provides the conclusions of the paper.

II. ARCHITECTURE

This section presents the framework used for the extraction of the dataset. It also covers the training phase of the neural network. Figure 1 presents the architecture of the proposed framework.

![Fig. 1. Framework architecture deployed for RAN behaviour classification.](image)

A. Framework deployed

5G architecture is composed of two main components: edge and core. The edge has two components called distributed unit (DU) and centralized unit (CU). The functionality of each one is related to the 5G protocol stack layers. On the one hand, the CU is in charge of the higher layers such as the Service Data Adaptation Protocol (SDAP), the Packet Data Convergence Protocol (PDCP) and the Radio Resource Control (RRC). On the other hand, the DU is in charge of the lower layers, the Radio Link Control (RLC), the Medium Access Control (MAC) and the physical layer (PHY). These two units together with the antenna are responsible of the creation of the wireless medium access, called Radio Access Network (RAN). These elements allow the user to be connected.

Moreover, the 5G core is composed of the standalone (SA) components. These are the Access and Mobility Management Function (AMF), the Authentication Server Function (AUSF), the Session Management Function (SMF), The Network Slice Selection Function (NSSF), the Unified Data Management (UDM), the Policy Control function (PCF), the Application Function (AF) and the User Plane Function (UPF). The first seven components belong to the control plane, in charge of the user session control. The UPF is in the data plane and its mission is to forward the user traffic to Internet.

Above the 5G architecture, there is the pipeline developed to extract the data from the RAN and to label its behaviour. In this part of the framework there are 5 phases that allow a correct classification. The first step is the extraction of the 5G RAN metrics. The software used for the 5G SA network deployment, srsRAN, allows to monitor the RAN behaviour with several metrics (see section 3B). Through this step, these metrics are extracted in real-time and stored in JSON files. The second step of the process is to collect the metrics and put them through a pre-processing stage to make them suitable for use with neural networks. This includes cleaning, labelling and normalising the data.

In the model creation phase, referred to in the figure as the training phase, a dataset is created with the metrics which is divided into a training and a test datasets. This division is 70-30% respectively. Finally, the LSTM model is continuously trained with different hyperparameter configurations to find the best configuration to obtain the higher accuracy.

When the model is already created/trained, after pre-processing stage, the samples are introduced directly into the model and a label is obtained indicating the service being consumed by the UE. Therefore, the figure shows two different scenarios, one for the creation/training of the model and one for the inference/execution of such model. They differ in the creation and splitting of the dataset.

B. LSTM Architecture

Long Short-Term Memory was proposed by Sepp Hochreiter in 1997 [13]. It is a type of Recurrent Neural Network (RNN) designed to model temporal sequences more accurately than RNNs. The difference between these two architectures is that LSTM introduces a memory cell which is composed by four elements: an input gate, a neuron with a self-recurrent connection, a forget gate and an output gate. These elements interact with each other in order to produce an output of the hidden cell. These gates are used to adjust the cell state. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell, and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. Because of this, the LSTM model is able to keep or discard information over time.

An important feature of this type of network architecture is that it works with sequential data. This means that the inputs need to be ordered sequentially in order for the model to learn the pattern. In our case, the samples used to train the model are timesteps that indicate the state of the RAN at a current time. Therefore, when training the model, the samples cannot be shuffled. However, when training the model with sequences, they can be shuffled because the timesteps by which they are formed are ordered. Each sequence will be labelled with the
scenario of the last timestep. In this way a window is created in which the model will learn to label the last timestep entered.

III. IMPLEMENTATION

A. Testbed

In order to deploy the architecture proposed in the previous section, several tools have been employed. Firstly, to achieve a functional 5G SA network, srsRAN (version 23.10.1) [14] together with Open5GS (version 2.6.6) [15] has been used. The first software is needed in order to have a functional 5G SA RAN. Open5GS is used to have a fully functional 5G SA core. To obtain the radio segment, along with srsRAN, a transceiver capable of converting digital signals into analogue signals working with an antenna capable of transmitting on 5G frequencies are needed. Since the 5G network deployed for this research work is a private indoor network, the transceiver used is the Ettus B210 and the antennas are dipoles capable of transmitting in the frequency band 78.

Both softwares are deployed on the same host, this is a Intel Xeon CPU E5-2630 v4 architecture with 10 cores operating at 2.20 GHz as the base frequency and a 32 GB RAM memory. UBUNTU 20.04 has been used as operating system running a Linux kernel version 5.4.0 version. A ONEPLUS 8T has been used as the UE.

Regarding the creation of the model, Pytorch has been used as the Python framework for the configuration of the architecture and execution of the experiments.

B. Dataset

srsRAN provides a functionality in which is possible to monitor some metrics related to the connection between UEs and RAN. These metrics are:

- Physical Cell Identification (PCI).
- Radio Network Temporary Identifier (RNTI).
- Channel Quality Indicator (CQI).
- Bitrate.
- Modulation and coding scheme (MCS)
- Number of packets successfully sent.
- Number of packets dropped.
- Pusch Signal-to-interference-plus-noise ratio.
- Buffer status Report.

Some metrics such as bitrate, number of packets successfully sent and dropped and MCS are obtained in both the uplink and downlink channels. This makes a total of 14 metrics related to RAN’s behaviour.

C. Data collection

Firstly, it is necessary to obtain RAN metrics related to each scenario. To do so and to be able to classify each scenario correctly, once the UE is connected, it consumed only traffic related to that scenario. This makes the labelling process much easier to carry out. In this step, the metrics were raw, no pre-processing was added.

The scenarios defined for classification in this research work were: video streaming, video conference, VoIP and gaming. Table I shows the different applications used to collect data.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streaming</td>
<td>Youtube</td>
</tr>
<tr>
<td>Conference</td>
<td>Skype / Teams</td>
</tr>
<tr>
<td>VoIP</td>
<td>Skype / Teams</td>
</tr>
<tr>
<td>Gaming</td>
<td>Call of Duty mobile / PUBG</td>
</tr>
</tbody>
</table>

Metrics were collected every second to allow enough time for a major change to occur in the RAN part. We also tried to make the dataset varied. For example, in the case of the streaming scenario, the device was forced to consume the video using different video resolutions. Consequently, the model would not relate the streaming scenario to bandwidth usage.

The total number of samples collected of each scenario is presented in the table II. A total of 32795 samples were collected in order to create the dataset. The dataset is balanced so that the model can equally learn the characteristics and patterns of the different scenarios. As mentioned before, each sample of the dataset represents a timestep of the state of the RAN and these have been obtained every second. This makes a dataset of 32795 seconds of RAN behaviour generating a total of 9 hours. The percentage of training and testing followed has been 70-30 where 70% of the samples have been used for training dataset and the other 30% for testing the model.

D. Data pre-processing

The second step is to pre-process the data to make it suitable for training a deep learning model. That is, to remove unnecessary metrics and select which features are going to be used for model creation.

From the metrics presented, PCI, CQI and RNTI were removed as these metrics are only for identification and do not provide information in the operation and performance of the RAN. So, in order to create a model capable of classifying the behaviour of the RAN, a total of eleven features were used.

Figure 2 shows the correlation between the 11 features used for the model creation. Knowing the correlation between the different metrics helps to identify which features will be most useful in the classification. The technique chosen to represent the correlation coefficient was the Spearman technique as it helps to obtain the relationship between two variables by covering also the non-linear relationships. An example of high correlation can be seen in the relationship between the number of packets sent successfully in downlink (dl_nof_ok) and the bitrate also in downlink (dl_brate). This makes sense...
since the higher the bitrate, the more packets are transmitted. Low correlation can be found in the relationship between the signal to noise level ($\text{pusch\_snr\_db}$) and the number of dropped packets ($\text{dl\_nof\_nok}$) as there is no relationship between these two metrics.

Figure 4 shows the correlation between each feature and the target, which in this case is the scenario. Clearly, it can be seen how the behaviour of some metrics can help to differentiate between different scenarios making it easier for the LSTM model to learn and distinguish between them. If we focus attention on metrics such as $\text{dl\_mcs}$, we can see how the behaviour of this feature differs significantly in the different scenarios. Another metric where clear differences can be seen is in the number of packets sent successfully in the uplink ($\text{ul\_nof\_ok}$). This is obvious because when the user is consuming streaming, practically no packets are transmitted as almost all the traffic is downloading, while in the video conference scenario, the user transmits video which consumes bandwidth in the uplink.

After studying the correlation between the metrics, these were normalised to improve training performance. This serves to ensure that the gradient descent algorithm converges more quickly towards the global minimum, making a reduction in the training process and also it helps with computational complexity by making memory usage much more efficient.

Before proceeding with the creation of the model architecture, each feature has been normalised between 0 and 1 by doing a min-max normalisation and the targets were encoded using label encoding.

### E. Proposed LSTM

Figure 3 presents the LSTM architecture proposed. The model consists of three primary components. The input layer has a size of eleven neurons to accommodate the features chosen in the previous section. The second component is formed by two hidden layers. The first one with a size of 32 neurons and the second with a size of 48 units. Each layer has a configured dropout of 0.5 so it helps with reducing overfitting problems. Finally, the third component is formed by an output layer with a size of four neurons which are the four possible scenarios. The final label obtained is the scenario with the highest probability out of the four possible scenarios in the output layer.

An important aspect of LSTM models is that one can select how far back in time the model will look in order to classify or predict the current sample. This is configured with the sequence size and is what will determine the ability to classify the different scenarios. The size refers to the total number of samples or timesteps that constitute this sequence. A small sequence size will make the model unable to find the necessary patterns to differentiate between scenarios and perform a correct classification. A large size will make the model too complex and inefficient. Finding the optimal size is one of the challenges when creating such models. So, to create the model, the different batches consisted of shuffled sequences of n timesteps. The idea behind the classification is that the model is able to classify a sequence according to the target associated to the last sample in this sequence.

To classify all samples, the sequences have been created in the form of a window. In this way, the sequences are like a window that scrolls through all the timesteps. This makes the number of sequences equal to the number of samples in the test dataset.

### IV. Empirical validation

This section presents the results obtained from the evaluation of the proposed LSTM model. The metrics chosen to
represents the evaluation of the model are: accuracy, precision, f1-score and recall.

- **Accuracy**: ratio of the number of correctly classified instances to the total number of instances.
- **Precision**: number of positive class predictions that actually belong to the positive class.
- **Recall**: number of positive class predictions made out of all positive examples in the dataset.
- **F1-score**: harmonic mean of precision and recall.

The first step to find the chosen model was to perform a series of experiments where the hyperparameters represented in table III were varied. This way it is tested how effective the model can be and with which configuration. After several experiments, the configuration which obtained better results was obtained. It can be seen in table III. With these values, LSTM model results obtained are shown in table IV. The columns indicate each validation metric. The last column, support, refers to the number of samples of each class used for testing.

On the one hand it is validated that the model is able to perform classifications with a 93% precision, with the conference scenario being the worst performer with 88%. On the other hand, the recall obtained by the model exceeds 96% in all classes except for the gaming scenario. These results are
better understood with the confusion matrix shown in figure 5.

![Confusion Matrix](image-url)

Fig. 5. Confusion Matrix obtained when testing.

The rows of the matrix indicate the actual classes and the columns represent the predicted classes. The sum of the row values gives the total number of timesteps tested for each scenario (support value). If we look at the gaming class row, we notice that the model has correctly classified 1997 samples, 47 gaming samples have been classified as “VoIP”, 286 as “Conference” and 211 as Streaming. In this case, the model was able to correctly classify 79% of the gaming scenario samples. This is indicated by the recall parameter in table IV. Conversely, looking at the gaming column, we see that when the model classified the unknown samples as gaming, 1979 were gaming, 19 were VoIP and 78 were classified as conference. This is what the 95% precision obtained in table IV refers to. The average accuracy obtained with the dataset for testing was 93% overall, matching the precision. With these results, it is clear that the most difficult class to classify is the gaming class, since, according to its recall, it can be confused with other scenarios.

Among the related work presented in the introduction section, [12] is the most similar since it uses KPIs in order to classify 5G services. However, those KPIs were synthetically created instead of collecting them from a real 5G network. Consequently, our 93% accuracy achieved is not comparable with their 97% accuracy obtained.

V. CONCLUSIONS

This paper has presented an LSTM model capable of classifying traffic of 5G radio access networks in different scenarios for improved QoS management. For this purpose, a framework has been designed where RAN metrics of a 5G network were obtained and introduced into a LSTM model. The proposed model is capable of identifying how RAN behaves by classifying four different scenarios: streaming, video conferencing, VoIP and gaming. Thus, a real indoor 5G network has been deployed using srsRAN and Open5GS software to obtain the necessary metrics to understand the behaviour of the RAN. A dataset of 32795 samples of the four different scenarios represented by a total of 11 features has been created. The proposed model was able to identify scenarios with a 93% accuracy. Future work includes adding features to the dataset to investigate if more scenarios can be classified; trying to improve the classification of the gaming scenario to obtain a much more precise model and adding more UE devices to the scenarios.

ACKNOWLEDGMENTS

This work is funded partly by the European Commission under two projects: 6G BRAINS: Bringing Reinforcement learning Into Radio Light Network for Massive Connections (Grant Agreement Number 101017226), ARCADIAN-IoT: Autonomous Trust, Security and Privacy Management Framework for IoT (Grant Agreement Number 101020259) and 5G-INDUCE (Grant Agreement Number 101016941).

REFERENCES

[15] Open5GS, “Open source implementation for 5g core and epc, i.e. the core network of lte/nr network (release-17).” https://open5gs.org/.