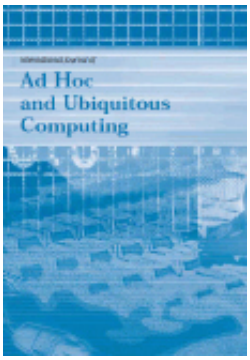



**International Journal of Ad Hoc and Ubiquitous Computing**

ISSN online: 1743-8233 - ISSN print: 1743-8225
https://www.inderscience.com/ijahuc

# 5G network traffic control: a temporal analysis and forecasting of cumulative network activity using machine learning and deep learning technologies

Ramraj Dangi, Praveen Lalwani, Manas Kumar Mishra

**DOI:** 10.1504/IJAHUC.2023.10052396

**Article History:**

| | |
|---|---|
| Received: | 07 September 2021 |
| Last revised: | 14 March 2022 |
| Accepted: | 18 March 2022 |
| Published online: | 16 December 2022 |

# 5G network traffic control: a temporal analysis and forecasting of cumulative network activity using machine learning and deep learning technologies

Ramraj Dangi*, Praveen Lalwani and Manas Kumar Mishra

VIT Bhopal University, India
Email: ramraj.dangi2019@vitbhopal.ac.in
Email: praveen.lalwani@vitbhopal.ac.in
Email: manaskumar.mishra@vitbhopal.ac.in
*Corresponding author

**Abstract:** In fifth generation (5G), traffic forecasting is one of the target areas for research to offer better service to the users. In order to enhance the services, researchers have provided deep learning models to predict the normal traffic, but these suggested models are failing to predict the traffic load during the festivals time due to sudden changes in traffic conditions. In order to address this issue, a hybrid model is proposed which is the combination of autoregressive integrated moving average (ARIMA), convolutional neural network (CNN) and long short-term memory (LSTM), called as ARIMA-CNN-LSTM, where we forecast the cumulative network traffic over specific intervals to scale up and correctly predict the availability of 5G network resources. In the comparative analysis, the ARIMA-CNN-LSTM is evaluated with well-known existing models, namely, ARIMA, CNN and LSTM. It is observed that the proposed model outperforms the other tested deep learning models in predicting the output in both usual and unusual traffic conditions.

**Keywords:** 5G; IoT; deep learning; traffic prediction.

**Reference** to this paper should be made as follows: Dangi, R., Lalwani, P. and Mishra, M.K. (2023) '5G network traffic control: a temporal analysis and forecasting of cumulative network activity using machine learning and deep learning technologies', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 42, No. 1, pp.59–71.

**Biographical notes:** Ramraj Dangi received his Bachelor of Engineering in Computer Science and Engineering from the Rajiv Gandhi Proudyogiki Vishwavidayalaya (MP) in 2015 and Master's in Computer Science and Engineering from the Samrat Ashok Technological Institute, Vidisha (MP) in 2017. He is currently pursuing his PhD in the Department of School of Computing Science and Engineering, VIT University, Bhopal (MP). His areas of research and interests are 5G, network slicing and optimisation.

Praveen Lalwani has completed his PhD in the Department of Computer Science and Engineering from the Indian Institute of Technology (ISM), Dhanbad, India. He has received his Master of Engineering in the Department of Computer Science and Engineering from UIT RGPV, India in 2012 and Bachelor of Engineering from the Computer Science and Engineering from LNCT, RGPV, India in 2009 with honours. He has qualified GATE examination more than five times. He has contributed articles and research papers in several refereed journals and conference proceedings of national and international reputes. He has also published several patents. He taught in IIITS, a Research Fellow in IIT, and SRM Chennai. Currently, he is teaching at VIT Bhopal University. He has received the fellowship in 2010 and 2014 and SERB Grant in 2022.

Manas Kumar Mishra currently works in the School of Computing Science and Engineering at the VIT Bhopal University. He has completed his PhD from the MNNIT. He does research in computer communications (networks). His most recent publication is 'Energy balanced data gathering approaches in wireless sensor networks using mixed-hop communication'. Successfully administrated teams in past as the Dean, Principal, Vice Principal (Academics), Deputy CoE, Head of the Department, etc. He is an able negotiator with demonstrated contribution in financial decision making process with roles like purchase in-charge of the University IT Resources. He played pivotal role in IT Infrastructure planning as a Professor in-charge of IT infrastructure, and also as a mentor for University LMS Design and Implementation.

## 1 Introduction

Most recently in three decades, rapid growth has been marked in the field of wireless communication with respect to the transition from first generation (1G) to fourth generation (4G) (Bhalla and Bhalla, 2010; Mehta et al., 2014). These devised networks are not able to meet the requirement of high bandwidth and low latency. To overcome these disadvantages, 5G network has been launched which provides a high data rate, improved QoS, low latency, high coverage, high reliability, and economically affordable services. 5G offers diverse services that can be classified into three sub categories:

1 Extreme mobile broadband (eMBB). It is a non-standalone architecture that offers high-speed internet connectivity, greater bandwidth, moderate latency, ultra HD streaming videos, augmented reality (AR)/virtual reality (VR) media and many more.

2 Massive machine type communication (eMTC) provides long-range and broadband machine type communication at a very cost-effective price with less power consumption. eMTC is a high data rate service, low power, extended coverage via less device complexity through mobile carriers for internet of things (IoT) applications.

3 Ultra-reliable low latency communication (URLLC) offers low-latency and ultra-high reliability, rich quality of service (QoS) which is not possible with traditional mobile network architecture. URLLC is designed for on-demand real-time interaction such as remote surgery, vehicle-to-vehicle (V2V) communication, Industry 4.0, smart grids, intelligent transport system, etc.

### *1.1 Evolution from 1G to 5G*

First generation (1G) cell phone was launched between the 1970s and 1980s. It was based on analogue technology which works just like a landline phone. In second generation (2G), the first digital system was offered in 1991 which provides improved mobile voice communication over 1G. At the time, when technology ventured from 2G GSM frameworks into third generation (3G) Universal Mobile Telecommunications System (UMTS) framework, users encountered higher system speed and quicker download speed making constant video calls. Fourth generation (4G) is purely mobile broadband standard. So, in digital mobile communication, it was observed information rate upgrades from 20 to 60 Mbps in 4G (Al-Namari et al., 2017). 5G is faster than 4G and offers remote-controlled operation over a reliable network with zero delays. It provides down-link maximum throughput of up to 20 Gbps (Agiwal et al., 2016). In addition, 5G provides unlimited internet connection at your convenience, anytime anywhere with extremely high speed, high throughput, low-latency, highly reliable, more scalable, and energy-efficient mobile communication technology (Buzzi et al., 2016; Dangi et al., 2021). The evolution of wireless mobile technologies is presented in Table 1.

### *1.2 Problem description*

Since, we know that the core network architecture is responsible for allocating resources to subscribers, a problem arises when there are a limited amount of resources available for distribution. In the event that resources are limited for various reasons like shortage of network stations, unavailability of adequate resources, etc. and the number of resources to allocate or the number of subscribers far exceeds the amount it can serve at that moment, then the devices might overload and fail to operate while trying to serve all the subscribers. Therefore, there is a need to predict the number of subscribers using network services at a certain point in time in order to efficiently scale the access and mobility management function (AMF) to limit the allocation and distribution of resources by the architecture. Scalability is one of the best features of a 5G network but when do we need to scale up and scale down the network resources. It is totally based on current traffic on the network. In this article, we are focusing to develop a scalable AMF 5G core network element. So, the AMF scalability is based on the number of user equipments (UEs) requests.

### *1.3 Author's contribution*

List of author's contribution as follows:

1 Data pre-processing:

- Read data from .txt files and convert them into non-null, interpolated and time-instanced data structures which can be used for further analysis.

2 Machine learning modelling:

- Apply autoregressive integrated moving average (ARIMA), 1D CNN, long short-term memory (LSTM) and a combined model of CNN-LSTM on the formatted data and predict the output at next time instance.
- Evaluate and analyse the results of each model.

3 Machine learning modelling (hybrid) approach:

- Apply ARIMA to the formatted data
- Apply a combined model of 1D CNN and LSTM on the residuals of ARIMA results
- Evaluate the model and compare the results with the previous approaches.

**Table 1** Summary of mobile technology

| *Generations* | *Access techniques* | *Transmission techniques* | *Error correction mechanism* | *Data rate* | *Frequency band* | *Bandwidth* | *Application* | *Description* |
|---|---|---|---|---|---|---|---|---|
| 1G | FDMA, AMPS | Circuit switching | NA | 2.4 Kbps | 800 MHz | 30 KHz | Voice | Let us talk to each other |
| 2G | GSM, TDMA, CDMA | Circuit switching | NA | 10 Kbps | 8,00 MHz, 900 MHz, 1,800 MHz and 1,900 MHz | 200 KHz to 1.2 MHz | Voice and data | Let us send messages and travel with improved data services |
| 3G | WCDMA, UMTS, CDMA 2000, HSUPA/HSDPA | Circuit and packet switching | Turbo codes | 384 Kbps to 5 Mbps | 800 MHz, 850 MHz, 900 MHz, 1,800 MHz, 1,900 MHz and 2,100 MHz | 1.2 MHz to 5 MHz | Voice, data, and video calling | Let us experience surfing internet and unleashing mobile applications |
| 4G | LTEA, OFDMA, SCFDMA, WIMAX | Packet switching | Turbo codes | 100 Mbps to 200 Mbps | 2.3 GHz, 2.5 GHz and 3.5 GHz initially | 3.5 MHz, 7 MHz, 5 MHz, 10 MHz and 8.75 MHz | Voice, data, video calling, HD television, and online gaming | Let us share voice and data over fast broadband internet based on unified networks architectures and IP protocols |
| 5G | BDMA, NOMA, FBMC | Packet switching | LDPC | 10 Gbpsds to 50 Gbpds | 1.8 GHz, 2.6 GHz and 30–300 GHz | 60 GHz | Voice, data, video calling, ultra HD video, virtual reality applications | Expanded the broadband wireless services beyond mobile internet with IOT and V2X |

**Table 2** Notations and abbreviations

| | |
|---|---|
| 3GPP | 3rd generation partnership project |
| ARIMA | Autoregressive integrated moving average |
| AMF | Access and mobility management function |
| ANN | Artificial neural network |
| CDR | Call details records |
| CN | Core network |
| CNN | Convolutional neural network |
| DL | Deep learning |
| LSTM | Long short-term memory |
| LTE | Long term evolution |
| MCG | Master cell group |
| MIMO | Multiple input multiple output |
| mMTC | Massive machine type communication |
| ML | Machine learning |
| MSE | Mean squared error |
| mmWave | Millimeter wave |
| NSA | Non standalone access |
| NVF | Network functions virtualisation |
| PCF | Policy control function |
| PDU | Protocol data unit |
| QoS | Quality of service |
| RAN | Radio access network |
| SA | Standalone access |
| SCG | Secondary cell group |
| SDN | Software-defined networking |
| SMAPE | Symmetric mean absolute percentage error |
| SMF | Session management function |
| SMS | Short message service |
| UDM | Unified data management |
| UE | User equipment |
| UPF | User plane function |
| URLLC | Ultra-reliable and low latency communication |
| VNF | Virtualised network function |
| VM | Virtual machine |

**Figure 1** Autoregressive integrated moving average (see online version for colours)

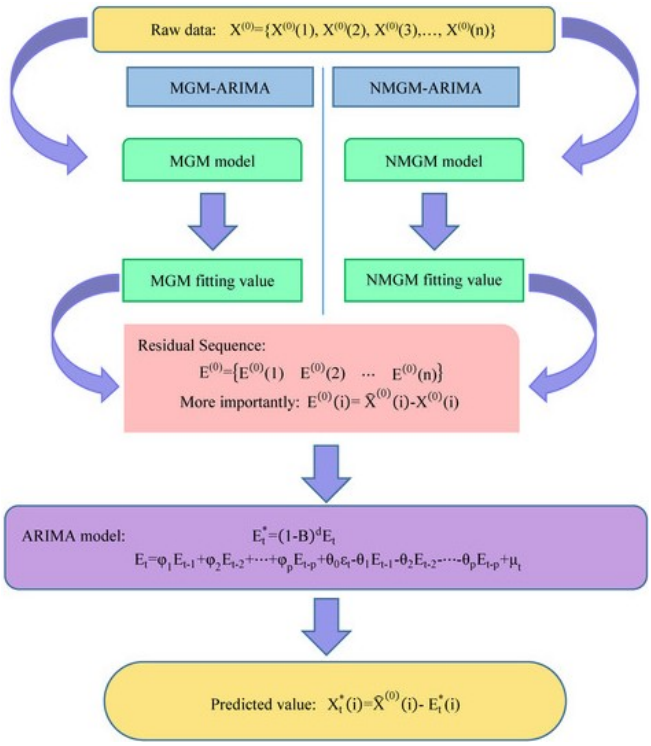


**Figure 2** Convolutional neural network

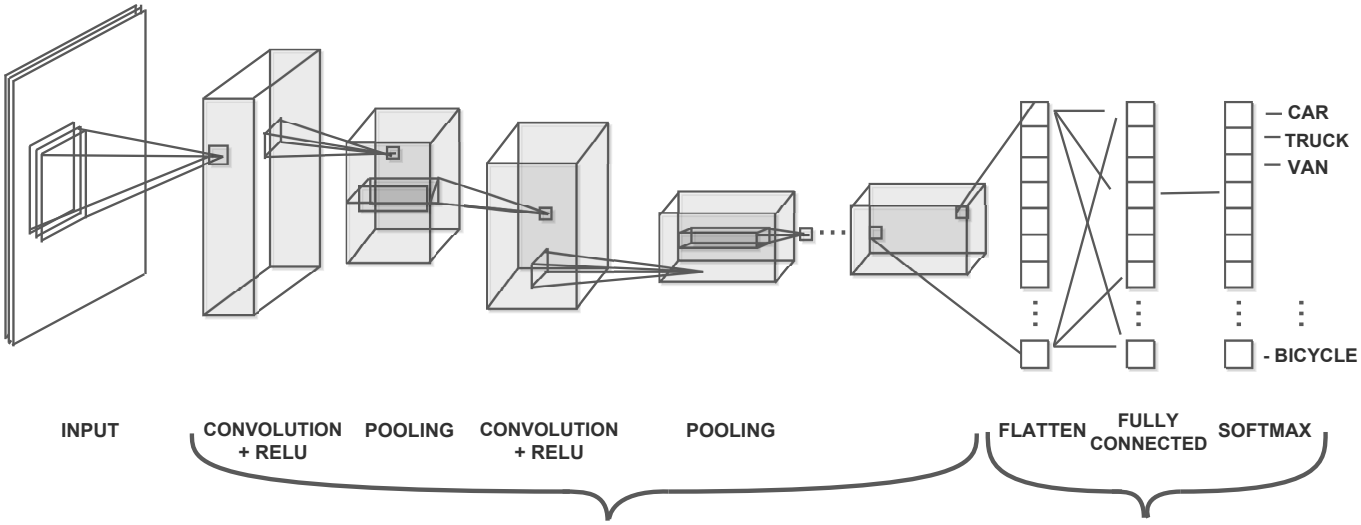
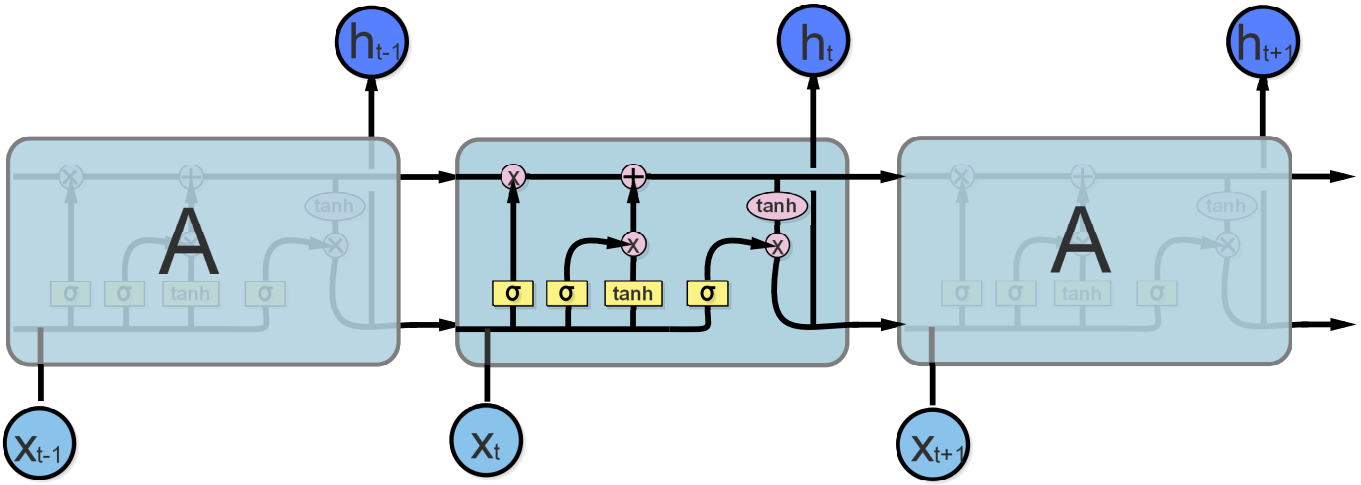



**Figure 3** Long short-term memory (see online version for colours)




### *1.4 Article organisation*

The remaining part of the paper is organised as follows. Section 2 shows the description of deep learning models, and notations and abbreviations. In Section 3, an overview of existing techniques is provided. Section 4 presents the proposed hybrid model, dataset description, data processing, and simulation framework. Analysis of proposed work with some existing well established models is presented in Section 5. Finally, Section 6 concludes the article and paves the path for future research direction.

## 2 Preliminaries

In this section, we have tried to describe the notations and abbreviations, as well as the various machine learning models, taken into consideration for the proposed methodology.

### *2.1 Notations and abbreviations*

The description of notations and abbreviations used in this article is presented in Table 2.

### *2.2 Machine learning models*

In this paper, both regression techniques and neural networks are proposed for the task of forecasting cumulative network activity over a time period. The regression model, ARIMA, is used as a linear model and CNN and LSTM is used as nonlinear models. A hybrid approach using all the stated models is also mentioned which drastically reduces the error rate.

#### *2.2.1 ARIMA*

The auto-regressive (AR) integrated moving average model is a forecasting algorithm that makes use of only the past values of the data in the time series to predict the future values. It is used on univariate data, where it uses its own time lags as predictors and constructs a linear regression

(LR) model out of it. In other words, it uses a linear combination of the lags of the data available and the linear combination of the lagged forecast errors of the same data to predict the values of that data at a point of time in the future.

An ARIMA model is constructed by combining both the equations and the full model can be written as.

$$y'_t = c + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t \quad (1)$$

where $y'_t$ is the differenced series.

This model is called the ARIMA ($p$, $d$, $q$) model, where $p$ represents an order of the AR part, $d$ is the degree of first differencing involved, and $q$ is the order of the moving average part.

Figure 1 represents a typical workflow of the ARIMA process explained above.

### 2.2.2 CNN

A convolutional neural network (CNN) is a class of deep neural networks most commonly utilised for image processing. Its architecture is based on shared-weights and translational invariance properties. They are most popularly used in analysing visual imagery. A CNN is able to properly capture *spatial* and *temporal* dependencies in the input matrix with the help of some specific filters and mainly by summarising a tensor or a matrix or a vector into a relatively smaller one. Thus, the number of parameters are also reduced and the weights are reused which leads to faster computation than a regular artificial neural network (ANN).

A CNN architecture consists of an *input layer*, an *output layer* and many hidden layers. The typically hidden layers of a CNN consists of a *convolutional layer* with an activation function, ReLu, followed by *pooling layers* and finally the *fully connected layer*. Figure 2 shows exactly how a 2D CNN works along with the various types of hidden layers used.

In our case, we used a 1D CNN instead of a regular CNN because the former is commonly structured in a way for it to be used for sequence problems, whereas, 2D CNNs are typically utilised for 2-dimensional data like images.

### 2.2.3 LSTM (RNN)

Long short-time memory is a type of artificial recurrent neural network (RNN) used in the field of deep learning. Unlike standard ANNs, LSTM has feedback connections and has the ability to process entire sequences of data. A typical LSTM unit consists of a *cell*, an *input gate*, an *output gate* and a *forget gate*. The cell remembers feature values over arbitrary time intervals and the three gates regulate data flow throughout the cell. LSTM networks are well suited for forecasting problems because of their architecture and thus is the most focused algorithm in this paper. Figure 3 shows a typical representation of an LSTM cell connected to former and latter cells.

## 3 Literature review

In this section, a plenty of recent approaches presented for traffic forecasting in 5G.

In 5G networks, AMF is the entry point. Therefore, the main focus is on diminishing the virtualized network functions (VNFs) which is hosting the AMF to manage the scalability of the 5G system. In 5G, scalability is not a hardware issue, it is like a software issue, in which, all the core network functions (CNFs) run as VNFs on the upper layer of virtualised infrastructure. One of the solutions for scalability provided by most of the investigators was a static threshold mechanism.

Dutta et al. (2016), Carella et al. (2016) and Alvi et al. (2017) have proposed threshold-based model for scalability. The idea behind this model is to launch a new instance during traffic load goes high and it automatically shuts down the instance, when traffic goes down. However, the static-based solutions are not good choice for mobile traffic. It is mainly taken into the consideration for commercial and open source cloud environment.

West (2018) and Open Source Cloud Computing (http://cloudstack.apache.org) have proposed a solution based on a static threshold. It also opted for various open-source cloud solutions. However, these solutions were not suitable for mobile networks, due to some limitations like reacting to unusual events takes time that depends on VNF scaling capacity and data centre architecture. Hence, an adaptive solution is required for the mobile network.

Arteaga et al. (2017) enhanced the scaling policy which was modelled according to the dynamic environment. The reason behind the improvement was the combination of Q-learning with a Gaussian process-based model. They also suggested dynamic solutions for scalability. But, in this technique reinforcement learning had been used to predict the correct decision, it took some time that was not acceptable in 5G technology which needs to handle both usual and unusual events.

Bilal et al. (2016) recommended a VNFs. In this approach, the authors provided the suggestions which were based upon the system-level information. They suggested a hybrid approach which is the combination of offline and online strategies by the help of which resources prediction can be done in both usual and unusual network conditions. In offline strategy, prediction equations were determined stationarily despite the weights for daily prediction, whereas, an exponential average estimator is applied in the online approach. Therefore, service level information of the mobile network with all attributes like the connected user and traffic load must be considered for proper scaling in the network. But, the accuracy of prediction can be enhanced by opting the hybrid deep learning models.

Abdellah and Koucheryavy (2020) proposed an LSTM-based deep learning model to predict the IoT device traffic. They used the optimisation technique for the learning process and predicted the best accuracy. The result

was evaluated in terms of root mean squared error (RMSE) and mean absolute percentage error (MAPE). However, performance was degraded when tested with residuals data.

Zhang et al. (2020) proposed a hybrid spatiotemporal network model that used CNN to predict the network traffic. In this model, they opted deformable convolution in the CNN model which improved the prediction results. During the simulation, their obtained result shows better than existing machine learning-based models in terms of MAE and RMSE. But, accuracy was less when test performed on residuals.

Nie et al. (2020) developed a reinforcement learning-based machine learning model to predict network traffic. They used a Monte Carlo Q-learning-based approach to find the best traffic predictions. Overall performance of this model was good but it did not perform well with residual data.

### *3.1 Advantage of proposed work over existing approaches*

List of advantage of proposed hybrid model over the existing techniques.

- High level data formatting resulting in a highly composed traffic data over regular intervals of time.
- Application of a hybrid technique involving both standard ML algorithm and popular deep learning sequence models.
- Use of time-series forecasting rather than multi-class classification used by most researchers.

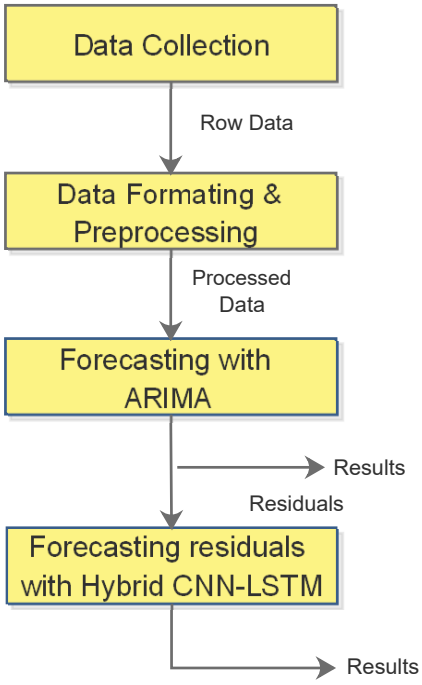
## 4 Proposed methodology

In this section, the proposed approach is conceded in order to forecast the cumulative traffic activity over a given time period. Our approach is an attempt to improvise over the already experimented methods for higher accuracy and lower error rates, so that, the allocation of resources would be as accurate as possible.

Illustration of Figure 4: the proposed methodology followed primarily consists of heavy data formatting followed by experimentation of different models and algorithms. Finally, the combination of the models are used which led to the lowest error estimations among all the experiments. In the experimentation, firstly, ARIMA a classical time forecasting model is applied, thereafter, hybrid deep learning models are applied over residuals of ARIMA which includes the combination of CNN and LSTM.

The hybrid approach uses ARIMA as a base model to forecast the cumulative network traffic over the time period, since, the formatted data had a repetitive pattern. It was observed during the experimentation, ARIMA performs best among the mentioned techniques. But, the ARIMA model is unable to predict some of the special cases when the network traffic was higher than usual. Therefore, a combination of CNN and LSTM on the residuals is applied to accurately predict the unusual outcomes and brought the error metrics to a minimum.

**Figure 4** Process flow of proposed approach (see online version for colours)




The major steps followed by the proposed approach for the network traffic forecasting are:

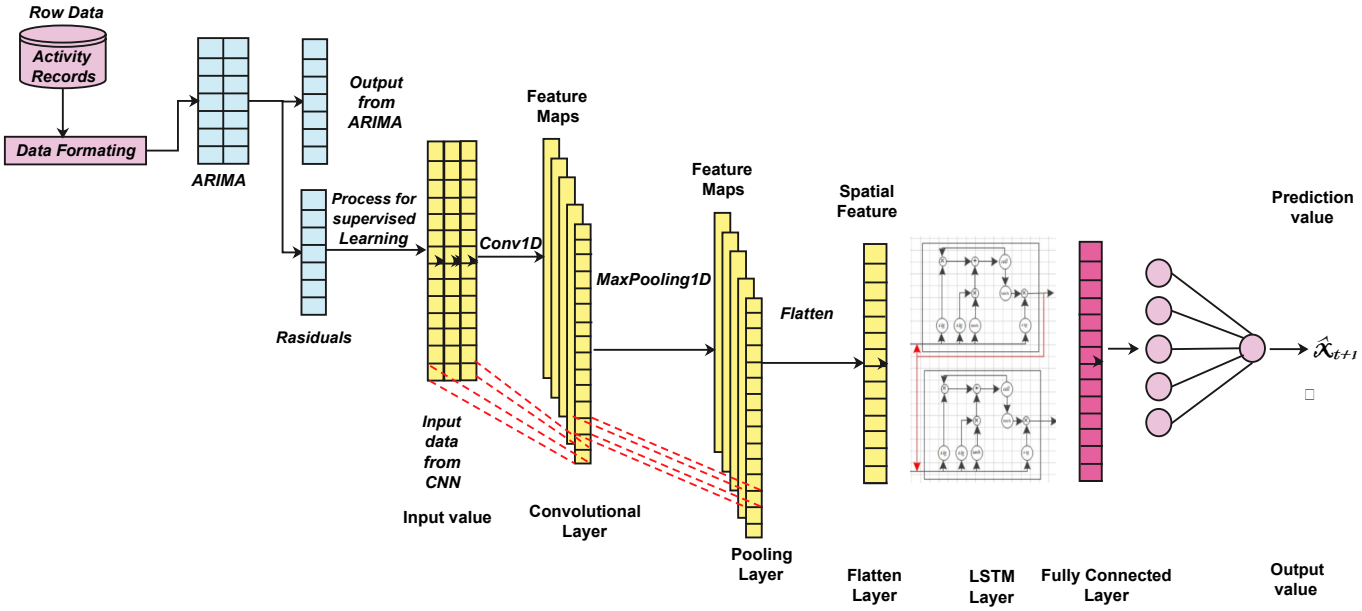
1 dataset collection and description
2 data formatting
3 fitting data to ARIMA and computing the residuals
4 designing the combined neural network
5 fitting the residuals to the neural architecture for prediction.

Each of the steps mentioned above are equally important in order to achieve a minimum error rate.

### *4.1 Dataset description*

Data is the most essential asset to a problem that requires a solution using deep learning techniques. In the proposed methodology, a dataset of call details records (CDRs) full of information regarding communication among various users to predict the cumulative traffic activity over a certain period of time is required. The cellular network operators (CNOs) are also able to collect these data easily using their respective infrastructure. In the proposed methodology, the telecommunications dataset is considered which is a collection of multi-source datasets of urban life in the City of Milan and the Province of Trentino. It shows the computation over the CDRs generated by Telecom Italia cellular network during the big data challenge in 2014.

**Figure 5** Complete flow of proposed model for traffic forecasting (see online version for colours)




The list of CDRs considered by Telecom Italia for the generation of the dataset is mentioned below.

1 *SMS-in:* Activity record generated for every time a user received an SMS.

2 *SMS-out:* Activity record generated for every time a user sent an SMS.

3 *Call-in:* Activity record generated for every time a user received a call.

4 *Call-out:* Activity record generated for every time a user issued a call.

5 *Internet:* Activity record generated for every time a user started or ended an internet connection.

All the aforementioned CDRs are based on the following three properties:

1 *Square ID:* The IDs of the square grids into which the City of Milan was divided into [1–10,000].

2 *Timestamp:* The beginning of the time interval elapsed from the Unix Epoch on 1st January 1970 at UTC and expressed in terms of milliseconds.

3 *Country code:* The phone country code of a nation. The meaning assumed by it differs according to context.

The final datasets are generated by combining all this anonymous information, with a temporal aggregation of time slots of ten minutes. The number of records in the datasets $S'i(t)$ follows the rule:

$$S'i(t) = Si(t)k \tag{2}$$

where $k$ is a constant defined by Telecom Italia, which hides the true number of calls, SMS, and connections.

The data was segregated into different text files, in which, each represents the records for a single day starting from 23:00 PM 10th October 2013 to 22:50 PM 1st January 2014 spanning for two months.

### 4.2 Data formatting

Extensive data formatting was done in order to meet the problem statement requirement. As stated above, the data for each day was segmented into different files, so there was a need to combine all the data into a single file for easier processing. The data also contained a large amount of missing data. In order to resolve these disputes and further pre-process the data, we have taken the following steps for each file in the directory.

1 *Interpolation:* To fill the huge amount of missing data, the linear interpolation technique has been taken in the proposed work.

2 *Aggregation:* To calculate the total cellular activity, aggregation is performed over all the given CDR columns and converted into a single column vector. This represented the total activity at a given time, for a given cellular ID and square grid ID.

3 *Conversion:* To make better use of the timestamps from the raw data, we converted it into date-time objects explicitly.

4 *Resampling:* Firstly the cell ID and the square grid ID columns are removed, thereafter, the date-time column is set as the index of the dataset. Then, we merged all the parameters and generates the total activity at a specific time without any other feature. Finally, re-sampled the total activity by computing the summation of activities for a specific date-time object and for every ten minutes elapsed.

All the aforementioned steps were performed repetitively for each text data file and combined into a single dataset object for easier and faster simulation. It reduces the size of the dataset. It was observed that the original dataset contains a total of 19.6 GB of data with more than 300 million samples and after the data formatting process, nearly 350 KB with around 9,000 data samples were obtained.

### 4.3 Proposed model for traffic forecasting

In this subsection, a detailed description of the proposed framework for accurately predicting the cumulative network traffic at a given time is provided.

During model formation, it was observed that the pattern of network activity throughout two months is very simple, thus, ARIMA was a very good option as a base model for forecasting the CDR initially. However, ARIMA is a linear model and so it will serve only as a generalised model. Now, in order to account for the nonlinearity present in the data, we propose a hybrid model that consists of CNN model and LSTM models which is executed on residuals of ARIMA, termed as ARIMA-CNN-LSTM.

A 1D CNN model is able to perform efficient convolution operations on 1D time-series of the residuals of the ARIMA results and filter out the spatial and temporal relationships. Then, an LSTM model is used to map the long time temporal dependencies of the filtered data from the CNN model. Figure 5 shows the detailed step-by-step workflow of the proposed methodology.

Initially, the ARIMA model is used to forecast the cumulative network traffic and the residuals obtained from it is passed on to the hybrid model as input. The hybrid model is built as a sequential model using the high-level Keras API with TensorFlow as the backend. The model starts with a 1D CNN or Conv1D layer along with the supportive 1D max pooling layer and a flatten layer. It also accompanies a repeated vector layer which repeats the output from the preceding model in order to provide the upcoming LSTM layer with the required input dimensions, to be exact, three-dimensions. Finally, a fully connected time distributed layer is added to apply the operation to each and every timestep of the 3D tensor we obtained from the previous steps.

### 4.4 Pseudo code of proposed framework

Algorithm 1 shows the details of proposed work. In Algorithm 1, the Telecom Italia network traffic dataset is taken as an input and obtains the output in terms of RMSE and SMAPE.

**Algorithm 1** Proposed method for network traffic forecasting at a specific time instance

**Result:** Traffic activity at a certain time instance along with the residuals

**Input:** Raw data collected from Harvard Data Store in the form of text files

**Output:** Predicted network traffic activity evaluated with RMSE and SMAPE and optimised residuals

**Procedure**

**Step 1:** Data formatting:

- 1.1 Collect the text file and convert the contents into a Pandas DataFrame
- 1.2 Linearly interpolate the dataset to fill up missing data
- 1.3 Aggregate the different CDRs to form a single cumulative network traffic
- 1.4 Convert the timestamps from data to Python date-time objects
- 1.5 Resample the dataset with the date-time as index column with 10 minutes interval
- 1.6 Repeat steps 1.1 to 1.5 for all the 60 text files in the directory

**Step 2:** Predictive modelling:

- 2.1 Apply ARIMA to the formatted data for each timestep and evaluate the predictions
- 2.2 Convert the residuals from the previous step into a supervised time-series data
- 2.3 Apply a hybrid of 1D-CNN and LSTM models as per Table 5 to the converted residuals
- 2.4 Evaluate the results by comparing the original residuals to the resultant residuals

## 5 Result and analysis

Consistent evaluation of time series forecasting models is very important. In this section, we define how we have evaluated our forecasting models.

### 5.1 Evaluation setup

In order to evaluate the performance of our model, we have used a walk-forward validation method. It is a method where each timestep in the dataset will be enumerated and a forecasting model will be constructed on the historical data prior to the timestep along with the forecast being compared to the actual data at that specific time instance. The observation will then be added to the list of historical data mentioned above and then this process is repeated for every data point present in the whole dataset.

The walk-forward validation method opted for validation because it is very similar to realistic scenarios, where the forecasting model will be updated with every new observation available from the continuous forecast updates.

In the performance analysis, the forecasts will be evaluated using two metrics commonly used for time series forecasting, i.e., RMSE and symmetric mean absolute percentage error (SMAPE)

#### 5.1.1 Root mean squared error

In statistics, RMSE is used to tell how close the regression/predicted line is to the original line. It can

be measured by estimating the average of the squared difference between the predicted or estimated values and the original values. Thereafter, the squared root of the resulting value has been taken. It is mentioned in equation (3).

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \quad (3)$$

RMSE is a popular metric used for regression tasks and is also used for evaluating time series forecastings. The advantage of using RMSE is that it penalises large errors and provides the errors in the same units as the forecast data.

**Table 3** Training hyperparameters – CNN

| | |
|---|---|
| Initial learning rate | 0.001 |
| Num. of epochs | 50 |
| Activation function | ReLU |
| Conv1D filters | 64 |
| Conv1D kernel size | 1 |
| MaxPool1D size | 1 |
| Feedforward hidden layers | 2 |
| Optimisation algorithm | Adam |
| Loss function | MSE |

**Table 4** Training hyperparameters – LSTM

| | |
|---|---|
| Initial learning rate | 0.001 |
| Num. of epochs | 100 |
| Activation function | ReLU |
| LSTM layers | 1 |
| LSTM units | 100 |
| Feedforward hidden layers | 1 |
| Optimisation algorithm | Adam |
| Loss function | MSE |

**Table 5** Training hyperparameters – CNN-LSTM hybrid

| | |
|---|---|
| Initial learning rate | 0.001 |
| Num. of epochs | 100 |
| Activation function | ReLU |
| Conv1D filters | 72 |
| Conv1D kernel size | 1 |
| MaxPool1D size | 1 |
| Repeated vector layer | 1 |
| LSTM layers | 1 |
| LSTM units | 200 |
| Time distributed dense layers | 2 |
| Optimisation algorithm | Adam |
| Loss function | MSE |

*Note:* The obtained value of RMSE is always positive. If the obtained value is near zero, it is derived from the square of Euclidean distance. It can never be zero in the case of real-world datasets since having an RMSE of 0 would mean a perfect fit for the data.

#### 5.1.2 Symmetric mean absolute percentage error

SMAPE is one of the most important forecast error metric, but it is not used commonly. Unlike RMSE, SMAPE is used majorly for Forecasting and not for general regression tasks. It is mentioned in equation (4).

$$SMAPE = \frac{100}{n}\Sigma_{i=1}^{n}\frac{2|\hat{Y}_i - Y_i|}{(|Y_i| + |\hat{Y}_i|)} \quad (4)$$

SMAPE is calculated by twice the absolute difference between the actual and the forecasted value, further divided by the sum of the same values. This value is summed over all the data points fitted to the forecasting model and the final result gives us the total SMAPE.

#### 5.1.3 Training hyperparameters

A number of deep learning networks have been used in modelling different architectures which required their own set of hyperparameters. We used fairly simple architectures owing to the results of the ARIMA forecasting, thus preventing complex training paradigms.

Illustration of Tables 3, 4 and 5: all these tables contain the detailed information regarding the various deep neural architectures taken into the consideration for the proposed methodology.

### 5.2 Performance analysis

#### 5.2.1 Comparative analysis of various deep learning models in terms of RMSE and SMAPE

The obtained results of various deep learning models are presented in Table 6. It contains the numerical values obtained after performing the devised experiments on the dataset. It can be evident that ARIMA works best on the data followed by CNN-LSTM hybrid which led us to choose a hybrid architecture consisting of training different models.

**Table 6** Numerical evaluation results

| *Metrics* | *Model architecture* | | | | |
|---|---|---|---|---|---|
| | *ARIMA* | *CNN* | *LSTM* | *CNN-LSTM* | *ARIMA-CNN-LSTM* |
| RMSE | 0.021124 | 0.021398 | 0.021091 | 0.020778 | 0.024384 |
| SMAPE | 4.177608 | 6.611342 | 6.064976 | 5.463721 | 1.705518 |

From the obtained results, it was observed that the proposed model performance was nearly equivalent to other tested models in terms of RMSE, but, it perform far better than the other tested deep learning models in terms of SMAPE. Hence, it proves that the proposed model outperforms the tested deep learning architectures and is able to provide a very accurate result over time.

#### 5.2.2 Graphical results

In this subsection, obtained results of various experiment are presented in graphical form.

**Figure 6** Residuals plot from ARIMA model (see online version for colours)

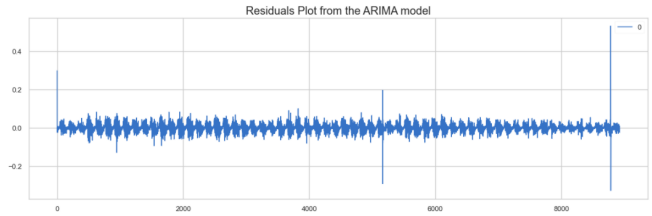



**Figure 7** CNN training – loss curve (see online version for colours)

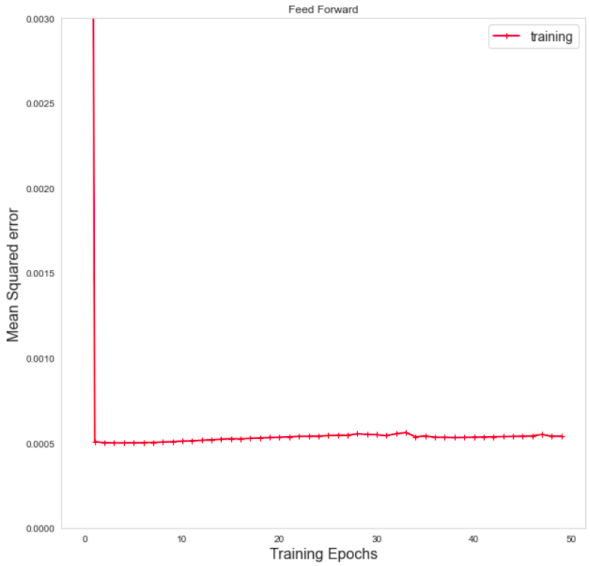



**Figure 8** LSTM training – loss curve (see online version for colours)

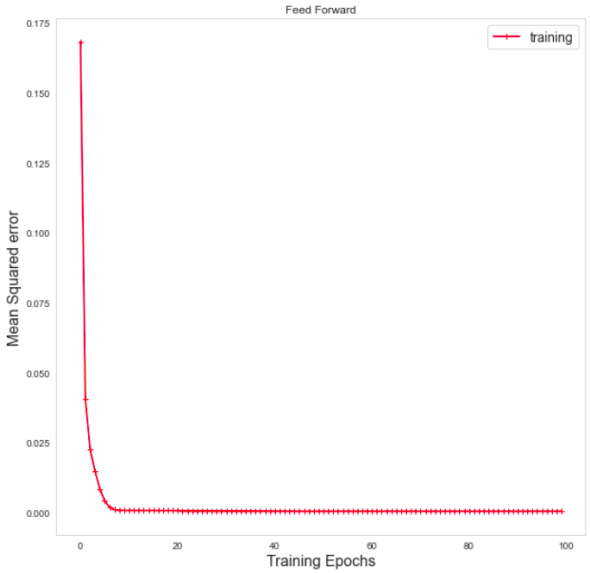



Illustration of Figure 12: it shows the distribution of the initially formatted data over the total period of two months. As we can see from the plot, the data has a repeated pattern with very few irregularities which eventually led ARIMA to show such good results. The deep learning models in turn produced lesser accurate results than ARIMA is mostly because they tend to generalise the output model as far as possible, owing to the factor that the architecture of each model is relatively simple.

**Figure 9** CNN-LSTM training – loss curve (see online version for colours)

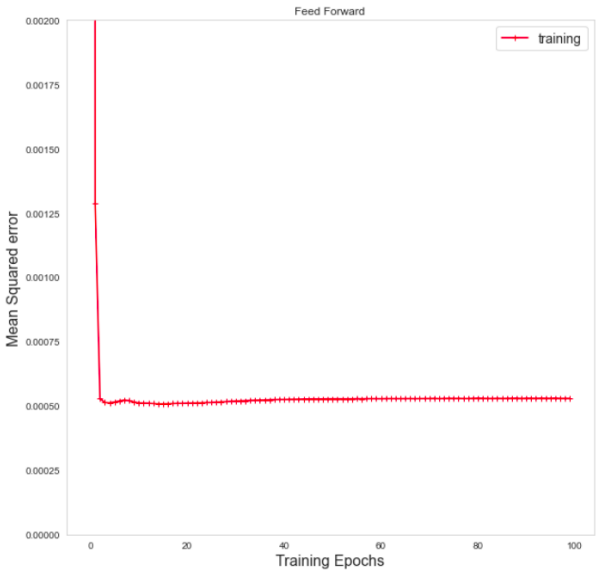



**Figure 10** Hybrid CNN-LSTM training – loss curve (see online version for colours)

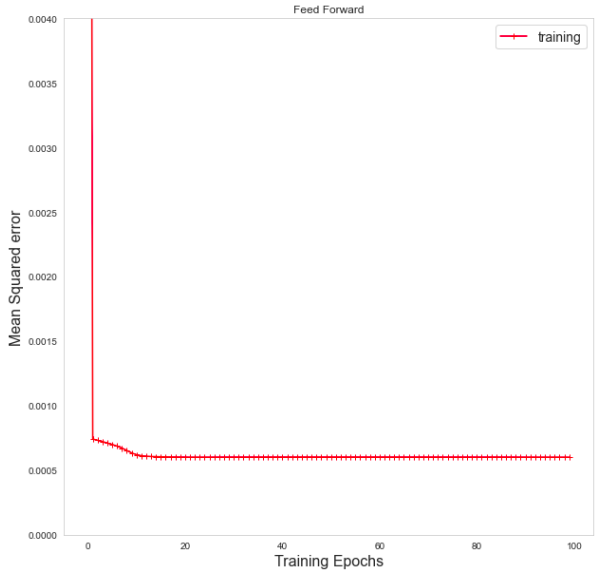



**Figure 11** Hybrid residual results (see online version for colours)

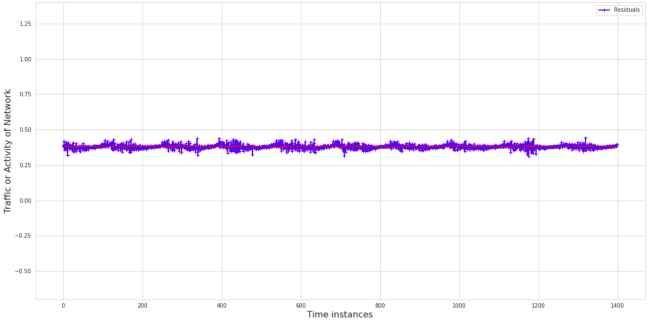


**Figure 12** Formatted data distribution over two months (see online version for colours)


**Figure 13** ARIMA – predicted vs. actual for two months (see online version for colours)

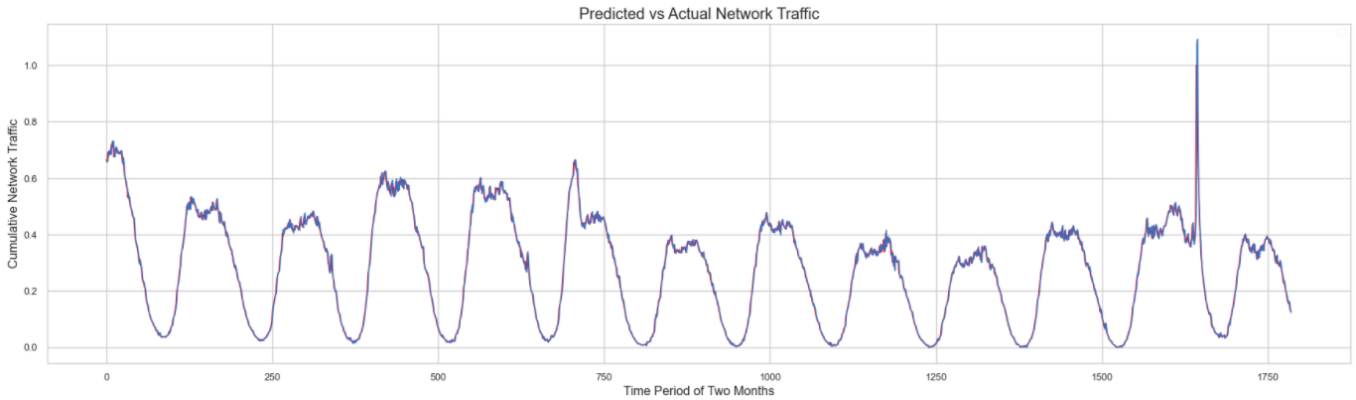



**Figure 14** CNN-1D forecasting (see online version for colours)

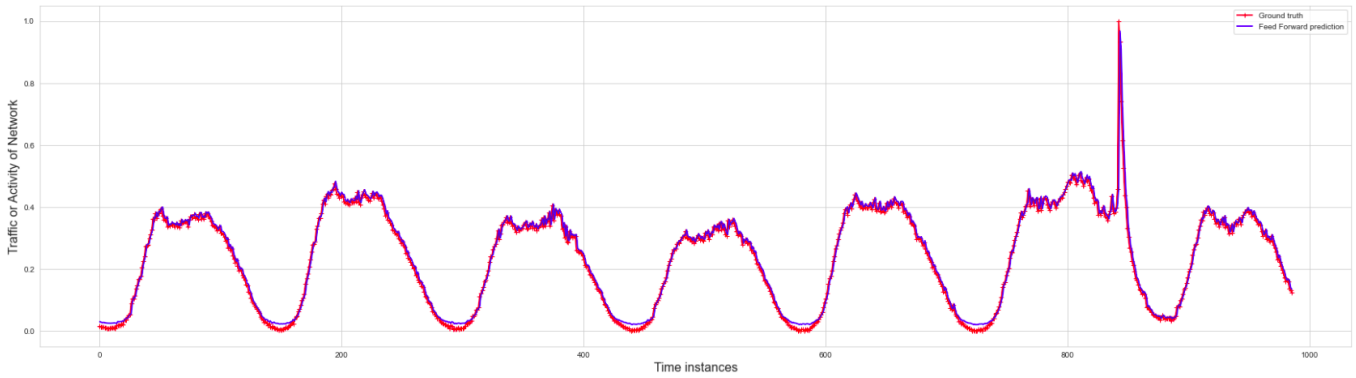



**Figure 15** LSTM forecasting (see online version for colours)

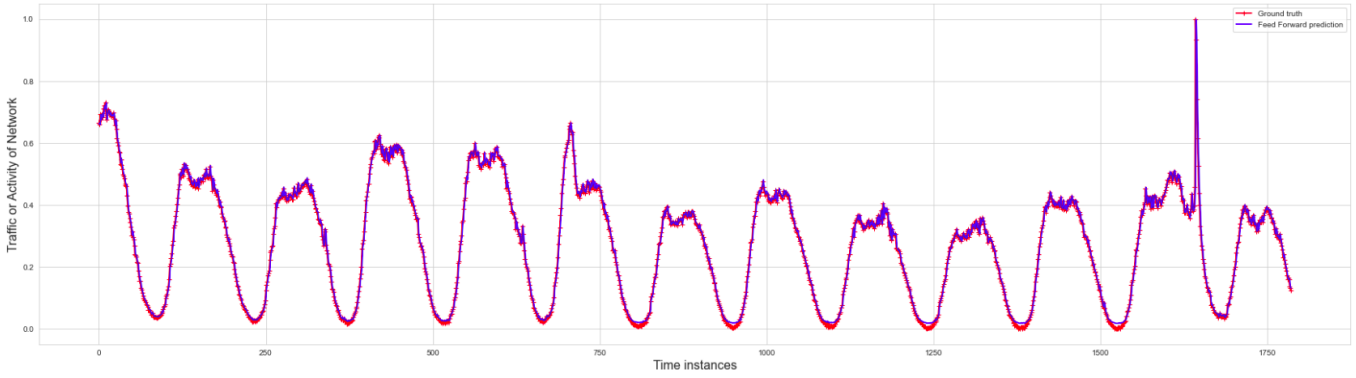


**Figure 16** CNN-LSTM forecasting (see online version for colours)

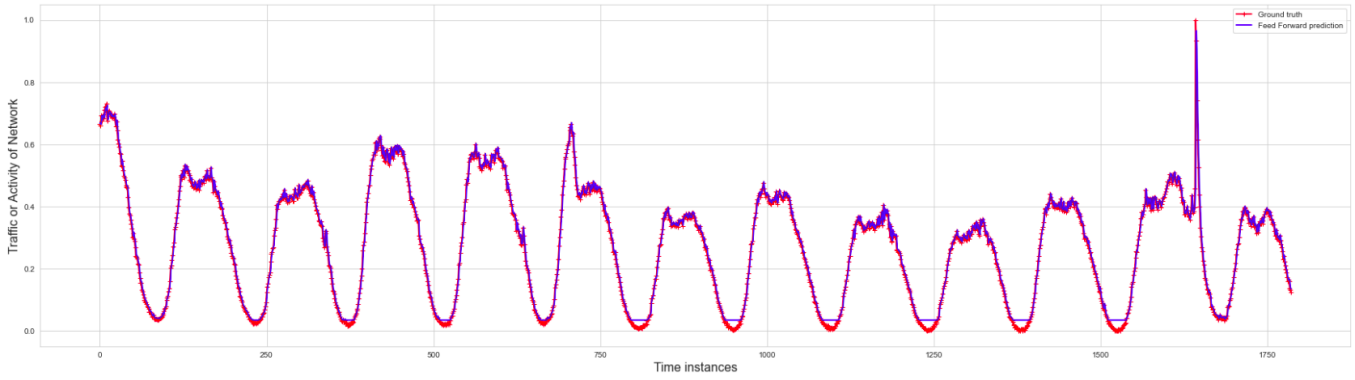



Illustration of Figure 13: it shows the final result obtained from ARIMA in comparison to the actual data. It is evident that ARIMA works best when looking at the graph which shows very little error in prediction when compared to the actual data distribution.

Illustration of Figure 6: the plot for residuals from the ARIMA forecasting is also shown in Figure 6. We observed very few but unexpected deviations from the original data through the residuals plot which was evident from our previous results.

Illustration of Figures 7, 8, 9 and 10: represents the training loss curve of the CNN, LSTM, CNN-LSTM and the hybrid proposed models with their architectures as mentioned in Tables 3, 4 and 5. The loss curves are very steep and not gradual mostly because of the nature of the data to be so simple. Due to this, even with a relatively simpler architecture, the deep learning models were easily able to converge to a very low value.

Illustration of Figures 14, 15 and 16: shows the predictions of each of the models mentioned above, respectively. If we look into each of them minutely, we can see that the predictions for the downhills were getting more sloppy with change in the deep learning model. We surmise that this is the result of the generalisation of the deep learning models.

Figure 11 shows the results of the hybrid model where the residuals were reduced to only a very small amount of error rates. The initial forecasting result of the hybrid model will be the same as Figure 13. Along with that, the hybrid architecture was able to properly predict the unexpected situations which led to the very slight error in ARIMA.

## 6 Conclusions

5G is launched to provide high data to the users. But, the network still faces high traffic issues because it is still new to the world. In order to enhance the network services, researchers have worked upon and have provided various solutions to it using the various networking techniques as well as leveraging machine learning. These various solutions have been updated over time due to advancement in technology and ML as a field. Most of the researchers are still working upon various direct and related problems to further increase the efficiency of the solution. In order to further increase the efficiency and results of the existing solutions, a new hybrid technique utilising machine learning model was proposed which is a combination of ARIMA forecasting technique, and deep learning architectures consist of CNN and LSTM. In this research article, the cumulative network traffic is forecasted over specific intervals of time to scale up and down the network effectively. In addition, correctly predicted the availability of 5G network resources by utilising the traffic load changes over a period of two months. In the comparative analysis, the proposed model was evaluated with an existing forecasting models known as ARIMA, and two of the most popular deep sequence models, namely, 1D-CNN and an LSTM. Moreover, the obtained output was compared to the results of the existing techniques and it was observed that our proposed model outperformed over the other existing well known techniques.

## References

Abdellah, A.R. and Koucheryavy, A. (2020) 'Deep learning with long short-term memory for iot traffic prediction', in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, pp.267–280, Springer, Cham.

Agiwal, M., Roy, A. and Saxena, N. (2016) 'Next generation 5G wireless networks: a comprehensive survey', *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 3, pp.1617–1655.

Al-Namari, M.A., Mansoor, A.M. and Idris, M.Y.I. (2017) 'A brief survey on 5G wireless mobile network', *Int. J. Adv. Comput. Sci. Appl.*, Vol. 8, No. 11, pp.52–59.

Alvi, A.B., Masood, T. and Mehboob, U. (2017) 'Load based automatic scaling in virtual IP based multimedia subsystem', in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, pp.665–670.

Arteaga, C.H.T., Risso, F. and Rendon, O.M.C. (2017) 'An adaptive scaling mechanism for managing performance variations in network functions virtualization: a case study in an NFV-based EPC', in *2017 13th International Conference on Network and Service Management (CNSM)*, IEEE, pp.1–7.

Bhalla, M.R. and Bhalla, A.V. (2010) 'Generations of mobile wireless technology: a survey', *International Journal of Computer Applications*, Vol. 5, No. 4, pp.26–32.

Bilal, A., Tarik, T., Vajda, A. and Miloud, B. (2016) 'Dynamic cloud resource scheduling in virtualized 5G mobile systems', in *2016 IEEE Global Communications Conference (GLOBECOM)*, IEEE, pp.1–6.

Buzzi, S., Chih-Lin, I., Klein, T.E., Poor, H.V., Yang, C. and Zappone, A. (2016) 'A survey of energy-efficient techniques for 5G networks and challenges ahead', *IEEE Journal on Selected Areas in Communications*, Vol. 34, No. 4, pp.697–709.

Carella, G.A., Pauls, M., Grebe, L. and Magedanz, T. (2016) 'An extensible autoscaling engine (AE) for software-based network functions', in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, IEEE, pp.219–225.

Dangi, R., Lalwani, P., Choudhary, G., You, I. and Pau, G. (2021) 'Study and investigation on 5G technology: a systematic review', *Sensors*, Vol. 22, No. 1, p.26.

Dutta, S., Taleb, T. and Ksentini, A. (2016) 'QoE-aware elasticity support in cloud-native 5G systems', in *2016 IEEE International Conference on Communications (ICC)*, IEEE, pp.1–6.

Mehta, H., Patel, D., Joshi, B. and Modi, H. (2014) '0G to 5G mobile technology: a survey', *J. of Basic and Applied Engineering Research*, Vol. 1, No. 6, pp.56–60.

Nie, L., Ning, Z., Obaidat, M.S., Sadoun, B., Wang, H., Li, S., Guo, L. and Wang, G. (2020) 'A reinforcement learning-based network traffic prediction mechanism in intelligent internet of things', *IEEE Transactions on Industrial Informatics*, Vol. 17, No. 3, pp.2169–2180.

Open Source Cloud Computing [online] http://cloudstack.apache.org (accesssed 30 August 2021).

West, C. (2018) *Web Services Auto Scaling* [online] https://aws.amazon.com/autoscaling (accesssed 30 August 2021).

Zhang, D., Liu, L., Xie, C., Yang, B. and Liu, Q. (2020) 'Citywide cellular traffic prediction based on a hybrid spatiotemporal network', *Algorithms*, Vol. 13, No. 1, p.20.