


## Article

# Long-Term Data Traffic Forecasting for Network Dimensioning in LTE with Short Time Series

Carolina Gijón , Matías Toril , Salvador Luna-Ramírez , María Luisa Marí-Altozano   
and José María Ruiz-Avilés 

Instituto de Telecomunicaciones (TELMA), Universidad de Málaga, CEI Andalucía TECH, 29071 Málaga, Spain; mtoril@ic.uma.es (M.T.); sluna@ic.uma.es (S.L.-R.); mlma@ic.uma.es (M.L.M.-A.); jmruiz@ic.uma.es (J.M.R.-A.)

\* Correspondence: cgm@ic.uma.es

**Abstract:** Network dimensioning is a critical task in current mobile networks, as any failure in this process leads to degraded user experience or unnecessary upgrades of network resources. For this purpose, radio planning tools often predict monthly busy-hour data traffic to detect capacity bottlenecks in advance. Supervised Learning (SL) arises as a promising solution to improve predictions obtained with legacy approaches. Previous works have shown that deep learning outperforms classical time series analysis when predicting data traffic in cellular networks in the short term (seconds/minutes) and medium term (hours/days) from long historical data series. However, long-term forecasting (several months horizon) performed in radio planning tools relies on short and noisy time series, thus requiring a separate analysis. In this work, we present the first study comparing SL and time series analysis approaches to predict monthly busy-hour data traffic on a cell basis in a live LTE network. To this end, an extensive dataset is collected, comprising data traffic per cell for a whole country during 30 months. The considered methods include Random Forest, different Neural Networks, Support Vector Regression, Seasonal Auto Regressive Integrated Moving Average and Additive Holt–Winters. Results show that SL models outperform time series approaches, while reducing data storage capacity requirements. More importantly, unlike in short-term and medium-term traffic forecasting, non-deep SL approaches are competitive with deep learning while being more computationally efficient.

**Keywords:** mobile network; traffic forecasting; network dimensioning; time series; supervised learning



check for updates

**Citation:** Gijón, C.; Toril, M.; Luna-Ramírez, S.; Marí-Altozano, M.L.; Ruiz-Avilés, J.M. Long-Term Data Traffic Forecasting for Network Dimensioning in LTE with Short Time Series. *Electronics* **2021**, *10*, 1151.

<https://doi.org/10.3390/electronics10101151>

Academic Editor: Christos J. Bouras

Received: 25 March 2021

Accepted: 3 May 2021

Published: 12 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In future 5G networks, it is expected that the rapid traffic growth and the coexistence of services with very different requirements will lead to constantly changing traffic patterns and network capacity requirements [1,2]. As a result of such dynamism, network (re-)dimensioning will become a critical task in these networks. As a result, operators will have to revise and constantly update their capacity plans to predict capacity bottlenecks in advance, avoiding troubles before user experience is degraded. To make it easier, smart capacity planning tools are developed in a Self-Organizing Networks (SON) framework [3,4]. Such tools, often implemented in the Operation Support System (OSS), require accurate prediction of upcoming traffic demand and network capabilities to anticipate traffic variations, so that network configuration can be timely upgraded to guarantee an adequate level of user satisfaction in this changing environment.

In the radio interface, capacity bottlenecks are detected by comparing traffic forecasts with some pre-defined threshold reflecting cell capacity, so that an alarm is activated to notify the future lack of resources. Then, different re-planning actions can be executed depending on how much in advance the problem is detected. Imminent problems detected with short-term forecasts often trigger temporary changes of network parameters (e.g., a more efficient voice coding scheme [5], new handover margin settings for traffic sharing between adjacent cells [6] or naïve packet schedulers for a lower computational load [7]).

Such quick actions, dealing with fast fluctuations of traffic demand, tend to be provisional, so that parameters return to their original values when normal network state is recovered, or act as temporary solution if the problem persists in the hope of more stable solutions relying on network capacity extensions. In contrast, long-term forecasts predict the lack of resources ahead in time (e.g., several months), so that more future proof solutions can be implemented (e.g., bandwidth extension [8], license extension for the maximum number of channel elements and/or simultaneous users [9] or new carriers/co-sited cells).

In the literature, the earliest works address circuit-switched traffic prediction by deriving statistical models based on historical data (e.g., Auto Regressive Integrated Moving Average—ARIMA) [10,11]. More modern approaches tackle packet-data traffic forecasting with sophisticated models based on supervised machine learning to take advantage of massive data collected in cellular networks [12,13]. Several models have been proposed to predict traffic in the short term (seconds, minutes) based on deep learning [14,15]. With these models, advanced dynamic radio resource management schemes and proactive self-tuning algorithms can be implemented [16]. However, some re-planning actions (e.g., deployment of a new cell) may take several months to be implemented (e.g., radio frequency planning, site acquisition, civil works, licenses, installation/commissioning, pre-launch optimization. . .) [17]. Thus, the upcoming traffic must be predicted with much longer time horizons (i.e., several months) [18]. For this purpose, a monthly traffic indicator is often computed per cell from busy-hour measurements, limiting the number of historical data samples used for prediction [19,20]. Moreover, some studies [21,22] have shown that the influence of past measurements quickly diminishes after a few weeks, due to changes in user trends (e.g., new terminals, new hot spots. . .) and re-planning actions by the operator (e.g., new site, equipment upgrades. . .). As a consequence, long-term traffic forecasting relies on short and noisy time series, different from those used in short-term traffic forecasting, based on minute/hourly data. It is still to be checked if SL methods outperform classical time series analysis with these constraints.

In this work, a comprehensive analysis is carried out to compare the performance of SL against time series analysis schemes for predicting monthly busy-hour data traffic per cell in the long term. For this purpose, a large dataset is collected during 30 months from a live Long Term Evolution (LTE) network covering an entire country. All prediction techniques considered here are included in most data analytics packages and have already been used in several fields. Hence, the main novelty is the assessment of well-established SL methods for long-term data traffic forecasting based on short and noisy time series taken from current mobile networks offering a heterogeneous service mix. Specifically, the main contributions are:

- The first comprehensive comparison of the performance of SL algorithms against classical time series analysis in long-term data traffic forecasting on a cell basis, relying on short and noisy time series. Algorithms are compared in terms of accuracy and computational complexity.
- A detailed analysis of the impact of key design parameters, namely the observation window, the prediction horizon and the number of models to be created (one per cell or one for the whole network).

The rest of the document is organized as follows. Section 2 presents related work. Section 3 formulates the problem of predicting monthly busy-hour data traffic per cell, highlighting the properties of the time series involved. Section 4 describes the considered dataset. Section 5 outlines the compared forecasting methods. Section 6 presents the performance assessment. Finally, Section 7 summarizes the main conclusions.

## 2. Related Work

Traffic forecasting in telecommunication networks can be treated as a time series analysis problem. In the earliest works, circuit-switched traffic prediction is addressed by deriving statistical models based on historical data. Linear time series models, such as Auto Regressive Integrated Moving Average (ARIMA), capture trend and short-range dependen-

cies in traffic demand. More complex models, such as Seasonal ARIMA (SARIMA) [10,23] and exponential smoothing (e.g., Holt–Winters) [24,25], include seasonality. These can be extended with non-linear models, such as Generalized Auto Regressive Conditionally Heteroskedastic (GARCH) [11], to reflect long-range dependencies.

The previous works show that it is possible to predict cellular traffic at different geographical scales (e.g., network operator [10], province [23], cell [24]) and time resolutions (e.g., minutes [11], hourly [24], daily [10], monthly [23]), provided that traffic is originated by circuit-switched services (e.g., voice and text messages). However, predicting packet data traffic is much more challenging [26]. As pointed out in [25], data traffic is more influenced by abnormal events and changes in network configuration than circuit-switched traffic. In [27], short-term traffic volume in a 3G network is predicted via Kalman filtering. In [28], ARIMA is used to predict the achievable user rate in four cells located in areas with different land use. In [29], application-level traffic is predicted by deriving an  $\alpha$ -stable model for 3 different service types and then dictionary learning is implemented to refine forecasts. As an alternative, more modern approaches tackle data traffic forecasting with sophisticated models based on Supervised Learning (SL). Most efforts have been focused on short-term prediction (seconds, minutes) to solve the limitations of time series analysis approaches to capture rapid fluctuations of the time series. A common approach is to use deep learning to model the spatio-temporal dependence of traffic demand. The temporal aspect of traffic variations is often captured with recurrent neural networks based on Long Short-Term Memory (LSTM) units [14,30–32]. Alternatively, in [33], a deep belief network and a Gaussian model are used to capture temporal dependencies of network traffic in a mesh wireless network. The spatial dependence is captured by different approaches. In [14], the scenario is divided into a regular grid, and a convolutional neural network is used to model traffic spatial dependencies among grid points. A similar approach is considered in [34], where extra branches are added to the network for fusing external factors such as crowd mobility patterns or temporal functional regions. In [35], convolutional LSTM units and 3D convolutional layers are fused to encode the spatio-temporal dependencies of traffic carried in the grid points. Alternatively, other authors model spatial dependencies of traffic carried in different cells. In [32], a general feature extractor is used with a correlation selection mechanism for modeling spatial dependencies among cells and an embedding mechanism to encode external information. In [15], to deal with an irregular cell distribution, the spatial relevancy among cells is modeled with a graph neural network based on distance among cell towers. A graph-based approach is also considered in [36], where traffic is decomposed in inter-tower and in-tower. Deep learning schemes such as LSTM [37], convolutional neural networks [38] or recurrent neural networks [39] have also been applied to coarser time resolutions (i.e., an hour) to extend the forecasting horizon to several days.

The above works show that advanced deep learning models perform well if data are collected with fine-grained time resolution to build long time series (i.e., thousands of samples) of correlated data. However, long-term traffic prediction performed in radio planning tools relies on short and noisy time series, which might prevent operators from using complex deep learning models. As an alternative, it is appropriate to check if simpler SL methods outperform classical time series analysis for time series with these constraints. For this purpose, a large dataset containing data collected on a per-cell basis during years is required. Such information is an extremely valuable asset for operators, which is seldom shared. For this reason, to the authors' knowledge, no recent work has evaluated long-term traffic forecasting in mobile networks considering SL techniques.

### 3. Problem Formulation

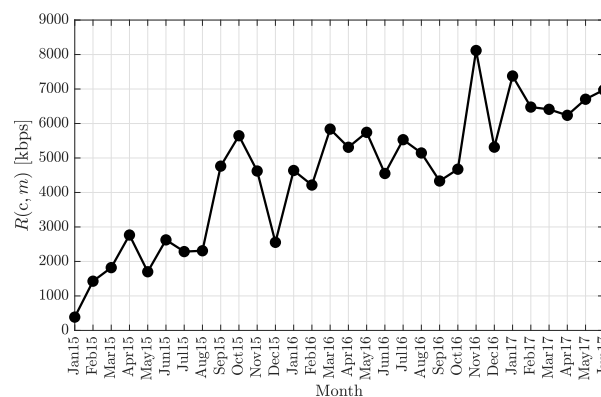
The problem of forecasting traffic  $R$  carried in a cell  $c$  with a time horizon  $t_0$  at time  $t$  from historical data collected during an observation window  $k$  can be formulated as

$$\hat{R}(c, t + t_0) = f(R(c, t), R(c, t - 1), \dots, R(c, t - k + 1)), \quad (1)$$

where  $t - k + 1$  denotes the oldest available data sample.

The way to tackle traffic prediction strongly depends on the time granularity of data, which determines two key factors. A first factor is the length of available time series,  $k$ . Note that, when training a prediction model, the number of observations must be higher than the number of model parameters [40]. Thus, complex deep learning models with hundreds of internal parameters cannot be considered if a short time series (i.e., dozens of samples) is available due to data aggregation on a coarse time resolution (e.g., monthly data).

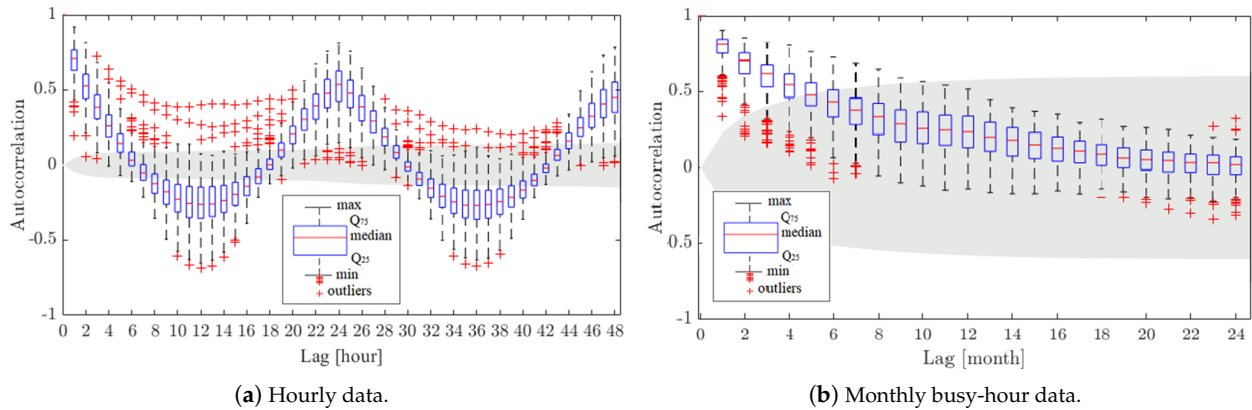
A second factor is data predictability, which may be degraded by the aggregation operation needed to compute monthly busy-hour traffic. To illustrate this fact, Figure 1 shows the evolution of monthly busy-hour traffic in the Down Link (DL) from a live LTE cell. As described later in Section 4, data aggregation is performed by selecting the busy-hour per week and then averaging traffic measurements per week in a month. Weekly and monthly aggregation eliminates hourly/daily fluctuations and the impact of sporadic events (e.g., cell outage, cell barring. . .), leaving only monthly variations needed for detecting capacity issues. The latter consist of: (a) a trend component, influenced by the traffic growth rate, (b) a seasonal component, given by the month of the year and (c) a remainder component, due to abnormal events taking place locally (e.g., new nearby site, new hot spot. . .) or network wide (e.g., launch of new terminals, change of network release. . .). In the example of the figure, it is observed that the trend component prevails over the seasonal component, causing that the time series is not stationary. Additionally important, network events cause sudden trend changes, which decrease the value of past knowledge and, ultimately, degrade the performance of prediction algorithms.



**Figure 1.** Evolution of monthly busy-hour traffic in a cell.

For a deeper analysis of how time resolution affects predictability, Figure 2a,b show a box-and-whisker diagram of the autocorrelation function of the DL traffic pattern in 310 cells from a live LTE network on an hourly and monthly basis, respectively. To ease comparison, two seasonal periods are represented in both cases (2 days for hourly data and 2 years for monthly data). The shadowed area depicts the 95% confidence interval for each lag, suggesting that values outside this area are very likely a true correlation and not a statistical fluke. The smaller interval width for the hourly series are due to a larger number of samples stored in those series. In Figure 2a, the interquartile boxes show a cyclical pattern with maxima (strong positive autocorrelation) at lags 24 and 48 and minima (strong negative correlation) at lags 12 and 36. Such a behavior reveals that hourly traffic is seasonal. Moreover, in most lags, the whole box is out of the shadowed area, suggesting that past information is relevant for the current traffic value. In contrast, Figure 2b shows that the autocorrelation of monthly traffic does not suggest a seasonal pattern, due to the absence of local maxima or minima. On the contrary, it quickly diminishes to 0, so that, in most cells, only information from lags 1 to 4 is significantly correlated with the current traffic value. This fact suggests that, even for time series with the same available observation

window, monthly busy-hour data are less predictable than hourly data. It is especially relevant that, unexpectedly, correlation with lags 12 and 24 (i.e., with data collected in the same month of previous years) is not significant, reducing data predictability. A closer analysis (not presented here) shows that seasonality is neither observed in the de-trended monthly traffic series.



**Figure 2.** Autocorrelation of cell DL traffic.

The above analysis confirms that long-term traffic forecasting for network dimensioning, based on monthly busy-hour traffic data, requires a separate analysis from short-term traffic forecasting based on higher time resolution data.

#### 4. Dataset

The analysis carried out in this work requires a large dataset including the largest number of cells and network states. The considered dataset contains data collected from January 2015 to June 2017 (i.e., 30 months) in a large live LTE network serving an entire country. The network comprises 7160 cells (also known as evolved Nodes B) covering a geographical area of approximately 500,000 km<sup>2</sup>, including cells of different sizes and environments, with millions of subscribers. Analysis is restricted to the DL, as it has the largest utilization in current cellular networks.

In the network, traffic measurements are gathered on a per-cell and hourly basis. As explained in Section 1, such raw data are pre-processed to obtain a single traffic measurement per cell/month to be stored in the long-term for network dimensioning tasks. The resulting dataset consists of 7160 time series (1 per cell) with 30 measurements (1 per month) of the monthly DL traffic volume in the busy-hour, expressed as a rate (i.e., in kbps). The monthly DL traffic volume carried in a cell  $c$  and month  $m$  during the busy-hour,  $R(c, m)$ , is calculated as follows:

1. The average DL traffic volume (in kbps) and the average number of active users (i.e., those users demanding resources) are measured and collected per cell  $c$  and hour.
2. The weekly busy-hour is selected per cell as the hour with the highest number of active users in week  $w$ . Each week belongs to a month  $m$ . The DL traffic volume (in kbps) during that busy-hour is selected as the weekly DL traffic volume per cell, week and month,  $R(c, w, m)$  (in kbps).
3. Finally, the monthly busy-hour DL traffic volume per cell and month,  $R(c, m)$ , is computed as the average of  $R(c, w, m)$  across weeks in month  $m$ , as

$$R(c, m) = \frac{1}{N_w(m)} \sum_{w \in m} R(c, w, m), \quad (2)$$

where  $N_w(m)$  is the number of weeks in month  $m$ . For simplicity, a week riding two months is considered to belong to only one month (the month including more days of that week).



The considered dataset combines a large temporal and spatial scale (30 months, entire country) with a fine-grained space resolution (cell), which guarantee the reliability and significance of results. Moreover, it allows to test prediction methods with different observation windows and time horizons, broadening the scope of the analysis.

## 5. Traffic Forecasting Methods

Six prediction methods are tested. A first group consists of classical statistical time series analysis schemes, namely Seasonal Auto Regressive Integrated Moving Average (SARIMA) and Additive Holt–Winters (AHW). A second group consists of state-of-the-art SL algorithms, namely Random Forest (RF), Artificial Neural Networks (ANN) and Support Vector Regression (SVR). The basics of these techniques are outlined next:

- SARIMA computes the current value of a time series difference as the combination of previous difference values and the present and previous values of the series. As detailed in [41], a SARIMA process is described as  $SARIMA(p,d,q)(P,D,Q)_m$ .  $(p,d,q)$  describe the non-seasonal part of the model, where  $p$  is the auto-regressive order,  $d$  is the level of difference and  $q$  is the moving average order, with  $p$ ,  $d$  and  $q$  non-negative integers.  $(P,D,Q)_m$  describe the seasonal part of the model, where  $P$ ,  $D$  and  $Q$  are similar to  $p$ ,  $d$  and  $q$ , but with backshifts of the seasonal period  $m$  (e.g., for monthly data,  $m = 12$ ).
- AHW calculates the future value of a time series with recursive equations by aggregating its typical level (average), trend (slope) and seasonality (cyclical pattern) [42]. These three components are expressed as three types of exponential smoothing, with smoothing parameters  $\alpha$ ,  $\beta$  and  $\gamma$ , respectively. As in SARIMA, the seasonal period is denoted as  $m$  (for monthly data,  $m = 12$ ). In this work, Additive Holt–Winters (AHW) is chosen, since, as shown in Figure 1, the seasonal effect is nearly constant through the time series (i.e., it does not increase with the level).
- RF is an ensemble learning method where several decision trees are created with different subsets of the training data (also known as aggregating or bagging). To reduce the correlation among trees, a different random subset of input attributes is selected at each candidate split in the learning process (feature bagging) [43]. To avoid model overfitting, trees are pruned. Then, predictions obtained from different trees are averaged to perform a robust regression [44].
- ANN is a statistical learning method inspired by the structure of a human brain. In ANN, entities called nodes act as neurons, performing non-linear computations by means of activation functions [45]. Two ANNs are considered in this work. The first one, denoted as ANN–MLP, is a feed-forward network (i.e., without memory) consisting on a Multi-Layer Perceptron (MLP) [46]. The second one, denoted as ANN–LSTM, is a deep recurrent network (i.e., with memory) based on LSTM units capable of capturing long-term dependencies thanks to the use of information control gates [47]. The architectures of such networks (number of layers, number of neurons per layer, activation functions, etc.) are detailed in Section 6.1.
- SVR maps a set of inputs into a higher dimensional feature space to find the regression hyperplane that best fits every sample in the dataset. For this purpose, a linear or non-linear (also known as *kernel*) mapping function can be used. Unlike traditional multiple linear regression, SVR neglects all deviations below an error sensitivity parameter,  $\epsilon$ . Moreover, the regularization parameter,  $C$ , restricts the absolute value of regression coefficients. Both parameters control the trade-off between regression accuracy and model complexity (i.e., the smaller  $\epsilon$  and larger  $C$ , the better the model fits the training data, but overfitting is more likely) [44].

Several issues must be considered when using these methods for long-term data traffic forecasting in cellular networks:

1. *Observation window*: time series corresponding to recently deployed cells may not have many historical measurements. Thus, it is important to check the capability of the methods to work with small observation windows. Such a feature is especially

critical during the network deployment stage, when network structure is constantly evolving (e.g., new cells are activated every month). At this early stage, robust traffic forecasting is crucial to avoid under-/over-estimating traffic in the new cells.

2. *Number of models*: recursive models such as SARIMA, AHW and ANN-LSTM are conceived to build a different model per cell based on historical data of that particular cell. Thus, the short period available in data warehouse systems for long-term forecasting (typically, less than 24 months) may jeopardize prediction capability in these methods, since it is always necessary to have more observations than model parameters [40]. In contrast, in RF, ANN-MLP and SVR, a single model can be derived for the whole network from historical data of all the cells. The latter ensures that enough training data are available to adjust model parameters, avoiding model overfitting. Likewise, sharing past knowledge across cells in the system increases the robustness of predictions in cells with limited data or abnormal events.
3. *Time horizon*: The earlier a capacity bottleneck can be predicted, the more likely the problem will be fixed without any service degradation, since some network re-planning actions (e.g., build a new site) may take several months. Such a delay forces operators to foresee traffic demand several months in advance (referred to as multi-step prediction). In classical time series analysis methods, such as SARIMA and AHW, multi-step prediction is carried out recursively by using a one-step model multiple times (i.e., the prediction for the previous month is used as an input for predicting the following month). Such a recursive approach reduces the number of models needed, but quickly increases prediction errors originated by the use of predictions instead of observations as inputs [48]. This is a critical issue when using recursive methods for series with large random components, as those used in long-term forecasting. In contrast, SL algorithms have the ability to directly train a separate model for each future step. Such an approach does not entail an increase of computational load if the set of steps predicted is small (e.g., 3 and 6 months ahead).
4. *Interpretability*: Ideally, prediction models should be simple enough to have an intuitive explanation of their output values [49]. Models built with SARIMA and AHW are easier to understand, since their behavior is described by a simple closed-form expression, whereas models built with RF, ANN and SVR cannot be explained intuitively. In long-term traffic forecasting, interpretability is not an issue, and is thus neglected in this work.

## 6. Performance Assessment

This section presents the evaluation of the different forecasting approaches with the dataset described in Section 4. For clarity, model construction is first explained, assessment methodology is then detailed and results are presented later.

### 6.1. Model Construction

Assessment is carried out with IBM SPSS Modeler [50], a commercial tool for predictive analytics extensively used in several fields [51–53]. The tool has a visual interface that allows users to leverage statistical and data mining algorithms without programming. Likewise, it offers an expert mode to find the optimal settings for certain model hyperparameters. These features are extremely valuable for network planners without a deep knowledge of prediction methods. SPSS Modeler 18.1 includes all the forecasting methods considered in this work except to ANN-LSTM, which is implemented by using *Keras* python library [54]. Table 1 summarizes the hyperparameter settings selected for the six prediction methods. To get the best of classical time series approaches,  $p$ ,  $d$ ,  $q$ ,  $P$ ,  $D$  and  $Q$  in SARIMA and  $\alpha$ ,  $\beta$  and  $\gamma$  in AHW must be set on a cell basis. For this purpose, the Expert mode offered by SPSS Modeler is used to find the optimal settings per cell. In this mode, the input data series is first transformed when appropriate (e.g., differentiating, square root or natural log) and model parameters are then estimated by minimizing the one-step ahead prediction error. Such an Expert mode is also used to set the number of

neurons per layer in ANN–MLP. In models comprising neural networks, overfitting is avoided through the use of early stopping callbacks and validation data. The rest of the hyperparameters are fixed manually following a grid search in the parameter space. The reader is referred to [54,55] for further information on the algorithms in SPSS Modeler and Keras, respectively.

**Table 1.** Methods of hyperparameter settings.

<b>SARIMA</b>	
$p, d, q, P, D, Q$	Automatically set by Expert mode
$m$	12
<b>AHW</b>	
$\alpha, \beta, \gamma$	Automatically set by Expert mode
$m$	12
<b>RF</b>	
No. of trees	10
Maximum depth	Until leaves have 10 samples
No. of predictors per tree	$\sqrt{N_p}$ ( $N_p$ : total number of predictors)
<b>ANN–MLP</b>	
No. of layers	3 (1 hidden layer)
No. neurons in hidden layer	Automatically set by Expert mode
Activation function	Hyperbolic tangent
Error function	Sum of squares
Training algorithm	Gradient descend
Training data split	70% for training/30% for validation
Training stop rule	Accuracy in the validation dataset is higher than 90% or does not improve across iterations
<b>ANN–LSTM</b>	
No. of layers	3 (2 LSTM layers + 1 output layer)
No. LSTM units per layer	10
Activation function	Rectified Linear Unit
Error function	Mean average error
Training algorithm	Adaptive Moment Estimation (Adam)
Training data split	70% for training/30% for validation
Training stop rule	100 epochs or accuracy in the validation dataset does not improve in 3 consecutive epochs
Batch size	64
<b>SVR</b>	
Error sensitivity parameter, $\epsilon$	0.1
Regularization parameter, $C$	10
Kernel	Linear
Training algorithm	Sequential minimal optimization

To check the sensitivity of methods to the data collection period, the five approaches are tested in 3 different cases. In a first case, it is assumed that the operator collects traffic data on a cell basis during 24 months. In the second and third cases, it is assumed that the operator collects traffic data during 18 and 12 months, respectively (e.g., network deployment stage). This reduction in the observation window is an important constraint for SARIMA and AHW, which require dozens of samples to predict monthly data from time series with some randomness, and these considered in this work [40]. Hence, only SL methods are compared in these cases.

To check the impact of prediction time horizon, two horizons are considered: 3 months (i.e., traffic is forecast 3 months in advance) and 6 months (i.e., 6 months in advance). Thus, the combination of factors results in 6 cases, hereafter referred to as cases 12-3, 12-6, 18-3,



18-6, 24-3 and 24-6, where the first number denotes the number of months when data are collected by the operator and the second number denotes the prediction horizon. From the operator point of view, the less data stored and the more in advance traffic is forecast (i.e., case 12-6), the better, but, from an accuracy point of view, the opposite is likely true (i.e., case 24-3).

Forecasting methods are trained and tested emulating the behavior of network planning tools. For clarity, an example of how the operator would apply the methods in each case is given in Figure 3. Specifically, the example illustrates how to predict traffic carried out in June 2017 with a 3-month time horizon (i.e., the prediction is made on March 2017) based on measurements from previous months. Figure 3a–c show the timeline used for SARIMA/AHW in case 24-3, RF/ANN-MLP/ANN-LSTM/SVR in case 24-3 and RF/ANN-MLP/ANN-LSTM/SVR in case 12-3, respectively. Figure 3a,b correspond to case 24-3 (i.e., 24-month collection period), where traffic in June 2017 is predicted in March 2017 based on historic data from April 2015 to March 2017. In SARIMA and AHW, a model is computed per cell with all data samples available (i.e., 24 datasamples), which is then applied recursively to predict traffic carried 3 months later (June 2017). This results in  $N_c$  models, with  $N_c$  the number of cells. In contrast, in SL methods, a single model is created for the whole network using data from all cells. First, the model is trained with available data. The training dataset consists of  $N_c$  datum points (1 per cell). Each datum point comprises 21 predictors, which are the monthly traffic in a specific cell from April 2015 to December 2016, and an output, which is the monthly traffic in that cell in March 2017. Note that not all months can be used as predictors to train the model to allow the 3-month horizon. Once the model is trained, traffic in June 2017 in a cell of the scenario is predicted by using the network-wide model with a new datum point containing traffic of that cell from July 2015 to March 2017 as predictors.

Similarly, Figure 3c illustrates the timeline in case 12-3 (12-month collection period). Now, only data from April 2016 to March 2017 are necessary. In this case, SARIMA and AHW methods cannot be used due to the limited length of the time series. In SL approaches, the process is identical to that explained in case 24-3, except for a reduction in the number of input attributes in every datum point from 21 to 9 months (3 months less than the new observation window). Likewise, datum points from case 18-3 have 15 traffic measurements as input datum points.

Data forecasting for cases 24-6, 18-6 and 12-6 (i.e., with a 6-month horizon) follows a similar procedure with a 6-month gap between the end of the observation window and the target/predicted month (e.g., in case 24-6, traffic in June 2017 is predicted in December 2016 based on measurements from January 2015 to December 2016).

It should be pointed out that, as stated in Section 5, ANN-LSTM is conceived to build a model per time series (i.e., per cell). However, the short length of the time series for the three considered observation windows (e.g., 12/18/24 samples) does not allow to have as many datum points (i.e., time lapses) as parameters. Note that the ANN-LSTM model with the architecture in Table 1 has 1331 parameters. To circumvent this problem, in this work, ANN-LSTM is used as the other SL methods, by creating a single model for the whole network, as explained in Figure 3b,c. To this end, a single time series is generated by concatenating time series from the different cells in the network and only time lapses where the target month is the output are considered (in the example above, June 2017).

Finally, it is worth mentioning that in this work, all available time series are used for both training and testing the SL models. However, training and test datasets used for each target month are disjoint due to a shift in the observation window. This workflow emulates how data become available in a network planning tool as time progresses.

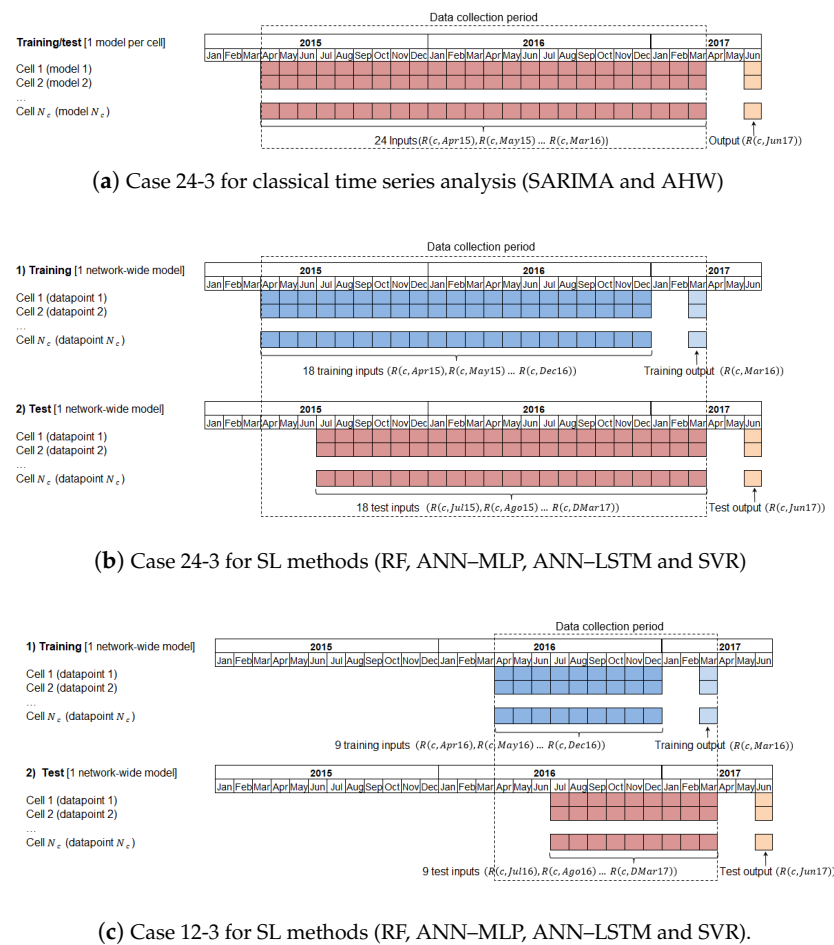


Figure 3. Timeline of prediction methods in cases 24-3 and 12-3.

## 6.2. Assessment Methodology

Three experiments are performed sequentially. Results of each experiment motivate the execution of the following one.

### 6.2.1. Experiment 1—Selection of the Observation Window

The first experiment aims to determine how many months of data (i.e., 12 months, 18 months or 24 months) are required to forecast cellular traffic with enough accuracy with the 2 considered prediction horizons (i.e., 3 and 6 months). For this purpose, the 6 considered methods are evaluated in cases 12-3, 12-6, 18-3, 18-6, 24-3 and 24-6. The data to be forecast are cell traffic in June 2017, as shown in Figure 3. As explained above, a model is created per cell with TSA methods, whereas a single network-wide model is built with all SL methods. Note that assessing case 24-6 (the most data-demanding case) requires collecting data 30 months in advance, and thus June 2017 is the only possible target month for this experiment in the considered dataset.

### 6.2.2. Experiment 2—Method Comparison

The second experiment aims to check: (a) how much time in advance prediction can be made (3 or 6 months), (b) the dependence of prediction accuracy on the target month and (c) which is the best prediction algorithm. For this purpose, cell traffic from July 2016 to June 2017 (i.e., for a year) is forecast 3 or 6 months in advance with an observation window of 12 months. Models used to predict traffic in the different months are created as explained in Section 6—A, but with the corresponding months displaced (e.g., in case 12-3, if the target month is May 2017, the observation window starts/ends a month before than

in Figure 3c). SARIMA and AHW approaches cannot be included in this experiment due to the small observation window (12 months).

### 6.2.3. Experiment 3—Creation of Specific Models for High-Traffic Cells

In capacity planning, accurate traffic prediction is especially important in cells with a high traffic, as these are more likely to suffer capacity problems. The aim of this experiment is to evaluate the possibility of creating a differentiated model for such cells. The idea is to discard underutilized cells, which often show noisy traffic measurements, when training the model. To this end, a model to forecast traffic in June 2017 is trained only with data from high-traffic cells. In this work, a cell  $c'$  is considered as a high-traffic cell if its traffic when the time prediction is made (e.g., in case 12-3, 3 months before, in Mar 2017) exceeds the 85th percentile of the monthly cell traffic in the network, i.e.,  $R(c', \text{Mar}17) > P_{85}^{R(c, \text{Mar}17)}$ .

In all the experiments, prediction accuracy is measured with two main figures of merit:

1. The Mean Absolute Error (*MAE*) computed as

$$MAE = \frac{1}{N_c} \sum_c |\widehat{R}(c, m) - R(c, m)|, \quad (3)$$

where  $\widehat{R}(c, m)$  is the predicted traffic for month  $m$  in cell  $c$ , and

2. The Mean Absolute Percentage Error (*MAPE*) computed as

$$MAPE = \frac{1}{N_c} \sum_c \left| 100 \frac{\widehat{R}(c, m) - R(c, m)}{R(c, m)} \right|, \quad (4)$$

Additionally, two secondary indicators are used for a more detailed assessment: (a) the *bias*, computed as the mean error

$$bias = \frac{1}{N_c} \sum_c (\widehat{R}(c, m) - R(c, m)), \quad (5)$$

and (b) the execution time, as a measure of computational load.

## 6.3. Results

### 6.3.1. Experiment 1

Table 2 breaks down the performance of the prediction methods in cases 12-3, 18-3 and 24-3 (3-month horizon). As explained above, SARIMA and AHW are not tested in cases 12-3 and 18-3. Results show that, in case 24-3, SARIMA achieves the worst performance, with an extremely large *MAPE* (=43.25%) and *MAE* (=2069.72 kbps). An analysis not shown here reveals that SARIMA fails to capture trend and level changes in historical data for some time series. Such a behavior may be due to the reduced number of historical data samples in the considered time series, which prevents the model from eliminating noise effect, thus causing a significant error when estimating future traffic. AHW outperforms SARIMA, but it still performs poorly (*MAPE* = 29.28% and *MAE* = 1780.71 kbps). Such a poor performance is due to the recursive nature of these models, where noisy input data severely degrade the accuracy of predictions beyond the next step. All SL techniques outperform AHW, with *MAPE* below 28% and *MAE* below 1600 kbps. Thus, it can be concluded that SL techniques are more accurate than SARIMA and AHW. When comparing case 24-3 (the longest tested window) against case 12-3 (the shortest tested window), it is observed that, unexpectedly, SL algorithms perform similarly or even better in case 12-3 (e.g., for ANN-MLP, *MAE* increases from 1023.55 kbps with 12 months to 1339.91 kbps with 24 months). This fact confirms that the influence of past measurements quickly diminishes in the long term due to changes in user trends and re-planning actions by the operator. This statement is reinforced by results from case 18-3, in which algorithm obtain any significant improvement compared to case 12-3.

Table 3 shows the comparison between cases 12-6, 18-6 and 24-6 (6-month prediction horizon). In case 24-6, all SL techniques but SVR again outperform both SARIMA and AHW (SVR outperforms SARIMA, but not AHW). When comparing cases 12-6 and 24-6, all SL algorithms achieve a better *MAE* in case 12-6, as in cases 12-3 vs. 24-3 (e.g., in SVR, *MAE* decreases from 2517.43 kbps in case 24-6 to 1372.89 kbps in case 12-6, a 45.47% in relative terms). Results from case 18-3 only improve performance of SVR. Nonetheless, such a method is still the worst SL approach.

**Table 2.** Impact of observation window for 3-month horizon.

Window	12 Months		18 Months		24 Months	
Indicator	<i>MAPE</i> [%]	<i>MAE</i> [kbps]	<i>MAPE</i> [%]	<i>MAE</i> [kbps]	<i>MAPE</i> [%]	<i>MAE</i> [kbps]
SARIMA	-	-	-	-	43.25	2069.72
AHW	-	-	-	-	29.28	1780.71
RF	23.75	1017.55	24.25	1027.20	23.29	1236.44
ANN-MLP	24.28	1023.55	25.33	1046.25	24.90	1339.91
ANN-LSTM	22.65	976.69	25.61	1076.16	23.38	999.15
SVR	25.78	1070.03	24.70	1020.59	27.86	1572.81

**Table 3.** Impact of observation window for 6-month horizon.

Window	12 Months		18 Months		24 Months	
Indicator	<i>MAPE</i> [%]	<i>MAE</i> [kbps]	<i>MAPE</i> [%]	<i>MAE</i> [kbps]	<i>MAPE</i> [%]	<i>MAE</i> [kbps]
SARIMA	-	-	-	-	590.17	16,708.55
AHW	-	-	-	-	30.88	1902.44
RF	23.94	1048.63	25.14	1053.56	22.76	1199.32
ANN-MLP	24.23	1055.93	27.46	1277.13	23.58	1250.32
ANN-LSTM	22.23	1034.30	24.59	1041.39	29.55	1253.69
SVR	30.81	1372.89	27.59	1163.40	37.66	2517.43

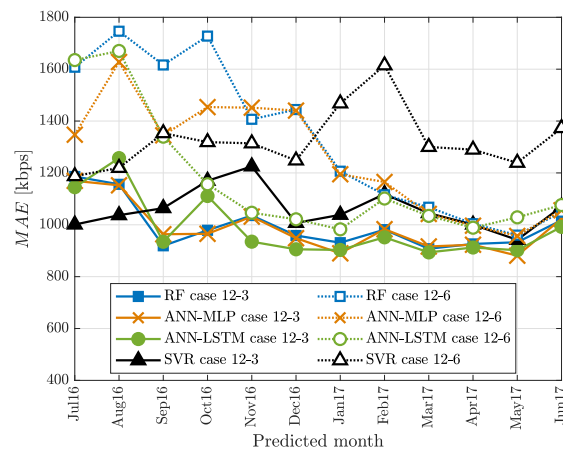
From the above results, it can be concluded that: (a) SL approaches outperform SARIMA and AHW when predicting traffic in cellular networks in the long term and (b) there is not much benefit on storing more than 1 year of traffic measurements (unless SARIMA, AHW or SVR are the only option). Thus, AHW and SARIMA are not considered in further experiments. Likewise, a 12-month time horizon is chosen as optimal for SL methods.

### 6.3.2. Experiment 2

Table 4 presents the average *MAPE*, *MAE* and *bias* across months for each algorithm and case. For a more detailed analysis, Figure 4 breaks down the *MAE* obtained in cases 12-3 (solid lines) and 12-6 (dotted lines) for each target month. Recall that cases 24-3 and 24-6 are not considered based on the conclusions in experiment 1. Table 4 show that, in case 12-3, RF, ANN-MLP and ANN-LSTM perform similarly ( $\overline{MAPE} \approx 27\%$  and  $\overline{MAE} \approx 1000$  kbps), outperforming SVR ( $\overline{MAPE} = 30.26\%$  and  $\overline{MAE} = 1059.86$  kbps). Moreover, Figure 4 shows that prediction accuracy for the former algorithms degrades significantly when predicting traffic in July and August 2016 (summer holidays) compared to the rest of months (working months). This might be due to isolated events taking place during summer months in the country where data was collected (e.g., tourism, festivals, etc.) that change traffic patterns unpredictably, making information collected 3 months in advance not representative of the traffic in the months to come. In contrast, SVR shows a more stable behavior during summer months (i.e., *MAE* does not degrade).

**Table 4.** Summary of method and time horizon comparison across months.

Case	12-3			12-6		
	$\overline{MAPE}$ [%]	$\overline{MAE}$ [kbps]	$\overline{bias}$ [kbps]	$\overline{MAPE}$ [%]	$\overline{MAE}$ [kbps]	$\overline{bias}$ [kbps]
RF	27.63	994.21	160.51	40.09	1329.31	719.20
ANN-MLP	27.73	987.28	134.61	38.71	1256.40	572.46
ANN-LSTM	26.37	986.88	134.27	31.87	1173.39	287.73
SVR	30.26	1059.86	−251.32	36.71	1327.15	−531.02

**Figure 4.** MAE time evolution.

By comparing cases 12-3 and 12-6 in Table 4, it is observed that, for all algorithms, there is a significant degradation in accuracy if traffic predictions are made more than 3 months in advance (e.g., for ANN-LSTM,  $\overline{MAE}$  and  $\overline{MAPE}$  increase by 18.89% and 20.85% in relative terms, respectively). Moreover, dotted lines in Figure 4 show a strong variation in  $MAE$  across months in case 12-6. Thus, when possible, it is recommended to use a 3-month prediction horizon.

ANN-LSTM shows the best overall results, with a  $\overline{MAPE}$  of 26.37% and a  $\overline{MAE}$  of 986.68 kbps in case 12-3, and the smallest degradation in accuracy from case 12-3 to case 12-6. Nonetheless, for a 3-month horizon, ANN-MLP and RF algorithms can be used alternatively with similar prediction accuracy ( $\overline{MAPE} \approx 27.70\%$  and  $\overline{MAE} \approx 990$  kbps). Remember that, unlike ANN-LSTM, the latter algorithms are included in most data analytics tools (such as, e.g., SPSS modeler), which provide automatic schemes that relieve network operators from the complex hyperparameter tuning task.

It should be pointed out that, even for the best method and case, prediction results are not very accurate (i.e.,  $\overline{MAE} \approx 1000$  kbps, or, expressed more intuitively, a deviation of 0.39 GB per hour and cell). This error can be explained by re-planning actions taken by the operator in the considered network area during the data collection period, which lead to unpredictable traffic changes in neighbor cells. For a closer analysis, Figure 5 illustrates traffic prediction from January 2017 to June 2017 with a 3-month horizon for two cells, referred to as Cell A and Cell B, with the compared methods. No significant re-planning actions were taking in the surroundings of Cell A during the data collection period, whereas Cell B is a cell with a new neighbor cell deployed in October 2016. In Figure 5a, it is observed that, for Cell A, all methods approximate real traffic quite well (e.g.,  $MAE$  fluctuates between 4 and 900 kbps for ANN-MLP, and between 314 and 824 kbps for RF). In contrast, in Cell B, the abrupt decrease in traffic in October 2016, caused by the deployment of the nearby cell, leads to large prediction errors even for all models.

It is also remarkable that  $\overline{bias}$  values in Table 4 for case 12-3 are negative or close to 0, i.e., models tend to underestimate traffic. This behavior is risky especially for high-traffic cells, which are more likely to suffer from capacity problems and, hence, require re-dimensioning actions. For a closer analysis of bias, Figure 6 shows the error scatter plot



of error versus measured cell traffic obtained when predicting traffic carried in March 2017 with ANN-LSTM in case 12-3 (the combination of month/algorithm/horizon with the lowest MAE in this experiment). The regression line shows that the more loaded cells are, the more negative the error is. This trend points out the need for a more accurate model for high-traffic cells. This problem will be addressed in experiment 3.

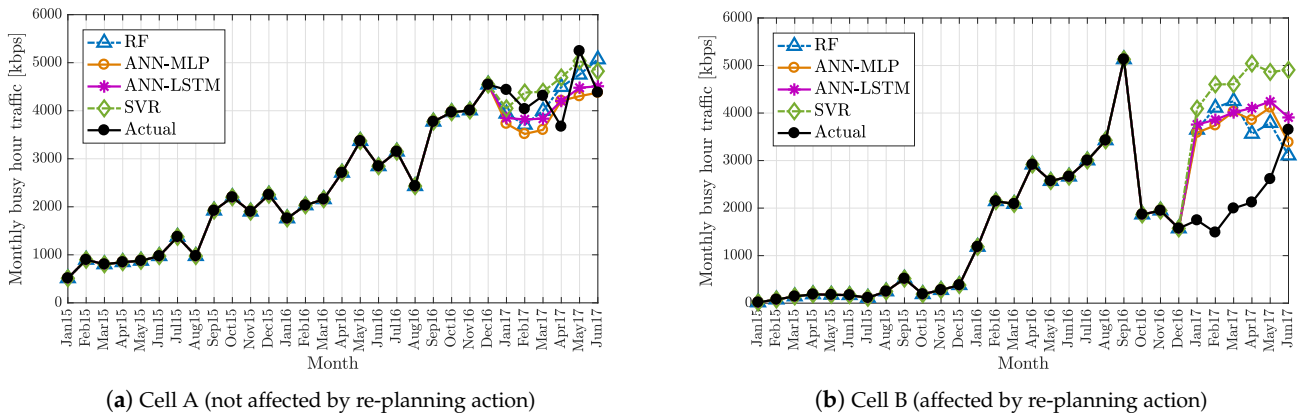


Figure 5. Example of the impact of re-planning actions on cell traffic.

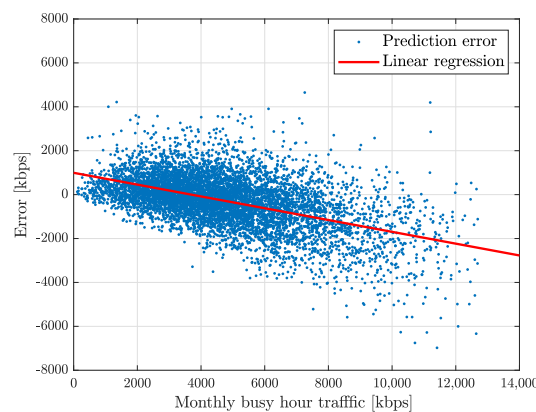


Figure 6. Error vs. monthly busy-hour traffic (ANN-LSTM, case 12-3).

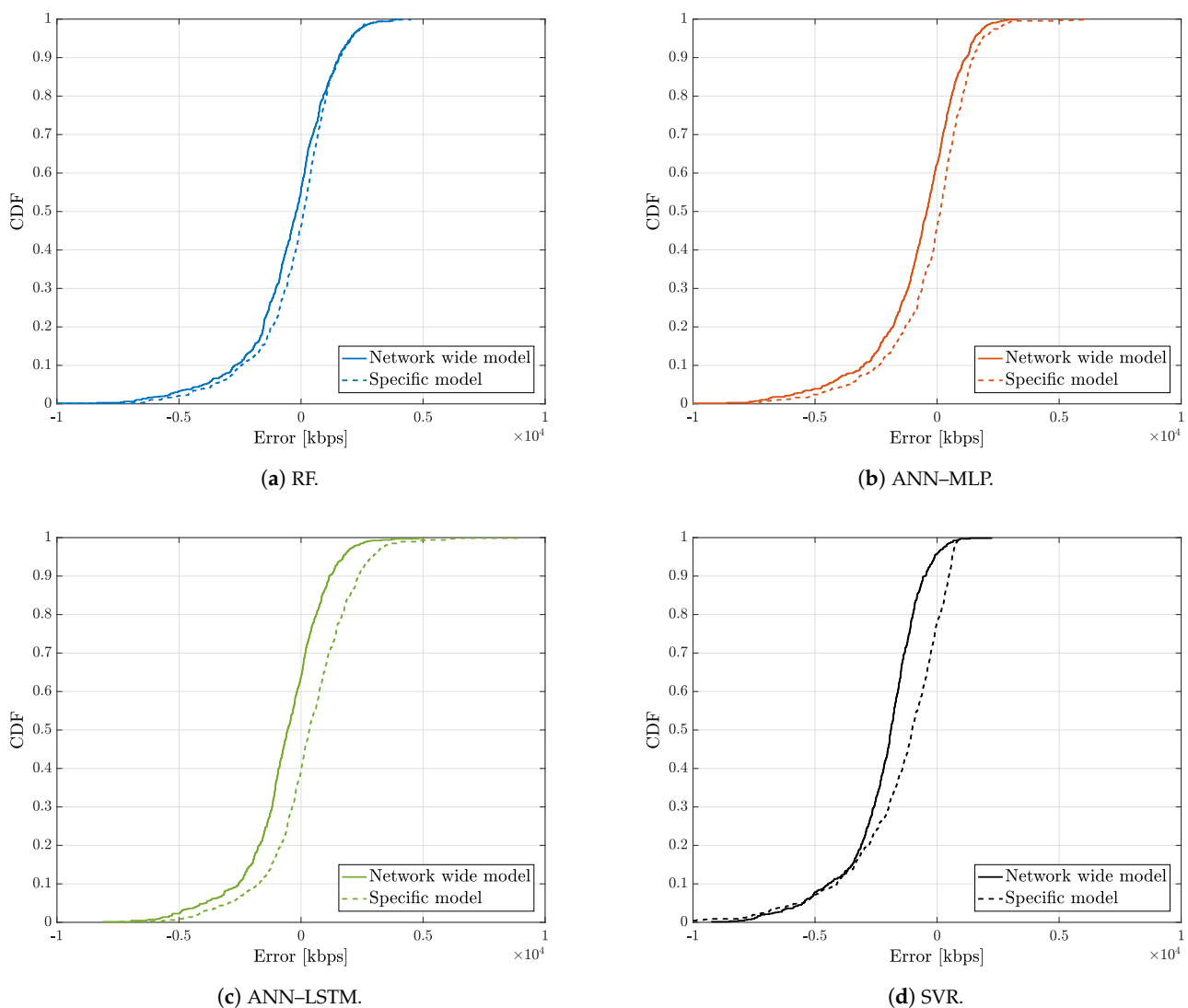
### 6.3.3. Experiment 3

Table 5 breaks down MAPE and MAE in case 12-3 for the 1074 (15%) cells with the largest traffic in March 2017, obtained with two different models: (a) the model built in experiment 1 (denoted as network-wide model), and (b) a specific model trained exclusively with data collected from the high-traffic cells (denoted as specific model). Note that the same set of cells is tested to assess both models. To isolate the effect of building a differentiated model, those high-traffic cells that were affected by re-planning actions before June 2017 have not been considered when evaluating the accuracy of the models. In the table, it is observed that the specific model outperforms the network-wide model in RF, ANN-MLP and SVR methods, whereas the specific model created with ANN-LSTM does not improve the network-wide model built with this method. SVR experiences the largest improvement with the specific model in absolute terms, decreasing MAE from 2223.88 kbps to 1725.20 kbps (22.42% in relative terms) and MAPE from 20.44% to 14.19% (29.11% in relative terms). Nonetheless, it is still the worst method. Specific models created with RF and ANN-MLP show the best results, achieving MAPE  $\approx$  11% and MAE  $\approx$  1200 kbps.

**Table 5.** Performance of network-wide and specific models for high-traffic cells in case 12-3.

Model	Network-Wide		Specific	
Indicator	MAPE [%]	MAE [kbps]	MAPE [%]	MAE [kbps]
RF	12.46	1339.88	11.26	1212.18
ANN-MLP	12.31	1374.55	11.35	1232.98
ANN-LSTM	12.21	1311.72	13.27	1356.13
SVR	20.44	2223.88	14.49	1725.20

For a closer analysis, Figure 7a–d show the error cumulative distribution functions (CDF) obtained with the different methods for high-traffic cells with the network-wide model (solid lines) and the specific model (dashed lines) in case 12-3 for the selected target month (June 2017). It is observed that, for all methods, error curves with the specific models are shifted to the right compared to those with the network-wide models, whose median values are closer to 0. Thus, the specific models for high traffic cells not only increase prediction accuracy, but also reduce the bias. Nevertheless, CDFs show that prediction error is still negative in many high-loaded cells ( $\approx 45\%$  for RF, ANN-MLP and ANN-LSTM and  $\approx 75\%$  for SVR). This issue should be addressed by other means (e.g., models based on predictors other than cell traffic).



**Figure 7.** Error cumulative distribution functions for SL methods in high-traffic cells (case 12-3).

#### 6.4. Computational Complexity

The different methods are compared by their total execution time. For SARIMA and AHW, execution time comprises computing a cell-specific model and extending it until the target month, which must be repeated for all cells in the system. For SL algorithms, execution time comprises training a network-wide model with the series from all cells and computing predictions from historical traffic values (i.e., no model extension is needed).

For SARIMA and AHW, it is expected that running time grows linearly with the number of predictors (months),  $N_m$ , and the number of models built (cells),  $N_c$ . Thus, their worst-case time complexity is  $O(N_c N_m)$ . For SL methods, running time increases with the number of datum points (cells),  $N_c$ , and the number of predictors/inputs (months),  $N_m$ . Specifically, the worst-case time complexity for the back propagation algorithm used to train ANNs with  $N_m$  inputs, 1 output and  $N_l$  hidden layers is  $O(N_c N_m N_l N_n N_i)$  where  $N_n$  is the size of the hidden layers and  $N_i$  is the number of iterations. The time complexity of sequential minimal optimization used to train SVR is quadratic with the training set size,  $O(N_c^2)$ . Likewise, the worst-case time complexity of RF is given by the time of building a complete decision tree,  $O(N_m N_c \log N_c)$ .

Table 6 summarizes the time taken to train the models in the different approaches in experiment 1 (7160 cells) in a centralized server with Intel Xenon octa-core processor, clock frequency of 2.4 GHz and 64 GB of RAM. The number of predictors per case is shown on the upper part of the table. Results show that, in cases 24-3 and 24-6, AHW and SARIMA are the most time-consuming methods, since they build a model per cell. Among SL methods, ANN-MLP is the fastest approach, whereas ANN-LSTM is the lowest approach. In SL methods, it is observed that, the longer data collection period, the larger number of predictors, and hence the larger runtime. For a certain collection period, the larger the time horizon, the lower number of predictors in the model, and hence the lower runtime. In contrast, in SARIMA and AHW, the longer the time horizon, the larger runtime, since the model must be extended until the target month. Nonetheless, execution times in all cases are negligible in long-term traffic forecasting, where traffic must be predicted at most once a month.

**Table 6.** Execution times (s).

Case	12-3	12-6	24-3	24-6
Number of predictors				
SARIMA and AHW	-	-	24	24
SL algorithms	9	6	21	18
Execution times [s]				
SARIMA (per cell)	-	-	0.56	0.61
AHW (per cell)	-	-	0.65	0.72
RF (entire network)	5	4	8	6
ANN-MLP (entire network)	3	2	6	4
ANN-LSTM (entire network)	140	98	313	245
SVR (entire network)	12	10	14	13

## 7. Conclusions

Accurate long-term traffic forecasting will be crucial for network (re)dimensioning in 5G networks. However, monthly busy-hour time series are short and noisy, which makes long-term traffic prediction a challenging task. In this work, a comparative study has been carried out to assess different approaches for predicting cellular traffic in the long term (i.e., several months in advance). Six methods have been compared, including classical time series analysis schemes (SARIMA and AHW) and supervised machine learning algorithms (RF, ANN-MLP, ANN-LSTM and SVR). To this end, three experiments have been carried out with a dataset taken from a live LTE network, covering an entire country (7160 cells) and traffic data for two and a half years.

Results have shown that SL methods outperform classical time series analysis in terms of accuracy and required storage capacity. Specifically, with SL algorithms, traffic carried per cell can be predicted with a  $MAE \approx 1000$  kbps with a 3-month time horizon and a 12-month observation window (i.e., data collection period). It has also been shown that it is convenient to develop specific models for high-traffic cells, where prediction accuracy is critical. Overall, RF and ANN–MLP have shown the best results, providing acceptable accuracy ( $MAPE \approx 11\%$ ) to detect capacity bottlenecks in high-load cells with a 3-month prediction horizon by using data from the past 12 months. It is remarkable that these non-deep algorithms perform very similar to deep neural networks based on LSTM units, used to model time dependencies in short- and medium-term traffic forecasting. This is due to the monthly busy-hour aggregation of data when forecasting traffic for network dimensioning, which reduces time series length and predictability compared to series used in short/medium-term forecasting. Nonetheless, none of the considered algorithms is extremely accurate, especially for summer months, due to changes in user trends, social events or temporary re-planning actions by the operator.

The use of SL algorithms proposed in this work, based on the creation of a single model for the whole network, is suitable for a centralized network dimensioning tool running in the OSS, where historical traffic measurements are currently stored. Alternatively, SARIMA and AHW could also be executed in base stations if historical measurements were available in each cell.

Future work will extend models to account for events that drastically change traffic patterns in a cell (e.g., new neighbor site, equipment upgrades, social events, etc.). Likewise, the values of nearby cells (e.g., co-sited cells) can be jointly predicted via multi-task learning to cope with noise in time series.

**Author Contributions:** The contributions of authors are: Conceptualization, C.G. and M.T.; data curation, M.L.M.-A. and J.M.R.-A.; formal analysis, C.G.; funding acquisition, C.G., M.T. and S.L.-R.; investigation, C.G., M.T. and S.L.-R.; methodology, C.G., M.T. and J.M.R.-A.; project administration, M.T. and S.L.-R.; resources, M.L.M.-A.; software, M.L.M.-A.; supervision, M.T. and S.L.-R.; validation, C.G.; visualization, C.G.; writing—original draft, C.G.; writing—review and editing, M.T., S.L.-R., M.L.M.-A. and J.M.R.-A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been funded by Junta de Andalucía (UMA18-FEDERJA-256), the Spanish Ministry of Science, Innovation and Universities (RTI2018-099148-B-I00) and the Spanish Ministry of Education, Culture and Sports (FPU17/04286).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ericsson. *Ericsson Mobility Report*; Ericsson: Stockholm, Sweden, 2018.
2. NGMN. 5G White paper. In *White Paper*; NGMN: Frankfurt am Main, Germany, 2015.
3. Moysen, J.; Giupponi, L.; Mangues-Bafalluy, J. A mobile network planning tool based on data analytics. *Mob. Inf. Syst.* **2017**. [[CrossRef](#)]
4. Net2Plan. The Open Source Network Planner. Available online: <http://www.net2plan.com/index.php> (accessed on 12 September 2020).
5. Toril, M.; Ferrer, R.; Pedraza, S.; Wille, V.; Escobar, J.J. Optimization of half-rate codec assignment in GERAN. *Wirel. Pers. Commun.* **2005**, *34*, 321–331. [[CrossRef](#)]
6. Gijon, C.; Toril, M.; Luna-Ramirez, S.; Mari-Altozano, M.L. A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 9414–9424. [[CrossRef](#)]
7. Fattah, H.; Leung, C. An overview of scheduling algorithms in wireless multimedia networks. *IEEE Wirel. Commun.* **2002**, *9*, 76–83. [[CrossRef](#)]
8. Onur, E.; Deliç, H.; Ersoy, C.; Çağlayan, M.U. Measurement-based replanning of cell capacities in GSM networks. *Comput. Netw.* **2002**, *39*, 749–767. [[CrossRef](#)]
9. García, P.A.; González, A.Á.; Alonso, A.A.; Martínez, B.C.; Pérez, J.M.A.; Esguevillas, A.S. Automatic UMTS system resource dimensioning based on service traffic analysis. *EURASIP J. Wirel. Commun. Netw.* **2012**, *2012*, 323. [[CrossRef](#)]
10. Shu, Y.; Yu, M.; Yang, O.; Liu, J.; Feng, H. Wireless traffic modeling and prediction using seasonal ARIMA models. *IEICE Trans. Commun.* **2005**, *88*, 3992–3999. [[CrossRef](#)]

11. Zhou, B.; He, D.; Sun, Z. Traffic modeling and prediction using ARIMA/GARCH model. In *Modeling and Simulation Tools for Emerging Telecommunication Networks*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 101–121.
12. Li, R.; Zhao, Z.; Zhou, X.; Ding, G.; Chen, Y.; Wang, Z.; Zhang, H. Intelligent 5G: When cellular networks meet artificial intelligence. *IEEE Wirel. Commun.* **2017**, *24*, 175–183. [[CrossRef](#)]
13. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [[CrossRef](#)]
14. Huang, C.W.; Chiang, C.T.; Li, Q. A study of deep learning networks on mobile traffic forecasting. In Proceedings of the IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017; pp. 1–6.
15. Fang, L.; Cheng, X.; Wang, H.; Yang, L. Mobile demand forecasting via deep graph-sequence spatiotemporal modeling in cellular networks. *IEEE Internet Things J.* **2018**, *5*, 3091–3101. [[CrossRef](#)]
16. Bui, N.; Cesana, M.; Hosseini, S.A.; Liao, Q.; Malanchini, I.; Widmer, J. A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1790–1821. [[CrossRef](#)]
17. Mishra, A.R. *Fundamentals of Network Planning and Optimisation 2G/3G/4G: Evolution to 5G*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2018.
18. Grillo, D.; Skoog, R.A.; Chia, S.; Leung, K.K. Teletraffic engineering for mobile personal communications in ITU-T work: The need to match practice and theory. *IEEE Pers. Commun.* **1998**, *5*, 38–58. [[CrossRef](#)]
19. Northcote, B.S.; Tompson, N.A. Dimensioning Telstra’s WCDMA (3G) network. In Proceedings of the 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD), Miami, FL, USA, 3–4 December 2010; pp. 81–85.
20. Jailani, E.; Ibrahim, M.; Ab Rahman, R. LTE speech traffic estimation for network dimensioning. In Proceedings of the 2012 IEEE Symposium on Wireless Technology and Applications (ISWTA), Bandung, Indonesia, 23–26 September 2012; pp. 315–320.
21. Toril, M.; Luna-Ramírez, S.; Wille, V. Automatic replanning of tracking areas in cellular networks. *IEEE Trans. Veh. Technol.* **2013**, *62*, 2005–2013. [[CrossRef](#)]
22. Xu, F.; Lin, Y.; Huang, J.; Wu, D.; Shi, H.; Song, J.; Li, Y. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE Trans. Serv. Comput.* **2016**, *9*, 796–805. [[CrossRef](#)]
23. Yu, Y.; Wang, J.; Song, M.; Song, J. Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient. In Proceedings of the 2010 International Conference on Intelligent System Design and Engineering Application, Changsha, China, 13–14 October 2010; Volume 1, pp. 980–983.
24. Tikunov, D.; Nishimura, T. Traffic prediction for mobile network using Holt-Winter’s exponential smoothing. In Proceedings of the 2007 15th International Conference on Software, Telecommunications and Computer Networks, Dubrovnik, Croatia, 27–29 September 2007; pp. 1–5.
25. Bastos, J. Forecasting the capacity of mobile networks. *Telecommun. Syst. (Springer)* **2019**, *72*, 231–242. [[CrossRef](#)]
26. Li, R.; Zhao, Z.; Zhou, X.; Palicot, J.; Zhang, H. The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice. *IEEE Commun. Mag.* **2014**, *52*, 234–240. [[CrossRef](#)]
27. Jnr, M.D.; Gadze, J.D.; Anipa, D.K. Short-term traffic volume prediction in UMTS networks using the Kalman filter algorithm. *Int. J. Mob. Netw. Commun. Telemat.* **2013**, *3*, 31–40.
28. Bui, N.; Widmer, J. Data-driven evaluation of anticipatory networking in LTE networks. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2252–2265. [[CrossRef](#)]
29. Li, R.; Zhao, Z.; Zheng, J.; Mei, C.; Cai, Y.; Zhang, H. The learning and prediction of application-level traffic data in cellular networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3899–3912. [[CrossRef](#)]
30. Hua, Y.; Zhao, Z.; Liu, Z.; Chen, X.; Li, R.; Zhang, H. Traffic prediction based on random connectivity in deep learning with long short-term memory. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 27–30 August 2018; pp. 1–6.
31. Trinh, H.D.; Giupponi, L.; Dini, P. Mobile traffic prediction from raw data using LSTM networks. In Proceedings of the IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1827–1832.
32. Feng, J.; Chen, X.; Gao, R.; Zeng, M.; Li, Y. DeepTP: An end-to-end neural network for mobile cellular traffic prediction. *IEEE Netw.* **2018**, *32*, 108–115. [[CrossRef](#)]
33. Nie, L.; Jiang, D.; Yu, S.; Song, H. Network traffic prediction based on deep belief network in wireless mesh backbone networks. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–5.
34. Assem, H.; Caglayan, B.; Buda, T.S.; O’Sullivan, D. ST-DenNetFus: A new deep learning approach for network demand prediction. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Dublin, Ireland, 10–14 September 2018; pp. 222–237.
35. Zhang, C.; Patras, P. Long-term mobile traffic forecasting using deep spatio-temporal neural networks. In Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Los Angeles, CA, USA, 25–28 June 2018; pp. 231–240.



36. Wang, X.; Zhou, Z.; Xiao, F.; Xing, K.; Yang, Z.; Liu, Y.; Peng, C. Spatio-temporal analysis and prediction of cellular traffic in metropolis. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2190–2202. [[CrossRef](#)]
37. Wang, J.; Tang, J.; Xu, Z.; Wang, Y.; Xue, G.; Zhang, X.; Yang, D. Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
38. Zhang, C.; Zhang, H.; Yuan, D.; Zhang, M. Citywide cellular traffic prediction based on densely connected convolutional neural networks. *IEEE Commun. Lett.* **2018**, *22*, 1656–1659. [[CrossRef](#)]
39. Qiu, C.; Zhang, Y.; Feng, Z.; Zhang, P.; Cui, S. Spatio-temporal wireless traffic prediction with recurrent neural network. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 554–557. [[CrossRef](#)]
40. Hyndman, R.J.; Kostenko, A.V. Minimum sample size requirements for seasonal forecasting models. *Foresight* **2007**, *6*, 12–15.
41. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
42. Winters, P.R. Forecasting sales by exponentially weighted moving averages. *Manag. Sci.* **1960**, *6*, 324–342. [[CrossRef](#)]
43. Efron, B. Bootstrap methods: Another look at the jackknife. In *Breakthroughs in Statistics*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 569–593.
44. Hastie, T.; Tibshirani, R.; Friedman, J.; Franklin, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd ed.; SPRINGER Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2001.
45. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
46. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Hoboken, NJ, USA, 1994.
47. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
48. Ing, C.K. Multistep prediction in autoregressive processes. *Econom. Theory* **2003**, *19*, 254–279. [[CrossRef](#)]
49. Lipton, Z.C. The Mythos of Model Interpretability. *Queue* **2018**, *16*, 30:31–30:57. [[CrossRef](#)]
50. IBM. *IBM SPSS Modeler 18.1.1 User's Guide*; IBM: Armonk, NY, USA, 2017.
51. Yaffee, R.A.; McGee, M. *An Introduction to Time series Analysis and Forecasting: With Applications of SAS® and SPSS®*; Elsevier: Amsterdam, The Netherlands, 2000.
52. Devi, B.; Rao, K.; Setty, S.; Rao, M. Disaster prediction system using IBM SPSS data mining tool. *Int. J. Eng. Trends Technol. (IJETT)* **2013**, *4*, 3352–3357.
53. Meng, X.H.; Huang, Y.X.; Rao, D.P.; Zhang, Q.; Liu, Q. Comparison of three data mining models for predicting diabetes or prediabetes by risk factors. *Kaohsiung J. Med. Sci.* **2013**, *29*, 93–99. [[CrossRef](#)]
54. Chollet, F. Keras. Available online: <https://keras.io> (accessed on 15 September 2020).
55. IBM. *IBM SPSS Modeler 18.1 Algorithms Guide*; IBM: Armonk, NY, USA, 2017.