Neural Networks for Cellular Base Station Switching

Neural Networks for Cellular Base Station Switching

Igor Donevski[1], Greta Vallero[1], Marco Ajmone Marsan[1,2]
[1] *Electronics and Telecommunications Department, Politecnico di Torino, Italy*
[2] *IMDEA Networks Institute, Spain*

***Abstract*—The adoption of base station sleep modes is considered one of the most effective approaches for the reduction of the energy consumption of radio access networks. Sleep modes allow the switch-off of base stations in periods of low traffic, and their successive switch-on when traffic increases. The selection of the appropriate instants to switch base stations on and off requires an accurate prediction of the traffic loads in the near future. In this paper we explore the performance of machine learning techniques for traffic prediction and for the selection of the instants when to switch off and on base stations, considering a heterogeneous network portion comprising one macro cell and six small cells within the macro cell coverage. We experiment with two machine learning approaches. The first aims at short term traffic load estimation, and from this derives the best combination of switching decisions. The second performs both traffic estimation and switching optimization at one time. For both approaches we develop artificial neural network implementations based on the Dense Neural Network and Recurrent Neural Network paradigms. Testing the two approaches on real traffic data, we observe very good performance in terms of both quality of service and energy saving.**

***Index Terms*—Cellular Base Station Switching; Machine Learning; Recurrent Neural Networks**

I. Introduction

Green networking, or energy efficient networking, has been a hot research topic for over 15 years, since the publication of the paper by Gupta and Singh at Sigcomm 2003 [1]. The intense research that followed targeted the design of energy-parsimonious approaches for both wired and wireless networks. In the case of wireless networks, base stations (BSs) were identified as the main source of energy consumption in radio access networks (RANs), and the introduction of sleep modes to switch off BSs in periods of low traffic, and switch them on again when traffic increases, was identified as the most effective approach to save energy. Surveys of the many proposals in this domain can be found, e.g., in [2]–[4].

The relevance of BS sleep modes is bound to increase: the yearly measurements and forecasts of the Cisco Visual Networking Index [5], [6] point out that the amount of traffic per smartphone per month has risen 38% from 2015 to 2016. In 2022, mobile data will amount to 71 percent of the total IP traffic. What is most relevant for sleep modes is the fact that the discrepancy between idle and busy hours is growing: the traffic in the busy hour is expected to increase by a factor 4.8 between 2017 and 2022, while the daily traffic is forecast to increase by a factor 3.7.

In spite of this, mobile network operators (MNOs) have been reluctant to introduce BS sleep modes in their network management algorithms, because most of the research proposals assume that the traffic load pattern along time is exactly known (see e.g., [7]), and MNOs fear that errors in traffic estimation can lead to poor quality of service (QoS). This makes the accurate short term prediction of traffic an extremely important prerequisite for the adoption of BS sleep modes.

MNOs collect traffic patterns at the BSs of their RAN over long time periods, in order to be able to make wise choices about network upgrades and new technologies introduction. Traffic prediction can thus exploit these time series, and, given the traffic load behavior up to a recent time instant, try to predict the traffic load over a short time interval, and decide whether some BSs should be put to sleep according to the traffic forecast.

In this paper we exploit artificial neural networks (ANNs) for short term traffic prediction, and we investigate their accuracy in a heterogeneous network scenario comprising one macro BS that gives origin to one macro cell within which 6 micro BSs define 6 small cells whose purpose is to handle traffic hot spots. These 6 micro BSs can be switched on and off according to the level of traffic demand.

The rest of this paper is organized as follows. Section II explains in more detail the issues in BS switching. Section III provides information about the RAN scenario that we consider, the available data, the energy consumption model, and defines the optimization problem of BS switching. Section IV contains a detailed explanation of the machine learning approaches that are tested as a solution to the problem. Section V presents numerical results, and finally Section VI concludes the paper.

II. Base Station Sleep Modes

BS sleep modes are a very promising approach that comes with some implementation issues. One problem, that is typical of the case of a single tier BS coverage, is the risk of coverage holes. Indeed, in spite of partial overlaps of cells, it is possible that when a BS is switched off, some area is excluded from service. This hindrance does not appear in new generation RANs that are implemented with a collection of technology generations creating cells of different sizes and capacities. In these new RANs, hierarchies of cells exist, and form dense and heterogeneous networks (HetNets) in which larger Macro Base Stations (MBSs) overlap with Small cell Base Stations (SBSs). In dense urban areas it is expected that high capacity scenarios can be achieved through extreme BS densification [8]. In such cases, switching off an underutilized SBS contributes only in the decrease of serviceable capacity, as the traffic that cannot be served by the SBS can be forwarded to the MBS. If planned well, a SBS shut-off lowers the energy spending

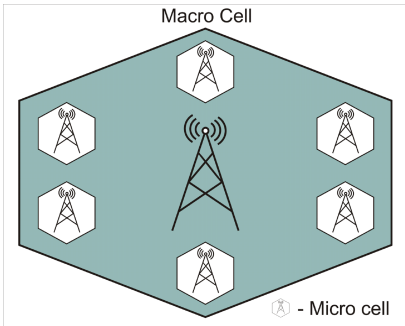

Fig. 1: The considered HetNet scenario

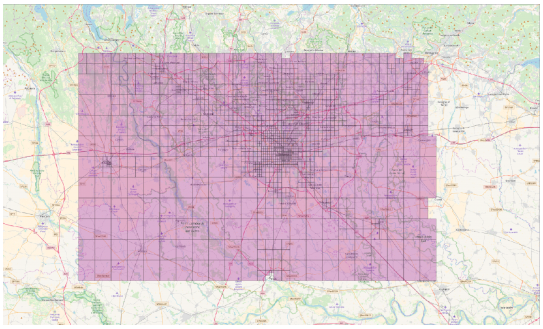

Fig. 2: The rectangular areas in the Milan metropolitan area that are taken as cellular base stations

whilst keeping network operation nominal. This makes sleep modes particularly well suited to dense RANs [9].

Many suggested implementations [9] recognize a plethora of culprits for the complexity of BS switching. Our work recognizes the issue of switching aversion, a consequence of the proneness of current equipment to failure when switched very often. This deters MNOs to commit to a switch decision, unless confident that the BS's service will not be needed for at least one hour. For such guarantee to be provided, prior knowledge of the future traffic demand, and accurate traffic estimation, are essential, hence an advanced understanding of the areas service history and a good estimation tool are necessary. In addition to estimation, the optimization problem for deciding the most efficient combination of switched off BSs needs to be solved. A problem which is easy to solve for small numbers of BSs, but can become difficult when the number of BSs is large.

To address both estimation and optimization at once requires a robust algorithm such as offered by machine learning (ML) algorithms. More specifically, Artificial Neural Networks (ANNs) are interesting as a very powerful tool for solving complex problems. Previous results in the literature include a very simple case of optimization [10], as well as a case where data caching at BS is assumed [11]. In comparison to the aforementioned cases, our work focuses only on the switching aversion as an issue; which narrows the applicable ANN models to the ones analyzed in Section IV.

III. The Scenario

We consider a portion of a HetNet comprising one MBS and six SBSs within the MBS coverage, as illustrated in Figure 1.

A. The traffic data

This work uses real traffic data that is provided in confidentiality by an Italian MNO. Within the provided data, Internet traffic overwhelms the other services in terms of load, ergo it is considered central to the analysis. To describe the spatial distribution of traffic, data volumes are reported for 1419 rectangular areas of variable size covering the metropolitan area of the city of Milan, as shown in Fig. 2. From here on, each rectangular area will be referred to as a BS, and a group of seven neighboring areas where one is chosen to be the MBS will be referred as a "scenario".

The traffic data spans exactly two months from March 1st 2015 to April 30th 2015. The time granularity of the available data corresponds to 15-minute time slots. For the purpose of this research, in order to avoid switching BSs too often, data is aggregated to a granularity of sixty minutes and switching is allowed at the beginning of every hour.

The available data allows testing in different scenarios based on human activity: business, industrial and residential; or based on significant local infrastructure: the San Siro football stadium, highway sections, and popular touristic areas such as the Duomo of Milan. Additionally, the available two-month period offers diversity in the analysis as it includes two irregular phenomena: Catholic Easter and beginning of Daylight Saving Time. Overall, the available data comprises $24 \cdot 61 - 1 = 1463$ time slots for each of the 1419 BSs.

B. Energy Consumption Model

To describe the energy consumption of a BS, we use the standard linear model proposed in [12]. The BS energy consumption can be mainly accredited to the power amplifier, the RF transceiver, the baseband processor, the DC-DC converter, the cooling, and the AC/DC power supply. By computing the impact of each element on the BS consumption, the model separates the load independent elements from the load dependent ones. The load dependency mainly amounts to the power amplifier element in each transceiver, that behaves in a near linear fashion. Finally, the total BS consumption is obtained by accounting for the number of transceivers on the BS. As a result, the total input power needed at the BS is:

$$P_{in} = \begin{cases} N_{TRX}\left(P_0 + \Delta p \; P_{out}\right), & 0 < P_{out} \leq P_{max} \\ N_{TRX} \; P_{sleep}, & P_{out} = 0 \end{cases} \quad (1)$$

where P_{out} is defined as:

$$P_{out} = \rho \; P_{max}, \qquad 0 \leq \rho \leq 1 \quad (2)$$

and where N_{TRX} is the number of transceivers on the BS; P_0 is the idle power consumption per transceiver; Δp is the slope of the load dependent elements; P_{out} is the instantaneous transmission power of the transceiver; P_{max} is the maximum possible transmission power of a transceiver; ρ is the BS traffic

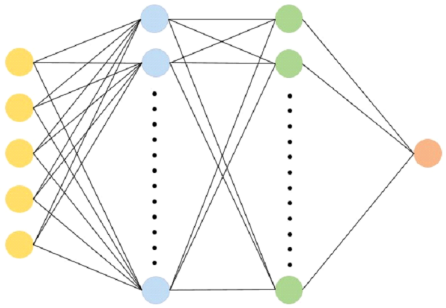

Fig. 3: The architecture of the Dense Neural Network for Two-Step-DNN

load normalized to 1; P_{sleep} is the amount of energy spent in sleep mode, which we assume negligible.

The problem of BS switching in the case of one MBS and N_s SBSs simplifies to turning off SBSs that have load ρ_i less than a predefined constant T while not exceeding the total traffic supported by the MBS ρ_M to which the load ρ_i is transferred. This is expressed in the following optimization problem where τ_i is the trigger for MBS i, taking values 0 for OFF or 1 for ON [15].

$$\text{Maximize:} \quad \sum_{i=1}^{N_s} \tau_i \left[\rho_i - T\right] \tag{3}$$

$$\begin{aligned} \text{Subject to:} \quad & \rho_M + \textstyle\sum_{i=1}^{N_s} \rho_i\big(1-\tau_i\big) \leq 1 \\ & 0 \leq \rho_M \leq 1 \\ & 0 \leq \rho_i \leq T, \qquad\qquad i \in 1,2,3,...N_s \\ & \tau_i \in \{0,1\} \qquad\qquad i \in 1,2,3,...N_s \end{aligned} \tag{4}$$

IV. Neural Networks for Base Station Switching

In the investigated context, any proposed solution must estimate the traffic at the next time slot (the next hour) as well as decide the best switching combination as in the optimization problem presented in Section III. Furthermore, given the limited amount of available data, the estimation can only be short term. Considering the available options, we decided to experiment with two ML approaches. In the first case, the ML algorithm outputs the estimated traffic demand samples, which are used to take switching decisions (two-step approaches). In the second case, the ML algorithm performs both estimation and switching decisions together (unified approaches): the output of the ANNs is the configuration of the RAN. For both types of approaches, two types of ANNs were tested, Dense Neural Networks and Recurrent Neural Networks, resulting in a total of 4 solutions, that we further explain in the subsections that follow.

A. Dense Neural Networks for Estimation (Two-Step-DNN)

When using a DNN for estimation, a single ANN is trained to estimate the traffic demand of the considered portion of a HetNet composed by one MBS and 6 MBSs.

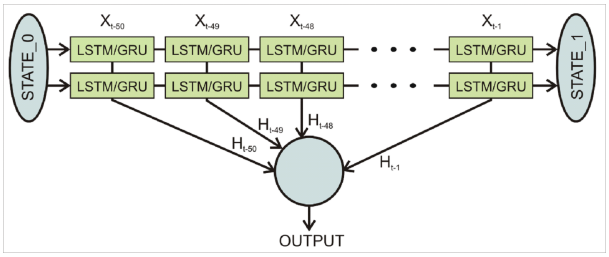

Fig. 4: The unwrapped form of the Recurrent Neural Network for Two-Step-RNN

In this case, the ANN receives as input the past traffic demand on a given BS at times $t-1$, $t-24$, $t-1-24$, $t-2\cdot 24$, $t-1-2\cdot 24$ and provides as output the forecast traffic demand at time t on that BS.

We decided to use these five features as input, since we expect them to be sufficient to provide reliable estimates without risking overfitting the model by inputting too much information [13]. The architecture consists of an input layer, two hidden layers, and a single output neuron layer, as shown in Fig. 3. Each neuron is activated by a sigmoid function, with the exception of the last output neuron which is left unrestricted. As a standard procedure, back-propagation of a gradient descent optimizer is used for training, and the squared difference in cost between the estimate and the desired output is taken as the error.

B. Recurrent Neural Networks for Estimation (Two-Step-RNN)

In this case, a single model is employed as a Milan-wide estimator as it is trained on all areas. A three dimensional matrix is fed as input, whose size is computed as [batch-size · number-of-batches, sequence-length, feature-size]. The first value of batch-size · number-of-batches is generally variable, to allow for flexibility during training, while the sequence length and the feature size define the architecture of the RNN. The sequence length is set to 50, aiming to fit each consecutive time slot in the past, up to the one where estimation is needed; this makes the oldest value at the RNN input the same as in the DNN above [13]. As the feature-size is inherently 1, the load of the BS we are estimating, the final matrix has a size of $[\cdot, 50, 1]$, leading to the architecture shown in Fig, 4. This architecture is created by passing the mentioned matrix through a two-layered LSTM or GRU cells with size 200, where the output of each cell from the second layer is tied to an output neuron deciding the final value.

The internal states of the RNN start with an input of all zeros in state_0 and ends with state_1. The output state is non-zero and carries useful information. Even though sequence-length is a finite number, state_1 can be exploited by implementing a fully tied state passing mechanism to pass some information through the RNN indefinitely. However, such approach of state linking follows very strict rules regarding batching, where every new batch must be the next-in-line for estimation.

In more detail, when training with batch number i, we save the corresponding output state state_i. This state_i is then used as input (the state_0) of the next batch $i+1$. The next in line training batch $i+1$ will be able to utilize state_i as a starting non-zero state leading to better estimations for the output value. However, this does not guarantee that each passed state is fully useful indefinitely. Unfortunately, it cannot carry information about a phenomenon for which it has not been trained within the band of 50 sequential inputs that are given to the RNN at each step.

C. Dense Neural Networks for Full Decision (Unified-DNN)

The simultaneous traffic estimation and switching decision is an inherently more complex problem than only estimation. Hence, a more careful approach is taken towards both the architecture of the ANN and the data feeding it.

In this case, a single model is trained to get the configuration of the cluster. When the configuration of the network at time t is estimated, the machine receives as inputs the past traffic demand and the status (ON/OFF) at $t-1$, $t-24$, $t-1-24$, $t-2\cdot 24$, $t-1-2\cdot 24$, of each SBS. In addition, the traffic demands at $t-1$, $t-24$, $t-1-24$, $t-2\cdot 24$, $t-1-2\cdot 24$, of the MBS are also used as input features.

The internal network consists of 3 hidden layers with 65 neurons each; in this way we try to obtain a balance between the power of the network and the time to convergence towards usable results. The final output layer consists of twelve output neurons. Each neuron is used to express the certainty of the DNN in deciding towards a switching decision. If output O_i has larger value than output O_{i+6}, the SBS i will be kept on. Otherwise, it will be turned off, as shown in (5), where the value th is an offset which allows to induce a bias towards keeping the SBS, on or turning it off.

$$SBS_i = \begin{cases} ON, & O_i \geq O_{i+6} - th \\ OFF, & Oi < O_{i+6} - th \end{cases} \tag{5}$$

D. Recurrent Neural Networks for Full Decision (Unified-RNN)

This unified solution is a slight modification of the RNN used for estimation, with the difference that it is capable to decide if a SBS needs to be turned off or kept on. It was implemented with input dimensions of $[\cdot, 50, 7]$, because each element in the sequence-length of 50 comprises 7 features for the load of each BS in a network scenario. The output consist of 12 neurons to present the best switching optimization, like in the Unified-DNN implementation. Finally, as in Unified-DNN, the main expected benefit of using a unitary solution is the possibility of scaling the output as in (5), by manipulating th.

V. Testing and Results

For each BS, the available data was divided in two parts which were never mixed: the first 1127 timeslots were used for training, and the final 336 hours for testing. In addition to the use of Dropout [14], overfitting was carefully avoided by following the evolution of the performance metrics during training, when fed with both the training and the testing sets. The testing was done on chosen scenarios where the number of SBSs under one MBS is 6. As congested areas are of the highest interest 11 different scenarios in the metropolitan area were considered: *Residential* (a residential area in the south-west), *Business* (a business area near the city center), *Polimi* (the "Politecnico di Milano"), *Highway* (a busy section of the south-western highway), *FS* (the main railway station "Milano Centrale"), *San Siro* (the occasionally very busy football stadium of San Siro), *Residential2* (a residential area in the north), *Duomo* (the city center), *Industrial* (a central industrial neighbourhood), *Linate* (the Linate airport), and *Rho* (the Rho neighbourhood); all highlighted in Figure 5.

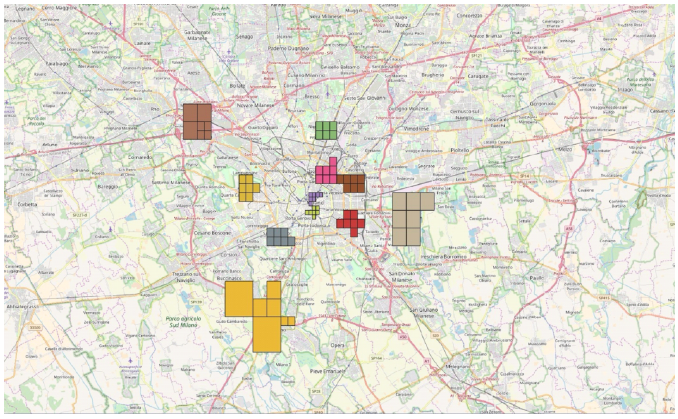

Fig. 5: The location of the 11 tested scenarios

The key metrics for measuring the performance of all solutions are *lost coverage* and *lost potential efficiency*. The lost coverage, denoted with C_L, measures the percentage of available traffic that is lost due to SBS in sleep mode. The lost potential efficiency E_L measures the percentage of extra energy spent relative to the whole potential energy saving in the optimal scenario. Note that the $E_L = 0$ is not an asymptote of the system, because losses in coverage can lead to better than ideal energy efficiency resulting in negative numbers for E_L.

Note that the available BS capacity is such that an optimization algorithm fed with all past and future data would be able to achieve $C_L = E_L = 0$. In the case of realistic assumptions, the quality of an algorithm is measured by the values achieved by C_L and E_L. Note however that, since our goal is to provide the best possible energy efficiency while not significantly affecting the QoS in the area, we strive to first achieve very low values for C_L.

A. Testing Two-Step Solutions

We first test the performance of the two-step solutions, where an ANN is only used for traffic estimation, and switching decisions are optimally derived from the ANN estimations, using the optimization problem in (3).

The application of the Two-Step-DNN to our traffic data for each of the 11 scenarios provides the results shown in

Two-Step-DNN		
	Efficiency Loss %	Coverage Loss %
Business	0.94	1.00
Duomo	0.56	1.09
FS	3.66	0.62
Highway	2.33	1.00
Industrial	0.51	1.17
Linate	1.07	1.39
Polimi	2.71	0.72
Residential	3.44	1.12
Residential2	2.54	0.90
Rho	0.41	1.25
SanSiro	-0.78	1.08

Fig. 6: The performance of the Two-Step-DNN implementation for each of the 11 tested scenarios

Two-Step-RNN		
	Efficiency Loss %	Coverage Loss %
Business	0.71	0.72
Duomo	0.54	0.78
FS	2.08	0.47
Highway	1.64	0.83
Industrial	0.75	0.70
Linate	1.68	1.18
Polimi	1.41	0.65
Residential	3.15	0.82
Residential2	2.64	0.85
Rho	-0.18	0.96
SanSiro	-0.81	1.23

Fig. 7: The performance of the Two-Step-RNN implementation for each of the 11 tested scenarios

Unified-RNN											
	1	2	3	4	5	6	7	8	9	10	11
0											
CL	2.01	1.57	1.26	1.7	0.71	1.74	2.15	1.39	1.49	3.43	2.89
EL	-0.06	-2.64	-3.15	-0.49	3.76	-1.41	-2.44	-0.52	-2.74	-5.08	-3.83
0.1											
CL	1.6	1.21	1.03	1.43	0.58	1.51	1.6	1.06	1.42	2.83	2.69
EL	2.92	-0.46	-1.42	1.57	5.14	-0.89	0.64	1.51	-1.78	-3.09	-3.26
0.2											
CL	1.17	0.86	0.79	1.13	0.41	1.12	1.21	0.77	1.2	2.25	2.46
EL	6.13	2.19	0.06	3.65	7.17	-0.18	3.07	2.99	-0.14	-0.59	-2.45
0.3											
CL	0.86	0.64	0.5	0.93	0.29	1.03	0.89	0.63	1	1.71	1.87
EL	9.24	3.63	2.8	5.44	9.32	0.32	5.93	5.07	1.46	1.59	-1.04
0.4											
CL	0.48	0.47	0.45	0.74	0.24	0.8	0.55	0.43	0.72	1.39	1.42
EL	13.04	4.95	4.54	7.93	11.02	0.94	8.88	6.78	3.44	3.59	0.05
0.5											
CL	0.24	0.31	0.39	0.67	0.16	0.66	0.41	0.3	0.49	0.94	1.21
EL	16.53	7.63	6.07	9.33	13.56	1.57	11.31	8.57	6.32	7.04	1.4
0.6											
CL	0.18	0.24	0.29	0.51	0.11	0.47	0.33	0.23	0.36	0.53	0.99
EL	20.43	9.58	8.93	12.09	15.27	2.64	14.49	11.4	8.87	11.52	2.96
0.7											
CL	0.1	0.06	0.15	0.22	0.09	0.41	0.09	0.14	0.19	0.33	0.73
EL	24.22	13.39	12.68	15.35	18.34	4.13	19.02	13.69	13.42	15.21	4.8
0.8											
CL	0.07	0.04	0.13	0.05	0.09	0.19	0	0.09	0.04	0.21	0.39
EL	28.58	18.12	17.62	19.14	20.52	5.79	25.25	17.84	20.38	20.13	8.76
0.9											
CL	0.07	0	0.02	0	0.07	0.05	0	0	0.01	0.1	0.18
EL	35.03	26.38	26.46	29.58	26.29	9.33	35.6	23.99	35.69	27.44	17.92

Fig. 8: The performance of the Unified-DNN implementation for each of the 11 tested scenarios, for each threshold setting from 0 to 0.9

Fig. 6. We can see that the area with the lowest C_L is the scenario over the main train station of Milan (labeled FS), which exhibits a very regular traffic pattern, so that predictions are easier. Whether this value is acceptable depends on the MNO policies, but in general we aim at lost coverage values lower that 0.1 %, even sacrificing somewhat the lost potential efficiency. The overall average performance metrics obtained with the Two-Step-DNN, considering all 11 scenarios, are $E_L = 1.58$, $C_L = 1.03$. We point out that the scenarios where performance is worse are Rho, Linate and San Siro, since their traffic pattern is irregular.

The application of the Two-Step-RNN yields the performance values presented in Fig. 7. The overall averages of the two performance metrics are: $E_L = 1.24$, $C_L = 0.84$. These two values represent the closest pair to the (0,0) point achieved in our study. In all scenarios the energy efficiency loss is always below 3.5%, while the lost coverage varies between 0.47% and 1.23%. Also in this case, the three worst performing scenarios are San Siro, Linate and Rho.

B. Testing Unified Solutions

The main expected benefit of using unitary solutions is the possibility of considering the output directly as switching decisions. In addition, these solutions allow the use of the offset th to drive the final performance of the switching scheme. Note that in Figures 8 and 9 column numbers correspond to: 1 Residential, 2 Business, 3 Polimi, 4 Highway, 5 FS, 6 SanSiro, 7 Residential2, 8 Duomo, 9 Industrial, 10 Linate and 11 Rho.

Fig. 8 reports the E_L and C_L values for the Unified-DNN implementation over all areas (identified by columns), varying the offset value from 0.0 to 0.9. It is possible to notice that when the offset increases, lower values of C_L are achieved at the cost of higher values of E_L. When the values of the offset are large, the SBSs tend to be kept ON, resulting in higher energy consumption, but lower lost traffic. Three scenarios achieve less than 0.1% coverage loss with acceptable loss in energy efficiency, when the offset is 0.9. The overall average performance with offeset 0.9 is $E_L = 37.48$ and $C_L = 0.02$. The value of the average energy efficiency loss can be considered acceptable, given the almost perfect performance in terms of coverage.

When we consider the performance of the unified-RNN, reported in Fig. 9, it possible to notice that it fails to provide less than 0.1% coverage for the case of Rho, with any offset value among those we considered. Nevertheless, with offset equal to 0.9, the average across all scenarios is below 0.1%; indeed, we get: $E_L = 26.7$, $C_L = 0.05$.

Finally, Figure 10 reports for the considered algorithms C_L versus E_L. The 2 considered two-step solutions (DNN and RNN) are shown as points, whereas the 2 considered unified solutions are shown as curves, which are generated by varying the offset value from 0 to 0.9. In the plot, the Two-Step-RNN

Unified-DNN											
	1	2	3	4	5	6	7	8	9	10	11
0											
CL	1.87	1.45	1.06	1.72	0.85	1.29	1.84	1.39	1.28	2.81	1.99
EL	0.49	-0.54	-1.56	-0.82	2.46	-0.81	-0.44	0.24	-0.56	-2.83	-1.9
0.1											
CL	1.47	1.11	0.8	1.19	0.75	1.15	1.32	1.08	1.2	2	1.78
EL	3.98	1.41	0.36	1.69	3.83	-0.36	3.04	2.42	0.51	-0.35	-1.22
0.2											
CL	0.96	0.86	0.66	0.78	0.53	0.96	1.02	0.94	0.92	1.67	1.64
EL	7.59	3.76	2.58	4.33	6.65	0.15	5.82	3.97	2.1	1.63	-0.38
0.3											
CL	0.73	0.61	0.47	0.58	0.42	0.76	0.82	0.68	0.74	0.89	1.25
EL	10.17	6.03	5.86	5.9	8.58	0.71	8.92	6.68	3.62	5.14	0.94
0.4											
CL	0.52	0.51	0.34	0.44	0.36	0.68	0.44	0.55	0.63	0.76	1.03
EL	13.81	7.59	8.48	8.07	10.71	1.22	11.84	9.14	5.67	7.02	2.14
0.5											
CL	0.23	0.44	0.25	0.26	0.18	0.51	0.25	0.43	0.5	0.51	0.75
EL	17.47	9.34	11.38	10.89	13.7	1.99	15.31	11.49	8.11	9.81	3.65
0.6											
CL	0.13	0.31	0.14	0.19	0.1	0.4	0.17	0.35	0.28	0.25	0.58
EL	21.57	12	14.12	13.67	17.07	2.68	20.03	14.13	12.77	13.44	5.52
0.7											
CL	0.08	0.11	0.1	0.04	0.08	0.27	0.11	0.18	0.13	0.13	0.22
EL	25.65	17.08	17.89	18.7	19.68	4.33	26.02	17.62	19.14	17.63	8.41
0.8											
CL	0.02	0.05	0.1	0.02	0.07	0.23	0.05	0.16	0.01	0.09	0.12
EL	34.07	23.95	22.53	28.22	24.85	7.08	34.58	22.99	32.33	25.57	13.11
0.9											
CL	0	0	0.05	0.02	0.06	0	0	0.04	0	0.01	0.02
EL	45.77	36.14	30.55	44.87	36.71	13.63	43.68	33.88	52.31	47.3	27.4

Fig. 9: The performance of the Unified-RNN implementation for each of the 11 tested scenarios, for each threshold setting from 0 to 0.9

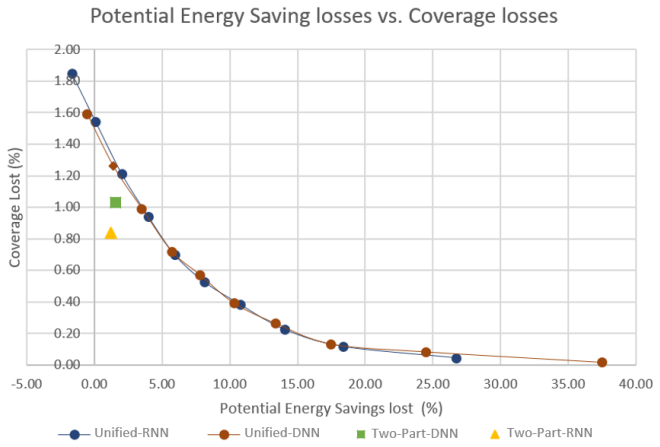

Fig. 10: C_L versus E_L; the two-step solutions are shown as points, whereas the unified solutions are shown as curves generated by varying the offset value from 0 to 0.9

shows the power of RNNs when used for simple time series estimation by being the algorithm with the best efficiency to coverage ratio.

Although the unified algorithm's ratios are not outperforming the Two-Step-RNN, unified solutions offer the added flexibility of being adjustable, thanks to the introduction of the offset, which permits giving weight to Coverage loss. We also see that both unified scenarios perform similarly. As regards implementation complexity, we wish to remark that Unified-DNN solutions are the least computing intensive among the considered alternatives.

VI. Conclusions

In this paper we investigated the effectiveness of four different classes of ANNs in the short-term estimation of the traffic load of BSs in HetNets, with the objective of creating on-line sleep algorithms for energy saving.

Two of the considered ANNs only deal with traffic estimation, and BS switching decisions are based on their outputs. The other two ANNs deal both with traffic estimation and switching decisions, so their output can be directly used to drive the HetNet management algorithm.

All the considered options proved quite effective, but with the second class of ANNs we achieved the goal of saving a substantial amount of energy with minimal QoS reduction. In particular, the amount of served traffic is at least 99.9% of the requests, while achieving 63% of the possible energy savings.

References

[1] M. Gupta and S. Singh, "Greening of the Internet," Sigcomm 2003, Karsruhe, Germany, August 2003.

[2] Z. Hasan, H. Boostanimehr and V. K. Bhargava, "Green Cellular Networks: A Survey, Some Research Issues and Challenges," in IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 524-540, Fourth Quarter 2014

[3] Budzisz et al., "Dynamic Resource Provisioning for Energy Efficiency in Wireless Access Networks: A Survey and an Outlook," in IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2259-2285, Fourthter 2014.

[4] A. De Domenico, E. Calvanese Strinati, A. Capone, Enabling Green Cellular Networks: A Survey and Outlook, Computer Communications, vol. 37, 1 January 2014, pp. 524

[5] Index, Cisco Visual Networking. "Global mobile data traffic forecast update, 2016-2021." (2017).

[6] Index, Cisco Visual Networking. "Forecast and Trends, 2017-2022." (2018).

[7] M. Ajmone Marsan, L. Chiaraviglio, D. Ciullo and M. Meo, "Optimal Energy Savings in Cellular Access Networks," 2009 IEEE International Conference on Communications Workshops, Dresden, 2009, pp. 1-5.

[8] Lpez-Prez, David, et al. "Towards 1 Gbps/UE in cellular systems: Understanding ultra-dense small cell deployments." IEEE Communications Surveys & Tutorials 17.4 (2015): 2078-2101.

[9] Han, Fengxia, et al. "Survey of strategies for switching off base stations in heterogeneous networks for greener 5G systems." IEEE Access 4 (2016): 4959-4973.

[10] Zeng, Ruhong, et al. "An artificial neural network based cell switch-off algorithm in cellular system." Computer and Communications (ICCC), 2016 2nd IEEE International Conference on. IEEE, 2016.

[11] Wang, Luhao, Shuang Chen, and Massoud Pedram. "Context-driven power management in cache-enabled base stations using a Bayesian neural network." 2017 Eighth International Green and Sustainable Computing Conference (IGSC). IEEE, 2017.

[12] Auer, Gunther, et al. "D2. 3: Energy efficiency analysis of the reference systems, areas of improvements and target breakdown." Earth 20.10 (2010).

[13] Lee, K. Y., Y. T. Cha, and J. H. Park. "Short-term load forecasting using an artificial neural network." IEEE Transactions on Power Systems 7.1 (1992): 124-132.

[14] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Lear.1 (2014): 1929-1958.

[15] Donevski I. "Energy Saving in Cellular Base Stations Under the Control of Supervised Neural Networks", Master thesis, Politecnico di Torino, Dipartimento di Elettronica e Telecomunicazioni (DET), Turin, Italy, 2018