# Multimedia Services Placement Algorithm for Cloud-Fog Hierarchical Environments

Fillipe Santos[a], Roger Immich[b], Edmundo R. M. Madeira[a]

[a] *University of Campinas. Campinas, SP, Brazil*
[b] *Federal University of Rio Grande do Norte Natal, Brazil*

**Abstract**

With the rapid development of mobile communication, multimedia services have experienced explosive growth in the last few years. The high quantity of mobile users, both consuming and producing these services to and from the Cloud Computing (CC), can outpace the available bandwidth capacity. Fog Computing (FG) presents itself as a solution to improve on this and other issues. With a reduction in network latency, real-time applications benefit from improved response time and greater overall user experience. Taking this into account, the main goal of this work is threefold. Firstly, it is proposed a method to build an environment based on Cloud-Fog Computing (CFC). Secondly, it is designed two models based on Autoregressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM). The goal is to predict demand and reserve the nodes' storage capacity to improve the positioning of multimedia services. Later, an algorithm for the multimedia service placement problem which is aware of data traffic prediction is proposed. The goal is to select the minimum number of nodes, considering their hardware capacities for providing multimedia services in such a way that the latency for servicing all the demands is minimized. An evaluation with actual data showed that the proposed algorithm selects the nodes closer to the user to meet their demands. This improves the services delivered to end-users and enhances the deployed network to mitigate provider costs. Moreover, reduce the demand to Cloud allowing turning off servers in the data center not to waste energy.

*Keywords:* Cloud-to-Fog Networks, Multimedia Services, Placement Strategies

## 1. Introduction

In recent years there has been a rapid proliferation of a wide range of real-time multimedia services, such as video on demand, video conferencing, broadcast of interactive 3D environments, high definition videos, streaming video with 4k/8k resolution Ultra-high-definition video (UHD), among others [1]. These services already account for the majority of global traffic and by 2021/2022 will flood mobile networks requiring unprecedented high speed and low latency [2, 3].

According to technical reports provided by Cisco Systems [4], 73% of all global IP traffic generated on the Internet in 2019 was related to video traffic over IP and 1% related to gaming traffic, with projections that these percentages will be 82% and 4%, respectively, for the years 2021/2022. In fact, the adoption of the 5th Generation of Wireless Systems (5G) will allow this growth to be even greater due to its high bandwidth capacity and low latency. Also, the most recent studies on consumer habits during the pandemic, and what may remain later, show that multimedia service traffic peaked at $\approx 60\%$ higher than January levels when the lockdowns started in some countries [5, 6].

Notwithstanding the numerous advantages that CC offers, such as scalability, security, and flexibility, multimedia services require constant, continuous flow of packets with low latency [7], requirements that it occasionally does not provide [8].

Adapting these services in this environment is a non-trivial task [9]. By the way, even with the improvements in wireless technologies offered by the 5G, reliable, high-quality video delivery still poses several challenges, such as dealing with a large number of heterogeneous devices and meeting the increasing requirements of users. To overcome these and other problems, it is desirable to use a distributed architecture that stores and processes services logically between the Cloud and the data source [10].

FG and Edge Computing (EC) present themselves as a joint solution to meet these latency-sensitive services, where the management of all resources occurs in a coordinated and tiered manner, from the Cloud to the end devices. The main goal is to allocate Cloud resources physically closer to end users [11]. These architectures share similar benefits compared to CC, including a reduction in latency to milliseconds, decreased network congestion, and real-time recognition of users' geographic location [12].

Nodes are hierarchically organized in tiers in these environments, from the Cloud to the end-users. They are also namely fog or edge nodes. These nodes are infrastructures that can provide resources for services that can be

*Corresponding author
*Email addresses:* fillipe@lrc.ic.unicamp.br (Fillipe Santos), roger@imd.ufrn.br (Roger Immich), edmundo@ic.unicamp.br (Edmundo R. M. Madeira)

executed in a distributed and independent way as microservices, available closer to end-users [13]. On the one hand, nodes belonging to the same tiers have similar networks (latency, download, and upload) and computational (storage and processing) resources. On the other hand, nodes of different tiers have distinct features. The non-availability of these environments for simulation makes the evaluation of these algorithms a challenge. Also, the hierarchical, distributed, and heterogeneous nature of computational instances makes the positioning of the multimedia services in these environments a challenging task [14, 15].

Placing multimedia services in these environments can be related to the Facility Location Problem (FLP) [16]. In short, this problem refers to optimal placement of facilities (resources) that an organization has to meet the clients' requirements so while considering a set of constraints like the distance between resources and clients or competitors' facilities [17]. In this problem, the facilities may or may not have capacity constraints for storage, which categorize the problems into capacitated and uncapacitated variants [18]. The Capacitated Facility Location Problem (CFLP) is the basis for many practical applications, where the facilities have a limit on the number of customers it can serve. For an Uncapacitated Facility Location Problem (UFLP) however, the assumption made is that an arbitrary number of customers can be connected to a facility.

To further improve the service delivery, algorithms based on traditional and deep learning models, such as ARIMA and LSTM, can be adopted to extract features from the telecommunication activity dataset and find correlations among them. The goal is to predict future demand and reserve the nodes' storage capacity to improve the positioning of multimedia services. The ARIMA model is used to understand time series or predict a point in the future. Otherwise, LSTM model is a special type of Recurrent Neural Networks (RNN) capable of learning long-term dependencies [19].

In previous work [20], we have introduced a novel algorithm to the Multimedia Microservices Placement Problem (MMPP) modeled as CFLP. We extend this algorithm by considering the network traffic prediction in this work. All contributions are summarized below:

- The design and implementation of a new method to build an environment based on CFC. Nodes are organized hierarchically in tiers in this environment, from Fog to Cloud.
- It is designed two models based on ARIMA and LSTM to solve the traffic forecasting problem. The goal is to predict demand and reserve the nodes' storage capacity to improve the positioning of multimedia services.
- An algorithm for the mul**T**imedia serv**I**ce **P**lacemen**T** problem based on facility l**O**cation aware of data traffic **P**rediction (TIPTOP).The goals are to select the minimum number of nodes, considering their hardware capacities for providing multimedia services in such a way that the latency for servicing all the demands is minimized.

The performance assessment was carried out in MultiTierFogSim using two months of real-world mobile network traffic data in Milan, Italy. First, the proposed algorithm is evaluated considering six snapshots selected in particular days and hours based on the traffic intensity to assess the performance of the Multimedia Service Placement Problem based on Facility Location (SMART-FL) algorithm in different circumstances. Later, the performance assessment is based on predicted mobile traffic one month. In this case, the results are compared considering four strategies to place multimedia services and are evaluated in terms of latency, package delivery, attempted requests, and network usage.

The results show that the proposed algorithm can balance the fog nodes' geographical location, hardware capacity, and the users' location. The positioning becomes more efficient through data traffic prediction due to previously reserved nodes' storage. Furthermore, using the information obtained in this work, it is possible to implement a strategy for shutting down servers in the Cloud to save energy. It is worth mentioning that, the proposed scheme can be adapted for other services, e.g., Traffic Control System (TCS), Internet of Things (IoT) applications, augmented reality, and others that require CC, FG, and EC as well as service migration.

The remainder of this work is organized as follows. Section 2 gives an overview of the main related work. Section 3 presents the design of Cloud-Fog hierarchical environments. Section 4 presents the formulation for the multimedia services placement problem. Section 5 describes how the prediction models are analyzed and implemented. Section 6 details the experimental method and results. Finally, Section 7 presents the conclusion and future work.

## 2. Related Work

There are several proposals in the literature to optimize multimedia delivery over hierarchical networks. This section is divided into two parts to cover all aspects of this work, namely Cloud-Fog hierarchical environments (2.1) Multimedia services placement in Cloud-Fog hierarchical environments (2.2).

### *2.1. Cloud-Fog hierarchical environments*

Cloud-Fog hierarchical environments have also been proposed to optimize various issues related to this domain, from identifying the most appropriate grouping of base stations to share Cloud resources or even minimizing the distance between servers and access points throughout the city [21, 22].

A proposal for positioning fog nodes to reduce the costs associated with their deployment and maintenance considering variable demands in time is proposed in [16]. The

set of selected nodes compose the hierarchical environment. Based on actual data, the results show that there is an improvement in end-user service that can be achieved in conjunction with minimizing costs by deploying a smaller number of servers in the infrastructure. Besides, costs can be reduced further if a limited blocking of requests is tolerated. However, like previous work, the type of workload is simulated, and the number of tiers and nodes are limited by one and three, respectively.

A solution to the base station grouping problem for sharing Cloud Radio Access Network (C-RAN) resources is proposed in [21]. The solution aims to group neighboring base stations with complimentary traffic patterns so that the traffic volume processed in C-RAN is balanced, requiring fewer resources. The results show that this collation scheme reduces deployment cost by 12,88%. The data set used was made available by Telecom Italia [22]. However, the number of tiers and nodes are limited by two and four, respectively. In our proposed method, the number of tiers varies.

A framework for partitioning a set of base stations into groups and processing the data in a shared data center is proposed in [23]. This partitioning and scheduling framework saves up to 19% of computing resources for a one in 100 million probability of failure. However, the adoption of only one data center can result in delays between the distant base stations and the data center. Also, the type of workload is unrealistic, and the number of tiers and nodes are limited by two and five, respectively. Further, it is not considered balanced workload and location preference. In contrast, we experiment with publicly available real-world mobile traffic data set and, as mentioned, the number of tiers varies, taking into account the balanced workload and location preference.

The placement of edge servers considering capacity constraints is modeled as a Capacitated Location-Allocation problem by [24]. The goal is to minimize the distance between servers and access points throughout the city. The results show that the proposed algorithm can provide optimal solutions that minimize distances and offer a balanced workload with sharing according to node capacity constraints. As in previous work, they used an unrealistic data set and the number of tiers is limited to seven, without balanced workload and location preference.

A proposal for locating fog nodes with limited battery support for mobile users capable of processing high demands with low latency restrictions is proposed by [25]. The conclusion is that the heuristic solution produces accurate results when compared to data generated by the Integer Linear Programming (ILP), thus allowing significant energy savings for end-users. However,the number of tiers is limited to seven, without balanced workload and location preference.

In this work, a method is proposed to creating Cloud-Fog hierarchical environments. The method uses a bottom-up approach, starting from a set $BS = \{bs_1, bs_2, ..., bs_{bs}\}$ of base stations and organizing new nodes hierarchically into tiers, producing a hierarchical Cloud-Fog environment. Fog nodes are grouped to facilitate resource pooling and work collaboratively, reducing effort in obtaining an optimal node towards service deployment. These groups of fog nodes are also linked to enable service migration whenever necessary.

Table 1 lists the related works and classifies their contributions with respect to five characteristics. The first column represents the related works. The second column represents the type of workload. The third column refers to the number of tiers. The fourth column shows the number of nodes able to services. The fifth column refers to if the work considers balanced workload. Finally, the sixth column indicates if the work considers location preference.

**Table 1:** Comparison table of related works.

| *Research work* | *Workload* | *Number of tiers* | *Number of nodes* | *Balanced workload* | *Location preference* |
|---|---|---|---|---|---|
| C da Silva el at. [16] | Simulated | 1 | 3 | ✓ | ✓ |
| Chen el at. [21] | Phone call | 2 | 4 | ✓ | ✓ |
| Bhaumik el at. [23] | Simulated | 2 | 5 | × | × |
| Lähderanta el at. [24] | Simulated | 7 | 4 | × | × |
| Silva el at. [25] | Phone call | 7 | N | × | × |
| **Proposed solution** | Phone call | N | N | ✓ | ✓ |

### *2.2. Multimedia services placement in Cloud-Fog hierarchical environments*

Many research efforts have been conducted to address the issue of reducing latency to deliver multimedia services in the context of CC, FG, and EC. They range from changes in the Cloud architecture to the use of fog/edge nodes to reduce network delay and improve Quality of Experience (QoE) [26, 27].

Table 2 lists the related works and classifies their contributions with respect to five characteristics. The first column represents the related works. The second column refers to if the work was performed in a Cloud-Fog hierarchical environment. The third column shows the number of nodes able to deploy multimedia services. The fourth column refers to if the work considers the nodes' hardware capacity. The fifth column indicates if the work considers multiple requests for multimedia services. Finally, the sixth column indicates if the work considers network traffic prediction. The symbol $'?'$ means that the authors did not include the information.

A Quality of Service (QoS)-aware service allocation problem for Cloud-Fog architectures as an integer optimization problem was proposed in [28]. The technique combines

**Table 2:** Comparison table of related works.

| Research work | Cloud-Fog hierarchical environments | Number of nodes | Node capacity | Multiple requests | Network traffic prediction |
|---|---|---|---|---|---|
| Souza el at. [28] | Cloud/Fog | 7 | ✓ | × | × |
| Fang Shi al. [26] | Cloud/Fog | ? | ✓ | ✓ | × |
| Kharel et al. [27] | Cloud/Fog | 84 | ✓ | × | × |
| Kryftis et al. [29] | Cloud | ? | × | × | ✓ |
| Mahmud et al. [30] | Fog | 4-10 | ✓ | ✓ | × |
| Sai et al. [31] | Cloud/Fog | 4 | ✓ | ✓ | × |
| **Proposed solution** | Cloud/Fog | 1160 | ✓ | ✓ | ✓ |

the Cloud-Fog operations and can accomplish high system capacity, granting low latency for requested services. The hierarchy of a layer is determined by capacity, vicinity, and reachability to end-users. The authors concluded that service distribution benefits among multi-tier fog nodes because they avoid the high delay access on the cloud layer. Nevertheless, the effect on time overhead created by the service distribution with a vast number of fog nodes for mobile users is not considered. In normal conditions, this may be analyzed. However, in an unusual situation, this could be a problem. For example, resources can be consumed by a large group of users. The proposed algorithm has a time and space complexity of $\mathcal{O}(mn \log \frac{n^2}{m} + nlog(m))$ and $\mathcal{O}(n+m)$, respectively; where $n$ and $m$ are the services requiring set size and IoT nodes, respectively.

A hierarchical content delivery network in a randomly distributed interference environment was proposed by [26]. The optimization problem is modeled as a combinatorial optimization problem. The authors concluded that the proposed algorithms improve the users' cache hit probability and provide more flexible cooperative transmission opportunities. However, a collaborative resource strategy in multi-tier fog nodes receives more attention. Also, the user preference model was not validated. The proposed algorithm has a time and space complexity of $\mathcal{O}(mn \log \frac{n^2}{m} log(m))$ and $\mathcal{O}(n+m)$, respectively; where $n$ and $m$ are the number of areas with service requests and nodes, respectively.

A hierarchical FG-based multimedia streaming that reduces latency and minimizes internet bandwidth consumption for passengers traveling in any vehicle is proposed by [27]. The result acquired from the simulation showed that the secondary and primary fog stations located at the edge were considered the better and best-case scenario to allocate multimedia services regarding QoS and QoE. However, the proposed hierarchical FG does not consider that the number of vehicles requesting multimedia services can be more significant in practice. Also, they do not consider that the demand for multimedia services and nodes' hardware capacity vary over time. The proposed algorithm has a time and space complexity of $\mathcal{O}(m(n + \log n))$ and $\mathcal{O}(n)$, respectively; where $n$ and $m$ are the number of vehicles and nodes, respectively.

A network architecture that utilizes novel resource prediction models for optimal selection of multimedia content provision methods is proposed by [29]. The authors also present two algorithms for the delivery of these contents. The results show a reduction in congestion and an 80% success rate of the services' transmission. The prediction engine is accurate, and in general, the content delivery process benefits from using the prediction model and algorithms. However, services are offered in the Cloud, where, as already discussed, CC does not provide a satisfactory QoE in areas with high demand for these services. Besides, new prediction models have emerged, such as LSTM networks that may present greater accuracy in network traffic prediction than models analyzed by the authors. The proposed algorithm has a time and space complexity of $\mathcal{O}(m + n \log n)$ and $\mathcal{O}(n + m)$, respectively; where $n$ and $m$ are the number of clients and nodes, respectively.

A QoE-aware application placement policy for distributed FG environments is proposed by [30]. The results indicate that the policy significantly improves data processing time, resource affordability, and service quality. However, a solution for service placement considering a large volume of data in the decentralized architecture in FG can cause network congestion [32]. The proposed algorithm has a time and space complexity of $\mathcal{O}(\log \frac{B}{M}(m + n)(m + n \log n))$ and $\mathcal{O}(B + m)$, respectively; where $B$, $m$, and $n$ are the number of clients, caches, and nodes, respectively.

An efficient collaborative content delivery and caching strategies in a 5G network is proposed in[31]. They propose a cache placement algorithm based on a greedy heuristic algorithm to solve this energy efficiency problem by optimizing cache placement. The authors conclude that the proposed system reduces the number of interference and improves the system throughput. However, implementing intelligent mechanisms that assess network and node conditions for dynamical deployment of these services could be more appropriate [33]. The proposed algorithm has a time and space complexity of $\mathcal{O}(m \log n + n \log n)$ and $\mathcal{O}(n+m)$, respectively; where $m$ and $n$ are the number of caches and clusters, respectively.

This work proposes an algorithm to select the minimum number of nodes for multimedia services placement aware of traffic prediction. It is taken into consideration their hardware capacities and network traffic predicted for providing multimedia services in such a way that the latency for servicing all the demands is minimized. The proposed algorithm has a time and space complexity of $\mathcal{O}(m \log m)$ and $\mathcal{O}(m)$, respectively; where $m = n_c \times n_f$, where $n_c$ and $n_f$ are the number of regions with multimedia services requests and nodes, respectively [34]. In this work, regions are grids of $235 \times 235$ $m^2$.

## 3. The design of Cloud-Fog Hierarchical Environments

The fast-growing mobile network data traffic poses great challenges for operators to increase their data processing capacity in base stations efficiently. With the Cloud Radio Access Network (Cloud-RAN) and FG, the data processing units can now be centralized in a data center and shared among several base stations. Also, clustering base stations can reduce the deployment cost and energy consumption with complementary traffic patterns to the same data center. To evaluate the proposed algorithm and contribute to this challenge, initially it is proposed a method to build an environment based on Cloud-Fog Computing.

This environment is based on the Cloud-Fog architecture, which provides a virtualized, hierarchically organized, distributed computing platform. All nodes are a small facility that hosts dedicated servers capable of processing end-user workload. The nodes closer to end-users are expected to have a smaller capacity, increasing towards the Cloud in the infrastructure, forming a Cloud-Fog Hierarchical Environment in which any device can access the Cloud.

The proposed method uses a bottom-up approach, starting from a set $BS = bs_1, bs_2, ..., bs_{bs}$ of base stations, and arranges new nodes hierarchically, from Fog to Cloud.

Initially, it is defines $\Xi$ as the connection between the base stations. For $\Xi$, one can consider the distance $R$, signal strength, traffic similarity, etc. Let $\mathbf{G} = (V, \mathcal{E})$ be an unweighted undirected graph, where $V$ contains all base stations, i.e., $V \equiv BS$; and $\mathcal{E}$ is the set of edges that represent the communication among base stations defined by $\Xi$. Method 1 and Figure 1 describe all the steps.

The stopping criterion is defined by $\mu$ as the number of subgraphs of the penultimate tier and must always be observed at steps two and five. When it is met, an upper-tier node that connects to the lower-tier nodes is added when the stopping criterion is met.

Considering **step 1**, nodes from other providers can be added as a vertex in $\mathcal{G}$. The nodes added (**steps 2-6**) can also be considered as service providers and can be a solution to the Network Planning Problem (NPP) [35]. Also, their geographic locations are not limited to the area of the region considered. That is, these nodes may be positioned geographically in other locations.

**Method 1:** Method to build a Cloud-Fog hierarchical environment.

**Input:** $BS$, $\Xi$, $\mu$

**Output:** $\mathcal{G}$

**Stopping criterion:** The number of subgraph is $\mu$.

**Stopping criterion is met:** Add an upper-tier node that connects to the lower tier nodes.

**1** - Define an undirected unweighted graph $\mathcal{G} = (BS, \mathcal{E})$.

**2** - Detect communities in $\mathcal{G}$.

**3** - For each community, an upper-tier node is added in $\mathcal{G}$, which communicates with all the base station nodes that the community belongs to. This step ends with removing the edges between all base stations.

**4** - Add edges between all nodes in the current tier.

**5** - Detect communities and remove edges between nodes of different communities.

**6** - For each subgraph, add an upper-tier node with edges between the node and the subgraph.

**7** - Go back to step **4**.

Community structure, also called clusters, groupings, or communities, are groups of vertices that are very likely to share common properties or have similar roles in the graph [36]. The communities detected in **step 2** and **5** are represented by set $S = \{s_1, s_2, ..., s_s\}$ and can be detected by any strategy [37].

The proposed environment can be simulated in any Cloud-Fog-Edge Simulator, where it benefits from several advantages related to Cloud and Fog, such as location awareness, analysis capability for processing, as well as service migrations to and from any layer.

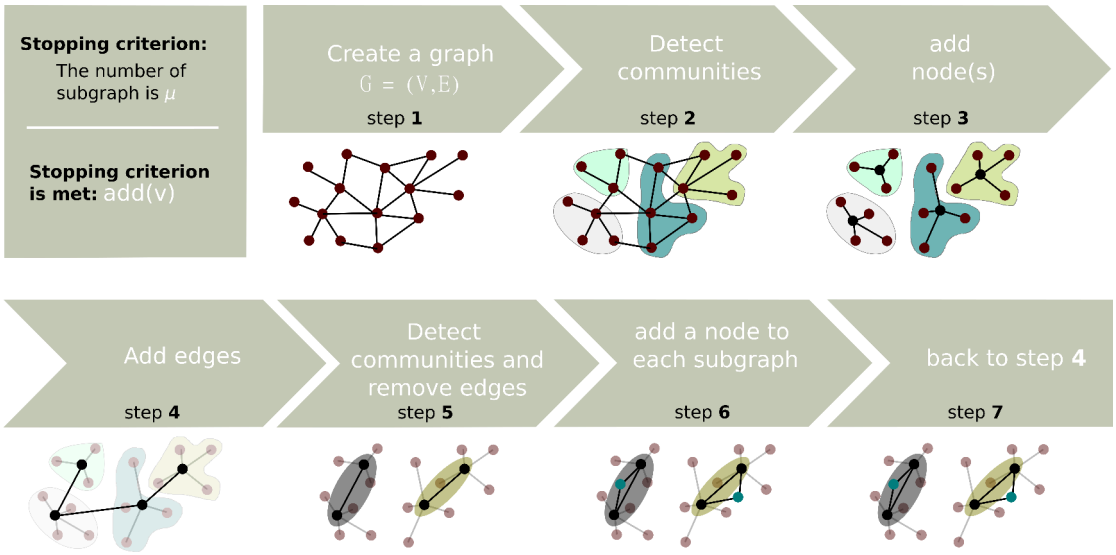



**Figure 1:** Method for design of Cloud-Fog Hierarchical Environments.

## 4. Multimedia services placement algorithm

First, the SMART-FL algorithm introduced in previous work [20] is presented. Later, an extension of this algorithm by considering the network traffic prediction is presented.

Addressing the multimedia services placement problem in Cloud-Fog hierarchical environments involves considering some specificities and criteria when designing the deployment strategies.

Regarding resource constraints, we consider that finite capabilities limit the nodes in terms of CPU, RAM, storage, and bandwidth. Let $\mathcal{L} = \{ell_1, ell_2, ell_3, ..., ell_x\}$ be the set of nodes capable of processing and storing multimedia services. This includes all nodes with maximum capacity $c_{max(\ell)}$, where $\ell \in \mathcal{L}$. Let $W = w_1, w_2, w_3, ..., w_z$ be the set of multimedia services; $U = u_1, u_2, u_3, ..., u_k$ the regions with multimedia services requests; and $k = |U|$ the number of regions with multimedia services. Then, $G_t = \{u_1^{w_1}, u_2^{w_2}, ..., u_k^{w_z}\}$ is the set of multimedia services $w \in W$ requested at time $t$ in region $u_k$;

Network links are bound by constraints such as latency, which need to be satisfied when deploying multimedia services. Variable $lat(\ell, u_k^{w_z}, t)$ is the latency offered by $\ell$ to service $w_z$ in region $u_k$ at time $t$, where $u_k^{w_z} \in G_t$. Variable $x(\ell, u_k^{w_z}, t) \geq 0$ represents the multimedia service $w$ requested in region $u_k$ processed by node $\ell$ at time $t$, where $u_k^{w_z} \subseteq G_t$.

In this way, when placing these services, we need to respect the resource requirements, i.e., ensure that the resources of the components deployed on the infrastructure nodes do not exceed their capabilities.

Therefore, a solution for the multimedia services placement problem in Cloud-Fog hierarchical environments is modeled as CFLP, namely SMART-FL, where (**i**) nodes are the potential facility sites, (**ii**) multimedia services requests are the demands, (**iii**) nodes' storage capacity and the users' demand are part of the constraint set, and (**iv**) multimedia services correspond to the type of service considered. An integer-optimization model can be specified as follows:

Minimize

$$\sum_{\ell \in \mathcal{L}} y(\ell, w_z, t) + \sum_{\ell \in \mathcal{L}} \sum_{u_k^{w_z} \in G_t} lat(\ell, u_k^{w_z}, t) \cdot x(\ell, u_k^{w_z}, t) \quad (1)$$

subject to

$$\sum_{\ell \in \mathcal{L}} x(\ell, u_k^{w_z}, t) = u_k^{w_z} \qquad \forall u_k^{w_z} \in G_t \quad (2)$$

$$\sum_{g^w \in G_t} x(\ell, u_k^{w_z}, t) \leq c_{max}(\ell, t) \cdot y(\ell, w_z, t) \; \forall \ell \in \mathcal{L} \quad (3)$$

$$x(\ell, u_k^{w_z}, t) \geq 0 \qquad \forall \ell \in \mathcal{L} \; and \; \forall u_k^{w_z} \in G_t \quad (4)$$

$$y(\ell, w_z, t) \in \{0, 1\} \qquad \forall \ell \in \mathcal{L} \quad (5)$$

$$w_z \in W \quad (6)$$

$$t \in [0, max\_simulation\_time] \quad (7)$$

The objective function 1 is composed of two parts. The first part selects nodes that minimize the associated costs. The binary variable $y(\ell, w_z, t) = 1$ indicates if multimedia service $w_z$ is deployed at node $\ell$ at time $t$, $y(\ell, w_z, t) = 0$ otherwise. The second part associates the latency and processing cost of the node $\ell$ to meet the multimedia service $w_z$ in region $u_k$ at time $t$. The constraint in Equation 2 requires that the service $w_z$ requested in region $u_k$ processed by node $\ell$ at time $t$ must be satisfied. The nodes' capacity is limited by the constraint in Equation 3. That is, if node $\ell$ is not activated, the demand satisfied by $\ell$ is zero. Otherwise, its capacity restriction is observed. Finally, the constraints in Equations 4-7 set the minimum values for the decision variables. The ILP model was coded using the Gurobi Optimizer solver [38]. Gurobi is a commercial mathematical programming solver. t is possible to implement shared-memory parallelism, which efficiently exploits any number

of processors and cores per processor. The solver uses an iterative process to converge on an optimal solution.

To place multimedia services aware of predicted mobile traffic, consider the performance of the SMART-FL algorithm. Let $t(n)$ be the current time. Let $A_{t(n)}$ be the set of nodes selected to provide multimedia services at time $t(n)$ (performed by SMART-FL algorithm). Let $P(A_{t(n+1)})$ be the set of nodes selected at $t(n)$ to provide multimedia services at $t(n+1)$, taking into account the traffic prediction at time $t(n)$ to $t(n+1)$.

Currently at $t(n+1)$, let $A_{t(n+1)}$ be the set of nodes selected at $t(n+1)$ to provide multimedia services at $t(n+1)$. Thus, $Y = A_{t(n)} \cap P(A_{t(n+1)})$ contains the **reserved nodes** to provide multimedia services at $t(n+1)$, case $Y \neq \{\emptyset\}$. Therefore, $\Gamma = Y \cup P(A_{t(n+1)})$ contains the **adequate nodes** to provide multimedia services at $t(n+1)$.

This means that the resources available by the nodes that provide multimedia services and that will be concluded at $t(n)$ belonging to $Y$ set are reserved to provide multimedia services at $t(n+1)$. Thus, multimedia services that will be provided in $Y$, requested at $t(n+1)$, will not compete for resources with concurrent services at $t(n+1)$. The advantage is that nodes in $\Gamma$ offer lower latency than nodes in $A_{t(n+1)}$. The SMART-FL algorithm multimedia aware of predicted mobile traffic is called TIPTOP.

Figure 2 illustrates an example considering two scenarios: without and with mobile traffic prediction. For the scenario without mobile traffic prediction, the nodes selected at $t(0)$ are $A_{t(0)} = \{E, C\}$. Otherwise, the nodes selected at $t(1)$ are $A_{t(1)} = \{B, C, D\}$. In this scenario, there are no reserved nodes to $t(1)$, and other services (i.e., services offered in TCS and IoT), requested at $t(1)$, are provided by $E$ node.

For the scenario with mobile traffic prediction, the selected nodes are $A_{t(0)} = \{E, C\}$, $A_{t(1)} = \{B, C, D\}$, and $P(A_1) = \{E, C, D\}$ for $t(0)$, $t(1)$ and $t(1)$ (predicted), respectively. In this way, $Y = \{B, C, D\} \cap \{E, C, D\} = \{C, D\}$; $\Gamma = Y \cup P(A_{t(n+1)}) = \{C, D\} \cup \{E, C, D\}$. Therefore, $Gamma = \{E, C, D\}$.

Considering that multimedia services provided by $A_{t(0)}$ will be concluded at $t(0)$ and $Y = \{E, C\}$ contains the reserved nodes to provide multimedia services at $t(1)$, the concurrent service that would be provided by node $E$ at $t(1)$ (scenario without mobile traffic prediction) is provided by node $B$. In this way, nodes in $Gamma$ offer lower latency than nodes in $A_{t(1)}$.

Services from the concurrent class are migrated to nodes of different tiers if, and only if, the storage capacity and latency offered are appropriate for such services. Otherwise, nodes in $Y$ are not reserved.

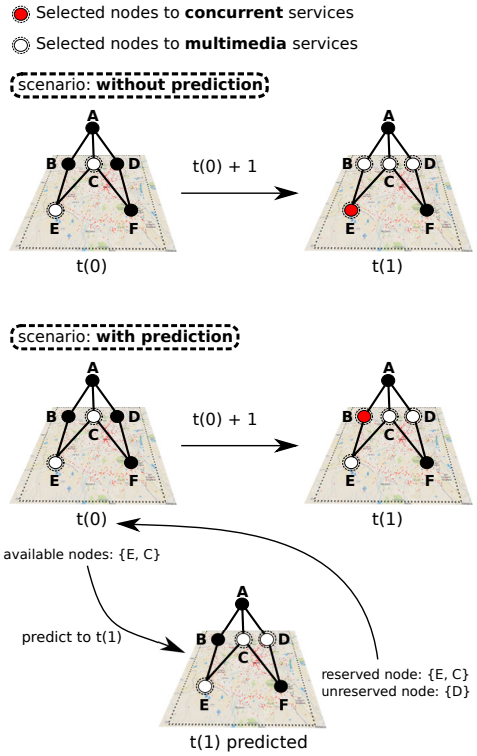



**Figure 2:** Multimedia services placement (scenario without and with mobile traffic prediction).

## 5. Improving real-time traffic forecast

Most time series prediction methods are based on the assumption that past observations contain all the information about the pattern of behavior of the time series, and that pattern is recurrent over time. In the literature, numerous methods describe the behavior of time series, and they are divided into two approaches: classical and learning [39]. On the one hand, the classical approach about the Exponential Smoothing and ARIMA methods, which use parametric statistics to transform their dataset into a known probability distribution, and therefore need to know the behavior of the time series [40]. On the other hand, deep learning models such as Gated Recurrent Unit (GRU) and RNN do not depend on prior knowledge of time series properties [41]. These models are simpler to be adjusted and demonstrate considerable performance even when applied to complex and highly non-linear series.

In this work, two models are designed, based on a classic parametric modeling ARIMA and a deep neural network architecture LSTM, namely Autoregressive Integrated Moving-prediction (ARIMA-PRED) and Long Short-Term Memory-prediction (LSTM-PRED), respectively, to solve the traffic forecasting problem. The comparisons with other baselines show the effectiveness of these methods [42, 43]. Besides, they are widely used for real-time predictions, such as forecasts of network traffic flows, been used in several studies [42, 44, 45]. In practice, the development of each regressive model differs (i) in the modeling of the time se-

ries for a set appropriate to the model (ii) and the choice of parameters most appropriate for each type of approach [46]. Performance evaluation for both models are discussed in Section 6.

### *5.1. ARIMA-PRED*

ARIMA model is used to understand time series or predict a point in the future. Any time series that exhibits patterns and is not random white noise can be modeled with ARIMA models [19].

The Autoregressive part (**AR**) of the method indicates that the variable of interest undergoes a regression on its previous values. The integrated part (**I**) indicates that the data values have been replaced with the difference between their current and previous values, making the series stationary (this process can be performed more than once). The Moving Average part (**MA**) indicates that the regression error is a linear combination of the error terms applied to past observations. The purpose of each component is to make the model fit the data as best as possible [47].

Non-seasonal ARIMA models are usually denoted by ARIMA**(p, q, d)**, where:

- **p**: number of lags of the autoregressive model.
- **d**: number of times the data has had past values subtracted.
- **q**: order of the moving average model.

The model can be written as follows:

$$y_t = \alpha + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \varepsilon t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - ... - \theta_q \varepsilon_{t-q} \quad (8)$$

where $\alpha$ is a constant, $\phi$ is an estimated coefficient, and $y$ deals with past entries (*lags*); $\theta$ is an estimated coefficient and $\varepsilon$ is the errors associated with past regressive predictions.

ARIMA models generally perform best when the series is relatively long and well behaved. If the series is very irregular, the results are generally inferior to those obtained by other methods, such as recurrent neural networks.

### *5.2. LSTM-PRED*

The LSTM-PRED is based on LSTM model, which processes data passing on information as it propagates forward. This model can recognize long-term patterns and dependencies, ideal for classifying, processing and predicting time series with time intervals of unknown duration. This is possible due to the ability to remove or add information to the state of the cell, regulated by structures called gates [48].

The cell's state, in theory, acts as a pathway that carries relevant information along the entire sequence chain. Information is added to or removed from the cell state through gates, which decide what information is allowed in the cell state. They learn what information is relevant

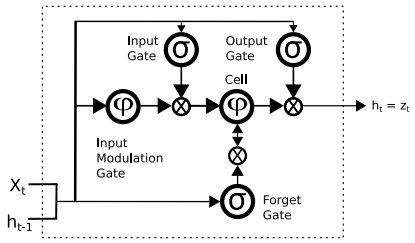



**Figure 3:** LSTM cell.

to keep or forget during training. In general, models based on LSTM networks has three gates:

- **Forget Gate**: removes information that is not useful to the cell state.
- **Input Gate**: adds useful information to cell state.
- **Output Gate**: extracts useful information from the current cell state to decide what the next hidden state should be.

Figure3 shows the structure of the LSTM cell.

LSTM networks can be applied to a variety of deep learning tasks, which primarily include prediction based on prior information. Examples include text prediction, business actions, and network traffic volume.

## 6. Evaluations and experiments

We show the performance of the TIPTOP using a real-world mobile network traffic data in Milan, Italy[22].

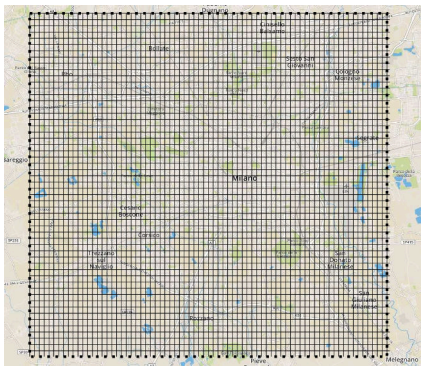
Section 6.1 describes the scenario used in this work. Section 6.2 shows the design and implementation of the method proposed to build an environment based on Cloud-Fog Computing. Section 6.3 discusses the development of the ARIMA-PRED and LSTM-PRED models. Section 6.4 shows the performance assessment of the SMART-FL considering six snapshots selected in particular days and hours based on the traffic intensity. Finally, Section 6.5 shows the performance assessment of the SMART-FL and TIPTOP compared with two algorithms considering one month of the predicted network traffic.

### *6.1. Scenarios description*

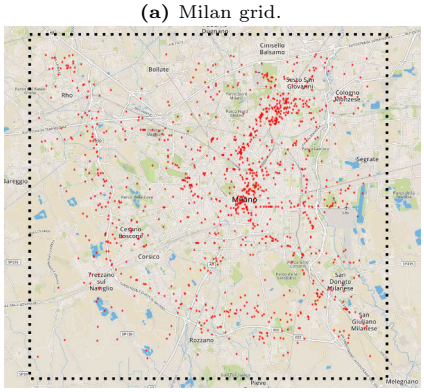
We experiment with publicly available real-world mobile traffic data sets, which contain two months of network traffic data (November/2013 to December/2013) released through Telecom Italia's Big Data Challenge [22]. The unique multi-source composition of the dataset makes it an ideal dataset to analyze various problems, including energy consumption, mobility planning, event detection, and many others. Also, it is the richest open multi-source data set ever released on two geographical areas. Figure 4 illustrates this scenario.

The geographical area is composed of a 100 $\times$ 100 grid, with a size of 235 $\times$ 235 $m^2$ each, illustrated in

Figure 4a. Every time a mobile user requests services to a telecommunication provider, a Call Detail Record (CDR) is recorded. This information is then compiled into 10-minute intervals. Furthermore, a base station set $BS = \{bs_1, bs_2, ..., bs_{bs}\}$ was obtained from CellMapper[1], which consists of the locations and coverage areas of active base stations observed in the two months periods, illustrated in Figure 4b. The grids were mapped to the base stations' coverage areas and aggregated the CDR amount per base station. It is considered that there are multimedia services requests in a grid if its CDR amount is above the average. The multimedia services requests are aggregated by region.




**(a)** Milan grid.




**(b)** Base station deploys in Milan.

**Figure 4:** Map view of the scenarios studied.

Table 3 shows the first five values of network traffic in cell #1. The first column (*timestamp*) represents a timestamp variable, starting on 2013-11-01 00:10:00 and ending on 2014-01-01 23:40:00. The second column (*traffic volume*) represents the amount of CDR generated.

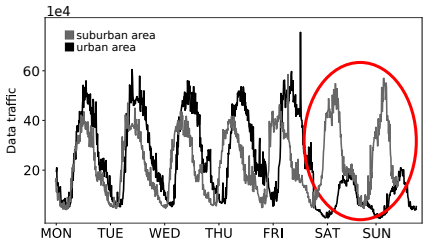
Given the temporal organization, these data can be modeled and evaluated as a time series, where the neighboring observations are dependent, and the interest is to analyze and model this dependency.

[1]https://www.cellmapper.net/map

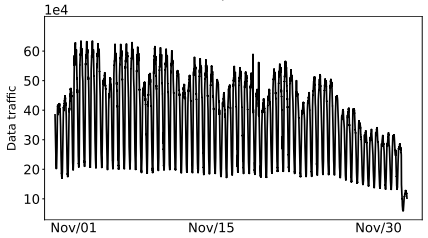
**Table 3:** The dataset structure.

| | timestamp | traffic volume |
|---|---|---|
| **0** | 2013-11-01 00:00:00 | 11.028366381681 |
| **1** | 2013-11-01 00:10:00 | 11.1271008756737 |
| **2** | 2013-11-01 00:20:00 | 10.8927706027911 |
| **3** | 2013-11-01 00:30:00 | 8.62242459098975 |
| **4** | 2013-11-01 00:40:00 | 8.00992746244576 |
| **...** | ... | ... |

Figure 5 shows the trends in three different traffic flows over a seven and thirty days period, both with 10 minutes granularity. Based on Figure 5a, the traffic data is higher during the working hours than at midnight and lower on weekends than on weekdays. The picture clearly shows that the data traffic exhibits certain periodicity (in daily and weekly patterns) due to regular working schedules. On the one hand, the urban area has lower traffic data on weekends than during the week due to urban mobility and regular working hours. On the other hand, the suburban area has regular traffic data during the seven days of the week. Figure 5b shows the average traffic volume over one month of November.




**(a)** Average traffic volume over one week (urban and suburban area).




**(b)** Average traffic volume over one month.

**Figure 5:** Average traffic volume of all base stations.

### *6.2. Proposed method application*

The design and implementation of the method proposed to build an environment based on Cloud-Fog Computing were applied to the existing cellular network described earlier. However, it can also be applied to other datasets.

The method starts with the definition of the set $BS$ and the parameters $\Xi$ and $\mu$. $BS$ represents the base stations in the scenario described above; $\Xi$ is defined as the radius $r = 3km$; and $\mu = 2$, but can be any value. Since $\Xi$ is defined as the radius $r = 3km$, this means that there is an edge between $bs_i$ and $bs_j \in BS$, if and only if, the distance between $bs_i$ and $bs_j$ is $3km$. It is worth mentioning that,

in this step, other forms of connection are possible, such as signal strength or traffic similarity.

Figure 6 illustrates the development of the hierarchical scenario resulting from each step performed. Figure 6a illustrates the graph $\mathcal{G} = (BS, \mathcal{E})$ modeled from step **1**. Figure 6b depicts the seven communities over the base stations (colored dots), represented by the letters A to G, from step **2**. Each community portrays a region. It is possible to notice that many regions (e.g., C, D, F, and G) are composed of an urban and suburban segment. This indicates that the base stations in these areas are potentially complementary due to traffic patterns. The Louvain heuristic is used to find the community set $S$. It is a fast algorithm $O(n+m \cdot logn+m)$, where $n$ and $m$ are the numbers of vertices and edges, respectively, to detect communities in large-scale networks based on modularity optimization [37]. This method aims to find partitions (structures composed of communities) that maximize the density of intra-group connections concerning the density of inter-group connections and find dense optimal sub-graphs in large graphs.

The environment obtained from the steps (**3-5**), the addition of an upper-tier node for each community detected, the addition of edges between all nodes in the current tier, and the detection of communities and removal of edges between nodes of different communities, respectively, are illustrated in Figure 6c. The step (**6**), adding an upper-tier node with links between the node and the subgraph for each subgraph found, is illustrated in Figure 6d. The stopping criterion is $\mu = 2$ and must be observed at each step. In this case, the stopping criterion is met, adding an upper-tier node that connects to the lower tier nodes, as shown in Figure 6e. Figure 6f illustrates the final Cloud-Fog hierarchical environment.

It is considered that nodes added of each tier from steps **(2)** to **(6)** are labeled cloudlet (**CL1**, **CL2**, **CL3**, **CL4**, **CL5**, **CL6** and **CL7**), regional Cloud (**RC1** and **RC2**) and Cloud (**CL**), respectively. Table 4 shows the numbers of nodes and average coverage areas per tier.

**Table 4:** Number of nodes and average coverage area per tier.

| Nodes | Number of nodes | Coverage areas ($m^2$ per node) |
|---|---|---|
| Base station | 1150 | 492.46 |
| Cloudlets | 7 | 9072.67 |
| Regional Cloud | 2 | 31754.37 |
| Cloud | 1 | 552250 |

Service requests belong to one of two classes in this environment: multimedia or concurrent. Let $W$ be the set that represents the services from the multimedia class and $J$ be the set that represents the services from the concurrent class. On the one hand, multimedia class' services are Video on Demand (VoD), interactive video 3D, high-definition, UHD (that includes 4K UHD and 8K UHD) video streaming, etc. On the other hand, concurrent class' services are offered in TCS and IoT. Thus, the network and nodes' resources, such as bandwidth and hardware capacity, are competed by both classes' services. The main idea is to create the most competitive environment.

Let $Lat = \{lat_1,\ lat_2,\ lat_3,\ ...,\ lat_r\}$ and $Mip = \{mip_1,\ mip_2,\ mip_3,\ ...,\ mip_r\}$ be the sets that represents the maximum latency and Millions of Instructions Per Second (MIPS) required for each service $w \in W$ and $j \in J$, based on [49, 50, 51]. Also, $V_t = \{v_1,\ v_2,\ v_3,\ ...,\ v_g\}$ be the set that represents the network traffic in time $t$ for each grid $g$. Thus, the service requirements are modeled for both classes in terms of three parameters: maximum service latency ($lat_i$ - in milliseconds); maximum amount of RAM for temporary storage ($mip_i$ - in $GB$); and maximum amount of MIPS ($v_g^t$). Therefore, for each $w \in W$ and $j \in J$:

$$w_i = \{lat_i,\ mip_i,\ v_g^t\}, \tag{9}$$

$$j_i = \{lat_i,\ mip_i,\ v_g^t\} \tag{10}$$

where, $0 \leq i \leq r$, $lat_i \in Lat$, $mip_i \in Mip$, and $v_g^t \in V_t$. The maximum storage required to perform any service requested at time $t$ in the grid $g$ is $v_g^t$, where $v_g^t \in V_t$. In that way, the storage required to perform any services varies according to network traffic, showing periodicity.

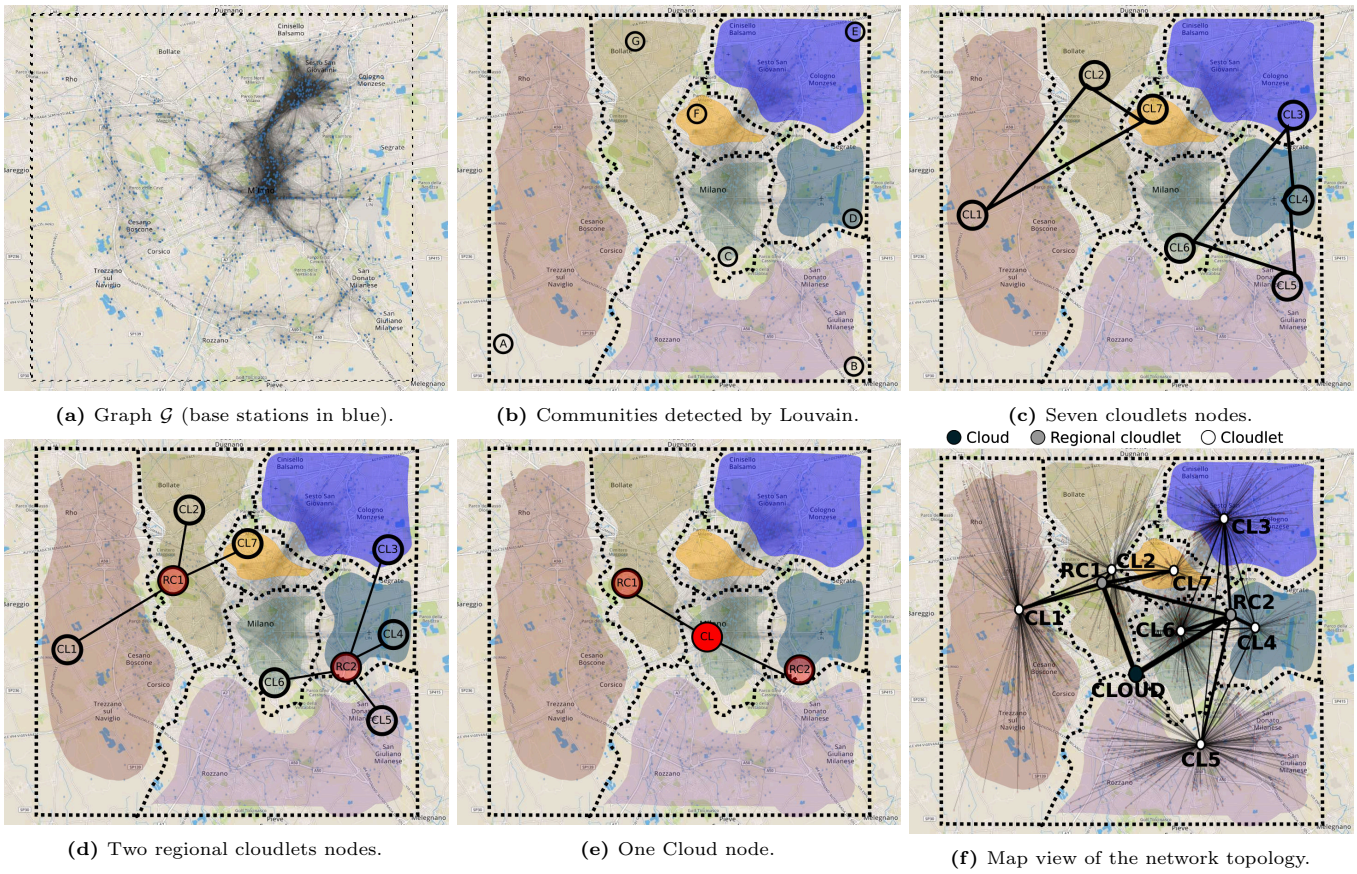
The nodes capable of storing and processing services are modeled in terms of five parameters: MIPS available ($mips$); storage available ($stor$ - in $GB$); RAM available ($ram$ - in $GB$); *uplink* ($up\_rate$) and *downlink* ($down\_rate$) rates offered, both in $Mb$. Table 5 shows the range of values for each parameter, based on [49]. Therefore, for each $ell \in \mathcal{L}$:

$$\ell = \{mips,\ stor,\ ram,\ up\_rate,\ down\_rate\} \tag{11}$$

**Table 5:** Range of values to model the nodes by tiers.

| Parameters | Values per tier | | | |
|---|---|---|---|---|
| | 1° | 2° | 3° | 4° |
| $mips$ | [2.8 - 5.3] | [5.3-7.8] | [7.8-10.2] | [10.2-20.5] |
| $stor$ | $100^2$ | $200^2$ | $400^2$ | $1000^2$ |
| $ram$ | 25 | 40 | 60 | 100 |
| $up\_rate$ & $down\_rate$ | 300 | 500 | 800 | 2000 |

It is worth noticing that all parameters, such as $lat_i$, $mip_i$, $mips$, $stor$, $ram$, $up\_rate$, and $down\_rate$, change dynamically over time [52]. Besides, higher-tier nodes have more network and hardware resources, such as bandwidth and hardware capacity, than lower-tier nodes. In contrast, lower-tier nodes support latency-sensitive services, not fully supported by CC. The performance assessment was con-




**(a)** Graph $\mathcal{G}$ (base stations in blue). **(b)** Communities detected by Louvain. **(c)** Seven cloudlets nodes.

**(d)** Two regional cloudlets nodes. **(e)** One Cloud node. **(f)** Map view of the network topology.

**Figure 6:** Designing of Cloud-Fog Hierarchical Environments.

ducted in a simulated environment on MultiTierFogSim[2].

### *6.3. Network traffic intensity*

Six snapshots selected in particular days and hours based on the network traffic intensity are analyzed to assess the performance of the SMART-FL algorithm in different circumstances in Milan - Italy. The main idea is to evaluate the positioning of the selected nodes in relation to their storage capacity, latency offered, and the demands' geographic location for multimedia services. To do that, it is applied the **K**-means algorithm to partition the network traffic intensity into 3-predefined distinct non-overlapping subgroups where each data point belongs to only one group. The network traffic intensity is grouped into low, medium, and high traffic intensity. Figure 7 illustrates a part of the training set during the first week of November.

[2]https://github.com/fillipesansilva/MultiTierFogSim.git

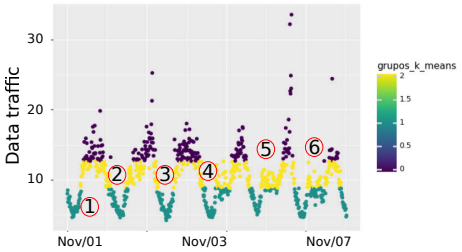



**Figure 7:** K-Means clustering in network traffic.

Figures 8-13 show the instantaneous multimedia service requests fragmentation measured in the Milan region for each snapshot. Each one of these figures presents two plots. Plots labeled "(a)" relate the snapshot of the multimedia service requests in the scenario as well as the nodes selected to receive the multimedia services. In these plots, every gray spot corresponds to one demand $d_r \in D_r$, where $r \in R$, for multimedia services that must be provided. The circles represent the nodes enabled to the placement of multimedia services. Plots "(b)" show the nodes' hardware capacity (x-axis) and latency (y-axis) at that moment. The resulting figures give a rough, yet intuitive, idea of the fog nodes selected in different possibilities. Additionally, the maximum acceptable delay in delivering multimedia services is less than 0.1 seconds [53] and the nodes' storage capacity and latency can fluctuate over time.

Figure 8a illustrates the first snapshot (Dec/22/2013 at 05:40). It is associated with the low traffic intensity group that occurs during dawn on weekends. Based on this, the nodes **CL1**, **CL3**, and **CL5** are selected. Figure 8b depicts that these Fog nodes have an appropriate hardware capacity and the lowest latency to meet this demand. Also, they are positioned geographically close to these regions, reducing the latency and may enhancing user experience.

Figure 9a illustrates the second snapshot (Nov/5/2013 at 10:20). It is associated with the medium traffic intensity group that occurs at certain times in the morning. In this case, all the fog nodes have a maximum acceptable delay in delivering multimedia services, i.e., less than 0.1 seconds. Also, all cloudlet nodes are low on hardware capacity (they may be running concurrent services, for example). Therefore, the nodes **CLOUD**, **RC1**, **and RC2** are selected. Figure 9b illustrates their characteristics at the moment analyzed.

Figure 10a illustrates the third snapshot (Nov/29/2013 at 11:30). It is associated with the medium traffic intensity group that occurs during the mornings on the weekends. Again, all nodes have a maximum acceptable delay in delivering multimedia services. As shown in Figure 10b, only the Cloud, and some cloudlet nodes have the hardware capacity to meet this demand. The nodes **CLOUD**, **CL2**, **CL3**, **CL4**, and **CL5** are selected.

Figure 11a illustrates the fourth snapshot (Nov/10/2013 at 5:00). It exhibits a medium traffic intensity during dawn on weekends. Based on Figure 11b, the Cloud node CL has high latency ($\geq$ 0.1 s) as CL1, CL5, and CL6 nodes have low hardware capacity to meet such demand. Therefore, based on demands' geographic location, nodes' hardware capacity, and latency, the nodes **RC1**, **RC2**, **CL2**, **CL3**, and **CL4** are selected.

Figure 12a illustrates the fifth snapshot (Dec/21/2013 at 13:40). It is associated with the high traffic intensity group that occurs during the working hours of weekdays. Figure 12b illustrates that all nodes have a maximum acceptable delay and reasonable hardware capacity. Again, based on demands' geographic location, nodes' hardware capacity, and latency, the nodes **CLOUD**, **RC1**, **RC2**, **CL1**, **CL2**, **CL4**, and **CL5** are selected.

Finally, Figure 13a illustrates the sixth snapshot (Dec /21/2013 at 14:30). Once again, it is associated with the high traffic intensity group that occurs during working hours on Fridays. Figure 13b illustrates that all nodes have a maximum acceptable delay ($\leq$ 0.1 s), but low hardware capacity. In this special case, all nodes are selected to meet as much of this demand as possible. Thereby, some regions will not be served, or some users maybe have their video-rate adapted, delivery with poor QoE due to the low nodes' hardware capacity.

Considering everything, it is possible to infer that the nodes' storage capacity and geographical location, number of nodes able to process tasks, amount demand for multimedia services, and network latency are all paramount factors to decide which nodes can deploy multimedia services. Also, selecting these nodes closer to the user using a distributed strategy may reduce the bandwidth consumption, resulting in lower costs and improving network efficiency, guaranteeing that most users who depend on these nodes are served, and improving the network deployed to mitigate provider costs.

### *6.4. Performance evaluation of the prediction models*

ARIMA-PRED generates forecasts through the information contained in the chronological series itself, through the ideal adjustments of its parameters. Determining them is a non-trivial task [54]. However, the *auto_arima*() function in Python automatically finds these ideal values. In general, $p + q <= 2$ [19]. Table 6 shows the four best combinations of these parameters generated in relation to the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics.

**Table 6:** Parameters ($p$,$q$,$d$).

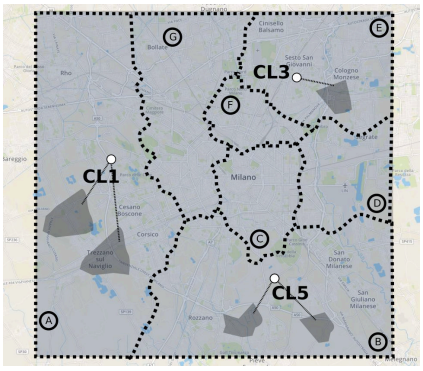
| **(p,q,d)** | **MAE** | **RMSE** |
|---|---|---|
| **(1,1,1)** | 1.092 | 3.645 |
| **(1,1,0)** | 1.108 | 3.985 |
| **(0,1,0)** | 1.175 | 4.732 |
| **(1,0,0)** | 1.190 | 4.355 |

LSTM-PRED, based on LSTM networks, processes data passing on information as it propagates forward. The training set technique was sliding window, commonly adopted in [55]. The sliding window size is $L = 144$, including the current timestamp. This value is based on the daily frequency observed in the series.

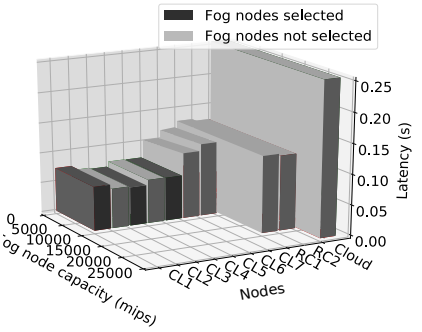
An exhaustive search technique called GridSearchCV is used, provided by the library *sklearn* to find the most suitable hyperparameters. This technique aims to test all possible combinations of the hyperparameters that were passed to them to find the most suitable configuration for the model in question. Table 7 shows the values of the estimated hyperparameters that optimize the model. The first column refers to the hyperparameters. The second column refers to the estimated values based on [43, 46]. The third column refers to the values that optimize the model. The average values for MAE and RMSE considering the hyperparameters that optimize the model are 0.97 and 1.35, respectively. Figure 14 describes the overall modeling process.

**Table 7:** Hyperparameters optimization.

| **Hyperparameters** | **Estimated values** | **Optimized model** |
|---|---|---|
| **Epochs** | [1500, 1700, 1800] | 1800 |
| **learning rate** | [0.001,0.01,0.1,0.0001] | 0.01 |
| **Optimizer** | [Nadam, Adam, RMSProp] | adam |
| **Loss function** | [logcosh, mae, mse, hinge, squared_hinge] | mae |
| **Activation function** | [relu, linear, sigmoid, hard_sigmoid, tanh] | sigmoid |
| **Number of hidden layers** | [1,2,3] | 2 |
| **Dimension of hidden layer** | [200, 400, 600] | 200 |
| **Sliding window** | 144 | |

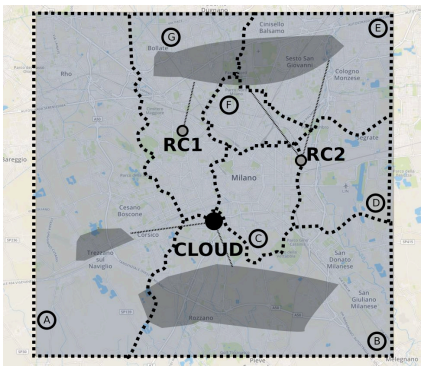



**(a)** Multimedia requests and selected nodes.

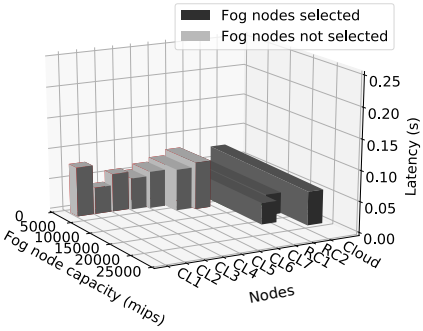



**(b)** Nodes' hardware capacity and latency.

**Figure 8:** Low traffic intensity.

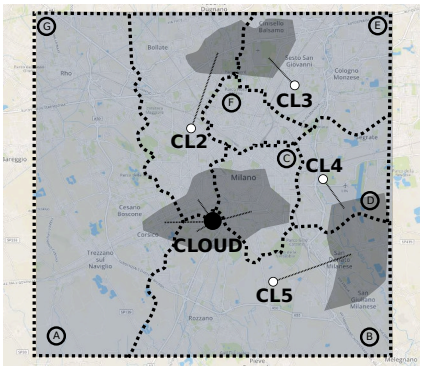



**(a)** Multimedia requests and selected nodes.

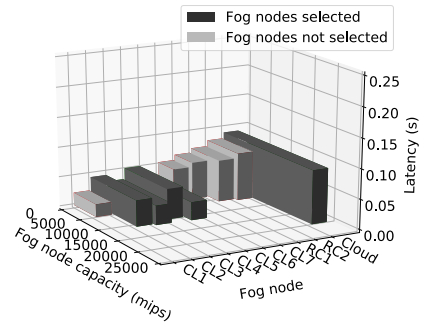



**(b)** Nodes' hardware capacity and latency.

**Figure 9:** Medium traffic intensity.

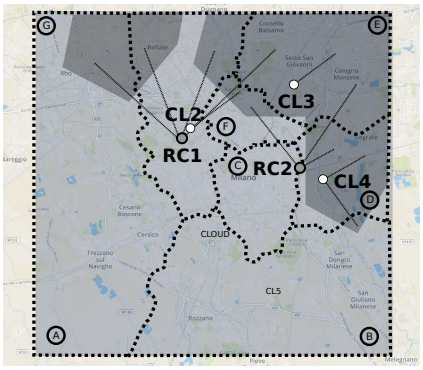



**(a)** Multimedia requests and selected nodes.

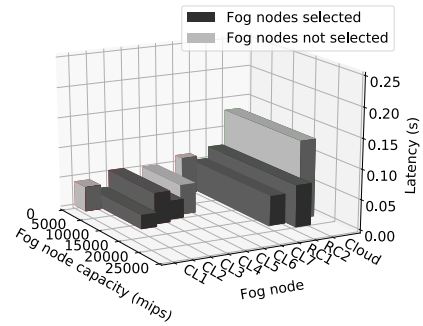



**(b)** Nodes' hardware capacity and latency.

**Figure 10:** Medium traffic intensity.

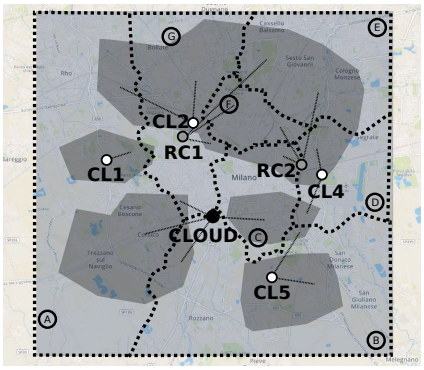



**(a)** Multimedia requests and selected nodes.

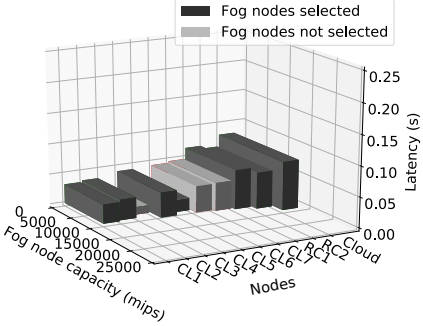



**(b)** Nodes' hardware capacity and latency.

**Figure 11:** Medium traffic intensity.

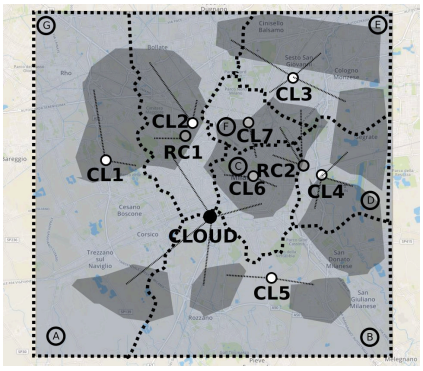



**(a)** Multimedia requests and selected nodes.

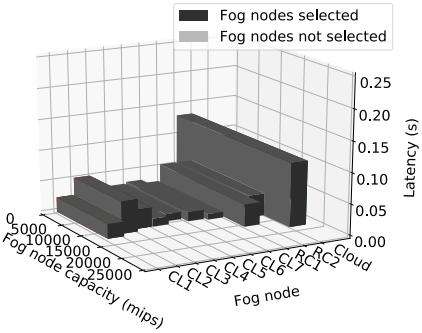



**(b)** Nodes' hardware capacity and latency.

**Figure 12:** High traffic intensity.




**(a)** Multimedia requests and selected nodes.




**(b)** Nodes' hardware capacity and latency.

**Figure 13:** High traffic intensity.

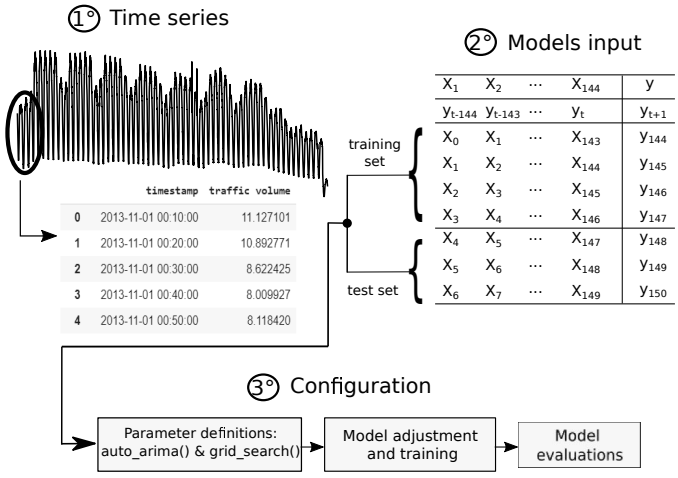



**Figure 14:** Overall modeling process.

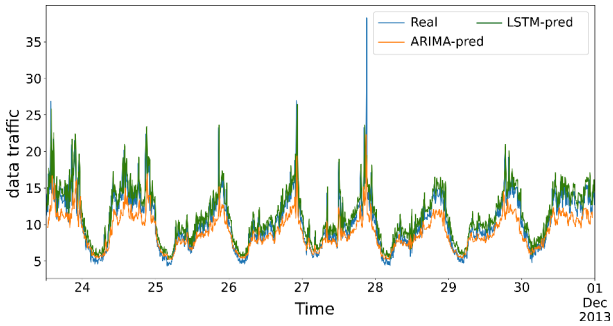



**Figure 15:** Prediction results considering ARIMA-PRED and LSTM-PRED models.

The traffic from the first 40 days is selected as the training data for both models, and the last 22 days' traffic is set to be test data. The MAE and RMSE are adopted as the evaluation metrics and defined as

$$MAE = (\frac{1}{N}) \sum_{i=1}^{N} |\hat{y}_i - y_i| \quad (12)$$

$$RMSE = \sqrt{\Sigma_{i=1}^{N} \Big(\frac{\hat{y}_i - y_i}{N}\Big)^2} \quad (13)$$

where $N$ is the sample size, $\hat{y}$ is the prediction and $y_i$ the true value.

Figure 15 shows the comparisons of traffic data predicted considering a one-month forecast of traffic data. It can be seen that the prediction results well match the trend of real data traffic even when the traffic becomes unstable due to regular working schedules and urban mobility.

As is possible to notice, the peaks of both in and out traffic can be effectively captured and predicted better by LSTM-PRED. Due to the irregularity of the time series, the result of the ARIMA-PRED is generally inferior to those obtained by other methods, such as those based on RNN. On the one hand, models based on ARIMA show better results when the series is relatively long and well behaved. Also, they depend on a greater knowledge of the components of the time series[46]. On the other hand, models based on RNNs can adapt better to learn the temporal dependencies of the context without much prior knowledge of the series components. Table 8 shows the comparison of MAE and RMSE considering both models.

**Table 8:** Experimental result

| | **MAE** | **RMSE** |
|---|---|---|
| **ARIMA** | 1.092 | 3.645 |
| **LSTM** | 0.97 | 1.35 |

Therefore, according to the analysis and the results presented, the prediction model based on the deep learning method (LSTM-PRED) was better than the traditional algorithm model (ARIMA-PRED). Thus, LSTM-PRED is utilized in the Model Deployment stage.

### *6.5. Network traffic prediction*

The performance evaluation is carried out considering one month of the predicted network traffic in Milan - Italy. The results are analyzed and compared in terms of (i) content delivery and packet delivery rate; (ii) average latency; and (iii) network usage in terms of (a) total volume of data transmitted during migration and (b) link usage. The results are presented with a 95% confidence interval and compared with four strategies:

1. **Dynamic Adjustment (DA) [27]:** The task for serving the client's media request is carried out in a hierarchical manner.
2. **QoE-Aware Placement (QoE-AP) [30]:** A linearly optimized mapping of applications and Fog instances that maximizes QoE.
3. **SMART-FL:** The multimedia services are positioned according to the **SMART-FL** algorithm.
4. **TIPTOP:** The multimedia services are positioned according to the **SMART-FL** algorithm aware of the predicted traffic volume.

DA strategy starts the search for resources by selecting tiers closest to the user, followed by the others in the hierarchy. If there are not enough resources in the current tier, the above tier is considered. QoE-AP presents a centralized service placement approach taking into account both QoE and node resources. SMART-FL and TIPTOP are solutions proposed by this work.

Figure 16 shows the content and packet delivery rate for the four placement strategies considering only requests for multimedia services.

As is possible to notice, DA strategy presents $\approx$ 100% of content delivery rate and $\approx$ 90% of packet delivery rate. This strategy searches for resources by selecting tiers closest to the end-users. When nodes are selected close to the end-users, the services are offered with high speed and connections with more reliable links, increasing the packet

delivery rate, which is not seen here. In fact, this approach selects the most appropriate tiers, not nodes.

QoE-AP strategy presents ≈ 83% of content delivery rate and ≈ 80% of packet delivery rate. For the same reasons given previously, the content and packet delivery rate are low because the services are offered by nodes not close to the end-users. This factor, among others, increases the delay and delivers such services with low QoE.

In general, both select adequate nodes to provide multimedia services. However, they do not exploit the benefits of positioning services in different tiers and close to the end-users. This can increase even more the content and packet delivery rate. The storage and computation of the services closer to the end-users are performed by the SMART-FL and TIPTOP. Both present themselves advantages superior to those discussed so far.

On the one hand, SMART-FL presents ≈ 100 % of content delivery rate and ≈ 96 % of packet delivery rate. On the other hand, TIPTOP presents *approx* 100 % of content delivery rate and ≈ 98 % of packet delivery rate, which is superior to all strategies. Both distribute the services across the network and close to users. The multimedia service placement by TIPTOP is even more adequate due to the set of nodes allocated in Γ.

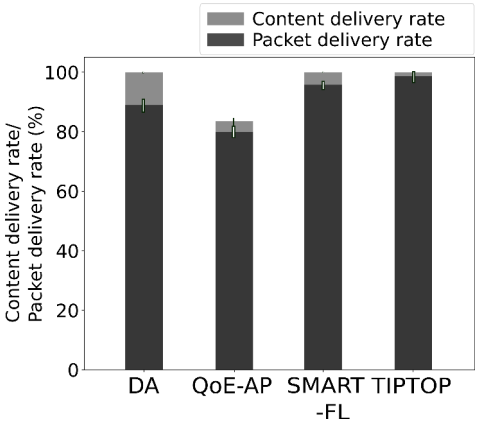



**Figure 16:** Content delivery and packet delivery rate.

Figure 17 shows the latency for the four placement strategies considering only requests for multimedia services.

All strategies have an average latency below the maximum acceptable [53]. QoE-AP offers services with average latency similar to DA. In contrast, DA presents content and packet delivery rate most appropriate than QoE-AP.

In general, both present average latency adequate to provide multimedia services. However, this is not sufficient to provide such content that also includes, among others, constant and continuous flow of packets delivery rate, not seen in both.

SMART-FL and TIPTOP strategies provide constant and continuous flow by selecting nodes distributed throughout the network and close to the end-users. This decreases the number of hops and reduces the latency to deliver these services through connections with more reliable links.

In comparison, the TIPTOP strategy offers the lowest average latency among all strategies. The reason is that the nodes in Γ are in tiers closer to the users and consequently have lower latency than the nodes in $A_{t(n+1)}$.

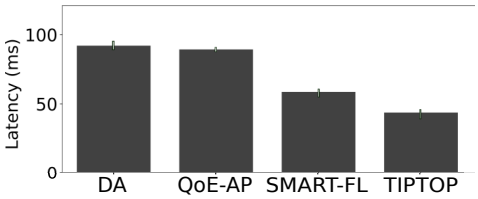



**Figure 17:** Latency.

Figure 18 shows the network usage in terms of (a) total volume of data transmitted during migration and (b) link usage, considering only requests for multimedia services.

DA strategy presents a higher network usage than all of them. This is due to the non-distribution of multimedia services at the network edge, which reduces the overall network usage because fewer channels for data transmission are utilized. Also, a cost of ≈ 29391592.17 Bits Per Second (BPS) is added due to multimedia services migrations to the most appropriate tiers due to variations in network resources and nodes' hardware capacity.

QoE-AP strategy presents an adequate average network usage. However, as discussed, it presents the lowest content and packet delivery rate between all strategies. Also, it presents a similar cost for multimedia service migrations. It seems that QoE-AP strategy evaluated in a strongly connected and fully Cloud-Fog-enabled scenario presents underestimated results.

As mention before, the benefits of distributing services across the network are not explored for both strategies, which can further reduce network usage since fewer channels for data transmission are used. This and others benefits are provided by SMART-FL and TIPTOP strategies.

SMART-FL strategy presents a network usage similar to DA and QoE-AP. However, it presents a better content rate, packet delivery rate, and latency. Likewise, a cost of ≈ 31267651.24 BPS is added. This cost occurs because of the same reasons given previously. It is a minimal cost about the advantages presented. This strategy has lower than maximum acceptable latency, adequate content and packet delivery rate, and relatively low network usage.

TIPTOP strategy presents even more satisfactory results. The network usage is lower than all strategies presented. Likewise, a cost of ≈ 34176269.96 BPS is added. This cost occurs because of the same reasons given previously. As mentioned before, it is a minimal cost about the advantages presented. Nodes reserved in Γ have advantages over nodes in $A_{t(n+1)}$. Therefore, of all the strategies, this is the one that most benefits from the available resources and environment.

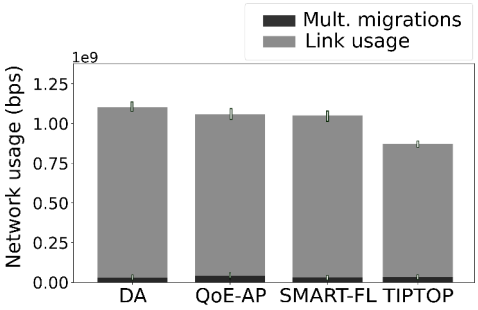



**Figure 18:** Network usage in terms of (a) total volume of data transmitted during migration and (b) link usage.

## 7. Conclusions and future work

The constant communication technology's advancements and the great availability of streaming video services require new methods to ensure the quality for end-users. To improve on this issue, this work first presented a process to design/create a hierarchical multi-tier Cloud-to-Fog network for the distribution of multimedia services. Moreover, this work also proposed an algorithm that selects the minimum number of nodes able to deliver multimedia services with low latency in multi-tier Fog nodes architecture. By the way, the traffic forecast improves the provision of these services. The performance assessment was carried out using two months of real-world mobile network traffic data in Milan, Italy.

Instead of concentrating data and computation in a small number of large Clouds, we consider that either portions or all of the Cloud services must migrate to fog nodes located closer to the user, meeting their needs regarding latency. This is one of the factors that form the primary motivation of this paper.

Keeping this analysis closer to the data source, especially for latency-sensitive services where every millisecond is essential, in addition to the advantages presented in this work, may improve the user experience and reduce overhead in the Cloud as a whole [56]. Moreover, reducing the demand on the Cloud allows turning off servers in the data center to save energy. As has been noted, the proposed solution can be used in a real-world network to cope with future challenges in providing seamless and, at the same time, high-quality multimedia services in hierarchical multi-tier Fog nodes.

Future work intends to extend the proposed method utilizing new approaches, such as Density Based Spatial Clustering of Application with Noise (DBSCAN), to find communities and analyze the energy consumption and network usage. The proposed algorithm for the multimedia services placement is based on an ILP model. The ILP model is logically based on linear equations. However, some decision variables, such as QoE and QoS, have a non-linear effect, which are not considered by the ILP model. More research could be carried out in this direction. Also, more dynamic evaluation scenarios will be used to prove the benefits of the TIPTOP.

## Acknowledgments

The authors acknowledge support from the Brazilian research agency CNPq, CAPES and INCT of the Future Internet for Smart Cities (CNPq 465446 / 2014-0, CAPES 88887.136422 / 2017-00, and FAPESP 2014 / 50937-1).

## 8. References

## References

[1] R. Immich, L. Villas, L. Bittencourt, E. Madeira, Multi-tier edge-to-cloud architecture for adaptive video delivery, in: 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud), 2019, pp. 23–30. doi:10.1109/FiCloud.2019.00012.

[2] J. M. Batallfa, P. Krawiec, C. X. Mavromoustakis, G. Mastorakis, N. Chilamkurti, D. Negru, J. Bruneau-Queyreix, E. Borcoci, Efficient media streaming with collaborative terminals for the smart city environment, IEEE Communications Magazine 55 (1) (2017) 98–104.

[3] F. Pisani, F. de Oliveira, E. S. Gama, R. Immich, L. F. Bittencourt, E. Borin, Fog computing on constrained devices: Paving the way for the future iot, Advances in Edge Computing: Massive Parallel Processing and Applications 35 (2020) 22.

[4] G. Forecast, Cisco visual networking index: global mobile data traffic forecast update, 2017–2022, Update 2017 (2019) 2022.

[5] A. Lutu, D. Perino, M. Bagnulo, E. Frias-Martinez, J. Khangosstar, A characterization of the covid-19 pandemic impact on a mobile network operator traffic, in: Proceedings of the ACM Internet Measurement Conference, 2020, pp. 19–33.

[6] T. Favale, F. Soro, M. Trevisan, I. Drago, M. Mellia, Campus traffic and e-learning during covid-19 pandemic, Computer Networks (2020) 107290.

[7] R. Immich, E. Cerqueira, M. Curado, Adaptive qoe-driven video transmission over vehicular ad-hoc networks, in: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2015, pp. 227–232. doi:10.1109/INFCOMW.2015.7179389.

[8] W. Zhu, C. Luo, J. Wang, S. Li, Multimedia cloud computing, IEEE Signal Processing Magazine 28 (3) (2011) 59–69.

[9] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, O. Rana, The internet of things, fog and cloud continuum: Integration and challenges, Internet of Things 3-4 (2018) 134 – 155. doi:https://doi.org/10.1016/j.iot.2018.09.005.
URL http://www.sciencedirect.com/science/article/pii/S2542660518300635

[10] Y. Cheng, Edge caching and computing in 5g for mobile augmented reality and haptic internet, Computer Communications 158 (2020) 24–31.

[11] T. Taleb, A. Ksentini, A. Kobbane, Lightweight mobile core networks for machine type communications, IEEE Access 2 (2014) 1128–1137.

[12] R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: A taxonomy, survey and future directions, in: Internet of everything, Springer, 2018, pp. 103–130.

[13] C. F. C. Solutions, Unleash the power of the internet of things, Cisco Systems Inc (2015).

[14] D. Rosário, M. Schimuneck, J. Camargo, J. Nobre, C. Both, J. Rochol, M. Gerla, Service migration from cloud to multi-tier fog nodes for multimedia dissemination with qoe support, Sensors 18 (2) (2018) 329.

[15] E. S. Gama, R. Immich, L. F. Bittencourt, Towards a multi-tier fog/cloud architecture for video streaming, in: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), 2018, pp. 13–14.

[16] R. A. C da Silva, N. L. S da Fonseca, On the location of fog nodes in fog-cloud infrastructures, Sensors 19 (11) (2019) 2445.

[17] R. Z. Farahani, M. Hekmatfar, Facility location: concepts, models, algorithms and case studies, Nature Publishing Group, 2009.

[18] A. Anas, Residential location markets and urban transportation. Economic theory, econometrics and policy analysis with discrete choice models, no. Monograph in Paper no. HT-FED2004-56887, Nature Publishing Group, 1982.
[19] G. P. Zhang, Time series forecasting using a hybrid arima and neural network model, Neurocomputing 50 (2003) 159–175.
[20] F. Santos, R. Immich, E. Madeira, Multimedia microservice placement in hierarchical multi-tier cloud-to-fog networks, in: 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), 2021, pp. 1044–1049.
[21] L. Chen, L. Liu, X. Fan, J. Li, C. Wang, G. Pan, J. Jakubowicz, et al., Complementary base station clustering for cost-effective and energy-efficient cloud-ran, in: 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2017, pp. 1–7.
[22] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, B. Lepri, A multi-source dataset of urban life in the city of milan and the province of trentino, Scientific data 2 (2015) 150055.
[23] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, T. Woo, Cloudiq: A framework for processing base stations in a data center, in: Proceedings of the 18th annual international conference on Mobile computing and networking, 2012, pp. 125–136.
[24] T. Lähderanta, T. Leppänen, L. Ruha, L. Lovén, E. Harjula, M. Ylianttila, J. Riekki, M. J. Sillanpää, Edge server placement with capacitated location allocation, arXiv preprint arXiv:1907.07349 (2019).
[25] R. A. C. da Silva, N. L. S. da Fonseca, Location of fog nodes for reduction of energy consumption of end-user devices, IEEE Transactions on Green Communications and Networking 4 (2) (2020) 593–605.
[26] F. Shi, L. Fan, X. Lai, Y. Chen, W. Lin, A hierarchical caching strategy in content delivery network, Computer Communications 179 (2021) 92–101.
[27] J. Kharel, S. Y. Shin, Multimedia service utilizing hierarchical fog computing for vehicular networks, Multimedia Tools and Applications (2018) 1–24.
[28] V. B. C. Souza, W. Ramírez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, G. Tashakor, Handling service allocation in combined fog-cloud scenarios, in: 2016 IEEE international conference on communications (ICC), IEEE, 2016, pp. 1–5.
[29] Y. Kryftis, G. Mastorakis, C. X. Mavromoustakis, J. M. Batalla, J. J. Rodrigues, C. Dobre, Resource usage prediction models for optimal multimedia content provision, IEEE Systems Journal 11 (4) (2017) 2852–2863.
[30] R. Mahmud, S. N. Srirama, K. Ramamohanarao, R. Buyya, Quality of experience (qoe)-aware placement of applications in fog computing environments, Journal of Parallel and Distributed Computing 132 (2019) 190–203.
[31] Y. Sai, D.-z. Fan, M.-y. Fan, Cooperative and efficient content caching and distribution mechanism in 5g network, Computer Communications 161 (2020) 183–190.
[32] K. Velasquez, D. P. Abreu, M. R. Assis, C. Senna, D. F. Aranha, L. F. Bittencourt, N. Laranjeiro, M. Curado, M. Vieira, E. Monteiro, et al., Fog orchestration for the internet of everything: state-of-the-art and research challenges, Journal of Internet Services and Applications 9 (1) (2018) 14.
[33] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, M. Dlodlo, From cloud to fog computing: A review and a conceptual live vm migration framework, IEEE Access 5 (2017) 8284–8300.
[34] W. Shu, A fast algorithm for facility location problem., J. Softw. 8 (9) (2013) 2360–2366.
[35] F. Luna, J. J. Durillo, A. J. Nebro, E. Alba, Evolutionary algorithms for solving the automatic cell planning problem: a survey, Engineering Optimization 42 (7) (2010) 671–690.
[36] J. Xie, B. K. Szymanski, Community detection using a neighborhood strength driven label propagation algorithm, in: 2011 IEEE Network Science Workshop, IEEE, 2011, pp. 188–195.
[37] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of statistical mechanics: theory and experiment 2008 (10) (2008) P10008.
[38] G. Optimization, Inc.,"gurobi optimizer reference manual," 2015 (2014).
[39] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[40] C. W. J. Granger, P. Newbold, Forecasting economic time series, Academic Press, 2014.
[41] R. Andrews, J. Diederich, A. B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-based systems 8 (6) (1995) 373–389.
[42] A. M. Nagy, V. Simon, Survey on traffic prediction in smart cities, Pervasive and Mobile Computing 50 (2018) 148–163.
[43] C. Zhang, H. Zhang, J. Qiao, D. Yuan, M. Zhang, Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data, IEEE Journal on Selected Areas in Communications 37 (6) (2019) 1389–1401.
[44] C. Zhang, H. Zhang, D. Yuan, M. Zhang, Citywide cellular traffic prediction based on densely connected convolutional neural networks, IEEE Communications Letters 22 (8) (2018) 1656–1659.
[45] C. Zhang, H. Zhang, J. Qiao, D. Yuan, M. Zhang, Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data, IEEE Journal on Selected Areas in Communications 37 (6) (2019) 1389–1401.
[46] P. Prettenhofer, G. Louppe, Gradient boosted regression trees in scikit-learn, IEEE Communications Letters (2014).
[47] J. Contreras, R. Espinola, F. J. Nogales, A. J. Conejo, Arima models to predict next-day electricity prices, IEEE transactions on power systems 18 (3) (2003) 1014–1020.
[48] K. J. Hunt, D. Sbarbaro, R. Żbikowski, P. J. Gawthrop, Neural networks for control systems—a survey, Automatica 28 (6) (1992) 1083–1112.
[49] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, R. Buyya, ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments, Software: Practice and Experience 47 (9) (2017) 1275–1296.
[50] M. Sinqadu, Z. S. Shibeshi, Performance evaluation of a traffic surveillance application using ifogsim, in: International Conference on Wireless Intelligent and Distributed Environment for Communication, Springer, 2020, pp. 51–64.
[51] S. S. N. Perala, I. Galanis, I. Anagnostopoulos, Fog computing and efficient resource management in the era of internet-of-video things (iovt), in: 2018 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2018, pp. 1–5.
[52] C. Puliafito, D. M. Gonçalves, M. M. Lopes, L. L. Martins, E. Madeira, E. Mingozzi, O. Rana, L. F. Bittencourt, Mobfogsim: Simulation of mobility and migration for fog computing, Simulation Modelling Practice and Theory 101 (2020) 102062.
[53] E. Liotou, D. Tsolkas, N. Passas, L. Merakos, Quality of experience management in mobile cellular networks: key issues and design challenges, IEEE Communications Magazine 53 (7) (2015) 145–153.
[54] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.
[55] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, I. Rojas, Window size impact in human activity recognition, Sensors 14 (4) (2014) 6474–6499.
[56] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM, 2012, pp. 13–16.