Koursioumpas, N., Barmpounakis, S., Stavrakakis, I. & Alonistioti, N. (2021). AI-driven, Context-Aware Profiling for 5G and Beyond Networks. *IEEE Transactions on Network and Service Management*, 1-12.

DOI: 10.1109/tnsm.2021.3126948

AI-driven, Context-Aware Profiling for 5G and Beyond Networks

Nikolaos Koursioumpas*, Sokratis Barmpounakis*, Ioannis Stavrakakis*, Nancy Alonistioti*
*Department of Informatics and Telecommunications, University of Athens, Greece

Abstract—**In the era of Industrial Internet of Things (IIoT) and Industry 4.0, an immense volume of heterogeneous network devices will coexist and contend for shared network resources, in order to satisfy the very challenging IIoT applications, requiring ultra-reliable and ultra-low latency communications. Although novel key enablers, such as Network Slicing, Software Defined Networking (SDN) and Network Function Virtualization (NFV) have already offered significant advantages towards more efficient and flexible network and resource management approaches, the particular characteristics of IIoT applications pose additional burdens, mainly due to the complex wireless environments, high number of heterogeneous network devices, sensors, user equipments (UEs), etc., which may stochastically demand and contend for the - often scarce - computing and communication resources of industrial environments. To this end, this paper introduces PRIMATE, a novel, Artificial Intelligence (AI)-driven framework for the profiling of the networking behavior of such UEs, devices, users and things, which is able to operate in conjunction with already standardized or forthcoming, AI-based network resource management processes towards further gains. The novelty and potential of the proposed work lies on the fact that instead of attempting to either predict raw network metrics in a reactive manner, or predict the behavior of specific network entities/devices in an isolated manner, a big data-driven classification approach is introduced, which models the behavior of any network device/user from both a macroscopic, as well as service-specific perspective. The extended evaluation at the last part of this work shows the validity and viability of the proposed framework.**

Index Terms—**context-aware profiling; machine learning; AI-driven networking; 5G; resource allocation**

I. INTRODUCTION

THE Internet of Things (IoT) has become one of the key concepts in the evolution towards the next generations of networks and communication systems offering a fertile ground for innovation, disruptive business models and novel ways of engaging diverse industries with other businesses, as well as end-customers. IoT can be applied in almost any sector of the economy, with often diverse participating stakeholders and numerous business applications and operations. Although the initial concept of IoT firstly appeared almost 20 years ago by Auto-Id Labs [1], the actual transition towards a true "Internet" of Things, where all devices will be part of a globally integrated system was introduced around a decade later [2]; the initial architectural principles and service requirements for specific types of applications -potentially involving multiple devices, types of equipment, sensors, actuators, etc.-, started emerging; nevertheless, the first attempts, mainly operated in an almost disconnected manner, due to the obstacles posed by the maturity of the available communication technologies.

Recently, IoT architectures began further evolving into smart, interconnected ecosystems with the adoption of novel communication enablers and technologies, primarily referring to the 5th generation of wireless and mobile networks (5G). Specifically, the Industrial IoT, -also in the context of the Industry 4.0, otherwise known as the fourth industrial revolution-, assumes the interconnection of massively deployed smart devices, network and/or computing elements, in industrial production environments, targeting high automation and control, as well as ultra-high reliability. Such complex industrial production environments examples comprise smart factories, smart plants, or business supply chains.

From the communication perspective, 5G networks have largely integrated the notions of Edge and Fog computing - i.e., distributed computing paradigms in the vicinity of the end users-, which gradually prove to be of utmost significance to the business and technical requirements of IIoT use cases. 3GPP in Release 16 [3] already provides a detailed overview of the support for Edge Computing, which enables the operator and third party services to be hosted close to the UE's access point of attachment, so as to achieve an efficient service delivery; this is realized on the one hand through the reduced load on the transport network, while on the other hand, on the reduced distance of the physical link -and as a result of the link propagation delays-, which is crucial for time-critical services. Additionally, ETSI has released a white paper on MEC for 5G Networks [4]; in this work, one of the described use cases focuses on MEC for Industrial IoT. The same work identifies the significance of both the Edge Cloud -as one of the key components for (Massive) IoT-, as well as Network Slicing, which allows offering dedicated resources for service tenants specifically tailored to their needs.

On top of the aforementioned technology enablers, solutions that exploit Artificial Intelligence (AI), Machine Learning (ML) and Data Analytics (DA) are very promising towards providing further gains in 5G and beyond complex network environments. In the context of 5G networks and beyond, ML and DA are already considered as structural entities and enablers towards the amelioration of numerous network operations. 3GPP has recently introduced a new core network function [3], namely the NetWork Data Analytics Function (NWDAF), which is responsible for providing network analysis information upon request from network functions (NFs), e.g., assisting the Policy Control Function (PCF) in selecting traffic steering and resource allocation policies. According to the literature, NWDAF is still very limited concerning its functionalities and capabilities.

Although AI and ML are considered vast domains, numerous applications and algorithms can be tailored to the needs of the Industrial IoT use cases such as predictive maintenance, Quality 4.0 - a term that references the future of quality and organizational excellence within the context of Industry 4.0 -, human-robot collaboration and predictive network and computing resource management, also in the context of 5G networking and Edge Computing mentioned earlier.

The last use case is the focus of this work, namely the predictive network and computing resource management. More specifically, the framework that is presented in this paper, namely PRIMATE (**PR**ofil**I**ng **M**ech**A**nism Based on Con**TE**xt), provides a detailed overview of a novel mechanism, which monitors and combines (processes) big volumes of diverse network- and user/device- oriented data and extracts profiles for UE and Things, based on past behavior in terms of device type, mobility patterns, service consumption, etc. PRIMATE's proposed framework advances the field of research by extending the functionality of the NWDAF by introducing complementary NFs (e.g Cluster & Profile Forecasting) capable of proactively providing new network analysis information. Industry 4.0 introduces numerous use cases with a plethora of diverse device types, such as sensors, smart monitors, remotely managed or autonomous robotic equipment, smart grid systems, embedded AI devices, tracking systems, etc., - referred to as "things" in the context of IIoT - all of which participate in complex manufacturing processes. It becomes, thus, a major challenge to efficiently differentiate and manage the resources allocated to those devices. The profiles that are extracted by the proposed framework are used for device behavior prediction and - as a result - can be used for the prediction of forthcoming network and/or computing resource requirements. This prediction can ultimately be exploited by network domain experts and administrators towards efficient and proactive resource planning. In order to achieve the latter, PRIMATE could be integrated with different resource allocation strategies such as Virtual Network Function (VNF) placement/(auto)scaling, mobility management algorithms by proactively predicting handover requests, etc. Finally, in latency-critical services – focused but not limited in the context of IIoT - , the forecasted profiles can be exploited for proactive resource allocation/cell association, towards ensuring that the respective challenging service demands are satisfied.

The novelty and high potential of the proposed approach lies on the fact that it departs from mainstream approaches that attempt to either predict raw network metrics in a reactive manner, or predict the behavior of specific network entities/devices according to individual network service requests in an isolated manner. Instead, a big data-driven classification approach is introduced by the proposed framework, which categorizes the behavior of any network user/device from both a macroscopic, as well as service-specific perspective. All devices/users are assigned a respective profile proactively, upon entering a specific domain. This assignment is based on a behavioral forecasting process -as it will be shown in the following sections - whose accuracy - in turn - increases the potential to efficiently manage the allocation of resources in a holistic manner, from an end-to-end perspective and for all participating users/devices of a specific domain.

The structure of the rest of the paper is as follows: Section II presents an overview of the existing work on profiling for networks and predictive behavior analysis of users and/or devices. Afterwards, Section III provides the details of the design, methodology, data model and ML algorithms used in the proposed framework. The following Section, IV presents the results of the evaluation of PRIMATE, while Section V concludes this paper and discusses the findings, next steps, as well as open research questions of the domain.

II. Related work

This chapter discusses the recent studies concerning AI/ML techniques for Industry 4.0, as well as studies that propose innovative mechanisms for analysing and predicting network behavioral patterns. Diez-Olivan *et al.* in [5] provide a comprehensive survey of the recent developments in data fusion and machine learning for industrial prognosis, placing an emphasis on the identification of research trends, niches of opportunity and unexplored challenges. Additionally, the authors provide a principled categorization of the utilized feature extraction techniques and machine learning methods for root cause analysis (descriptive), determine when the monitored asset will fail (predictive) or decide what to do in order to minimize the impact on the process at hand (prescriptive). In [6] Paolanti *et al.* describe a Machine Learning architecture for Predictive Maintenance, based on the Random Forest algorithm, which was tested on a real industry use case. The presented preliminary results showed a proper behavior of the approach on predicting different machine states with high accuracy. The study in [7] proposes a big data analysis framework for extracting network behaviors in cellular networks for Industry 4.0 applications providing novel insights into call usage and network utility. Apart from Industry 4.0-specific studies, there are numerous proposals that attempt to analyze and exploit user and network behavioral patterns, as well as introduce innovative mechanisms based on both supervised and unsupervised approaches [8] [9] [10] [11] [12]. In [8], a Context Extraction and Profiling Engine (CEPE) is introduced, which builds upon a Knowledge discovery in databases (KDD) framework catering for the extraction and exploitation of user behavioral patterns from network and service information. The methodology followed for the implementation of this mechanism is based on k-Means Clustering, Spectral Clustering, Naïve Bayes and Decision Tree learning algorithms. Extended evaluation results provided in this work prove the validity of the proposed solution. Valtorta *et al.* in [9], propose a methodology to process LoRaWAN packets in order to perform profiling of the IoT devices based on radio and network behavior. K-means clustering was chosen as a method for the implementation of the mechanism. The results showed remarkable clustering performance according to validation indices such as Silhouette and Devies-Bouldin indices. The authors in [10] introduce a framework for clustering, forecasting and managing traffic behaviors for large numbers of heterogeneous cells, with different statistical traffic characteristics. The methodology followed in this study comprises k-Means algorithm for traffic clustering and Auto Regressive (AR), Neural Networks and Gaussian Process for traffic forecasting. In [11], the authors propose algorithms for clustering cell towers based on their location, as well as for defining Baseband Units (BBUs) clusters, based on the prediction of mobility and traffic patterns. For the clustering of the cell towers, the authors propose Hierarchical Clustering, as well as an improved version of the Affinity Propagation with Traffic Awareness. As for the BBU clustering, Karneyenka *et al.* introduced three different Location Aware algorithms enhanced with mobility and handovers. Another work [12] utilizes mobile network data in order to analyze anomalous behavior of mobile wireless network based on k-Means and Hierarchical Clustering Techniques. Additionally, Parwez *et al.* discussed the effect of anomalous and anomaly-free data by experimenting on a prediction model where results showed that the error in prediction, - while training the model with anomaly-free data - largely decreases as compared to the case when the model was trained with anomalous data. Apart from the well-established and improved machine learning techniques there are numerous studies that introduce new and innovative solutions in order to exploit the aggregated traffic data from the network and perform context-based resource management policies [13], [14], [15], [16]. The work presented by Barmpounakis *et al.* in [13] provides a holistic context-based framework comprised of three individual mechanisms named Compass, CEPE and CIP that optimize in a complementary manner the radio access technology (RAT) selection and access traffic steering/switching/splitting (ATSSS) operations in 5G network environments. The extended evaluation results show the effectiveness of the proposed framework, while the architectural aspects of the same work discuss the viability of the framework in forthcoming 5G - and beyond - systems. In [14], the authors propose an implementation for distributing traffic flows across different network interfaces based on the characteristics of the flow. Xu *et al.* have evaluated two different approaches based on static and dynamic network selection clustering algorithms in a simplified MIMO scenario, with LoRa, WiFi and LTE network interfaces available. The dynamic clustering approach achieved an even better load balancing between two network interfaces (Wi-Fi, LTE). In [15], an SDN-enabled 5G VANET is introduced, where neighboring vehicles are clustered in an adaptive manner, according to real-time road conditions based of Angle of Arrival (AoA), Received Signal Strength (RSS) and inter-vehicular distance (IVD). Additionally, a dual cluster head scheme is introduced in order to improve the network's robustness and guarantee seamless communication. The results showed that the proposed design substantially improved 5G users' bit error rate and trunk link throughput rate. Last but not least, in [16], the use of a user-specific and adaptive cell clustering technique is proposed, based on mobility state estimation. As a result, the network customizes the cell cluster size separately for each user. Results show network performance enhancements in terms of mobility management, as well as throughput. As it becomes clear, numerous works have attempted to model the user traffic and mobility patterns, or the network's traffic characteristics via diverse ML-based approaches. However, to the best of our knowledge this is the first work that aggregates and correlates network-, service-, as well as user/device-oriented information from diverse layers of the network towards a holistic behavioral pattern identification, as well as forecasting. The proposed framework correlates formerly disconnected and uncorrelated information sources into a broader concept, namely the behavioral profile and manages to ultimately predict this profile throughout the users' activity for future, dynamically-defined time windows in the network. As it will be shown, this can prove of utmost significance for network resource management purposes, enabling the network administrators to proactively allocate resources in different segments of the network (infrastructure, virtualization, etc.), and especially for Industry 4.0 and IIoT use cases, where the heterogeneity of co-existing devices and the ultra-dense nature of the radio environments demonstrate highly complex behaviors.

III. The proposed framework

This section provides a detailed description of the proposed framework. Section III-A describes the data model that was used, Section III-B presents a brief overview of PRIMATE including all its structural components and modules. Finally, section III-C provides a detailed description of each methodology step followed for the implementation of PRIMATE's proposed framework along with a step-by-step algorithmic representation.

A. Data Modeling

This section presents the details of the data model designed and used by PRIMATE. Different data categories - from now on denoted as *data entities* - are identified, namely *Device*, *Service* and *Network*. All data entities are associated to a specific UE, which is characterised by a unique identifier, -i.e., the International Mobile Subscriber Identity (*IMSI*). Each one of the three data entity types is correlated with the other two, since each single UE is using a *Device* that executes a *Service* through a *Network* Radio technology. A visual representation of the data model is given in Figure 1.

The *Device* entity refers to a specific type of equipment that a user is using and comprises of data features that are related to geolocation (latitude, longitude), user velocity, battery status of the device, radio signal-related metrics (Reference Signal Received Power/Quality - RSRP/RSRQ and transmission power), as well as user-cell association information. The *Service* entity is related to the type(s) of active session(s) of the user. It is described by the service name, the transfer protocol used for the Transport Layer (TCP/UDP), as well as a number of service-related metrics, namely Uplink (UL) and Downlink (DL) packet size and inter-packet transmission interval. The *Network* entity refers to the type of RAT serving the specific user-cell association. The identified data features of *Network* entity are related to traffic load and end-to-end delay for both UL and DL stream (transmitted and received data in packets/bytes, UL/DL end-to-end delay). There are data features characterising a cell that is used in the RAT (transmission power of the cell, type of cell, allocated bandwidth, number of UEs connected on the cell).

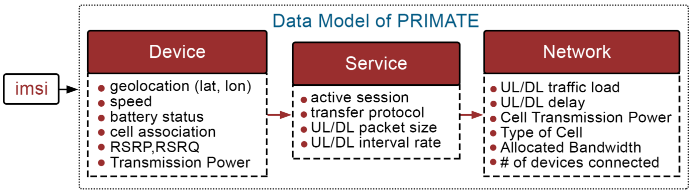


Fig. 1: data model of PRIMATE

B. Overview

In the current section, an overview of the PRIMATE framework is provided. As already discussed earlier, PRIMATE dynamically generates network-, service-, as well as user/device-related behavioral profiles, by processing contextual information that is aggregated from diverse network entities and segments as well as logging modules; ultimately, the aforementioned extracted knowledge is exploited in order to forecast network and service requirements, as well as UE-related behavior (e.g., device mobility, service consumption etc.). As a result, this extracted knowledge could be exploited by respective proactive network management and resource allocation schemes, generating considerable gains for the network. PRIMATE's step-by-step methodology, which is also visually depicted in Figure 2, is provided below via the list of modules, which comprise the overall framework, and their respective functionality.

1) **Aggregation and pre-processing of Contextual Information**: This module is responsible for aggregating and pre-processing all the available contextual information from the different network entities and segments as well as the logging modules.
2) **Hierarchical Agglomerative Clustering (HAC)**: This module exploits all the available contextual information (data points) provided by the previous module, in order to group similar data points in common clusters named *Labeled Clusters*.
3) **Dynamic Cluster Filtering & Profile Extraction**: This module is responsible for 3 main operations. More specifically, the module removes *Labeled Clusters*, which are considered outliers. This study defines the output of this operation as *Behavioral Clusters*. The module also maps the *Behavioral Clusters* to profiles (named *Behavioral Profiles*) and finally filters these produced profiles in order to further enhance the quality of the output.
4) **Training for Cluster Prediction**: This module exploits the extracted *Behavioral Clusters* in conjunction with the initial contextual information, in order to train an ensemble *Weighted Voting Model*. This model combines the predictions from 4 different well-known supervised ML models (called base models), namely a Decision Tree (DT), a Random Forest (RF), a k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM). The result of this module is the classification of newly introduced contextual information by predicting the respective clusters.
5) **Cluster & Profile Forecasting**: This module is developed to perform cluster and profile forecasting for predefined time windows. More specifically, the module attempts to forecast contextual information that will be provided as input to the already trained ensemble model, introduced in the previous step, in order to predict the respective clusters. These clusters are exploited by the dynamic cluster filtering & profile extraction module which is capable of extracting the final forecasted *Behavioral Clusters* and *Profiles*.
6) **Performance Evaluation**: This is the final module of the PRIMATE framework. It is responsible for collecting the predicted (Cluster & Profile Forecasting module) along with the ground truth (Profile Extraction & Dynamic Cluster Filtering) behavioral clusters and profiles in order to evaluate the performance of the forecasting module (*Step 5*).

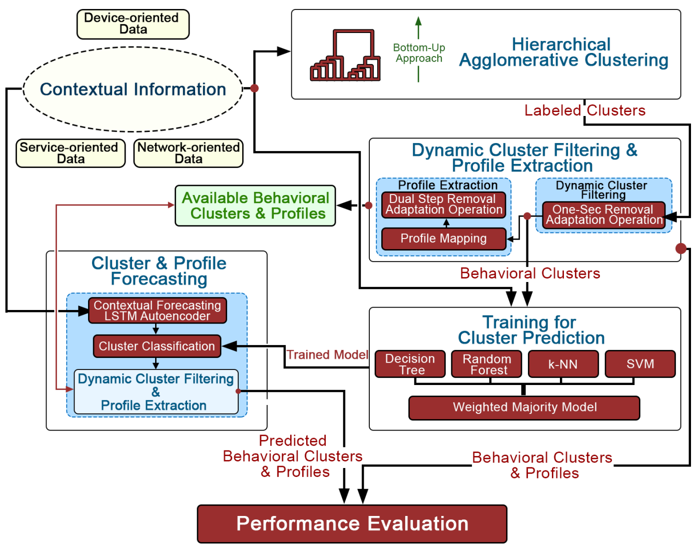


Fig. 2: Overview of PRIMATE

C. Methodology and Algorithm

1) Aggregation and pre-processing of Contextual Information: The first step in the development of PRIMATE is the aggregation and pre-processing of all the available contextual information from the different network entities and segments as well as the logging modules. This information is related to *Device*, *Service* and *Network* oriented data, which was thoroughly presented in Section III-A. This step is responsible for removing any potential irrelevant and/or redundant information present in the data set.

2) Hierarchical Agglomerative Clustering (HAC): The second step in the development of PRIMATE, as already discussed in III-B, is the extraction of the *Labeled Clusters* from the collected data features of the 3 different data entities presented in III-A. In order to successfully extract the set of *Labeled Clusters*, a well-known unsupervised clustering method used for grouping objects into clusters based on their similarity, is proposed in the current work. The selected clustering method is called Hierarchical Agglomerative Clustering (HAC). The name agglomerative relates to the "bottom-up" approach of this technique.

More precisely, each observation is initially considered to be a single-element cluster. At each step of the algorithm, the two clusters with the greater similarity are merged into a new cluster. This procedure is iterated until all points are merged into one single cluster. An indicative example of HAC is given in Figure 3. This method provides great flexibility, since having a pre-defined number of clusters is not a requirement. Additionally, HACs results are reproducible. In order to decide which clusters should be combined, a similarity measure is required. More specifically, the merging is based on an appropriate distance function, which is a measure of distance between pairs of observations and a linkage criterion, that specifies which clusters should be combined (linked) based on the distance information [17].

The HAC method was implemented for each one of the 3 data entities separately. At this stage a grid search process is introduced, in order to find an appropriate configuration (similarity measure, number of clusters) that best categorises the data features. The objective of the grid search process is to maximise a metric ranging from -1 to $+1$, that measures how similar a data feature is to its own cluster compared to other clusters, which is known as *Silhouette Value* [18]. A high silhouette value indicates that an object is well matched to its own cluster. The grid search process exhaustively searches for a combination of a similarity measure (distance function, linkage criterion) along with a number of clusters (selected from a predefined set of values), that results in the highest *Silhouette Value*. The final output of HAC is 3 labeled data sets (one data set per entity), in which all data features are mapped to *Labeled Clusters*.

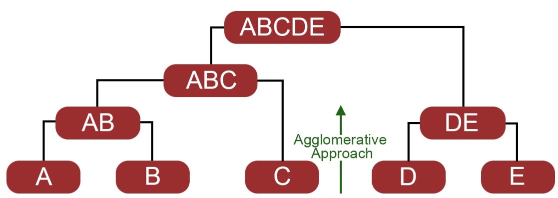


Fig. 3: Example of Hierarchical Agglomerative Clustering

3) Dynamic Cluster Filtering & Profile Extraction: The third step in the development of PRIMATE, is the implementation of 2 mechanisms named *Dynamic Cluster Filtering* and *Profile Extraction* respectively. The purpose of the first mechanism, is to remove the *Labeled Clusters*, which are considered to be outliers, from each one of the 3 data entities (Section III-A). In a given timeline in case of transitioning from one *Labeled Cluster* to a different one that has only 1 second of duration, the latter is considered to be falsely classified (outlier) and thus it needs to be replaced. Each outlier will be replaced by a *Labeled Cluster* that last appeared for more than 2 seconds. This replacement operation is named *One-Sec Removal Adaptation*. An indicative example is given in Figure 4. The final output of this mechanism is the filtered set of *Labeled Clusters* and it is called *Behavioral Clusters*. The second mechanism is responsible for a 2-fold operation. The first one is called *Profile Mapping* and is responsible for assigning labels to the *Behavioral Clusters*. This is achieved by mapping unique triplets into single unique labels. One triplet consists of three *Behavioral Clusters*, one for each of the 3 data entities that have been already presented in Section III-A.

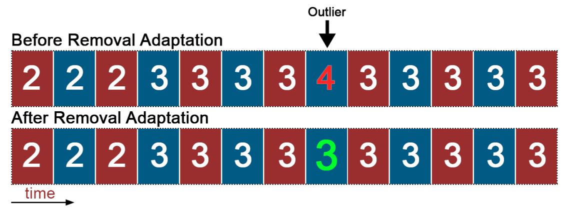


Fig. 4: Example of PRIMATE's One-Sec Removal Adaptation Operation

Each one of these unique labels is defined as a *Behavioral Profile*. An indicative example is given in Figure 5.

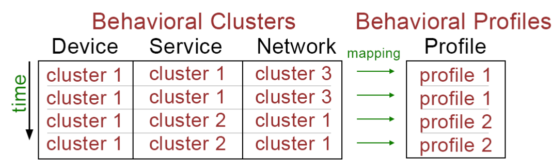


Fig. 5: Example of PRIMATE's Profile Mapping Operation

The second operation of *Profile Extraction* is called *Dual Step Removal Adaptation* and is responsible for the removal of *Behavioral Profiles*, which considered to be outliers. Once again, the same definition of the term outlier, that is used for the *Behavioral Clusters*, is also applied to *Behavioral Profiles*. This study considers these outlier *Behavioral Profiles* to be falsely extracted and thus they need to be replaced. The replacement method for the outlier *Behavioral Profiles* follows the same logic as the one presented for the *One-Sec Removal Adaptation*. This operation adds a second layer of filtering, after the first stage outlier filtering. The removal of the falsely extracted profiles may lead to additional reduction of the final *Behavioral Profiles* and as a result further enhance the quality of the output. Intuitively, the improvement in quality comes from the fact that fewer profiles with longer duration provide easier monitoring and management. The set of *Behavioral Clusters & Profiles* is being stored in a data pool named *Available Behavioral Clusters & Profiles* in order to be used as a mapping guide from the *Cluster & Profile Forecasting* module (Section III-C5).

4) Training For Cluster Prediction: The fourth step in the development of PRIMATE is the training of an ML model, which is capable of classifying contextual information - from each one of the 3 data entities - (Section III-A) into clusters. The selection of the model to be trained in order to predict clusters - and not profiles - was made in order to enforce the *Cluster & Profile Forecasting* module (Section III-C5) with the possibility to discover new triplets of clusters -and as a result new profiles (through the process of *Profile Mapping*). In case of profile prediction, such a scenario would not be possible and the number of profiles would be static. The heavy task of training the model is performed offline. Having trained the model offline, makes the prediction process suitable for near real-time operation. The trained model is stored and periodically updated in a database, in order to be exploited in a real-time manner by the *Cluster & Profile Forecasting* module. The development of this model is of high importance, as it provides PRIMATE the capability of exploiting contextual information in a supervised manner.

In order to properly train the aforementioned model, this work introduces an ensemble *Weighted Voting Model*, which combines the predictions from 4 different well-known supervised ML models (called base models), proposed by the literature [19], namely a Decision Tree (DT), a Random Forest (RF), a k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM). Ideally, the ensemble model will be able to achieve better performance than any of the base models and offer more robust predictions in different simulation scenarios. The term weighted derives from the fact that the RF algorithm along with the SVM has increased weighted vote importance (2-times the importance compared to the other base models) since they are the most robust algorithms among the 4 base ones considered. The performance of the aforementioned ensemble model has been evaluated using the *accuracy* and *f-1* score metrics. The final output of this module, named *Training for Cluster Prediction*, is a trained model, which will be exploited later on by the *Cluster and Profile Forecasting* module.

5) Cluster & Profile Forecasting: The fifth step in the development of PRIMATE is the forecasting of *Behavioral Clusters & Profiles*. This module comprises two different operations. The first operation, named *Contextual Forecasting - LSTM Autoencoder* relates to the forecasting of contextual information for each one of the 3 data entities described in Section III-A. The forecasting model is a Long Short Term Memory (LSTM) Autoencoder model for *multivariate multi-step time series* data [20]. The core functionality of this model is to forecast contextual information on multiple time-steps into the future (*multi-step*), using multiple prior time-steps of contextual information (historical contextual information). In this work, a time series model is used, since all the collected data points are equally distributed and indexed in time. The term *multivariate*, results from the fact that the collected contextual information consists of different data entities with different data features (Section III-A). Since PRIMATE is attempting to forecast time series of contextual information (future) using data, which is also structured in the same time series manner (past), the use of an Autoencoder is of high importance. The selected LSTM Autoencoder is an implementation of an Encoder-Decoder LSTM architecture and is designed to efficiently perform a sequence-to-sequence (seq2seq) prediction. A seq2seq approach imposes that the order in the data must be preserved when training the model and making predictions. The Encoder processes the input sequence of contextual information and encoding it into a fixed-length vector (*context vector*). The decoder takes as input the encoded vector, reconstructs the sequence and outputs the forecasted sequence of contextual information. In order to evaluate the forecasting capabilities of the proposed model, the Root Mean Squared Error (RMSE) is used as an error rate metric and is defined as the square root of the Mean Squared Error (MSE). The RMSE is directly interpretable in terms of units of measurement, and thus is a more suitable metric for a high performing model. Additionally, since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. The latter makes this metric quite appropriate in the case of forecasting, since large errors are particularly undesirable.

The second operation of the *Cluster & Profile Forecasting* mechanism is called *Cluster Classification*. This operation is responsible for exploiting the forecasted contextual information in conjunction with the exported *Weighted Voting* trained model from the previous step (Section III-C4) in order to perform cluster classification. The output of the model is the extracted clusters for each one of the 3 data entities. This set of extracted clusters may also contain outliers that need to be filtered. Once again, the outlier removal operation that was applied in this step is the *One-Sec Removal Adaptation*, introduced previously. The output of this removal operation is considered to be the *Predicted Behavioral Clusters*. These *Predicted Behavioral Clusters* should be mapped into the final profiles. In order to extract the final profiles, we initially map the filtered clusters into profiles and then add a second layer of filtering to remove any potential outlier profiles. This procedure is once again performed by applying the *Profile Extraction* mechanism introduced in the third step of PRIMATE (Section III-C3). In this case, the mechanism is using the *Available Behavioral Clusters & Profiles* pool as a mapping guide. The final extracted profiles of this mechanism are denoted as *Predicted Behavioral Profiles*. It is worth mentioning that it is possible for the *One-Sec Removal Adaptation* operation to output triplets of *Predicted Behavioral Clusters* that were not previously identified. As a result, the *Profile Extraction* mechanism will extract newly introduced *Predicted Behavioral Profiles*. These newly introduced behavioral clusters and profiles will be stored in the data pool introduced in III-C4.

6) Performance Evaluation: Finally, the sixth and final step in the development of PRIMATE is responsible for collecting and comparing the predicted *Behavioral Clusters & Profiles* along with the ground truth ones (extracted from step 3 of PRIMATE (Section III-C3)) in order to evaluate the performance of the proposed forecasting module.

In this final part of the study, an algorithmic representation (1) summarizes the methodology that was previously presented, while Table I provides the respective notation.

IV. Evaluation

This section presents the evaluation outcomes of the proposed framework. The necessary data that were used for the evaluation were generated using the discrete-event network simulator (NS-3) [21]. All the experiments were conducted using a single PC unit equipped with Ubuntu 16.04, Intel® Core™ i7-6800K Processor 6/12 and 32 GB of DDR4 RAM. The simulated topology chosen for this study was considered from a use case defined in one of the first 5G-related EU projects, namely METIS, [22], [23] and was implemented in NS-3. The simulated topology is depicted in Figure 6. The data produced by NS-3, simulate 10800 real-life seconds (3 Hours), while the simulation's estimated time of completion (ETC) is approximately 12 hours. The logging frequency of data is set to 1 second. The size of the area of experiments, in which the simulation was executed is equal to 400x200 m^2 with 10 equally sized sub-areas, in which a femto cell is placed

Parameter	Description
X	The Available unlabeled Contextual Information
X_d, X_n, X_s	Unlabeled Contextual Information for the Device, Network and Service data entities
$X^{HAC}/X_d^{HAC}, X_n^{HAC}, X_s^{HAC}$	HAC's labeled set / HAC's labeled dataset for the Device, Network and Service data entities
$X^{one\ \ sec}$	One-Sec Removal filtered set
$X_d^{one\ \ sec}, X_n^{one\ \ sec}, X_s^{one\ \ sec}$	One-Sec Removal filtered dataset for the Device, Network and Service data entities
$BP/BP^{(dual\_step)}$	Unfiltered set of behavioral profiles / Reduced set of Behavioral Profiles
WVM_d, WVM_n, WVM_s	Weighted voting models for the Device, Network and Service data entities
$X^{f(no\_lbl)}$	Unlabeled Forecasted Contextual Information
$X_d^{f(no\_lbl)}, X_n^{f(no\_lbl)}, X_s^{f(no\_lbl)}$	Unlabeled Forecasted Contextual Information for the Device, Network and Service data entities
X^f	Labeled Forecasted Contextual Information
X_d^f, X_n^f, X_s^f	Labeled Forecasted Contextual Information for the Device, Network and Service data entities
$X^{f(one\ \ sec)}$	One-Sec Removal filtered forecasted set
$X_d^{f(one\ \ sec)}, X_n^{f(one\ \ sec)}, X_s^{f(one\ \ sec)}$	One-Sec Removal filtered forecasted dataset for the Device, Network and Service data entities
BP^f	Unfiltered set of behavioral forecasted profiles
BC_d^f, BC_n^f, BC_s^f	Forecasted Behavioral Clusters for the Device, Network and Service data entities
$BP^{f(dual\_step)}$	Reduced set of forecasted Behavioral Profiles

TABLE I: PRIMATE's Framework Notation

in the middle. Additionally, there are 2 macro cells in the outer bounds of the area of experiments, which provide greater coverage if needed. There are 2 different mobility models implemented in the simulation. Each model has 2 parameters named velocity and path pattern. Both mobility models follow a Random-Walk process for their path selection. The velocity of the low as well as the medium mobility models is uniformly distributed in the ranges of $(0, 0.2]$ m/s and $(0.2, 1.2]$ m/s respectively. Table II summarizes the NS-3 parameters used in the simulated scenario.

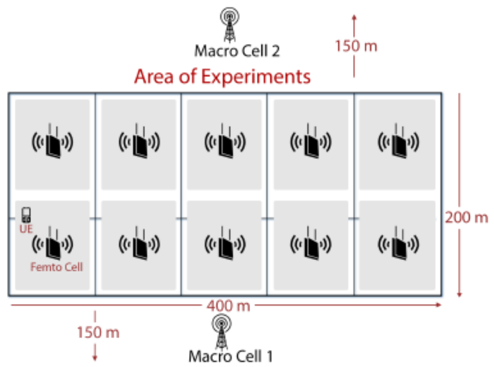


Fig. 6: NS-3 Virtual Topology

In the current simulation scenario a single eMBB UE consumes 3 different kinds of simulated services, namely Instant File Sharing (IFS), High Quality Live-Streaming (HQLS) and Fast Browsing (FB); the traffic modeling specifications of the service types are summarised in Table III [24]. During the simulation time, services are randomly assigned to the user and their duration is modelled following a Normal Distribution $X \hookrightarrow \mathcal{N}(\mu, \sigma^2)$), where $(\mu, \sigma^2) = \{(30, 10),(45, 10), (40,15)\}$ for each one of the abovementioned services respectively. The selection concerning the mean value and variance aims at offering a rather small but distinctive variation in the duration for each one of the services.

A. Hierarchical Agglomerative Clustering and Dynamic Cluster Filtering & Profile Extraction evaluation

This section focuses on the evaluation of the HAC along with the implemented filtering mechanisms introduced in Section III-C2 and III-C3 respectively. Overall, for the implemented grid search concerning the HAC, several number of clusters were tested in order to efficiently capture the variance in the data along with 1 distance function and 4 linkage criterion. More specifically, the number of clusters tested were selected from the set {*3, 4, 5, 6, 7*}, while the *Euclidean distance* was selected as distance function. Finally, the 4 linkage criteria were selected from the set {*Ward, Average, Complete, Single*}. The *ward* criterion refers to the minimisation of the variance of the clusters being merged, while the *average* criterion uses the average of the distances of each observation of the two sets. The *complete* linkage criterion uses the maximum distances between all observations of the two sets. Finally, *single* linkage uses the minimum of the distances between all observation of the two sets.

Figure 7 presents the *Silhouette Scores* for the best parameters selected based on the results of the implemented grid search. For the *Device* entity the Silhouette score is nearly 0.6. The selected number of clusters is set to 4 while the linkage criterion is set to *Average*. The *Silhouette Score* for this entity is rather low indicating clustering results of low accuracy. This is due to the fact that the data features of the *Device* are far less logically correlated (e.g no correlation between geolocation and battery status) and have no obvious repeating patterns for the HAC method to identify (e.g random-walk process for path selection, random velocity changes). Regarding the *Service* entity the number of clusters selected is set to 3 while the linkage criterion is set to *Ward*. Finally, the selected number of clusters for the *Network* entity is set to 3 with the *Ward* being the selected linkage criterion. For both the *Network* and *Service* data entities, the *Silhouette Scores* are almost identical and close to 1 indicating that the data entities have been separated into different clusters in an almost optimal manner. This can be explained by the fact that in the current simulated

Algorithm 1: PRIMATE's Framework

Input Data: $X = \{X_d, X_n, X_s\}$
Output Data: $BC_d^f, BC_s^f, BC_n^f, BP^{f(dual\_step)}$
Hierarchical Agglomerative Clustering
foreach *entity* $\in X$ **do**
  perform HAC
retrieve labeled unfiltered set
$X^{HAC} = \{X_d^{HAC}, X_n^{HAC}, X_s^{HAC}\}$
Dynamic Cluster Filtering & Profile Extraction
foreach *entity* $\in X^{HAC}$ **do**
  perform One-Sec Removal Adaptation Operation
retrieve filtered set
$X^{one\ \ sec} = \{X_d^{one\ \ sec}, X_n^{one\ \ sec}, X_s^{one\ \ sec}\}$
apply Profile Mapping to $X^{one\ \ sec}$ and retrieve set of behavioral profiles BP
foreach $p \in BP$ **do**
  perform Dual Step Removal Adaptation Operation
retrieve reduced set of behavioral profiles
$BP^{(dual\_step)}$
Training for Cluster Prediction
foreach *entity in* $X^{one\ \ sec}$ **do**
  perform training of the Weighted Voting Model
retrieve weighted voting models
WVM_d, WVM_n, WVM_s
Cluster & Profile Forecasting
foreach *entity* $\in X$ **do**
  perform forecasting using LSTM
retrieve unlabeled forecasted set
$X^{f(no\_lbl)} = \{X_d^{f(no\_lbl)}, X_n^{f(no\_lbl)}, X_s^{f(no\_lbl)}\}$
foreach *entity* $\in X^{f(no\_lbl)}$ **do**
  perform classification using trained models WVM_d, WVM_n, WVM_s
retrieve labeled unfiltered forecasted set
$X^f = \{X_d^f, X_n^f, X_s^f\}$
foreach *entity* $\in X^f$ **do**
  perform One-Sec Removal Adaptation Operation
apply Profile Mapping to $X^{f(one\ \ sec)}$ and retrieve set of unfiltered forecasted behavioral profiles BP^f
retrieve BC_d^f, BC_n^f, BC_s^f from $X^{f(one\ \ sec)}$
foreach $p \in BP^f$ **do**
  perform Dual Step Removal Adaptation Operation
retrieve reduced set of forecasted behavioral profiles
$BP^{f(dual\_step)}$
return set of forecasted behavioral clusters & profiles
$BC_d^f, BC_n^f, BC_s^f, BP^{f(dual\_step)}$

Parameter Description	Default Value
# of UEs	1
# of Femto Cells	10
# of Macro Cells	2
Mobility Models	2 (Low, Medium)
UEs' Transmission Power	20 dBm
Femto Cells' Transmission Power	20 dBm
Macro Cells Transmission Power	35 dBm
Macro Cells Downlink and Uplink Bandwidth	20 MHz
Femto Cells Downlink and Uplink Bandwidth	20 MHz

TABLE II: Parameters Used in the ns-3 Simulated Scenario

Service	Transfer Protocol	packet interval DL/UL	packet size DL/UL
IFS	UDP	2 / 2000 ms	600 / 12 bytes
HQLS	UDP	2 / 2 ms	1400 / 1400 bytes
FB	TCP	5 / 1000 ms	1100 / 12 bytes

TABLE III: Traffic modeling specifications of the simulated services

scenario there are 3 different simulated services (3 clusters for the entity *Service*) and the *Network* entity is comprised of data features concerning the traffic load of 3 services in the network (3 clusters for the entity *Network*). To this end, the rest of the evaluation is based on the aforementioned parameters' configuration.

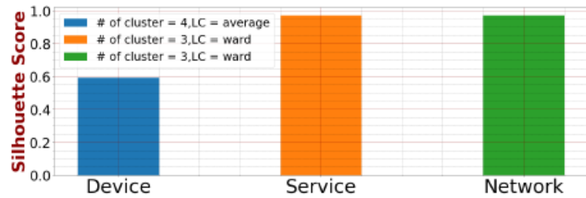
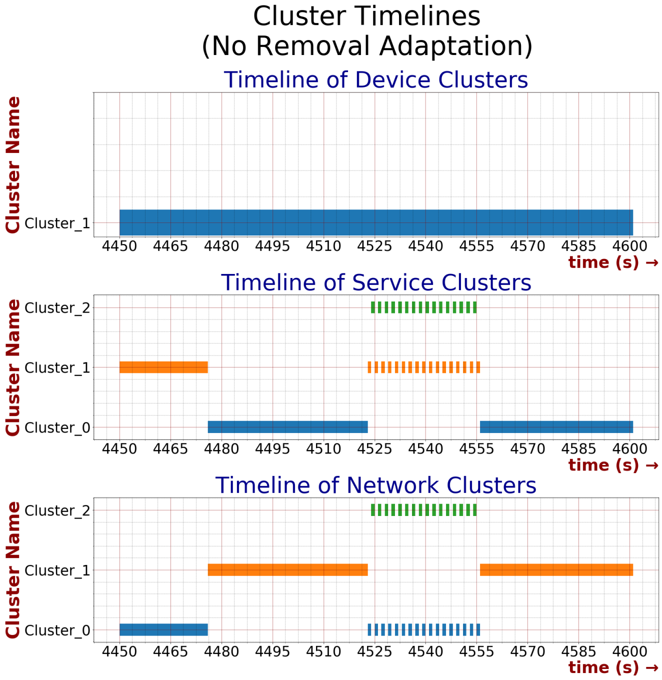


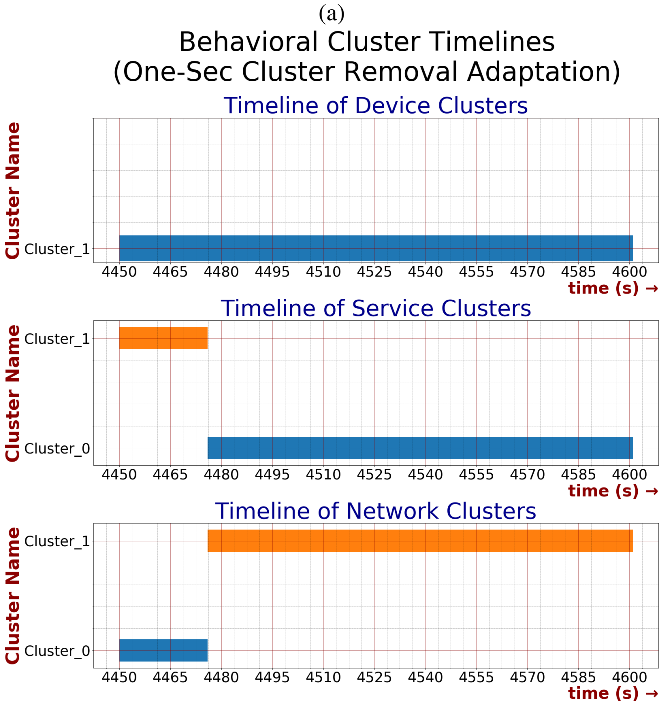
Fig. 7: HAC Performance (Best Silhouette Scores) for the 3 data entities: Device, Network, Service

The next part that is presented in this section relates to the filtering mechanisms introduced in Section III-C3. More specifically, Figure 8a depicts 3 vertically stacked excerpt timelines (one for each entity), in which all the different clusters, that were extracted from the HAC, are arranged based on their chronological occurrence. As it can be seen in the figure, the y-axis depicts the names of the different clusters and the x-axis depicts the duration of the timeline in seconds. In the time-window [4523,4550] the *Service* and *Network* data entities are rapidly changing clusters every second. These short-lasting clusters will be candidates for the outlier removal operation that follows. Figure 8b illustrates the behavioral cluster timelines as they are formed after the enforcement of the One-Sec Removal Adaptation Operation. As it can be inferred from the figure, the aforementioned operation manages to eliminate all the short-lasting clusters. Overall, the elimination process (*One-Sec Cluster Removal Adaptation*) for the entire simulation, results in 3, 2 and 2 behavioral clusters for the *Device*, *Service* and *Network* data entities respectively. The results of the filtering mechanism indicate that the output of the HAC could be improved (in terms of number of clusters) despite of the high *Silhouette Scores*. This can be explained by the fact that two (IFS, FB) out of the three simulated services have similar traffic load modeling, and - as a result - similar behavior. As a result, the One-Sec removal is able to identify this similarity and remove the extra cluster for both *Service* and *Network* data entities.

Figures 9a, 9b and 9c and present the Profile Timelines that result from the implementation of the *Profile Mapping* operation. More specifically, Figure 9a depicts over time the different profile occurrences, using as an input the clusters that were extracted from the HAC. Once again, the y-axis depicts the names of the different profiles, while the x-axis depicts the duration of the timeline in seconds (10800 secs in total).



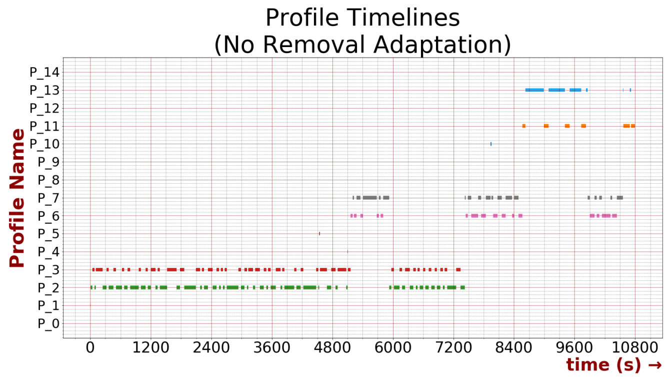
(a)



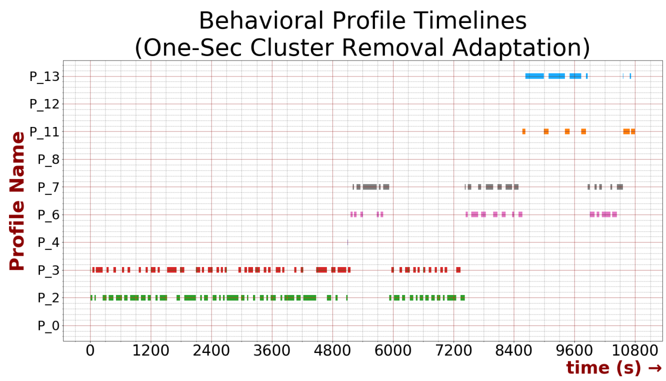
(b)

Fig. 8: Cluster Timelines: **(a)**: Timelines as extracted from HAC **(b)**: Timelines after the enforcing of the One-Sec Removal Adaptation operation (Behavioral Cluster Timelines)

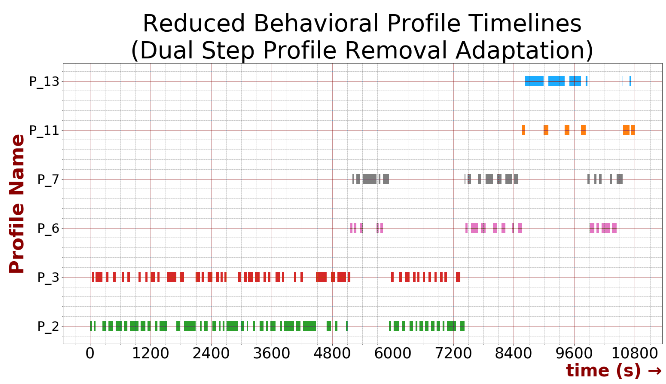
It must be noted that some profiles appear to have no duration (P_0, P_1, P_4, P_8, P_10, P_12, P_14). This inability to depict the duration for some of the profiles, is explained by the fact that it is minuscule for such a large scale timeline. Again, these short-lasting profiles will be candidates for the outlier removal operation that follows. Figure 9b depicts the behavioral profile timelines, as they are formed using the resulted clusters of the One-Sec Removal Adaptation operation as an input. As it can be inferred from the figure, there is a clear reduction in the number of the extracted profiles. Figure 9c depicts the reduced behavioral profile timelines, which result from the additional layer of filtering, namely *Dual Step Removal Adaptation* operation. The result is a clean timeline with 6 behavioral profiles in total. Finally, Figure 10 compares the different number of extracted profiles, under the above-mentioned filtering mechanisms. As it can be seen in the figure, there are 15 different profiles in total when no filtering mechanism is applied. After the application of the *One-sec* and *Dual Step Removal Adaptation* operations, the different number of profiles are clearly reduced and equal to 10 and 6 respectively. As previously discussed in Section III-C3, this reduction further improves the output quality of the the *Dynamic Cluster Filtering & Profile Extraction* module.



(a)



(b)



(c)

Fig. 9: Profile Timelines: **(a)**: with no Removal Adaptation **(b)**: with One-Sec Removal Adaptation (Behavioral Profiles) **(c)**: with the Dual Step Removal Adaptation (Reduced Behavioral Profiles)

Fig. 10: Dynamic Removal Adaptation Performance

Entity Name	Accuracy	f1-Score
Device	96%	92%
Service	92.72%	91.72%
Network	93.68%	89.68%

TABLE IV: Performance of the Weighted Voting Model

B. Cluster & Profile Classification evaluation

This section focuses on the evaluation of the Weighted Voting Model introduced in Section III-C4. The weighted model combines the predictions from 4 different base models (DT, RF, k-NN, SVM) in order to potentially improve on the performance delivered by each of these base models. The weight of each vote is equal to 1 for the DT and k-NN, while the voting weight for the case of RF and SVM models is doubled since they are the most robust algorithms among the rest. All base models have been optimized using a grid search process in order to identify the best parameters that result in the highest performance. Table IV depicts the performance of the aforementioned weighted model. The performance was measured using the accuracy and f1 score metrics. More specifically, as it is shown in the table, the models achieve accuracy equal to to 96%, 92.72% and 93.68% for the *Device*, *Service* and *Network* data entities, respectively. In the same manner, the f1 score for the 3 data entities is equal to 92%, 91.72% and 89.68% respectively. This high performing model will be exploited in the *Profile Forecasting* section that follows.

C. Cluster & Profile Forecasting evaluation

In this last part of the evaluation, the results from the cluster and profile forecasting module (Section III-C5) are presented. After extensive experimentation regarding the training phase of the 3 LSTM models (one per entity), the batch size was set to 64 and the total number of epochs to 250. In order to prevent any degradation in the performance of the validation set and as a result overfitting, an early stopping function was applied. This function monitors the MSE of the validation set and stops the training when a minimum value is reached. To increase the performance of the models the Stochastic Gradient Descent (SGD) with momentum was selected as an optimiser with learning rate, $lr = 0.001$. All models used L2 (weight decay) regularisers of 10 ^{6}. Finally, the time window selected for forecasting was set to 30 seconds using a 30 second prior time window of contextual information as an input. The split ratio for the dataset was set to 0.8, meaning that 80% of the available data was used for model training and the rest 20% for evaluation. As a result, 8640 seconds were used for the training phase and 2160 for the evaluation. After segmenting the training and evaluation set into 30 second time windows, we result with 288x30 and 72x30 seconds of data respectively.

Figure 11 depicts 3 excerpts from the performance evaluation of the 3 LSTM models. Figure 11a presents the RMSE for the Device's data feature named *velocity* for all the 30 second forecasted time windows. More specifically, each $t+i$ depicts the RMSE for the i_{th} second for each one of the 72 forecasted windows. The plot shows that the seconds in the range $[t+1, t+6]$ are easier to forecast while the seconds in the range $[t+10, t+16]$ are the hardest. The overall RMSE for this data feature is about 0.0045 m/s and indicates a good performance of the model since the average value of velocity for the evaluation dataset is equal to 0.129 m/s. In a similar way, Figure 11b presents the RMSE for the Services's data feature named *Downlink Packet Size* in bytes for all the forecasted windows. The easiest seconds to forecast are in the range $[t+1, t+6]$ while the hardest are the $t+14$ second and the range $[t+24, t+30]$. The overall RMSE results in 278.97 bytes, while the average value in the evaluation dataset is equal to 1118 bytes, which indicates an acceptable performance of the model. Finally, Figure 11c illustrates the RMSE for the Network's data feature named *Downlink Delay*. This time, the hardest seconds to forecast are included in the set $\{t+12, t+20, t+26\}$, while all the other seconds are forecasted with higher accuracy. The overall RMSE for this feature is at 0.001 seconds having an average value of 0.0047 seconds in the evaluation dataset, which once again indicates good forecasting capabilities. The fluctuations in the results are related to the stochastic nature of the LSTM model. Additionally, a larger data set as well as a larger prior time window could potentially further decrease the overall RMSE and increase the performance of the model.

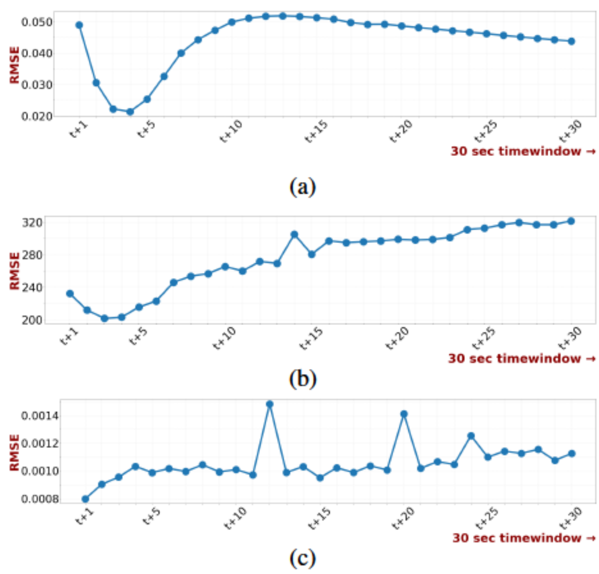
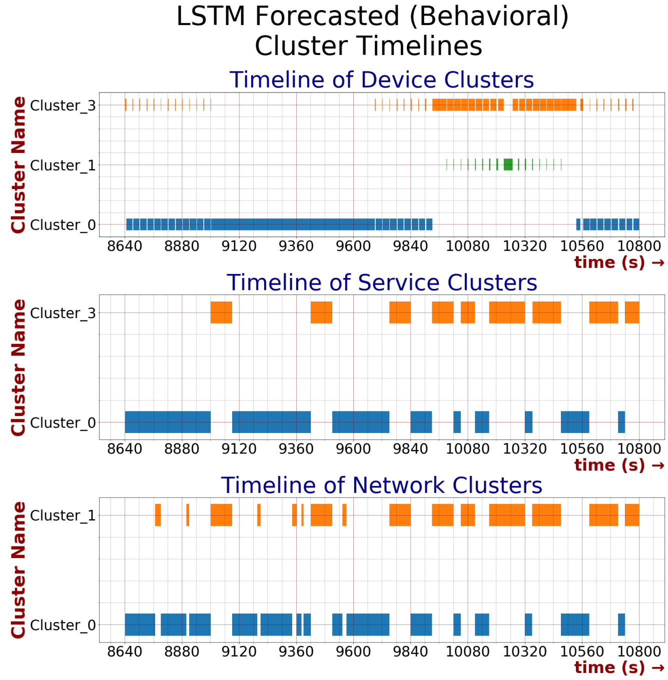


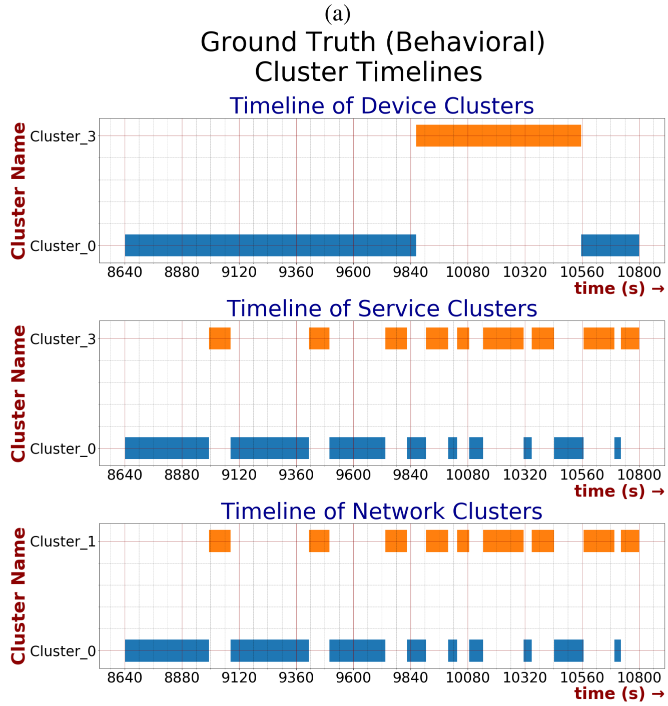
Fig. 11: Evaluation of LSTM using RMSE per timestep for specific data features: **(a)**: Device's Velocity in m/s **(b)**: Service's Packet Size in bytes **(c)**: Network's Downlink Delay in seconds

Having evaluated the performance of the LSTM models, Figure 12 illustrates the behavioral clusters (top), in comparison to the ground truth ones (bottom). More specifically, Figure 12a depicts the behavioral clusters, which are the forecasted cluster timelines for each one of the 3 data entities that were extracted with the use of the already introduced and evaluated (Section IV-B) *Weighted Voting* model for the 2160 (72x30) seconds of the evaluation dataset in conjunction with the *One-Sec Removal Adaptation* operation. Figure 12b depicts the actual behavioral cluster timelines (ground truth) for the same time period as they were extracted at a previous step of PRIMATE (Section IV-A). As it can be inferred from the two figures, there is a high similarity between them, indicating that once again the good performance of the forecasting mechanism. To be more precise, Figure 14 shows that the accuracy for the cluster forecasting is 88.29% for the Device entity, 87.71% for the Service entity and 82.59% for the Network entity.

The last step of the forecasting module is the application of the *Profile Extraction* mechanism in order to map the behavioral clusters into the final behavioral profiles. Figure 13 presents the final behavioral profiles and the actual ones in order to illustrate the performance of the *Profile Extraction* mechanism. Figure 13a, illustrates the behavioral cluster timelines for the 2160 (72x30) seconds of the evaluation dataset in conjunction with the *Dual Step Removal Adaptation* operation. In this particular evaluation phase, a newly-introduced profile, namely *NP_0* is falsely forecasted according to the ground truth timeline. However, the aforementioned profile illustrates the capability of the module to forecast profiles, which were never observed before. Figure 13b presents the actual behavioral profile timelines (ground truth) for the same time period, which were extracted and discussed in Section IV-A. Overall, the Profile Forecasting performance amounts to 71.25%, as it can be seen in Figure 14. The profile forecasting performance is lower than the forecasting accuracy for each one of the 3 entities. The reason for this decrease lies on the fact that in order to successfully forecast a final profile, firstly we need to accurately forecast the behavioral clusters for the 3 entities. Then, we use the *Profile Mapping* operation in order to extract the final profiles, based on which the profile forecasting accuracy will be calculated. Intuitively, this step lowers the final accuracy of the final forecasting. Overall, the results of PRIMATE illustrated the ability of the framework to forecast behavioral profiles with high accuracy. As presented in Section III-C, PRIMATE comprises multiple modules, which in turn offers high flexibility and adaptability in different environments. As a result, the performance of PRIMATE could be improved by exploiting different clustering techniques like k-Means and Hierarchical Density-based spatial clustering of applications with noise (HDBSCAN) [25]. Additionally, the Weighted Voting model could be enriched by including new base models like Naïve Bayes, Multilayer-Perceptron (MLP) and Stochastic Gradient Descent (SGD) that could potentially improve the performance of the voting model and as a result the performance of the Cluster & Profile Forecasting module.



(a)



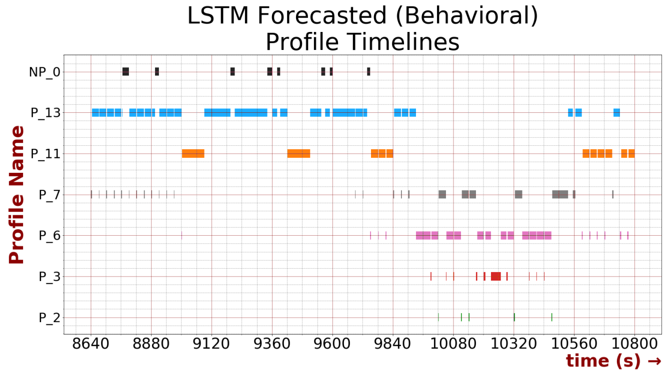
(b)

Fig. 12: Cluster Timelines: **(a)**: Forecasted (Behavioral) Clusters **(b)**: Ground Truth (Behavioral) Clusters

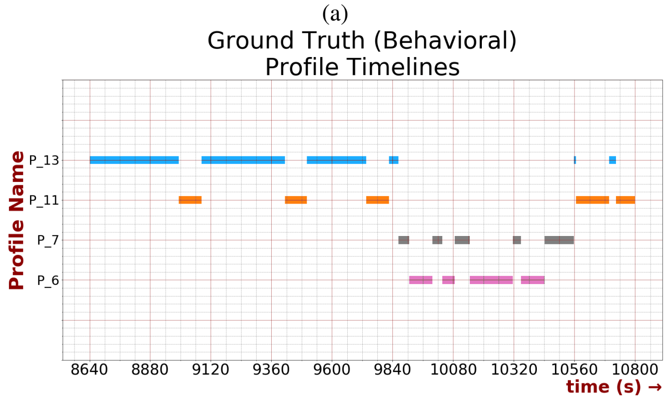
V. Discussion and Conclusions

This work presented a novel framework for AI-driven, profiling and network behavior forecasting of heterogeneous 5G and beyond devices and *things*. A comprehensive methodology was presented, which described step-by-step the modules and algorithms applied to the collected contextual network information towards profile extraction and forecasting. Novel definitions of entity clusters and profiles were introduced, described by a detailed data model. Lastly, a detailed evaluation study showed the effectiveness and viability of the proposed scheme.

As part of the next steps of this work, extended evaluation



(a)



(b)

Fig. 13: Profile Timelines: **(a)**: Forecasted (Behavioral) Profiles **(b)**: Ground Truth (Behavioral) Profiles

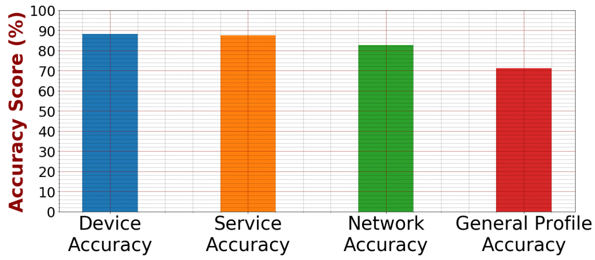


Fig. 14: Performance of Cluster & Profile Forecasting Mechanism

scenarios have already been planned in order to evaluate the performance of PRIMATE for more complex scenarios and network deployments, which will comprise heterogeneous types of *things*, users and service types.

Last but not least, one of the major next objectives of the specific work is to integrate the specific framework with a set of resource allocation algorithms in different layers of the network (such as radio resource management, VNF placement and scaling, etc.) and exploit the profile forecasting outputs for proactive resource management. Particularly, for beyond 5G network environments with several co-existing, heterogeneous types of devices and services, considerable gains are expected.

Acknowledgments

This work has been partially supported by EC H2020 5GPPP 5Growth project (Grant 856709).

References

[1] Auto-id labs homepage. [Online]. Available: http://www.autoidlabs.org/

[2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless- and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.

[3] 3rd Generation Partnership Project (3GPP), "System Architecture for the 5G System (Release 16)," Technical Specification (TS) 23.501, August 2020.

[4] ETSI, "MEC in 5G Networks, First edition," White Paper No. 28, June 2018.

[5] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0," *Information Fusion*, vol. 50, pp. 92 – 111, 2019.

[6] M. Paolanti, L. Romeo, A. Felicetti, A. Mancini, E. Frontoni, and J. Loncarski, "Machine learning approach for predictive maintenance in industry 4.0," in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2018, pp. 1–6.

[7] D. Jiang, Y. Wang, Z. Lv, S. Qi, and S. Singh, "Big data analysis based network behavior insight of cellular networks for industry 4.0 applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1310–1320, 2020.

[8] P. Magdalinos, S. Barmpounakis, P. Spapis, A. Kaloxylos, G. Kyprianidis, A. Kousaridas, N. Alonistioti, and C. Zhou, "A context extraction and profiling engine for 5g network resource mapping," *Computer Communications*, vol. 109, pp. 184 – 201, 2017.

[9] J. M. Valtorta, A. Martino, F. Cuomo, and D. Garlisi, "A clustering approach for profiling lorawan iot devices," in *Ambient Intelligence*, 2019, pp. 58–74.

[10] L. Le, D. Sinh, B. P. Lin, and L. Tung, "Applying big data, machine learning, and sdn/nfv to 5g traffic clustering, forecasting, and management," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 168–176.

[11] U. Karneyenka, K. Mohta, and M. Moh, "Location and mobility aware resource management for 5g cloud radio access networks," in *2017 International Conference on High Performance Computing Simulation (HPCS)*, 2017, pp. 168–175.

[12] M. S. Parwez, D. B. Rawat, and M. Garuba, "Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2058–2065, 2017.

[13] S. Barmpounakis, A. Kaloxylos, P. Spapis, C. Zhou, P. Magdalinos, and N. Alonistioti, "Data analytics for 5g networks: A complete framework for network access selection and traffic steering," 11 2018.

[14] L. Xu and R. Duan, "Towards smart networking through context aware traffic identification kit (trick) in 5g," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, 2018, pp. 1–6.

[15] X. Duan, Y. Liu, and X. Wang, "Sdn enabled 5g-vanet: Adaptive vehicle clustering and beamformed transmission for aggregated traffic," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 120–127, 2017.

[16] M. Joud, M. García-Lozano, and S. Ruiz, "User specific cell clustering to improve mobility robustness in 5g ultra-dense cellular networks," in *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2018, pp. 45–50.

[17] L. Rokach, *A survey of Clustering Algorithms*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2010. [Online]. Available: https://doi.org/10.1007/978-0-387-09823-4_14

[18] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257

[19] I. Sarker, A. S. M. Kayes, and P. Watters, "Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage," *Journal of Big Data*, vol. 6, 07 2019.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[21] Official ns-3 website. [Online]. Available: https://www.nsnam.org/

[22] The metis 2020 project – laying the foundation of 5g, 2016. [Online]. Available: https://www.metis2020.com

[23] Scenarios, requirements and kpis for 5g mobile and wireless system. [Online]. Available: https://www.metis2020.com/wp-content/uploads/deliverables/METIS_D1.1_v1.pdf

[24] G. Soos, D. Ficzere, and P. Varga, "Towards traffic identification and modeling for 5g application use-cases," *Electronics*, vol. 9, no. 4, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/4/640

[25] M. Ester, H. P. Kriegel, J. Sander, and X. Xiaowei, "A density-based algorithm for discovering clusters in large spatial databases with noise," 12 1996. [Online]. Available: https://www.osti.gov/biblio/421283