IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 February 2024, accepted 15 April 2024, date of publication 23 April 2024, date of current version 1 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3392624

RESEARCH ARTICLE

Machine Learning-Based Cellular Traffic Prediction Using Data Reduction Techniques

HEBA NASHAAT[1], (Member, IEEE), NIHAL H. MOHAMMED[1], SALAH M. ABDEL-MAGEID[2,3], AND RAWYA Y. RIZK[1], (Senior Member, IEEE)

[1]Electrical Engineering Department, Port Said University, Port Said 42523, Egypt
[2]College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia
[3]Computer Engineering Department, Al-Azhar University, Cairo 11884, Egypt

Corresponding author: Heba Nashaat (hebanashaat@eng.psu.edu.eg)

ABSTRACT Estimating and analyzing traffic patterns become essential in managing Quality of Service (QoS) metrics while assessing internet data traffic in cellular networks. Cellular network planners frequently apply various approaches to predict network traffic. However, most existing studies focus on using the available local data to jointly build prediction models, facing data security challenges and time complexity, especially with multi-dimensional datasets. Therefore, this paper proposes a framework to handle traffic prediction with the considerable potential of Machine Learning (ML) algorithms. An Adaptive Machine Learning-based Cellular Traffic Prediction (AML-CTP) framework is presented to select a suitable ML algorithm for multi-dimensional datasets. Its objective is to streamline and speed up the selection of an appropriate model for predicting network traffic load. The framework employs two density-based clustering algorithms to categorize similar nearby traffic into various clusters, considering data similarity and convergence. Additionally, it assesses data quality and homogeneity by training models with data samples from each cluster to accurately determine the most suitable machine learning model. The optimal model is selected from four supervised predicting algorithms, reducing training time and hardware complexity. Two case studies from a popular telecommunication equipment corporation in Egypt are implemented using real-life cellular traffic with multi-dimensional features. The case studies show that the framework can help reduce the computational cost of training the model and reduce the risk of overfitting. The experimental results show that selecting the best prediction model training could save up to 85% of computational time compared to two state-of-the-art techniques while achieving an accuracy of 98.8%.

INDEX TERMS 4G/LTE, KPIs, machine learning, traffic prediction.

I. INTRODUCTION

Nowadays, a rash growth of traffic data exists due to the rise of smartphone subscriptions and streaming video services. As a result, this influences the Quality of Service (QoS) of network users. Therefore, many network-level optimizations should be performed to maintain the best QoS for users. However, an optimization problem is challenging for the best QoS that adjusts the transmission power [1], [2], [3]. Steering traffic from congested cells frees the physical resource blocks (PRBs) of congested cells and is considered a network that optimizes resource allocation [4], [5]. Hence, predicting 4G and 5G LTE-A traffic became essential to significantly enhance telecommunications QoS [6], [7]. Accurate monitoring and prediction of mobile traffic help improve optimizer goals; consequently, any overload or congestion in any band can be detected [3], [8].

Over the past decades, Machine Learning (ML) has become a crucial backbone of information technology. However, with the significant increase in data sizes, training time for models can range from hours to weeks. Hence, it poses intense pressures across computation, networking,

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Fadda.

and storage [5]. In turn, this affects the choice of the ML model. Also, the nature of the data, which includes correlation, distribution, and homogeneity, influences the selection process of the final model. Since ML deals with the data and behaves accordingly without being programmed, data analysis is required to evaluate and estimate the benefits of optimizer goals. Practical data analysis leads to the successful choice of the most accurate prediction algorithm and saves computational time and memory usage [9], [10], [11].

Therefore, analyzing the similarity, closeness, and homogeneity in traffic data is essential to selecting the appropriate ML model. Moreover, clustering techniques are applied to large datasets, so the data is partitioned into clusters containing similar elements. Then, an extraction rule is estimated based on the pattern of occurrence of data tokens [12], [13]. In addition, for unknown traffic, dividing data into classes requires more information on the nature of the traffic. Thus, clustering methods are used to gain some perception of the structure of the data. The optimal clustering algorithm collects data in one cluster when the data is homogeneous and similar. A smaller sample size may be present in the data when the clusters are more heterogeneous. This sample size can sufficiently capture the underlying patterns and relationships in the data [14].

Therefore, this paper introduces the Adaptive Machine Learning-based Cellular Traffic Prediction (AML-CTP) framework. The AML-CTP aims to facilitate and speed up the selection of a suitable model for network traffic load prediction. The framework considers the similarity and convergence of the data by applying two density-based clustering algorithms to categorize similar near traffic in various clusters. In addition, it considers the data quality and homogeneity by training these models with data samples from each cluster to get the exact indication of the best ML model. Finally, the best model is chosen from four supervised predicting algorithms, reducing the training time and hardware complexity.

An overall view of the proposed AML-CTP framework is illustrated in Figure 1. It starts by preprocessing the collected data from real-life key Performance Indicators (KPIs). The prepossessing process includes analyzing the data, selecting uncorrelated traffic features, and data visualization [15], [16]. Then, the framework applies the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm [17], [18] and the Kernel Density Clustering (KDC) algorithm [19], [20] to split the data into clusters. Then, the output clusters are fed to the co-association matrix to check data similarity [21], [22], [23], [24]. After that, the AML-CTP extracts a data sample with the same probability distribution as the original population. Finally, these data samples are used to train the ML models and select the most suitable one.

Four ML prediction models are used to attain the best performance and aggregate predictions. These models are Linear Regression (LR), Support Vector Regression (SVR), Decision Tree (DT), and Light Gradient Boosting Machine Model (LGBM). Then, a separate validation dataset is evaluated to ensure the models can generalize well to new, unseen data to avoid the risk of overfitting and improve the overall accuracy and robustness of the model. Two case studies are examined using real-life KPIs collected through one of Egypt's most popular telecommunication equipment corporations. Both case studies utilize large, high-dimensional LTE-A key performance indicators datasets.

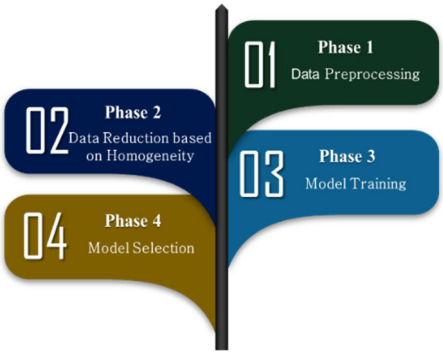


FIGURE 1. An overall view of the proposed framework.

The rest of this paper is organized as follows: Section II represents related work of cellular mobile traffic prediction models. Section III describes the proposed framework's details. Section IV introduces two real case studies and discusses the experimental results. Finally, the main conclusion and future work are presented in Section V.

II. RELATED WORK

Machine learning can play a valuable role in various aspects of telecommunication technologies, including enhancing services for end-users on the network. With the ongoing rise in the number of users and the corresponding increase in data size, it becomes necessary to analyze data using statistical processing methods to reduce the expenses associated with manual analysis. These methods are also effective in identifying errors and data noise. Therefore, incorporating machine learning techniques can optimize network operations and contribute to the early prediction of network failures, preventing significant degradation of the network's Quality of Service.

Initially, network operators gather available datasets from the cellular network operator for monitoring purposes [25]. A detailed examination of mobile datasets can be found in [26]. However, constructing predictive models with mobile datasets poses numerous challenges. The primary issue stems from the diversity of mobile traffic datasets. Additionally, varying data collection approaches can produce datasets of differing precision, adding complexity to the machine learning training process. Furthermore, the substantial growth in 5G and 6G introduces several challenges in handling big data and selecting the optimal prediction model. Consequently, identifying the most suitable prediction model is not always straightforward, particularly when dealing with extensive and high-dimensional datasets.

Existing research has explored the use of ML in optimizing the performance of wireless networks. For example, in [27], an experimental study assesses popular supervised and unsupervised learning methods. The experimental analysis in [27] identifies SVM, LR, and DT as the most accurate algorithms among the supervised learning approaches. Another study [28] examines existing techniques for traffic prediction that leverage machine learning, offering concise descriptions of traditional machine learning models and methods in this survey. This rationalizes utilizing these machine learning algorithms in the proposed framework for traffic prediction.

Moreover, another study [29] utilized the Radio Environment Map as the data source, which depicts a coordinate plane with X and Y axes, representing the spatial distribution and intensity of a specific feature using a thermal strip. Then, the authors augmented the dataset with information on the mobile drive test unit's speed to forecast the network performance. Chaudhary and Johari [30] introduced a novel ML-based algorithm for routing in wireless networks. They gathered a small and imbalanced dataset, employing random oversampling to augment sample numbers and balance the dataset. The proposed algorithm utilized supervised machine learning methods to predict the network type of the source and destination nodes. Also, Sliwa et al. [31] employed ML algorithms to predict data rates, focusing on the concept of utilizing cars as mobile sensors. The authors suggested a method that considers the current channel situation to determine the optimal time for the transaction.

In general, existing network traffic load prediction solutions [32], [33], [34], [35], [36], [37] mainly depend on Deep Learning (DL) [38] to build the prediction model. A deep learning model can handle high-dimensional traffic datasets. Also, they can usually obtain high accuracy for long-term prediction. During the last few years, different types of DL techniques have been applied to network traffic load prediction; those techniques mainly depend on deep learning, neural networks, or a combination of both techniques. Long short-term memory (LSTM) based solution is introduced in [32]. Also, LSTM and Convolutional Neural Networks (CNN) collaborative solutions are in [33]. In [34], Gated Recurrent Units (GRU) and CNN combined solutions are employed. Deep Belief Networks (DBN) based solutions in which estimation solutions based on DBN are introduced in [35] and [36]. A stacked Auto-Encoder (SAE) based solution that proposed a Downlink-based traffic prediction method that uses a Stacked Demising Auto-encoder (SDA) model to learn generic traffic features is in [37].

Moreover, the direct advantages of clustering to improve prediction accuracy through data quality enhancement are explored in [39]. However, these investigations focus on distinct, non-uniformly distributed data. The widely used K-means clustering algorithm is applied, generating multiple clusters for a finite dataset. There is no definitive optimal clustering of the data; specific clusters may be unnecessary, some might signify sampling noise, and specific information could be exclusive to a particular grouping. Nonetheless, the emergence of novel information from these clusters provides an opportunity to employ a different clustering algorithm in prediction tasks, mainly when dealing with homogeneous data.

However, the primary limitations of these systems involve substantial computational load and complex parameter setup. Ensemble learning approaches have also been incorporated, utilizing an integrated LGBM model through bagging and LGBM [41]. Nonetheless, ensuring the accuracy of predictions in ensemble learning-based solutions poses a challenge due to the requirement for genuine training data derived from the target environment [36]. Hence, further research is required to explore the significance of implementing machine learning in innovative architectural systems to gather performance data in wireless networks. It involves creating preprocessing methodologies to eliminate data with notable violations and developing machine learning models.

III. PROPOSED AML-CTP FRAMEWORK

This section presents the phases of the proposed AML-CTP framework. These phases mainly aim to investigate network performance, retrieve management information, and predict network traffic load through a fast and accurate model with less computational time and storage space. AML-CTP framework phases are illustrated in Figure 2. These phases are discussed in detail in the following subsections.

A. PHASE 1: DATA PREPROCESSING

As in Figure 2, the phase starts with data collection. After collecting data, it contains various underlying properties and information saved in a local database. First, the AML-CTP framework starts the cleaning phase by dealing with missing values. In many analyses, missing values can be a problem. Therefore, all missing values are replaced with the median. As a result, different cells vary significantly in their traffic loads. Feature selection effectively solves the problem of high-dimensional data analysis by removing irrelevant and redundant data. Hence, this can reduce computation time, improve learning accuracy, and better understand the learning model or data. Sklearn's SelectKBest chooses the most important features to include in the learning process [42]. It also analyzes the correlation between features as in [7] and [43]. The proposed AML-CTP framework normalizes the traffic load using Min-Max scaling to avoid significant data variance.

Finally, the proposed framework utilizes the Principal Component Analysis (PCA) [44] algorithm to reduce data dimensionality. PCA is a linear dimensionality reduction technique that extracts information from a high-dimensional space by projecting the data into a lower-dimensional subspace. It tries to keep the essential parts with more data variation and remove the non-essential parts with fewer variations. A vital part of using PCA is estimating how many components are needed to describe the data. Therefore, the cumulative explained variance ratio can exhibit the components needed [45]. PCA can help visualize the data in a 2D or

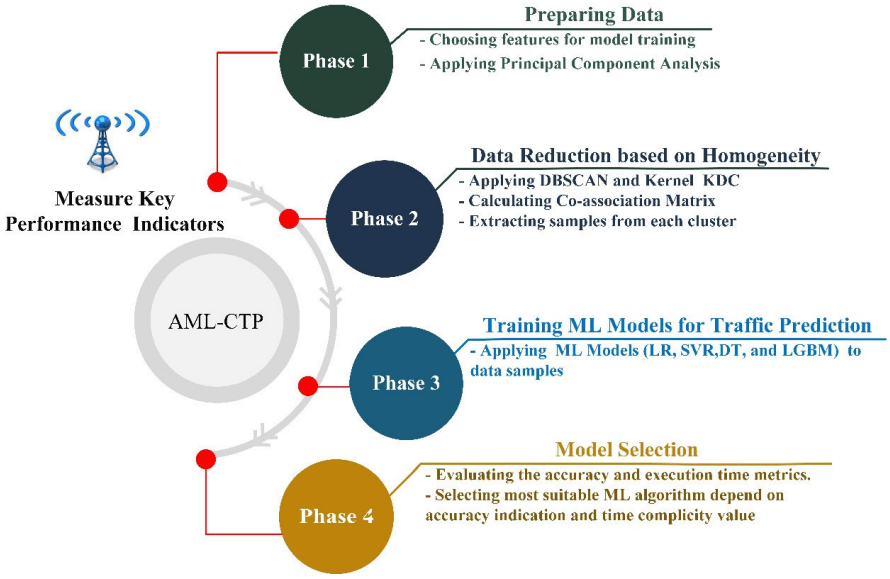


FIGURE 2. Proposed adaptive machine learning-based cellular.

3D space. Thus, data can be plotted to understand the underlying pattern better. Also, it removes noise by condensing many features into a few principal components. So, PCA speeds up ML algorithms, especially with high-dimensional data. By that, the most critical part of preparing data is finished. The next phase aims to decide which data sample can be used for training the ML models.

B. PHASE 2: DATA REDUCTION BASED ON HOMOGENEITY

Recently, ML-based approaches have been investigated to detect the dataset pattern and organize similar data types. The concept of similarity is closely related in the context of clustering algorithms. Therefore, the proposed AML-CTP framework groups together data points that are similar based on some measure of similarity or distance. It also utilizes a Co-Association technique, like a voting mechanism used to combine the clustering results.

Traffic Prediction (AML-CTP) framework The main idea for using more than one clustering algorithm is that the more often two packets fall into the same cluster when several clustering algorithms are applied, the more similar they are. The considered clustering algorithms use density functions. The algorithms used are Density-Based Spatial Clustering of Applications with Noise and Kernel density clustering. The first algorithm, DBSCAN, is used to identify the outliers (noise) as the points in low-density regions. The AML-CTP framework applies this algorithm and configures the main parameters. The first one is the Epsilon Value (eps); Epsilon is the circle's radius around a data point, and all other data points that fall inside the circle are considered neighborhood points. In other words, two points are neighbors if their distance is less than or equal to eps. Second, a minimum minPoints can be derived from several dimensions (D) in the dataset, as minPoints $\geq$ D + 1. Larger values are usually better for noisy data sets and will form more significant clusters [7].

The second one, the KDC algorithm, uses a kernel density estimator to identify clusters in a dataset. KDC also requires the specification of a bandwidth parameter, which determines the amount of softness applied to the density estimate. A bandwidth that is too small can result in overfitting and the identification of small, noisy clusters, while a bandwidth that is too large can result in underfitting and merging distinct clusters. One of the ways to determine bandwidth is Silverman bandwidth.

After considering cluster algorithms, the framework handles datasets with irregularly shaped clusters and noise in the data. Considered clustering algorithms: The Co-Association matrix compares the results of two clustering algorithms (BDSCAN and KDC). A N×N matrix (Co) is created to construct the co-association matrix, where the (r,s) position is either one if observations r and s belong to the same cluster and 0 otherwise. The average of all these matrices constitutes the co-association matrix. All elements of matrix Co are initialized to 0. then iterate over all pairs of data points and increment the corresponding; if both algorithms assign data points r and s to the same cluster, Co (r, s) is updated.

After that, a sample selection criterion is applied to extract the training samples from each cluster. The selection criteria are dependent on the Silhouette Score and homogeneity measures. Those techniques are used in clustering analysis to evaluate the quality of extracted samples from clusters.

Also, the Rand Index (RI) is measured to check similarity. The RI is calculated by comparing the number of pairs of data points that are either in the same cluster or different clusters. That can be done by selecting a random small sample of data points from each cluster with a high RI. Extracted samples with M number of sample rows of the training dataset

($Z = \{v_1, v_2, \ldots, v_N\}$) with N rows of the dataset for c_i clusters are presented as:

$$Z' = \{v_1, v_2, \ldots, v_M\} \ , where\ M < N \tag{1}$$

A sample is density maintained as follows:

$$\sum_{\delta \in c_j} pr\left(v \in Z' \mid v \in c_j\right) = \delta_j \mid c_j) \tag{2}$$

where δ_j is a constant for any cluster c_j , $j = \{1, \ldots, k\}$s

A sample is uniform, as in the following equation:

$$pr\left(v \in Z' \mid v \in c_i\right) = pr\ v'\left(\in Z \mid v' \in c_j\right) \quad if\ any\ v \in c_i, v' \in c_j \quad where, i, j \in \{1, 2, \ldots\} \tag{3}$$

Therefore, uniform sampling can be formulated as follows:

$$pr(v \in Z' | v \in Z) = \frac{M}{N} \tag{4}$$

The proposed framework uses the homogeneity measure to assess the degree to which clusters contain only samples of a single class. That score can be calculated as:

$$Homogeneity\ score = 1 - H(Z', C)/H(C) \tag{5}$$

where Z' is the true labels of the samples, C is the cluster labels assigned by the clustering algorithm, and The homogeneity measure ranges from 0 to 1, where a higher score indicates better clustering. A score of 1 indicates perfect homogeneity, meaning each cluster contains only samples of a single class. In contrast, 0 indicates the opposite. A large homogeneous population typically requires a smaller sample size for accurate results, while a more heterogeneous population may require a larger sample size [46].

Likewise, the Silhouette score can measure how similar an object is to its cluster compared to others.

$$Silhouette\ score = (\mu_2 - \mu_1)/max(\mu_1, \mu_2) \tag{6}$$

where μ_1 is the mean distance between the sample and all other samples in the same cluster. μ_2 is the mean distance between the sample and all other samples in the nearest neighboring cluster. It ranges from -1 to 1, where a higher score indicates better clustering. A score of 1 indicates that the object is well-matched to its cluster and poorly matched to neighboring clusters, while a score of -1 indicates the opposite [46].

After the homogeneity tests, the proposed AML-CTP framework understands the dataset and extracts the most suitable samples that can be used for training. Now, it is ready to apply different ML algorithms, further explained next.

C. PHASE 3: TRAINING ML MODELS FOR TRAFFIC PREDICTION

The main goal of the AML-CTP framework is network traffic load prediction. Therefore, the problem is considered a regression situation. The considered regression algorithms can substantially reduce the computational complexity of searching when dealing with big data. However, good accuracy is maintained, and model overfitting is avoided. Hence, the proposed framework utilizes different ML models from which the best model is chosen. These models are selected since they can handle high-dimensional data while retaining fast training time and high efficiency [47], [48]. Consequently, the AML-CTP framework conducts experiments with several machine learning algorithms, including Linear Regression with polynomial features, Support Vector Regression, Decision Tree algorithm, and LGBM. A brief explanation of each ML algorithm is given as follows:

The first assessed algorithm is a linear regression with polynomial features. It finds the relationship between a given dataset and predicted network traffic load. Therefore, it captures nonlinear correlations between variables By fitting a nonlinear regression line. A standard regression model takes the form as follows:

$$L = \theta_0 + \theta_1 v_1 + \theta_2 v_2 + \ldots\ldots \theta_M v_M \tag{7}$$

where L is the predicted network traffic load value of the dependent variable, θ_0 is the intercept, and $\theta_1, \theta_2 \ldots, \theta_M$ are the predicted regression coefficients representing the contribution of the independent predictor variables $v_1, v_2, \ldots, v_d$, respectively.

The second algorithm (SVR) guarantees the flexibility to define the acceptable error in our model and find an appropriate line to fit the data. The objective function of SVR is to minimize the coefficients. Instead, the error term is handled in the constraints; the absolute error is assumed to be less than or equal to a specified margin, called the maximum error, ϵ (epsilon). Therefore, epsilon is tuned to gain the desired accuracy. The proposed framework state's objective function and constraints for Minimize (Error) are as follows:

$$Min \frac{1}{2}\left[||\theta||^2\right] \quad where |L_i - \theta_i v_i| \leq \varepsilon \quad for\ i = \{1, 2, \ldots\ldots\ldots, M \tag{8}$$

where L_i is the target network traffic load value, θ_i is the coefficient, and v_i is the predictor (feature) [49].

The decision tree algorithm is also considered one of the embedded ML algorithms of the proposed framework. That algorithm is reasonably simple to understand and is effective. The primary goal of a decision tree is to split a data population into smaller segments. Then, binary trees are structured, where each node represents a test on a feature, and each leaf node holds an output. DT is induced in a top-down order to construct the tree. Building a decision tree is about locating the attribute that provides the maximum information gain. The information gain is calculated using the decrease in entropy after splitting a data set on an attribute and can be calculated as:

$$Max(Gain\ (v, L) \equiv Max(Entropy\ (v) - \sum\nolimits_{s \in D_L} \frac{|v_s|}{|v|} Entropy\ (v_s) \tag{9}$$

where v_s is the sum of every node value, v is the sum of the sample value, and Entropy(v_s) is the entropy of the current node.

As formulated in Equation (9), the key hypothesis of the mathematical model of the DT model is entropy, which takes a lot of information to describe a sample adequately. Therefore, the DT algorithm can be considered a regression algorithm with less computational time and good accuracy [50].

The proposed framework also considers the LGBM as one of the embedded MLs in the system. LGBM is a high-performance gradient-boosting machine learning algorithm. Uses histogram-based algorithms to speed up training and reduce memory usage. LGBM is highly efficient and accurate, supports parallel learning, and is suitable for large datasets. Additionally, LGBM contains two techniques that work together: Gradient-based One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which overcome the inadequacies of the histogram-based algorithm used in all GBDT (Gradient Boosting Decision Tree) frameworks [50], [51], [52]. Hence, these models are trained using the data samples output from Phase 2. Then, the obtained quality metrics are examined by the AML-CTP framework to choose the best model.

D. PHASE 4: MODEL SELECTION

After training ML models, the considered ML models are evaluated using the accuracy and execution time metrics. The first metric is R^2, a common measure of how well the curve fits the data. A value of one indicates a perfect fit between actual and predicted values, which value has the same propensity. The mathematical formula for computing R^2 is [50] and [53]:

$$R^2 = 1 - \frac{\sum_{i=1}^{M} (v_i - L_i)^2}{\sum_{i=1}^{M} (v_i - L_i^{mean})^2} \tag{10}$$

where v_i is the actual value for data samples, L_i is the predicted network traffic load.

L_i^{mean} is the mean of the observed data given by:

$$L_i^{mean} = \frac{1}{M} \sum_{i=1}^{M} v_i \tag{11}$$

Also, RMSE is used to measure how well the model fits predicted variables. Which is the average distance of a data point from the fitted line measured along a vertical line. The RMSE is calculated through the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M} (v_i - L_i)^2}{M}} \tag{12}$$

Finally, The MAPE is calculated as a percentage used to measure how close predictions are to the eventual outcomes. A smaller value of MAPE represents better prediction accuracy. The MAPE is determined by:

$$MAPE = 100 \times \sum_{i=1}^{M} \frac{v_i - L_i}{v_i} \tag{13}$$

Another metric evaluated is execution, which is related to data preprocessing. Therefore, the complexity analysis for LR, SVR, DT, and LGBM is demonstrated.

Assume that there are a p number of features. So, the complexity of the LR algorithm is calculated as $(O\left(\mathrm{M}p^2 + p^3\right))$ However, SVR computational time differs from that of LR. Hence, it has a complexity of $(O\left(M^3 + M^2 p\right)$. Furthermore, the overhead complexity for the Decision Tree algorithm is based on splitting the dataset. It can be formulated as $(O(\left(M^2 p\right))$. The LGBM has time complexity, like the gradient boosting method. It depends on the t number of trees, and the tree may have d as the depth. So, it can be calculated as $(O\left(mptd\right))$.

We introduce the criteria that adaptively accommodate the minimal computational delay and the maximum accuracy to select the best ML model. That can be a higher accuracy indication if both MAPE and RMSE must be close to 0, while R2 should be close to 1. A smaller time complicity value denotes a lower computational time. Based on insights from our industrial partner, a telecommunication carrier in Egypt, the framework outputs the model score with the following formulation:

$$Score_{Model} = \frac{\omega_{Ac} \times Accuracy_{Model}}{\omega_{Comp} \times Complicity_{Model}}$$
$$\text{where } \omega_{Ac} + \omega_{Comp} = 1 \tag{14}$$

In Equation (14), the Model symbol presents the embedded ML models {LR, SVR, DT, LGBM}. The ω_{Ac} and ω_{Comp} are weights that depend on the relative value of the accuracy indicator and complicity value, respectively.

Also, the AML-CTP framework adaptively sets the weights based on the network load and required prediction time. The network load is quantified using the number of active users registered. For instance, in light-load times when the traffic is expected to be low, the AML-CTP framework sets the accuracy weight as dominant. Alternatively, the framework favors computational complexity over model accuracy during beak time and sets the complicity weight as foremost. Finally, the ML model with the highest score should be selected as the best model. Then, the AML-CTP framework is ready to apply the selected ML model to the complete dataset. A full, detailed description of the AML-CTP framework flow chart is illustrated in Figure 3.

IV. EXPERIMENTAL EVALUATION

This section evaluates the AML-CTP framework when applied to real datasets of two case studies. The section is divided into three subsections. The first subsection explains the dataset used in the evaluation and the experimental setup. The results obtained from the empirical evaluation are presented and analyzed in the second subsection. Finally, the third subsection compares the proposed framework against two state-of-the-art techniques [48], [54].

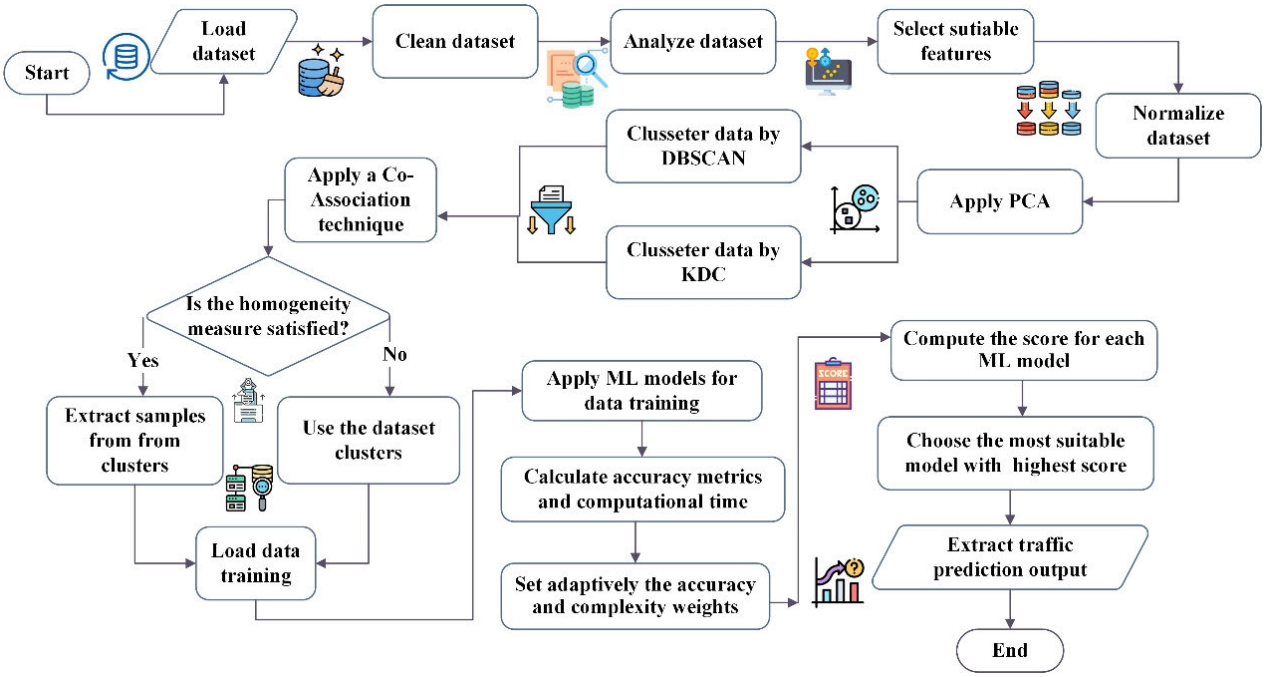


FIGURE 3. A flow process for the proposed AML-CTP framework.

A. DATASETS CHARACTERISTICS AND EVALUATION SETUP

The first case study dataset is KPI from an LTE mobile network. It is collected from the traffic data from one of Egypt's most popular telecommunication equipment corporations. Our previous work thoroughly analyzes the used dataset [7]. Traditionally, Key Performance Indicators are divided into radio network KPIs and service. In our analysis, we focus on three types of KPIs to observe the throughput of cell edge users and its correlation with traffic load across different frequency bands. These categories include Integrity KPIs, Utilization KPIs, and Traffic KPIs. Integrity KPIs measure the impact of eNBs on service quality, such as throughput for cells and users, as well as latency for served users. Utilization KPIs measure network utilization and resource distribution based on demands, encompassing uplink resource block utilization and downlink resource block usage rates. Traffic KPIs assess traffic volumes on LTE Radio Access Network, categorized by the type of traffic, including radio bearers, downlink traffic volume, and uplink traffic volume. The final dataset used in the first case study consists of 77 features and 259.223 rows [7].

Moreover, over four weeks, another dataset is collected for the second case study from 6,000 cells in 200 telecommunication network sites. It has 80 features and 443,136 rows.

Furthermore, we investigate applying the proposed framework to different geographic locations and telecommunication providers. That is because data collected from different geographic locations exhibit the same behavior, as discussed in [7]. Therefore, the provided data is considered a suitable representative sample. Moreover, the dataset is split into training and testing sets to prevent overfitting. Then, the experiments apply a double cross-validation process in which the selected models are tested and evaluated using a validation set to assess whether the model is overfitting to a particular set of training examples [7].

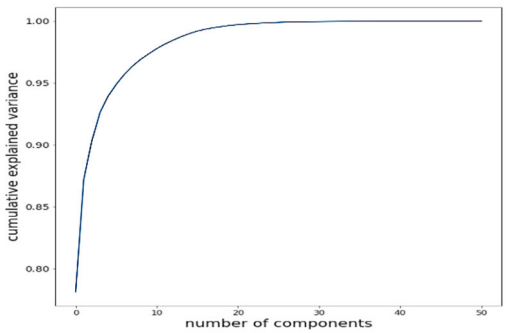
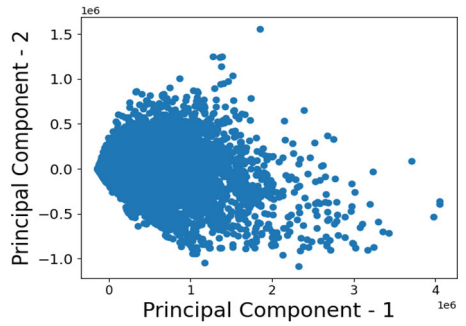


FIGURE 4. Figure Number of components according to the cumulative explained variance.

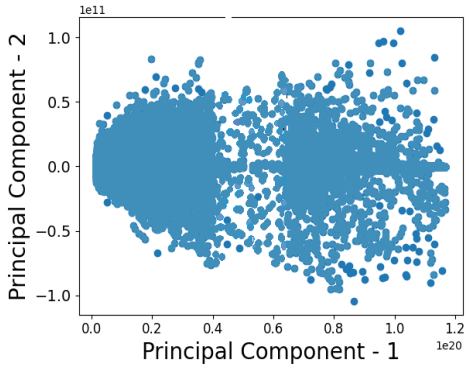
B. EMPIRICAL EVALUATION OF THE AML-CTP FRAMEWORK

As a start, the first phase of the framework is applied to analyze the data. The analysis eliminates 24 features from the data since these columns are not numerical and are used to identify the frequencies and the ID of each eNB.

Also, some of these columns describe information regarding user handover [11], [45], [55], which is outside of the scope of the analysis. A deeper investigation and further analyses of the collected dataset can be found in our previous work [7], [43]. Also, regarding missing values, all null values are compensated with the median value of the

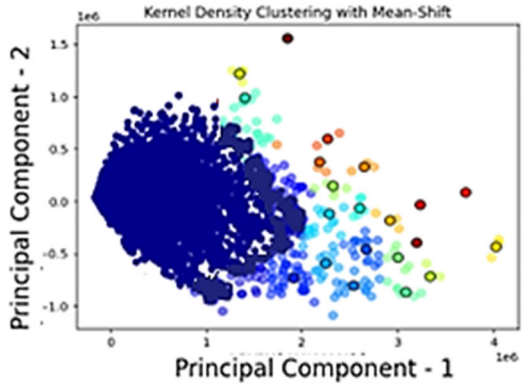


(a) First case study

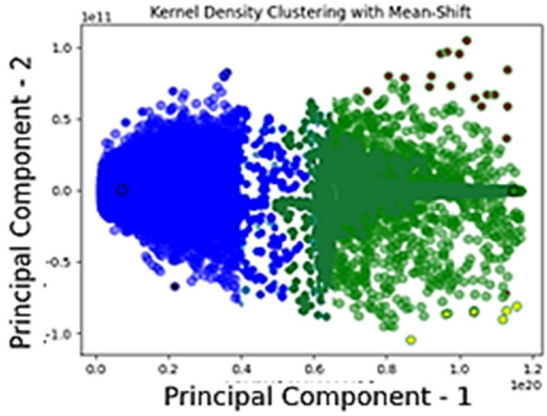


(b) Second case study

FIGURE 5. Two components of PCA of datasets.



(a) First case study



(b) Second case study

FIGURE 6. The output of the KDC algorithm for two case studies.

entire row. After calculating the feature score using Sklearn's SelectKBest [7], [43], the high-scored features are utilized to choose the final feature set. Therefore, the method that extracts a final set of 50 features will have the highest score. This final set is then used as the network traffic load prediction target. Each feature is further explained in detail in [7] and [43]. Finally, normalization is applied to the selected dataset to avoid significant data variance. As for large-scale multi-dimensional datasets like the collected data, visualization is an essential task. Hence, PCA is applied to diminish the data into a lower dimension. PCA estimates how many components are needed to describe the data.

The curve in Figure 4 quantifies the number of dimension variances that can be contained within the first N components. The first ten components represent approximately 95% of the data variance. Also, to fully describe the variance, a total of 50 components are needed.

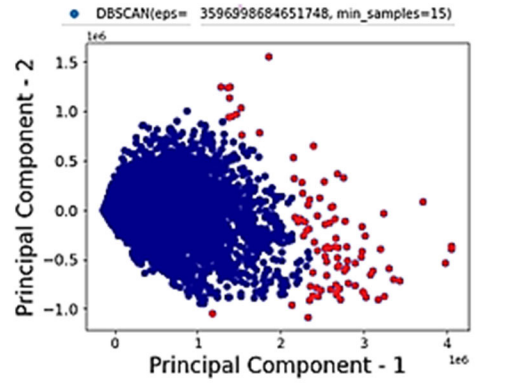
Consequently, the analysis considered a two- or three-dimensional projection (with two or three principal components), representing 87.17% of the data variance for two components, a more than acceptable percentage to represent the data [11].

Moreover, the plots in Figure 5 represent the complete datasets with two components using PCA. Figure 5(a), two-dimensional representations are sufficient to show that a hyperplane's fit to the data points distribution is excellent. Figure 5(b) shows the distribution of classes on a two-dimensional vector space.

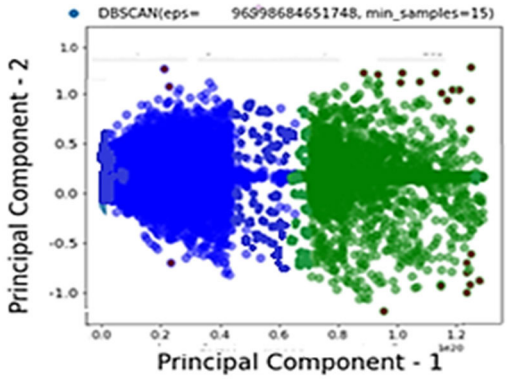
The considered clustering algorithms are applied to datasets. The detection method calculates the average distance between each point and its k nearest neighbors. Then, the average k-distances are plotted in ascending order on a k-distance graph. The optimal value for epsilon is the point with maximum curvature or bend (i.e., at the most significant slope). The calculated Epsilon value for our dataset in the first case study is 3596.999. Alternatively, the clustering minPoints is considered 15 since it represents a suitable value for the collected dataset.

In the first case study, the DBSCAN clustering algorithm's output is one cluster of 250,648 data rows shown in Figure 6(a) with dark blue color and represents 96.7% of data and noise of 8,575 data rows with red color. According to data density in the second case, the output of the DBSCAN clustering algorithm is two clusters of 198,565 data rows and 244,423 data rows represented in Figure 6(b) with green and blue color and noise of 1148 data rows with brown color. The KDC clustering algorithm is applied after applying the Silverman bandwidth formula in the first case study.

The output of the KDC clustering algorithm in first case study is 12 clusters with one main of 249,346 data rows that are represented in Figure 7(a) with blue color a and represent 96.3% of data. The rest of the 11 clusters contain the rest of the data with different colors, each centered with an 'o' character and a limited number of data rows.

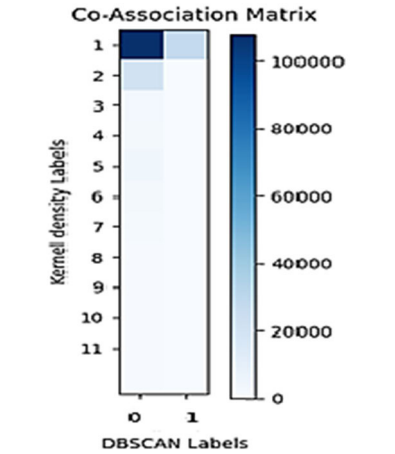


(a) First case study

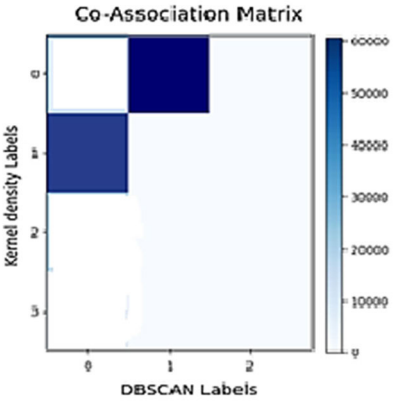


(b) Second case study

FIGURE 7. Output of DBSCAN clustering for two case studies.



(a) First case study



(b) Second case study

FIGURE 8. The CO-Associated matrix between the output clusters of DBSCAN and KDC clustering algorithms for two data sets.

After applying the Silverman bandwidth for the second case study, the Gaussian kernel is also used as a kernel function, and the output of the KDC clustering algorithm is 4 clusters with 205,136 data rows, 237,401, 1006, and 593 data rows. The main two clusters are represented in Figure 7(b), with blue and green colors representing 99.6% of data; the other two clusters contain the rest of the data with brown and yellow colors.

Figure 8 illustrates the density distribution of similarity measure scores for two case studies. The co-associated matrix of these scores is empirically determined using DBSCAN and KDC clustering algorithms for the respective case studies. Agreement and similarity between the labels assigned by the two clustering algorithms. Conversely, the remaining clusters exhibit significantly lower values of RI.

As anticipated, the co-associated matrix for the dataset in the first case study is generated for a single cluster and noise, as output by DBSCAN, and eleven clusters, the output of KDC. Notably, the data in the first cluster of both clustering algorithms predominantly overlap, as depicted in Figure 8 (a). The Rand Index is recorded at 0.99 in this first cluster, indicating a high level of In Figure 8 (b), nearly the entire dataset in the second case study can be categorized into two primary clusters, and the data within these clusters are identical. The DBSCAN algorithm divides the dataset into two clusters and identifies one noise cluster. Meanwhile, the KDC model reveals four clusters, with respective Rand Index values of 0.982 and 0.991 for the two main clusters. These high RI values indicate strong similarity results between the clusters.

TABLE 1. Homogeneity and silhouette.

	First case study	Second case study	
	Cluster	Cluster 1	Cluster 2
Homogeneity	0.93	0.94	0.973
Silhouette	0.953	0.956	0.98

TABLE 2. LR R^2 results using different polynomials with different degrees.

	POLY1	POLY2	POLY3	POLY4
R^2(%)	85.46	97.08	97.59	97.87

In Table 1, Homogeneity and Silhouette Score measures for clusters with high RI in both case studies are computed, verifying the data similarity in each cluster with high scores near one value. The AML-CTP framework, as presented, selects optimal samples from a dataset for training purposes. These samples should be manageable to ensure efficient processing, enabling statistical measures of a selected subset to

TABLE 3. LGBM'S optimized parameters.

Model	Hyperparameters	Grid
LGBM	Number of estimators	500
	Max Tree Depth	400
	Number of leaves	19
	Learning Rate	0.1

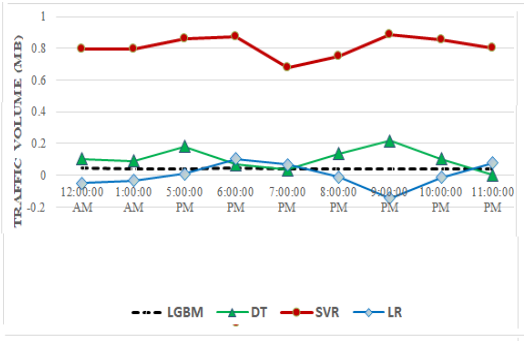


FIGURE 9. Difference between original traffic and predicted traffic during peak hours in the first case study.

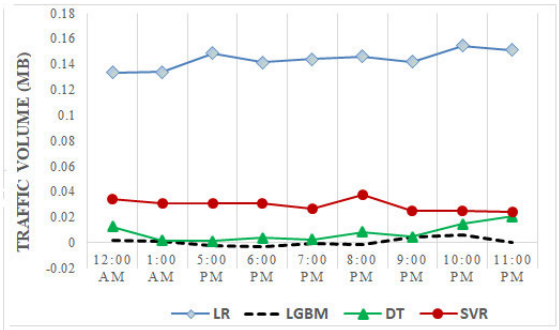
approximate a response from the entire dataset in instances where similar clusters exhibit a high RI, such as one cluster in the first case and two clusters in the second case, random samples of 1500 data rows are drawn from each cluster, maintaining the same distribution probability as the entire cluster.

LR with polynomial features, Support Vector Regression, DT algorithm, and LGBM models are trained using the data samples. The optimal parameters are obtained to maximize the performance of LR and LGBM. As for LR, the framework experimented with different degrees (first to fourth degrees) to determine the best quality metric degree [56]. The R^2% values are calculated with each model and presented in Table 2. The LR with polynomial features of the fourth degree achieved the highest accuracy in both cases.

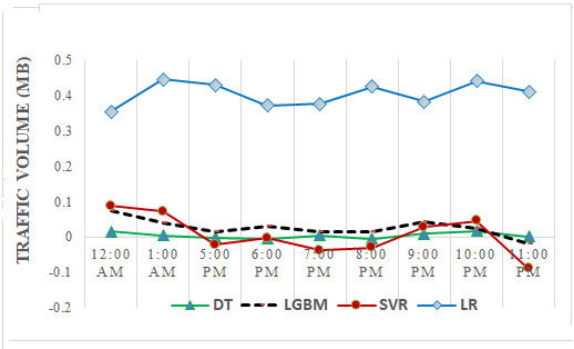
Regarding LGBM, the hyperparameter values are determined using Bayesian optimization hyperparameter tuning [57], [58]. Table 3 illustrates the final values of LGBM's hyperparameters.

Next, the quality metrics are evaluated for the embedded ML models (LGBM, LR with fourth-degree polynomial features, SVR, and DT) to identify the most suitable model. As Table 4 illustrates, the quality metrics include R2, MAPE, RMSE, and time complexity (comp). In the results of the initial case study, it is evident that the LGBM algorithm exhibits higher accuracy, accompanied by an acceptable computational time, aligning with the expectations of our industrial partner. Since LGBM is efficient for large datasets, it achieved the best performance in the first scenario.

Analyzing a selected sample from the two clusters in the second case study reveals that the LGBM algorithm achieves the most favorable accuracy indication among all other algorithms in the first cluster. As the first cluster does not contain noise (as explained in Section IV-B), LGBM sustained its



(a) First case study



(b) Second case study

FIGURE 10. Difference between original traffic and predicted traffic during peak hours in the second case study.

TABLE 4. Performance evaluation ordered by the accuracy indication (R^2%).

		Model	R^2%	MAPE	RMSE	Comp
First case study		**LGBM**	**98.9**	**0.03**	**0.412**	6.43
		DT	98.5	0.051	0.901	9
		SVR	97.3	0.331	0.611	7.2
		LR	90.4	1.5	1.269	15.7
Second case study	First cluster	**LGBM**	**98.7**	**0.035**	**0.441**	6.1
		DT	98.4	0.071	0.52	9
		SVR	98	0.1	0.599	15.7
		LR	91.1	1.54	1.366	8.5
	Second cluster	**DT**	**98.6**	**0.05**	**0.450**	12.3
		LGBM	**98.7**	**0.035**	**0.441**	6.1
		SVR	98.4	0.071	0.52	9
		LR	98	0.1	0.599	15.7

superior performance. The DT algorithm is the second-best performer. Also, as LR fundamentally assumes a linear relationship between features and the log odds of the target variable, the LR model performs the least favorably in both clusters since this assumption does not hold in the second scenario.

Interestingly, the DT algorithm yields the best quality metrics in the second cluster. Since the data in this cluster contains noise, and with LGBM being sensitive to noisy data and outliers, DT is considered the most suitable algorithm. Decision trees are relatively robust to outliers, making them more suitable for this cluster. Finally, SVR is an effective algorithm, especially with nonlinear data. Hence, it performed well in both clusters. It performs slightly better in the first cluster since it is less sensitive to outliers than other regression algorithms. In general, the results exhibit

TABLE 5. Performance evaluation of the proposed framework compared to other state-of-the-art techniques.

Dataset	Model	R^2% Train.	R^2% Test.	MAPE	RMSE	F1-score	Precision	Recall	T. Time	P. Time
1st case study	**AM-CTP**	**99.2**	**98.9**	**0.030**	**0.412**	**0.97**	**0.96**	**0.99**	**6.40s**	**3.10ms**
	ML	98.5	97.0	0.330	0.660	0.95	0.94	0.98	45.71s	12.32ms
	LSTM	95.4	93.0	0.541	1.120	0.94	0.92	0.96	72.23s	33.40ms
2nd case study	**AM-CTP**	**99.3**	**98.5**	**0.043**	**0.441**	**0.96**	**0.95**	**0.98**	**12.00s**	**4.37ms**
	ML	92.5	0.89	1.710	1.450	0.93	0.91	0.97	53.11s	20.00ms
	LSTM	94.6	92.0	0.610	1.220	0.94	0.94	0.97	77.40s	40.00ms

the strengths and weaknesses of each algorithm given the underlying data in each cluster and show that the suitability of each model may vary depending on various factors, such as the data's linearity and the noise's exitance.

Hence, the discrepancy between actual and predicted traffic volumes is assessed for each ML model during peak hours. As illustrated in Figure 9, LGBM demonstrates superior prediction accuracy compared to other algorithms, with DT ranking as the second-best performer. Negative values indicate instances. The results show that when the predicted value surpasses the original value, the variance between the original and predicted values falls within 1 to −0.2 MB. This approach significantly reduces the time required to identify the most accurate prediction module by carefully selecting relevant data samples, resulting in a 53.3%-time reduction in the first case study. The framework highlights the efficacy of LGBM in traffic quality prediction.

Similarly, Figure 10 illustrates the disparities between the actual and predicted traffic within each pair of clusters. The variations range from −0.01 to 0.16 MB in the first cluster and from −0.1 to 0.5 MB in the second cluster. These findings empirically validate the effectiveness of the proposed AML-CTP framework in adapting to changes in network load. In both case studies, the framework demonstrated a time-saving of over 85% when utilizing a sample of 1,500 data rows from similar clusters in the first case study and over 90% time reduction in the second case study. It is anticipated that even more significant time savings can be achieved with a more representative sample size relative to the total number of rows in the utilized samples.

C. COMPARISON WITH THE STATE-OF-THE-ART TECHNIQUES

This section compares the proposed framework with two state-of-the-art techniques [48], [54]. The first approach (ML) [48] applies traditional learning algorithms with LR with polynomial features to predict the network traffic load. Alternatively, the second approach (LSTM) [54] is a predictive model that aims to enhance and sustain network capacity using the LSTM algorithm and regression methods.

To compare the proposed framework with these approaches, we employed our datasets used in the experiments (first and second case studies) to evaluate the overall performance. Table 5 shows the results obtained from the experiments. During the evaluation, we report the quality metrics, including R2, MAPE, RMSE, FI score, precision, recall, and execution time for training (T. time) and prediction (P. time).

As the table shows, the proposed model produced the best prediction in the quality metrics during the experiments compared to the other methods. As for the prediction performance, the framework achieved the best traffic load prediction with the highest Precision, Recall, and F1-score without indication for overfitting, as shown from R^2 results in training (R^2% Train.) and testing (R^2% Test) shown in the table.

Also, since the proposed framework only considers relevant data samples to select the most suitable learning algorithm, it significantly reduces training and prediction time. The table shows that the AM-CTP could reduce prediction time by up to 91% in the first case study compared with the two other techniques. Similarly, the proposed framework could save more than 90% of training and prediction time in the second case study. For instance, the proposed model could achieve better results with less training time. It reduces the training time by up to 84.4% and 77.44% compared to the ML and LSTM frameworks, respectively.

V. CONCLUSION AND FUTURE WORK

The paper discusses the challenge of accurately predicting network traffic loads in cellular networks, especially when dealing with complex, multi-dimensional datasets. Therefore, it is advisable to explore multiple algorithms before selecting the most suitable one. Also, it introduces clustering methods to understand the data structure. So, applying a single algorithm with identical hyperparameter tuning across different datasets is challenging. This paper emphasizes the importance of understanding the data and its characteristics to select appropriate algorithms, with visualization aiding in this process. It presents a proposed AML-CTP framework to minimize runtime overhead by analyzing the data's nature in traffic prediction. The AML-CTP framework proposed combines two-density clustering techniques (KDC and DBSCAN) with a co-associated matrix to categorize unknown traffic into clusters. High-quality clusters are identified using metrics like RI, silhouette score, and homogeneity. Four high-speed, low-complexity algorithms are then fitted to these clusters for prediction. Two case studies using real KPI data from LTE-A networks demonstrate the framework's effectiveness, with one showing the superiority of the LGBM

model. The framework saves significant computational time in selecting suitable data samples and demonstrates the effectiveness of different models for different clusters. It suggests steering predicted traffic to cells or carriers with low traffic during peak hours to enhance the Quality of Service for users at cell edges.

REFERENCES

[1] G. Alsuhli, K. Banawan, K. Seddik, and A. Elezabi, "Optimized power and cell individual offset for cellular load balancing via reinforcement learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–7.

[2] K. Grochla and M. Slabicki, "Transmit power optimisation in cellular networks with nomadic base stations," *IET Commun.*, vol. 13, no. 18, pp. 3068–3074, Nov. 2019.

[3] A. Salah, H. M. Abdel-Atty, and R. Y. Rizk, "Joint channel assignment and power allocation based on maximum concurrent multicommodity flow in cognitive radio networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–14, Jul. 2018.

[4] Y. Ouyang, Z. Li, L. Su, W. Lu, and Z. Lin, "APP-SON: Application characteristics-driven SON to optimize 4G/5G network performance and quality of experience," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1514–1523.

[5] A. Pandey, T. R. Nair, and S. B. Thomas, "Combination of K-means clustering and support vector machine for instrument detection," *Social Netw. Comput. Sci.*, vol. 3, no. 2, p. 121, Jan. 2022.

[6] M. Nashaat, I. E. Shaalan, and H. Nashaat, "LTE downlink scheduling with soft policy gradient learning," in *Proc. 8th Int. Conf. Adv. Mach. Learn. Technol. Appl. (AMLTA)*, 2022, pp. 224–236.

[7] N. H. Mohammed, H. Nashaat, S. M. Abdel-Mageid, and R. Y. Rizk, "A framework for analyzing 4G/LTE—A real data using machine learning algorithms," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.*, 2021, pp. 826–838.

[8] S. M. M. AboHashish, R. Y. Rizk, and F. W. Zaki, "Energy efficiency optimization for relay deployment in multi-user LTE-advanced networks," *Wireless Pers. Commun.*, vol. 108, no. 1, pp. 297–323, Sep. 2019.

[9] E. T. Ogidan, K. Dimililer, and Y. K. Ever, "Machine learning for expert systems in data analysis," in *Proc. 2nd Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Oct. 2018, pp. 1–5.

[10] R. Rizk and H. Nashaat, "Smart prediction for seamless mobility in F-HMIPv6 based on location based services," *China Commun.*, vol. 15, no. 4, pp. 192–209, Apr. 2018.

[11] H. Nashaat, "QoS-aware cross layer handover scheme for high-speed vehicles," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 1, pp. 135–158, Jan. 2018.

[12] H. Huang, Z. Hu, Y. Wang, Z. Lu, X. Wen, and B. Fu, "Train a central traffic prediction model using local data: A spatio-temporal network based on federated learning," *Eng. Appl. Artif. Intell.*, vol. 125, Oct. 2023, Art. no. 106612.

[13] S. T. Nabi, Md. R. Islam, Md. G. R. Alam, M. M. Hassan, S. A. AlQahtani, G. Aloi, and G. Fortino, "Deep learning based fusion model for multivariate LTE traffic forecasting and optimized radio parameter estimation," *IEEE Access*, vol. 11, pp. 14533–14549, 2023.

[14] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," *PLoS ONE*, vol. 14, no. 11, Nov. 2019, Art. no. e0224365.

[15] S. Chahboun and M. Maaroufi, "Principal component analysis and machine learning approaches for photovoltaic power prediction: A comparative study," *Appl. Sci.*, vol. 11, no. 17, p. 7943, Aug. 2021.

[16] P. Ghasemi, M. Aslani, D. K. Rollins, and R. C. Williams, "Principal component neural networks for modeling, prediction, and optimization of hot mix asphalt dynamics modulus," *Infrastructures*, vol. 4, no. 3, p. 53, Aug. 2019.

[17] S. Chakraborty, "Analysis and study of incremental DBSCAN clustering algorithm," *Int. J. Enterp. Comput. Bus. Syst.*, vol. 1, no. 2, Jul. 2011.

[18] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, "A density based algorithm for discovering density varied clusters in large spatial databases," *Int. J. Comput. Appl.*, vol. 3, no. 6, pp. 1–4, Jun. 2010.

[19] Q. Lin and J. Son, "A close contact identification algorithm using kernel density estimation for the ship passenger health," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 6, Jun. 2023, Art. no. 101564.

[20] B. Guo, L. Tian, J. Zhang, Y. Zhang, L. Yu, J. Zhang, and Z. Liu, "A clustering algorithm based on joint kernel density for millimeter wave radio channels," in *Proc. 13th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2019, pp. 1–5.

[21] Y. Jia, S. Tao, R. Wang, and Y. Wang, "Ensemble clustering via co-association matrix self-enhancement," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2023.

[22] N. Arinik, V. Labatut, and R. Figueiredo, "Characterizing and comparing external measures for the assessment of cluster analysis and community detection," *IEEE Access*, vol. 9, pp. 20255–20276, 2021.

[23] W.-C. Hong, "Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting," *Neural Comput. Appl.*, vol. 21, no. 3, pp. 583–593, Apr. 2012.

[24] D. E. Birba, "A comparative study of data splitting algorithms for machine learning model selection," KTH Royal Inst. Technol., Tech. Rep., 2020.

[25] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, 2014.

[26] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[27] B. Mahdy, H. Abbas, H. Hassanein, A. Noureldin, and H. Abou-zeid, "A clustering-driven approach to predict the traffic load of mobile networks for the analysis of base stations deployment," *J. Sensor Actuator Netw.*, vol. 9, no. 4, p. 53, Nov. 2020.

[28] D. Alekseeva, N. Stepanov, A. Veprev, A. Sharapova, E. S. Lohan, and A. Ometov, "Comparison of machine learning techniques applied to traffic prediction of real wireless network," *IEEE Access*, vol. 9, pp. 159495–159514, 2021.

[29] J. Riihijarvi and P. Mahonen, "Machine learning for performance prediction in mobile cellular networks," *IEEE Comput. Intell. Mag.*, vol. 13, no. 1, pp. 51–60, Feb. 2018.

[30] S. Chaudhary and R. Johari, "ORuML: Optimized routing in wireless networks using machine learning," *Int. J. Commun. Syst.*, vol. 33, no. 11, p. e4394, Jul. 2020.

[31] B. Sliwa, T. Liebig, R. Falkenberg, J. Pillmann, and C. Wietfeld, "Efficient machine-type communication using multi-metric context-awareness for cars used as mobile sensors in upcoming 5G networks," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, 2018, pp. 1–6.

[32] C. Gijón, M. Toril, S. Luna-Ramírez, M. L. Marí-Altozano, and J. M. Ruiz-Avilés, "Long-term data traffic forecasting for network dimensioning in LTE with short time series," *Electronics*, vol. 10, no. 10, p. 1151, May 2021.

[33] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 124–161, 1st Quart., 2016.

[34] H. D. Trinh, L. Giupponi, and P. Dini, "Mobile traffic prediction from raw data using LSTM networks," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1827–1832.

[35] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019.

[36] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Interactive temporal recurrent convolution network for traffic prediction in data centers," *IEEE Access*, vol. 6, pp. 5276–5289, 2018.

[37] L. Nie, D. Jiang, L. Guo, and S. Yu, "Traffic matrix prediction and estimation based on deep learning in large-scale IP backbone networks," *J. Netw. Comput. Appl.*, vol. 76, pp. 16–22, Dec. 2016.

[38] M. Nashaat, A. Ghosh, J. Miller, and S. Quader, "TabReformer: Unsupervised representation learning for erroneous data detection," *ACM/IMS Trans. Data Sci.*, vol. 2, no. 3, pp. 1–29, Aug. 2021.

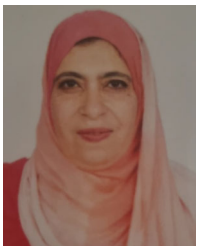
[39] H. Huang, X. Zhu, J. Bi, W. Cao, and X. Zhang, "Machine learning for broad-sensed internet congestion control and avoidance: A comprehensive survey," *IEEE Access*, vol. 9, pp. 31525–31545, 2021.

[40] H. Xia, X. Wei, Y. Gao, and H. Lv, "Traffic prediction based on ensemble machine learning strategies with bagging and LightGBM," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[41] W. Wang, Y. Bai, C. Yu, Y. Gu, P. Feng, X. Wang, and R. Wang, "A network traffic flow prediction with deep learning approach for large-scale metropolitan area network," in *Proc. NOMS IEEE/IFIP Netw. Operations Manage. Symp.*, Apr. 2018, pp. 1–9.

[42] P. Shrivastava and S. Patel, "Selection of efficient and accurate prediction algorithm for employing real time 5G data load prediction," in *Proc. IEEE 6th Int. Conf. Comput., Commun. Autom. (ICCCA)*, Dec. 2021, pp. 572–580.
[43] N. H. Mohammed, H. Nashaat, S. M. Abdel-Mageid, and R. Y. Rizk, "A machine learning-based framework for efficient LTE downlink throughput," in *Artificial Intelligence for Sustainable Development: Theory, Practice and Future Applications*. Springer, 2021, pp. 193–218.
[44] A. Ghosh, M. Nashaat, J. Miller, and S. Quader, "Context-based evaluation of dimensionality reduction algorithms—Experiments and statistical significance analysis," *ACM Trans. Knowl. Discovery from Data*, vol. 15, no. 2, pp. 1–40, Jan. 2021.
[45] S. M. M. A. Hashish, R. Y. Rizk, and F. W. Zaki, "Joint energy and spectral efficient power allocation for long term evolution-advanced," *Comput. Electr. Eng.*, vol. 72, pp. 828–845, Nov. 2018.
[46] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *Proc. IEEE 7th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2020, pp. 747–748.
[47] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "DBSCAN: Past, present and future," in *Proc. 5th Int. Conf. Appl. Digit. Inf. Web Technol. (ICADIWT)*, Feb. 2014, pp. 232–238.
[48] P. Fu and X. Hu, "Biased-sampling of density-based local outlier detection algorithm," in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Aug. 2016, pp. 1246–1253.
[49] X. Wu, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.
[50] A. M. Ahmed, A. Rizaner, and A. H. Ulusoy, "A decision tree algorithm combined with linear regression for data classification," in *Proc. Int. Conf. Comput., Control, Electr., Electron. Eng. (ICCCEEE)*, Aug. 2018, pp. 1–5.
[51] M. Osman, J. He, F. M. M. Mokbal, N. Zhu, and S. Qureshi, "ML-LGBM: A machine learning model based on light gradient boosting machine for the detection of version number attacks in RPL-based networks," *IEEE Access*, vol. 9, pp. 83654–83665, 2021.
[52] D. Zhang and Y. Gong, "The comparison of LightGBM and XGBoost coupling factor analysis and prediagnosis of acute liver failure," *IEEE Access*, vol. 8, pp. 220990–221003, 2020.
[53] R. F. Abdel-Kader, R. M. Ramadan, and R. Y. Rizk, "Rotation invariant face recognition based on hybrid LPT/DCT features," *Int. J. Electr. Comput. Eng.*, vol. 2, pp. 1613–1618, Jan. 2008.
[54] R. L. Devi and V. Saminadan, "Machine learning based traffic prediction system in green cellular networks," in *Proc. 1st Int. Conf. Comput. Sci. Technol. (ICCST)*, Chennai, India, Nov. 2022, pp. 593–596.
[55] H. Nashaat and R. Rizk, "Handover management based on location based services in F-HMIPv6 networks," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 12, pp. 5028–5057, 2015.
[56] A. H. Ali, M. G. Yaseen, M. Aljanabi, S. A. Abed, and C. Gpt, "Transfer learning: A new promising techniques," *Mesopotamian J. Big Data*, pp. 29–30, Feb. 2023.
[57] M. Gamal, R. Rizk, H. Mahdi, and B. Elhady, "Bio-inspired based task scheduling in cloud computing," in *Machine Learning Paradigms: Theory and Application*, A. E. Hassanien, Eds. Cham, Switzerland: Springer, 2019, pp. 289–308.
[58] R. Attia, A. Hassaan, and R. Rizk, "Advanced greedy hybrid bio-inspired routing protocol to improve IoV," *IEEE Access*, vol. 9, pp. 131260–131272, 2021.

HEBA NASHAAT (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer and control engineering from Suez Canal University, in 2001 and 2006, respectively, and the Ph.D. degree in computer and control engineering from Port Said University, Egypt, in 2011. Recently, she has been the Executive Director of the Software Engineering Unit (SWEU), Faculty of Engineering, Port Said University, and the Manager of the Electronic Learning Center (ELC), Port Said University. She is currently an Associate Professor with the Electrical Engineering Department, Port Said University. She is the Former Executive Director of the Network Infrastructure Center, Port Said University. Her research interests include computer networking, including mobile networks, cloud computing, and the Internet of Things.



NIHAL H. MOHAMMED received the B.Sc. degree in computer and control engineering from Suez Canal University, in 1992, and the M.Sc. degree in computer and control engineering from Port Said University, in 2017. She is currently the Head Manager in the general authority of educational buildings, Egypt, in the computer maintenance and power section, which is responsible for making electrical and networking infrastructure of educational buildings. Her research interests include mobile networking and using AI in it.

SALAH M. ABDEL-MAGEID received the M.S. and Ph.D. degrees in systems and computer engineering from Al-Azhar University, in 2002 and 2005, respectively. He performed his postdoctoral research with the Computer Science and Engineering Department, School of Engineering, Southern Methodist University, Dallas, TX, USA, in 2007 and 2008. He was a member of the Tool for Extensive Management and Performance Optimization (TEMPO) Project at Cairo University and Vodafone Egypt as an Industrial Partner in 2014 and 2015. He is currently a Professor with the Computer Engineering Department, College of Computer Science and Engineering, Taibah University, Saudi Arabia. His research interests include mobile computing, cellular networks, sensor networks, cognitive radio networks, vehicular ad-hoc networks, IoT protocols and security, robotics navigation, and deep learning.

RAWYA Y. RIZK (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computers and control engineering from Suez Canal University, in 1991, 1996, and 2001, respectively.

She has been the Manager of CISCO Academy, Faculty of Engineering, PSU, since 2010. She was the Executive Director of PSU Network Infrastructure, from 2010 to 2014. She was the Chief Information Officer (CIO) of Port Said University (PSU), Egypt, from 2014 to 2021. She was the Head of the Electrical Engineering Department, PSU, from 2017 to 2021. She was the Vice President of Postgraduate and Research, from 2021 to 2025. She is currently a Professor of computers and control with the Electrical Engineering Department, PSU. Her research interests include computer networking, including mobile networking, cloud computing, the IoT, sensor networks, and the fields of AI. She is a reviewer of many international communication and computer journals, such as IEEE Transactions on Intelligent Transportation Systems, IEEE Access, *IET Communications*, *IET Wireless Sensor Systems*, *IET Networks*, and *The Journal of Supercomputing*.