Avrupa Bilim ve Teknoloji Dergisi
Sayı 27, S. 370-385, Kasım 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 27, pp. 370-385, November 2021

Research Article

A Hybrid Algorithm for Changepoint Aware Long-Term Seasonality Detection of Mobile Network Base Stations

Y. Tarık Kranda[1], Ruya Samli[2*]

[1] Istanbul University-Cerrahpasa, Faculty of Engineering, Department of Computer Engineering, İstanbul, Turkey, (ORCID: 0000-0002-5291-2277), ykranda@ogr.iu.edu.tr

[2*] Istanbul University-Cerrahpasa, Faculty of Engineering, Department of Computer Engineering, İstanbul, Turkey, (ORCID: 0000-0002-8723-1228), ruyasamli@iuc.edu.tr

(First received 1 May 2021 and in final form 12 September 2021)

(**DOI:** 10.31590/ejosat.931099)

ATIF/REFERENCE: Kranda, Y. T. & Samli, R. (2021). A Hybrid Algorithm for Changepoint Aware Long-Term Seasonality Detection of Mobile Network Base Stations. *European Journal of Science and Technology*, (27), 370-385.

Abstract

Automated capacity planning for mobile networks requires long-term forecasting of traffic demand by using historical patterns. To decide the correct time of investment and correct capacity expansion size or to improve the accuracy of forecasting algorithms with exogenous features, both seasonal decomposition, and seasonal period identification improves decision accuracy. We design a hybrid algorithm to calculate these features on live network data with improved accuracy which uses piecewise Seasonality Trend Decomposition with Loess (STL) decomposition and Prophet library's regression with Laplace prior under the hood. Combining both methods with the awareness of their weak and strong parts and leveraging overall output with changepoint and similarity analysis help us to improve our accuracy around 18.6% comparing the average of single usage of these methods. We also provide and present some special cases that increase problem complexity and decrease decomposition accuracy.

Keywords: Mobile Network Seasonality Detection, Changepoint Awareness, Network Capacity Planning, Piece-wise Regression, Time Series Decomposition

Mobil Ağ Baz İstasyonlarının Değişim Noktalarının Uzun Dönem Sezonsallık Tespiti için Hibrid Bir Algoritma

Öz

Mobil ağlar için otomatik kapasite planlaması, geçmiş kalıpları kullanarak trafik talebinin uzun vadeli tahminini gerektirir. Doğru yatırım zamanına, doğru kapasite genişletme boyutuna karar vermede veya dışsal etkilere sahip tahmin algoritmalarının doğruluğunu iyileştirmede hem mevsimsel ayrıştırma hem de mevsimsel dönem tanımlama işlemleri karar doğruluğunu artırır. Bu çalışmada bu işlemleri, altyapısında parçalı Loess ile Mevsimsel Trend Ayrışımı (Seasonality Trend Decomposition with Loess – STL) ayrıştırması ve Prophet Kütüphanesi'nin Laplace önsele sahip regresyon yaklaşımını kullanan ve canlı ağ örnekleri üzerinde daha yüksek doğrulukla gerçekleştiren hibrid bir algoritma tasarlanmıştır. Her iki yöntemi de zayıf ve güçlü parçalarının farkındalığıyla birleştirmek ve değişim noktalarının benzerlik analizi ile tespit edilmesi üzerine geliştirilen çözüm, bu yöntemlerin tek başlarına elde ettiği ortalama başarımı yaklaşık %18,6 oranında artırmaktadır. Ayrıca çalışma kapsamında, problemin karmaşıklığını artıran ve ayrıştırma doğruluğunu azaltan bazı özel durumlar da sunulmuştur.

Anahtar Kelimeler: Mobil Ağ Sezonsallık Tespiti, Değişim Noktası Farkındalığı, Mobil Ağ Kapasite Planlama, Parçalı Regresyon Analizi, Zaman Serisi Bileşenlerine Ayırma

1. Introduction

Many industries are turned into digitalized processes so the number of mobile internet users is rapidly growing in recent years. This is followed by the growing demand for Machine-to-Machine (M2M) communication and Internet of Things (IoT) transformation for businesses and individuals. Global Covid-19 pandemic effect is another acceleration source for the growing demand for digitalized businesses and relevant mobile application usage. Every single day new internet services and applications are deployed, so the load to be managed by Mobile Network Operators (MNO) is increasing. Regarding GSM Association (GSMA) Intelligence Mobile Economy Research Report, by the year 2025, there will be 1.2 billion new subscribers being served by mobile networks (GSMA, 2020). The ratio of mobile subscribers over the regional population is depicted in Figure 1.

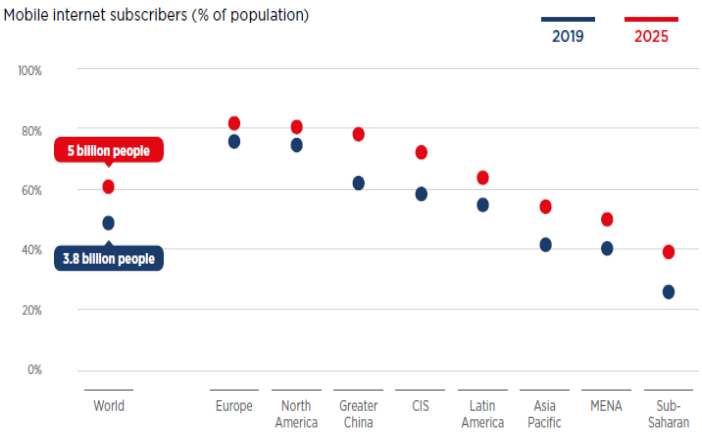

Figure 1. Mobile internet subscriber's percentage of population by 2025 (GSMA Intelligence, 2020)

The variety of services corresponds to a different amount of traffic generation on the network, and each has different customer experience requirements. Throughput-sensitive applications such as video streaming or latency-centric applications such as online gaming which are served through the same mobile base stations exist. Variance on multiple services and their customer expectation increases the complexity of the mobile network capacity planning process which is aimed to make correct investments at the correct time to manage the balance between Capital Expenditure (CAPEX) and Return-on-Investment (ROI). As depicted in Figure 2, the second-highest response of MNOs for infrastructure renewal and network transformation decisions is improving customer experience (GSMA, 2020). One another key point of this research worth mentioning is also declining CAPEX as much as possible to keep MNOs profitable.

All these increasing demands, smart ROI management pressure, improving customer experience requirements to decrease churn threat push MNOs to digitalize their internal services and make data-centric intelligent decisions. In MNO organizations, there are regional and central network investment planning teams who are in charge of capacity investment decisions by considering all the above dimensions. A generalized regular process of network capacity planning lifecycle is shown in Figure 3. The number of consumed data sources is increasing over time in the process. To digitalize decision process with more data-centric approach, all process is being softwarized to rely on data analysis and demand forecasting. Similar to all data-oriented solutions, the process starts with data collection operations which consume data from multiple data sources. The most commonly utilized data source is Element Management Systems (EMS) logs which provide performance counters to calculate Key Performance Indicators (KPI) about resource utilization statistics. For instance, the number of users connected to Base Station (BS), carried amount of traffic in Megabytes, Physical Resource Block (PRB) utilization for LTE cells, and so on. All these statistics are labeled with timestamps that indicate certain time intervals (i.e., every 5, 15, 30, or 60 minutes). Besides network KPIs, there are some other data sources to utilize for improved investment accuracy such as geolocated customer complaints from Customer Relationship Management (CRM) systems, crowd-sourced quality samples collected from subscriber User Equipment (UE), signal propagation modeling tool exports (binned coverage signal levels), transport layer statistics and so on. It is common to use EMS-generated KPIs to use for traffic forecasting. High-quality forecasting for traffic and demand requires some preprocessing operations such as data cleaning, missing data imputation, anomaly healing, and daily or weekly aggregation of data for long-term regressions.

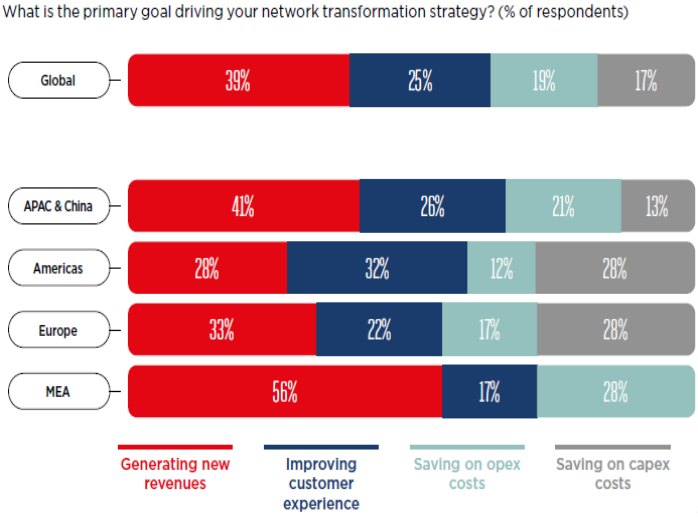

Figure 2. The primary goal driving MNOs' network transformation strategy (GSMA Intelligence, 2020)

Multiple key factors affect the overall forecasting and inherently investment decision accuracy. These factors are listed as the quality of data, having enough data to cover targeted seasonality search, being aware of change points when identifying seasonal factors, impacts of special days, and so on. In concise, it is significant to understand features of time series data and decompose it correctly. MNO policies and business rules are applied on forecasted traffic and other utilization KPIs and it is followed with investment suggestions. At the last phase of the lifecycle, all suggestions are evaluated regarding to pre-defined constraints and orchestrated. Conflicting suggestions which may be impacting each other in low distances are resolved concerning MNO policies and priorities. The distilled suggestion list is applied at investment periods sorted by ROI.

Detecting the seasonality impact of base stations regardless of changepoints in their history improves the accuracy of investment suggestion decisions. Especially, identifying the locations which can be compensated with temporary mobile cells, baseband units, and accumulators in small buses makes more saving than perpetual investment.

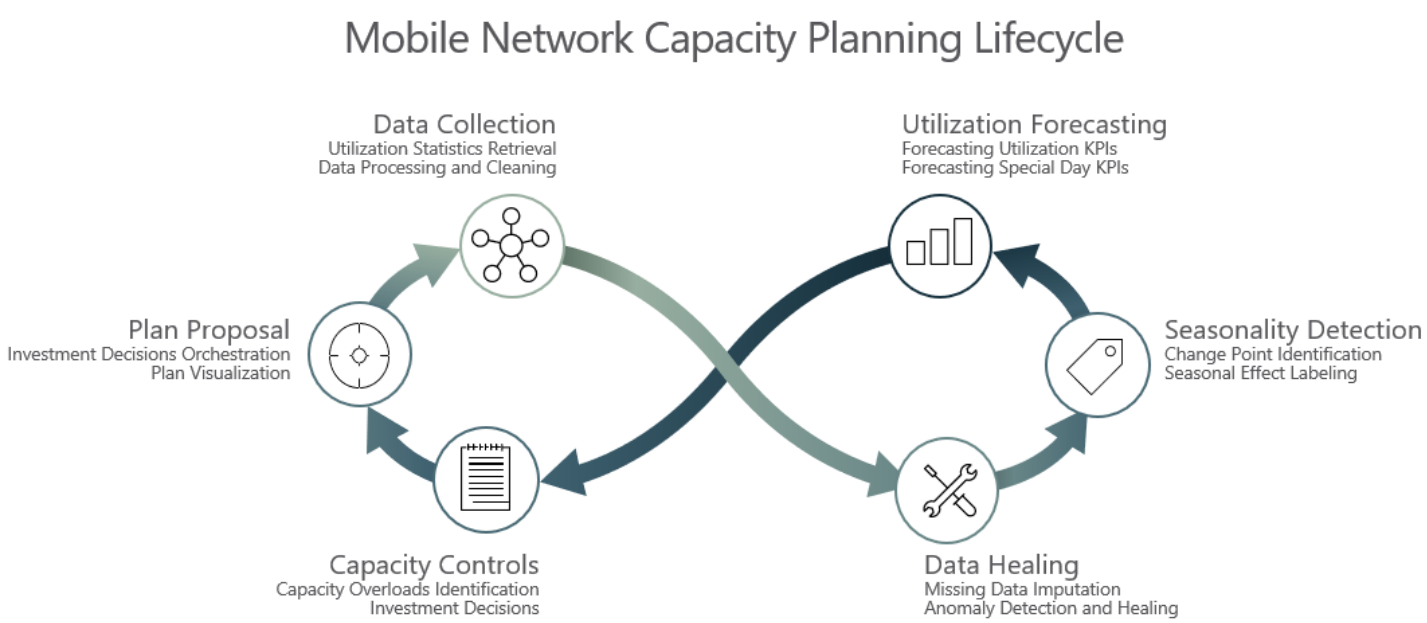

Figure 3. Mobile network capacity planning lifecycle

Finding the correct time of investment on high seasonal base stations for some sectoral capacity expansion is also another cost-saving factor for MNOs. Hence, they make expansions to correct base stations at the correct time to preserve customer experience. Unnecessary investment causes another cumbersome cost on the environment as well due to carbon footprint increment. Managing the capacity of a location with the correct number of stations decreases MNO's carbon footprint and saves energy. Centralized energy saving algorithms are also required to be aware of high season start-end dates for their predictive accuracy.

In this study, we process real network daily traffic data and develop a new hybrid solution to identify long-term seasonality impact over base stations. After this process, we design an algorithm for seasonality detection and decomposition of mobile network base station traffic data to extract seasonality impact on base stations by considering changepoint existence due to level shift or exponential trend increase.The algorithm consists of built-in changepoint search, change ratio analysis, and similarity controls mechanisms inside. It can calculate multiplicative seasonal factors by using two-years historical traffic data. Labeling base stations with decomposed seasonality impact helps to forecast algorithms as new features when predicting next year's traffic demand. We compare our results with 100 base stations with two years of historical data taken from a European Tier-1 operator. The dataset consists of 4119 base stations' data; 150 of them are labeled by three Subject Matter Experts (SMEs) from central planning teams, 100 of them are used as the test set to benchmark the proposed solution with pure Prophet library-based decomposition and STL solution. We introduce our dataset, distribution of base station characteristics and share our algorithm results with benchmarking. It is critical for the mobile network capacity planning lifecycle to have both these capabilities (being aware of changepoints and identifying seasonality impact on a base station) unified as one solution to be able to develop a software-oriented planning approach. So, the solution would be able to work as an unsupervised solution with offline analysis and learning capabilities on historical data.The outputs of algorithms (seasonal labeling of base stations and percentage-based amount of seasonal impact) can be supplied into forecasting algorithms including Neural Networks (NN) or Machine Learning (ML) based solutions or to rule-based network management policy applications. To the best of our knowledge, there are no other studies focused on long-term changepoint aware seasonality labeling of mobile base stations and tested with real network data under the supervision of SMEs.

The rest of the study is as folows. In Section 2, related works about the subject of the study are given. In Section 3, materials and methods used in this study are presented. In Section 4, experimental results are analyzed and discussed and finally the study is concluded in Section 5.

2. Related Works

Time series is a sequence of observations labeled with a timestamp as data points in successive order. Each time series has some characteristic features as dependent on the nature of the stochastic process that generates its observations. Time series may contain multiple seasonal cycles for different periods. For example, the hourly utility demand data exhibits both daily and weekly cycles (Gould et al., 2008). In the characteristics of base station traffic time series, it is common to see multiple seasonal components such as weekly, monthly, and yearly seasonality which means non-stationarity as it is analyzed in the "rationalization section" the study by Yu et al. (2010). Same patterns exist in our dataset as well, but for capacity planning, we focus on long-term, low-frequency seasonal components which are yearly seasonality. Mobile network traffic can be decomposed into multiple components in additive or multiplicative form as classical decomposition methods. It is formulated by Lakshmanan & Das (2017) such that time series Y to be modeled or forecast is viewed as (1) depending on additive or multiplicative model, where T represents the trend or the long-term direction, S represents the seasonality or a pattern which repeats periodicity and ϵ represents the random error and i.i.d (independent and identically distributed)

$$Y = T + S + \epsilon \quad \text{or} \quad Y = T \,.\, S \,.\, \epsilon \tag{1}$$

It can also be represented as low frequency and higher frequency elements as given by (2), where the trend component is low-frequency changes and seasonality is addressed as higher frequency changes.

$$Y = Y_{low} + S_{high} + \epsilon \quad \text{or} \quad Y = Y_{low}\ x\ S_{high}\ x\ \epsilon \tag{2}$$

Anomalies such as outliers and level shifts are quite common in time-series data. The presence of outliers and level shifts (changepoints) poses problems for the identification and estimation of Autoregressive Integrated Moving Average (ARIMA) models (Balke, 1993). In this study, we define our algorithm based on multiplicative decomposition as in (1). We complete many empirical analyses for two years of historical data and observed multiplicative seasonality characteristics in the traffic data. Since the mobile network traffic is highly non-stationary due seasonal characteristics besides changepoints, an algorithm to reveal seasonal impact in mobile traffic is required to consider both changepoints and seasonal movements together. At the inner sections of our algorithm, we apply log-transformation to turn our multiplicative characteristics into additive ones for easier calculations and evaluation for changepoints existence controls.

In this section, we analyze some previous studies about changepoints detection as a subset of anomaly detection and seasonality decomposition algorithms. The unique output of our algorithm is to find a satisfactory algorithm, working on live mobile network historical data as an offline and batch solution. The algorithm should be successful at detecting seasonal impact on base station traffic even if it includes changepoint in its history. There are studies about changepoints detection and seasonality detection separately on other domains and have not combined together for investigating mobile network traffic. It is beneficial for healthy forecasting of mobile traffic demand and accuracy of capacity investment decisions because many base stations have changepoints due dynamism of mobile networks. For instance, a newly deployed base station is open to steal traffic from existing neighbor base stations or Covid-19 pandemic quarantine alerts may cause unexpected level shifts in specific regions like business areas or leisure locations. We observe similar behavior in our dataset and implement an algorithm that combines piece-wise Bayesian regression-based change points distribution analysis, similarity search, and flexible STL decomposition.

2.1. Changepoints Detection

Changepoints are statistical changes of the probability distribution on the output of stochastic processes. Let $X = \{X_t,\ t = 1, \dots, T\}$ be the series of observed traffic volume samples, then if we say null hypothesis given by (3) is no change in the traffic, the alternative hypothesis given by (4) would be a change points existence at point k (Chen & Zhang, 2015).

$$H_0: P_{X_1} = P_{X_2} = \dots = P_{X_n} \tag{3}$$

where $\{1, \dots, n\}$ are intervals of X_t series and P_{X_n} is the probabilistic density function of interval n.

$$H_1: \exists\ 1 \le \mathrm{k} < \mathrm{n}, \begin{cases} P_{X_k}, & 1 \le \mathrm{k} \\ P_{X_n}, & k < n \end{cases} \quad P_{X_k} \neq P_{X_n} \tag{4}$$

Multiple studies address changepoint detection or anomaly detection algorithms on time series data to identify unexpected changes in series. Some of them are listed in the study of Aminikhanghahi & Cook, (2017). Changepoint detection algorithms are categorized into several categories mainly as online and offline algorithms. Online algorithms run concurrently with the process they are monitoring, processing each data point as it becomes available, and as a real-time solution, processing should be completed before the next data point arrives. On the other hand, offline algorithms analyze the entire data set at once, and there is no real-time decision during the run-time period (Downey, 2008). There are also other categories of changepoint detection algorithms such as univariate, multivariate, model-based, non-parametric as listed by Burg & Williams (2020). One of the initial approaches is the Cumulative Sum Control (CUSUM) chart method which is widely used and proposed by Page (1954). It is to detect a change in the mean of the distribution as a weighted sum of last k observations. It considers the change of cumulative sum to be over a threshold. There are multiple derivatives of the CUSUM algorithm where one of them is an intuitive approach which has the typical behavior of the log-likelihood ratio S_k given by (5) and s_i given by (6) showing a negative drift before changepoint, and a positive drift after changepoint. It is visualized in Figure 4.

Let null hypothesis $H_0: \theta = \theta_0$ and alternative hypothesis $H_1: \theta = \theta_1$, then

$$S_j^k = \sum_{i=j}^{k} s_i \tag{5}$$

$$s_i = \ln \frac{P_{\theta_1}(y_i)}{P_{\theta_0}(y_i)} \tag{6}$$

be the log-likelihood ratio for the observations from y_j to y_k.

There are some other approaches which are constructed over likelihood ratio with binary segmentation. Scott and Knott (1974) studied the consequences of using a well-known method of cluster analysis to partition the sample treatment means in a balanced design. They showed how a corresponding likelihood ratio test gives a method of judging the significance of the differences among groups obtained.

Probabilistic methods are used for changepoint detection as well by segmenting time series and generating probability outputs of each segment for changepoint existence. Downey proposed an algorithm with Bayesian approach in which the kernel of the algorithm is a system of equations that computes, for each index i, the probability that the last (most recent) changepoint occurred at i. He evaluates this algorithm by applying it to the changepoint detection problem and compares it to the Generalized Likelihood Ratio (GLR) algorithm (Downey, 2008). It is also mentioned in the Downey's research that a special type of changepoint problem which is tracking. The goal of the tracking problem is to partition a time series into stationary intervals and estimate the parameters

of the process in each interval. A simple approach is to use hypothesis testing to detect a changepoint, estimate the location of the change(s), and then use conventional techniques to estimate the parameters of the process in each interval.

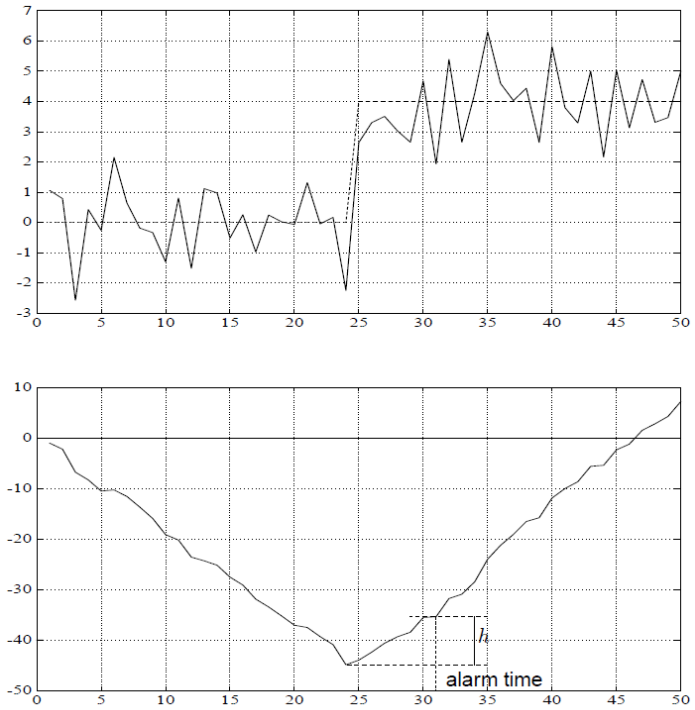

Figure 4. CUSUM log-likelihood ratio behavior to a change in the mean of a Gaussian sequence with constant variance (Basseville & Nikiforov, 1993)

Our goal in this study is not to define yet another changepoint detection algorithm, we used a Bayesian-based approach constructed on Prophet library's Bayesian approach where they specify a large number of changepoints and use a sparse prior as $Laplace(0,\tau)$. The parameter τ directly controls the flexibility of the model in altering its rate (Taylor & Letham, 2018). But we enhanced their automatic changepoint detection outputs by comparing the distribution of significant change ratios among years. Hence, we separated it from multiplicative seasonality changes, and this helps to distinguish real changepoints. The impact of the changepoint on-trend is identified and decomposed successfully. Normally this approach marks all multiplicative seasonal changes as a high chance of changepoints with a significant rate of changes and repeats real changepoints in future forecasting since it is not designed as dedicated to changepoint detection algorithm.

In parallel to the growing number of time series and streaming data, there is a growing demand to be able to estimate the location of multiple changepoints efficiently and accurately (Killick & Eckley, 2014). For multiple changepoints detection problems, Bayesian rules are also used. Let the series T is generated with a probability distribution of parameter θ. In this case, Bayes rule is

$$\mathrm{P}(\theta|T) = \frac{\mathrm{P}(\mathrm{T}|\theta)\, P(\theta)}{P(T)} \qquad (7)$$

where $\mathrm{P}(\theta|T)$ is posterior distribution, $\mathrm{P}(\mathrm{T}|\theta)$ is the likelihood and $P(\theta)$ is prior. So, a Bayesian Model to detect a single changepoint at C (as a discrete parameter, $C \sim Uniform\ (1,T)$) is targeted to find a posterior probability given by (7) which is validating hypothesis given by (4) at point C.

2.2. Seasonality Detection

Business time series often have multi-period seasonality as a result of the human behavior (Taylor & Letham, 2018). For automatic forecasting of traffic series for capacity planning of radio access networks, it is valuable to decompose the series into the level, growth, and seasonal components (Hyndman & Khandakar, 2008). It is not different for short-term and long-term time series generated by mobile networks traffic behavior (Nikravesh, et al., 2016; Cortez, et al., 2006; Tikunov & Nishimura, 2007; Sciancalepore, et al. 2017). There are different ways of modeling the seasonality component of the series. One of them is modeling the periodic effects as standard Fourier series given by (8) (Harvey & Shephard, 1993). It is similar for Trigonometric Exponential Smoothing State-Space model with Box-Cox transformation, Autoregressive Moving Average (ARMA) errors, Trend and Seasonal Components (TBATS) modeling (Livera, et al., 2011).

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos(\frac{2\pi nt}{P}) + b_n \sin(\frac{2\pi nt}{P}) \right) \qquad (8)$$

where P is the period length. For instance, $P = 12$ for monthly aggregated yearly data or $P = 24$ for half-month aggregation of yearly data.

There are filtering-based decomposition techniques with partial regression line fitting. Seasonal Moving Averages is one of them including weighted averaging as an alternative. Also, there are methods for seasonal decomposition of longer series which work over monthly or quarterly aggregated data such as the X-11 method, Seasonal Extraction in ARIMA Time Series (SEATS) method (Dagum & Bianconcini, 2016), or hybrid version of them as X-13ARIMA-SEATS. X-11 and SEATS methods are aware of unexpected sudden falls in the trend component if long enough data is provided to clearly distinguish strong seasonality patterns from the trend. But when the monthly aggregated data points are not so many similar to two years historical data in our dataset, it is not easy to model multiplicative yearly seasonality by using only X-11 or SEATS methods. As an alternative to these methods, there is STL for nonlinear relationships and flexible controlling of trend and seasonal components regardless of aggregation granularity of data unlike X-11 and SEATS methods. STL is a filtering procedure for decomposing time series into trend, seasonal and remainder components, respectively. STL has a simple design that consists of a sequence of loess smoother applications. The simplicity allows analysis of the properties of the procedure and allows fast computation, even for very long time series and large amounts of trend and seasonal smoothing. Other features of STL are the specification of amounts of seasonal and trend smoothing that range, in a nearly continuous way, from a very small amount of smoothing to a very large amount, robust estimates of the trend and seasonal components (Cleveland, et al., 1990). STL can only model components additively. To be able to model multiplicative seasonality log-transformation is required to be applied which frequently exists in our data set series for mobile networks. A Loess smoother is applied locally weighted polynomial regressions at each point in the dataset, with the explanatory variables being the values close to the point whose

response is estimated. The parameters for the STL procedure are obtained from the eigenvalue and frequency response analysis of a given time series. The iterated cycle is composed of two recursive procedures, the inner and the outer loop. If Y to be our traffic volume data and T represents the trend component or the long-term direction, S represents the seasonal component or a pattern which repeats periodically and ϵ represents the random error. In this case, the inner loop performs six basic steps (Theodosiou, 2011):

1- Detrending: Let k means the value of our components at k^{th}pass of the algorithm. At the first iteration, the trend is assumed to be zero. At iteration $k+1$, our traffic data is detrended with the components of the previous pass is given by (9) as

$$Y_t - T_t^{(k)} \tag{9}$$

2- Seasonal smoothing: Loess smoothing applied to the remainder of the first step to retrieve preliminary seasonal component, $\widetilde{S_t^{(k+1)}}$.

3- Filtering of smoothed seasonality: By applying a moving average to the preliminary seasonal component and then another Loess smoother, $\widetilde{T_t^{(k+1)}}$ is retrieved as the remaining preliminary trend component.

4- Detrending of smoothed seasonality: The additive seasonal component becomes the difference between these two preliminary components is given by (10) as

$$S_t^{(k+1)} = \widetilde{S_t^{(k+1)}} - \widetilde{T_t^{(k+1)}} \tag{10}$$

5- Deseasonalizing: To retrieve non-smoothed trend component for iteration k+1, seasonal component subtracted from original series is given by (11) as

$$Y_t - S_t^{(k+1)} \tag{11}$$

6- Trend smoothing: At the final step, the seasonally adjusted remainder series is smoothed with Loess to give an estimate of the trend component $T_t^{(k+1)}$

Hence, each pass of the inner loop applies seasonal smoothing that updates the seasonal component, followed by trend smoothing that updates the trend component.

There are hybrid solutions like Seasonal Hybrid Extreme Studentized Deviate (S-H-ESD) that is built on the ESD algorithm (Rosner, 1975) to address anomalies rather than changepoints in respect to seasonal patterns in series. In this study, we addressed a similar problem for changepoint aware seasonal impact extraction in mobile radio network traffic series. Our solution is built as a hybrid solution by using changepoint detections with the Bayesian approach of Prophet library, changepoint distribution similarities analysis for changepoint smoothing, and followed by STL decomposition. Hence, we successfully segregate the changepoint effect from actual multiplicative seasonal effects as an automated decomposition solution.

3. Material and Method

In this section, we explain the dataset utilized to develop our solution, validate its accuracy and its estimate parameters and the completed preprocessing operations required to feed the data into the algorithm and finally the workflow of the proposed algorithm.

3.1. Dataset

It is not so easy to work and analyze long-term historical patterns of mobile network data due to the lack of recorded historical data allowed for external access. There are some public datasets from mobile operators like Telekom Italia or China Mobile shared for hackathons and academic purposes, but they do not include long enough historical data like two years and more. Many MNOs just in recent years started to build their data warehouses or data lakes for long-term strategic analytical purposes. To analyze and develop our algorithm, we work on a proprietary dataset of a Tier-1 European MNO. Collected data includes two years historical Long-Term Evaluation (LTE) traffic volume data in Megabytes, base station ids, cell ids, sector ids, city, region, band and vendor information. Dataset statistics are given in Table 1.

Table 1. Dataset statistics

Dataset Breakdown	**Total Count**
Number of regions	3 regions
Number of cities	4 cities
Number of cells	20680 LTE cells
Number of sites	4119 sites
Number of freq. layers	4 bands
Number of vendors	3 vendors
Time Interval	2018-Aug-1 / 2020-Aug-1

We selected 150 base stations from overall dataset which have different characteristics in terms of seasonality, changepoint existence. Characteristic of each base station is labeled by three SMEs. Total of 50 of them used for exploratory analysis and parameters tuning for supervision. Remained 100 base stations are used for model accuracy testing and not used for any analysis or parameter tunning purposes.

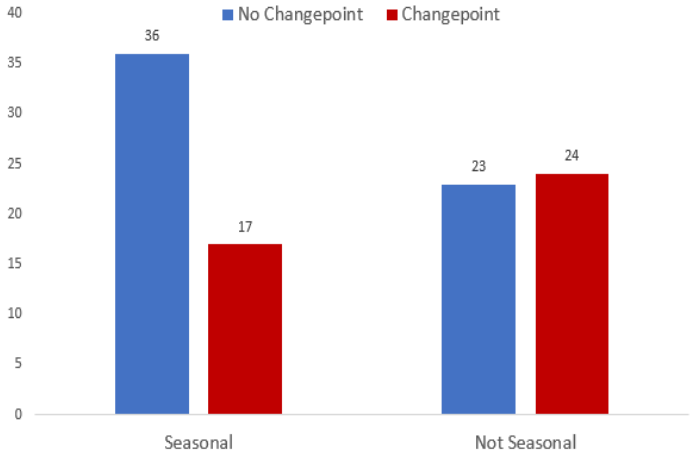

Figure 5. Distribution of labeled test base stations

Distributed number of categorized sites with SME labeling is shown in Figure 5. SMEs define 30% seasonal impact as a threshold to label a site as seasonal. So, let $\{1, \dots, n\}$ be intervals of $X_n^{(k)}$ traffic values serie of the base station k. If we denote seasonal component for a specific time (n) as $S_n^{(k)}$ and trend component as $T_n^{(k)}$, SMEs define a ratio threshold based on investment policies which are denoted with λ and defined as 0.30 in our study. To label a base station as seasonal ($Is\_Sea^{(k)} = 1$) or not seasonal ($Is\_Sea^{(k)} = 0$), we use the formula given by (12).

$$Is\_Sea^{(k)} = \begin{cases} 1, & \exists\, n \ni \dfrac{S_n^{(k)}}{T_n^{(k)}} \geq \lambda \\ 0, & \forall\, n \ni \dfrac{S_n^{(k)}}{T_n^{(k)}} < \lambda \end{cases} \quad (12)$$

If the ratio of seasonal and trend components of a base station go over 30% for any time interval, that base station is labeled as seasonal, otherwise it is labeled as non-seasonal even if it shows seasonal periodicity with lower levels. Non-seasonal base stations are not critical incapacity planning decisions when considering high season start-end dates. As displayed in Fig. 5, there are 53 base stations marked as seasonal and 47 as non-seasonal. 17 of the seasonal base stations have changepoints due to some reasons which may differ such as Covid-19 pandemic quarantines or traffic stealing of another near field cell investment. Changepoints can be in form of level shifts or trend increase or decrease.

3.2. Preprocessing

Some preprocessing operations are applied to make raw data more suitable for long-term seasonality and changepoint analysis. Collected data consists of cell level daily traffic volume and relevant cell planning data. Long-term seasonal periodicity and trend shifts are clearer when statistics are aggregated to get rid of misleading outliers. To expand the related base stations with new sectorial cell expansions or mobile base stations assignment at the correct time, the aggregation of daily data should not be too wide unnecessarily. Also, yearly seasonal characteristics are required to be preserved. So, the following steps are applied beforehand to submit data into the algorithm:

1- *Data cleaning*: There are some abnormal values due to some problems with EMS servers i.e., negative valued traffic volume data or missing cell id information, and so on.
2- *Missing data interpolation:* It is common to see missing daily traffic data due to some EMS server outages. Those values are interpolated with window-based (3 weeks before and 3 weeks after values) averaging of same days of weeks.
3- *Anomaly healing:* There are traffic values unexpectedly high or low due to incorrect reading of counters by Extract, Transform, and Load (ETL) platforms or EMS problems. These outlier days are detected with seasonal scaled Median Absolute Deviation (MAD), removed, and interpolated with correct values like in step 2. Weakly seasonality is considered in this step.
4- *Base Station level aggregation:* To decrease the processing time and discarding cell-level changes to focus more on location, cell-level data is aggregated into base station level daily data with summation of cell-based daily traffic volume under the same base station.
5- *Removal of new base stations:* Young base stations with less than 2 years of history are excluded from the dataset for yearly seasonality analysis.
6- *Storing processed data:* For future analysis, processed outputs are stored in the database.

Preprocessing operations are done automatically at once with some custom Python and SQL scripts and stored into PostgreSQL relational database. Batch analysis algorithm is consumed data stored in the database.

3.3. Algorithm

Preprocessed data are submitted into our hybrid solution for seasonal impact analysis. Hybrid solutions are aimed to utilize stronger functionalities of other solutions to yield much better outputs on specific problems. In this section, we provide the workflow of our hybrid algorithm that combines three steps under the hood which are using the flexibility of piece-wise Bayesian regression with a Laplacian (double exponential) sparse prior, rate of change-based energy distribution comparison for changepoint and seasonality search, and lastly decomposition of series components with STL's Loess smoother, respectively. To develop and test our hybrid solution, we utilized two external libraries coded in Python which are widely used in academia and industry. Initial Bayesian regression with sparse prior is built on top of Facebook's Prophet library (Taylor & Letham, 2018) which provides the ratio of changes distribution over time by using probabilistic Stan framework (Stan Development Team, 2020) at the backend. It is also explained by Erp et al. (2019). The other library is the statsmodels library (Seabold, et al., 2010) which we used for the STL algorithm in the last decomposition step.

The algorithm starts with aggregating daily base station traffic volume data into bi-weekly and monthly series to get rid of daily fluctuations and focusing more on yearly seasonal characteristics which are more critical for long-term investment decisions. Both aggregated series are log-transformed due to multiplicative seasonal characteristics of mobile radio networks. So, the seasonal impact over the base stations is a function of trend and showing multiplicative impact as a result of our explanatory analysis. A sample is shown in Figure 8.

We fit highly flexible Bayesian Lasso regression by using the Prophet library model on log-transformed traffic series to retrieve potential abnormal behavior on change ratio distribution. With the help of disabled seasonality, the ratio of change is explained inside the trend curve, hence seasonal changes are also retrieved in terms of change ratios. For every single month of two years, a potential changepoint is placed when the curve is fitted for further investigation of changepoints distribution and Laplace distribution swallows the weak changes. Both bi-weekly and monthly series are also decomposed with STL to check for trend-seasonal components changes. Both these indicators are used to decide if traffic series include changepoints and still show a seasonal characteristic or not. If history includes changepoint regarding the first indicator and also the second indicator, seasonal characteristic indicator, is positive, then a flexible trend curve is fitted to cover changepoint inside and subtracted from the main trend component for changepoint smoothing before STL decomposition. This helps changepoint impact to be segregated from seasonal movements and explained inside trend component. Prophet library tends to label high multiplicative seasonal rises as potential changepoints and STL library also tends to explain changepoints in the seasonal component. Also, Prophet library

explains seasonal component with Fourier series, and this creates a tendency for fitting seasonal impact as a multiplier of the previous year which has a lower trend component. This causes wrong fitting the level of seasonal rise for next year.

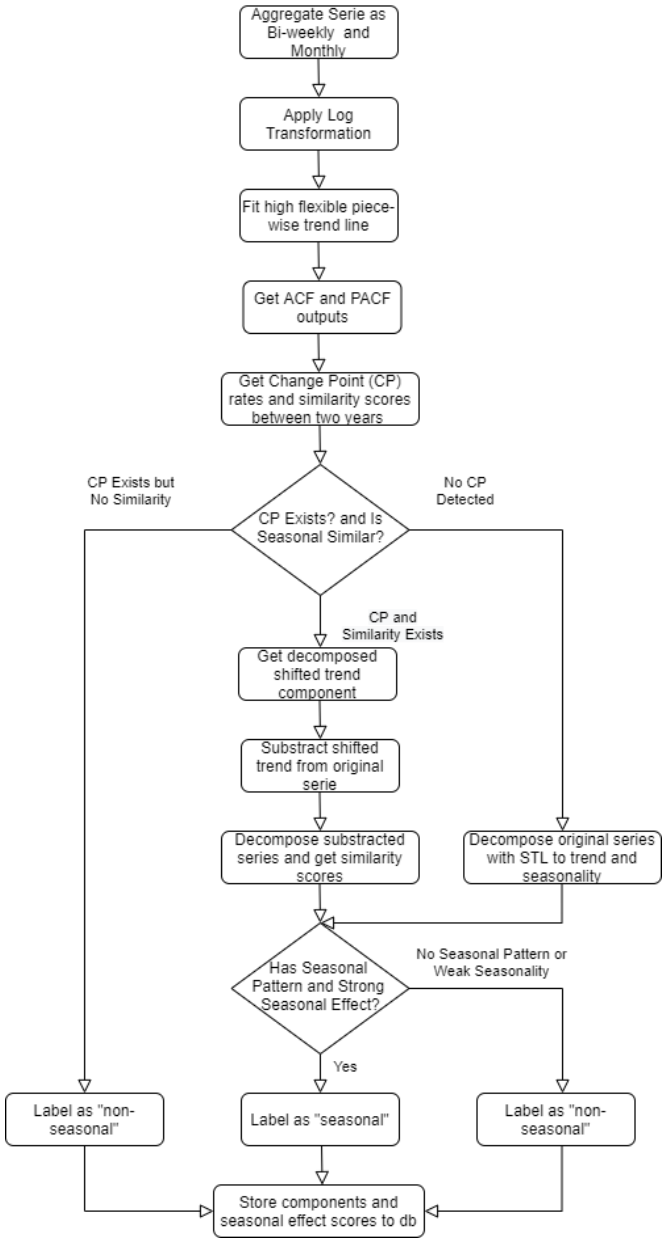

Figure 6. The main workflow of the algorithm

In our study, we yield five outputs, which are,

- strength of positive seasonal impacts in terms of Seasonal Power Index (SPI) values calculated with the formula given by (13),
- length of this seasonal periods in days,
- the overall SPI value of the base station in respect to formula given by (14),
- existence of changepoints in its history,
- label of the base station as “seasonal” or “not seasonal” by comparing its max SPI period with the 30% threshold definition of SMEs.

All 100 base stations in our testbed are benchmarked in these dimensions with previously done SME decisions and results are shared in section 4.

$$SPI_M = \frac{F_S(M)}{F_T(M)} \tag{13}$$

$$d_j(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{14}$$

3.1.1. Changepoint and Similarity Controls

The most common and widely used time series decomposition algorithms like seasonal moving averaging, STL, or Prophet library suffer from changepoints if they are not identified correctly. This situation causes some incorrect decomposition such as mapping the dramatic changepoint change into seasonal component rather than trend component, fitting trend component incorrectly by smoothing seasonal characteristics unnecessarily, or mixing the real changepoints with highly multiplicative seasonal changes. These types of problems are pretty common when we look at mobile radio networks due to their non-stationary structure under the impact of changing customer behaviors or ongoing investment-based expansions of networks. So, to be able to design automated capacity planning solutions, it is a must to cover changing story of every single base station. It is straightforward to decompose seasonal and trend components for all common decomposition methods if the base station has clear seasonality with additive or lower scaled multiplicative seasonality with no changepoints. But when the series is shorter similar to two years history in our dataset and characteristic variance is high on many base stations, common methods are required to be aware of these situations and improve with some additional controls and adjustments. To identify potential changepoints' existence and seasonal periodicity in the log, controls in Algorithm 1 are applied.

Algorithm 1 *change ratio controls*

```
procedure exec_change_ratio_controls (bw_series, sig_thr, rat_thr)
  set c_prio_sc ← 1.5, c_rng ←1.0, cp_in_y1 ← false, cp_in_y2 ← false
  set model ← init Prophet with no_sea, c_prio_sc, c_rng
  call model. fit with bw_series
  set rc_y1[ ], rc_y2[ ] ← split model.rate_changes[ ] yearly
  set max_y1 ← Max(Abs(rc_y1)), max_y2 ← Max(Abs(rc_y2))
  set num_sig_y1 ← 0.1, num_sig_y2 ← 0.1
  for each change ratio c1, c2 in rc_y1[ ], rc_y2[ ]
    if Abs(c1) > sig_thr then
      increment num_sig_y1
    end if
    if Abs(c2) > sig_thr then
      increment num_sig_y2
    end if
  end for
  if num_sig_y1 / num_sig_y2 > rat_thr or max_y1 / max_y2 > rat_thr
then
    set cp_in_y1 ← true
  elif num_sig_y2 / num_sig_y1 > rat_thr or max_y2 / max_y1 > rat_thr
then
    set cp_in_y2 ← true
  return cp_in_y1, cp_in_y2
end procedure
```

We fit highly flexible Bayesian Lasso regression by using the Prophet library model on log-transformed traffic series to retrieve potential abnormal behavior of change ratios. Prophet library itself has two types of growth models which are non-linear, saturating growth, and linear trend with changepoints, respectively. We used the latter one with flexible changepoint prior scale as a generalized additive model. The default components of Prophet library models are formulated in (12). Here $g(t)$ is the trend function which models non-periodic changes, $s(t)$ represents periodic changes (e.g., weekly, monthly or yearly seasonality), and $h(t)$ represents the effects of holidays which occur on potentially irregular schedules over one or more days and ϵ_t is for normally distributed error terms (Taylor & Letham, 2018). But since we focus on the comparison of change ratios between two years, if they are similar or not including seasonal changes, we do not fit seasonal characteristics At this step we only fit a growth model with flexible changepoint prior scale given by (15) and (16). The growth function consists of k as the growth rate, δ as the rate adjustments, m as the offset parameter. If we suppose there are S changepoints at times $s_j, j = 1, \dots, S$ then, the vector of rate adjustments as δ and δ_j is the change rate at the time s_j. Once the growth model is fitted with a flexible changepoint prior scale (default Prophet library value is 0.05), the model puts a sparse prior on δ as we do at this stage to get the clear ratio of changes. It is called sparse prior since the prior has double exponential (aka Laplace) distribution as given by (17). The parameter τ directly controls the flexibility of the model, due Laplace form. So, increasing this prior value makes the trend fitting more flexible. Hence, based on our empirical analysis over seasonal labeled base stations, we set it as $\tau = 1.5$, and by setting the changepoint range as 1.0, we look for 100% of the historical log-transformed traffic series. A change ratio over 0.8 is defined as significant change which also covers strong multiplicative seasonal upward and downward movements of traffic volume. The ratio of changes for the first and second year are compared by the number of significant ratios and maximum absolute values of ratios. In case of dissimilarity over a ratio distribution in both of these indicators are resulted as a potential changepoint indicator. There are some visualized samples in Section 4 for this scenario. We made an empirical analysis for changepoint including multiplicative seasonal base stations which are out of our test dataset and define this threshold as 3 times. Graph of a base station with multiplicative seasonal characteristics and ratio of changes of its log-transformed bi-weekly traffic is shown in Figures 8 and 12.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{15}$$

$$y(t) = g(t) + \epsilon_t, \tag{16}$$

$$\mathrm{g(t)} = (\mathrm{k} + \mathrm{a(t)}^{\tau}\delta)\mathrm{t} + (\mathrm{m} + \mathrm{a(t)}^{\tau}\gamma), \tag{17}$$

$$\delta_j \sim \mathrm{Laplace}(0, \tau) \tag{18}$$

Let P is seasonal period and $X = \{X_{S1}, X_{S2}, \dots, X_{SN} \mid N \leq P\}$ is the seasonal component index and $F_S(X)$, $F_T(X)$ are the decomposed functions of seasonality and trend components, respectively, then we calculate the strength of the seasonal impact of a month as SPI given by (18). When X = M where M means a certain month, the monthly SPI of that month is calculated as (13).

We checked for the existence of another indicator that helps us to distinguish base stations with seasonal similarities in terms of monthly SPI changes by using Jaccard distance as the complementary of Jaccard similarity score in (14) (Jaccard, 1912).

In our dataset there are base stations at high seasonal locations but also have level shifts in their history. Despite changepoints, they preserve seasonal characteristics, so the algorithm check for their characteristic similarity in terms of SPIs with the algorithm in Algorithm 2. To calculate monthly SPIs of two years, firstly we decompose log-transformed traffic series with STL decomposition algorithm, hence we have trend and seasonal components even if the existence of changepoint ruins seasonal component. SPI changes are binarized regarding their upward and downward direction and both binary direction vectors (as A and B) and SPI change powers are compared among two years. Jaccard distance is obtained for directional comparison if the same months of two years are moving in similar directions. Moreover, as another indicator, the change of monthly SPIs for two years is similar to each other with a delta threshold, then the base station also labeled as having seasonal characteristics.

Algorithm 2 *seasonal characteristics controls*

```
procedure exec_sea_char_control (m_series, delta_thr, spi_sim_thr,
jacc_thr)
   set num_similars ← 0, num_match_dir ← 0, num_match_pos ← 0
   set has_sea_pattern ← false
   set model ← init STL with yearly_sea
   set trend_m, sea_m ← call model.fit with m_series get decompose_res
   set spi_m[len(m_series)] ← null
   for each component tre_of_month, sea_of_month in trend_m, sea_m
      add spi_m ← sea_of_month / tre_of_month
   end for
   set spi_y1[ ], spi_y2[ ] ← split spi_m [ ] yearly
   for each index ix in spi_y1, spi_y2
      set spi_m_y1 ← spi_y1[ix], spi_m_y2 ← spi_y2[ix]
      set spi_m_y1_prev ← spi_y1[ix - 1], spi_m_y2_prev ← spi_y2[ix - 1]
      if Abs(spi_m_y1 - spi_m_y2) <= delta_thr then
         increment num_similars
      end if
      if (spi_m_y1_prev - spi_m_y1) * (spi_m_y2_prev - spi_m_y2) > 0
then
         increment num_match_dir
      end if
      if spi_m_y1 * spi_m_y2 > 0 then
         increment num_match_pos
      end if
   end for
   set jacc_sc_dir ← num_match_dir / 12, jacc_sc_pos ← num_match_pos
/ 12
   if num_similars >= spi_sim_thr and jacc_sc_dir >= jacc_thr
   and jacc_sc_pos >= jacc_thr then
      has_sea_pattern ← true
   return has_sea_pattern
end procedure
```

Monthly changes are compared among years and their similarity is calculated in terms of SPI change direction (upward, downward) and position (below trend or above trend). Both matches are compared with Jaccard distance. If the monthly decomposed pattern of a base station shows similarity above 25% ($spi\_sim\_thr = 0.25$) and less than 50% ($jacc\_thr = 0.5$) Jaccard distance, then it is accepted in our study a potential seasonal pattern regardless from changepoint. If the changepoint indicator is also positive, then a flexible trend covering the changepoint inside is calculated and subtracted from the main trend before STL decomposition. Hence, the impact of trend shift is not added to the seasonal component.

3.1.2. Decomposition and Seasonal Impact Calculations

As aforementioned above, decomposition of monthly and bi-weekly series are done with STL decomposition. The seasonal power index is calculated per month with the formula given by (18). But to calculate a common seasonal impact score another formula is used given by (14). This formula consists of seasonal maximum and minimum values over corresponding trend values at the same moments. It is also shown in Figure 7.

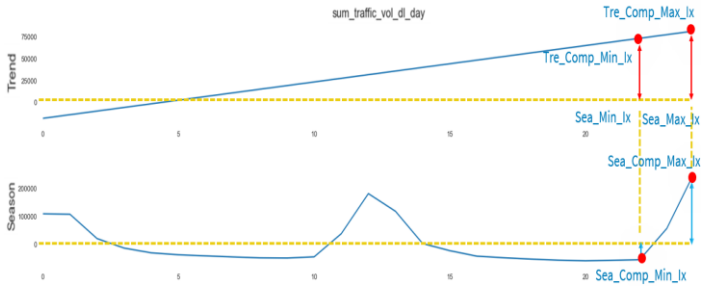

Figure 7. Base station seasonality power index parameters

Again, Let P is seasonal period and $X = \{X_{S1}, X_{S2}, \ldots, X_{SN} \mid N \leq P\}$ is the seasonal component index and $F_S(X)$, $F_T(X)$ are the decomposed functions of seasonality and trend components, respectively, then we calculate the total strength of seasonal impact for a base station SPI given by (19).

$$\begin{aligned} \text{Sea\_Max\_Ix} &= \text{ArgMax}(F_S(X)), \\ \text{Sea\_Min\_Ix} &= \text{ArgMin}(F_S(X)), \\ \text{Tre\_Comp\_Max\_Ix} &= F_T(\text{Sea\_Max\_Ix})), \\ \text{Tre\_Comp\_Min\_Ix} &= F_T(\text{Sea\_Min\_Ix})), \\ \text{Sea\_Comp\_Max\_Ix} &= F_S(\text{Sea\_Max\_Ix})), \\ \text{Sea\_Comp\_Min\_Ix} &= F_S(\text{Sea\_Min\_Ix})), \end{aligned}$$

$$SPI = \frac{Sea\_Comp\_Max\_Ix + |Sea\_Comp\_Min\_Ix|}{Tre\_Comp\_Max\_Ix + Tre\_Comp\_Min\_Ix} \quad (19)$$

4. Experimental Results and Discussion

In this section, we analyze specific use-cases for different characteristics of base station traffic data and hard to handle situations where our hybrid algorithm takes over shortcomings of pure Prophet library and STL-based approaches. In the end, we provide the accuracy results of the algorithm for the test set prepare by SMEs and retrieve outputs for the whole network.

4.1. Specific Cases to Handle

We can simply categorize base stations into two main categories as seasonal and non-seasonal stations in respect to the long-term seasonality component of traffic volume. But there are sub-categorized cases when it comes to extracting seasonal components in an automated fashion. For automated decomposition scenarios, it is not feasible to make an explanatory analysis for each base station with empirical methodologies since a typical Tier-1 network has more than a hundred thousand base stations. So, we explain some of these cases in this section over real network data and visualize the shortcomings of existing approaches of the two libraries underneath.

4.1.1. Aggressive Multiplicative Seasonality

The very first and common case to see is the aggressive multiplicative level for some base stations. Seasonal impact value changes as a function of the trend for this type of base station as seen in Figure 8. The main trend is in red just to supply an idea about linear trend growth. Mainly, the base station traffic trend needs to be saturated based on SME evaluations, so the final decomposition of trend is set to logistic growth by our algorithm taken from STL decomposition rather than Prophet library's logistics growth. In this sample base station traffic makes peaks around April, May, June periods each year, and the aggressiveness of the seasonality increases as a function of a trend which is also increasing as a function of time to a saturating point due to available resource blocks of the base station.

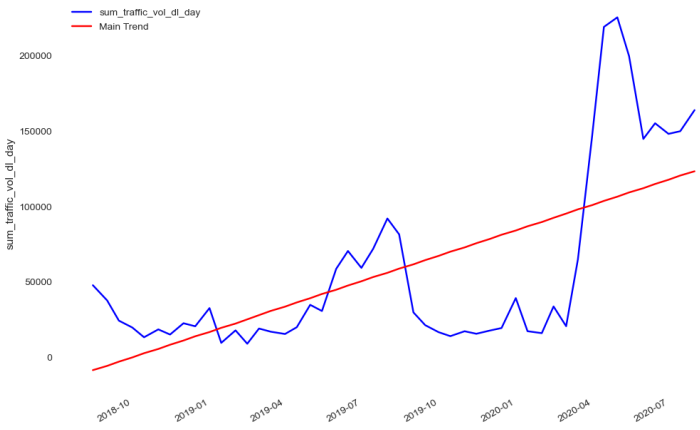

Figure 8. Base station sample with multiplicative seasonality

To be able to detect changepoints, a Laplace prior with higher flexibility causes this aggressive multiplicative seasonal boosting to be considered as strong changepoints. The distribution of change ratios by time for $Laplace(0, 1.5)$ can be seen in Figure 9 and Figure 10 for the multiplicative yearly seasonality model constructed on the Prophet library.

Keeping changepoint prior too flexible unnecessarily to grab any potential changepoints causes seasonal movements to be marked as potential changepoints and this results from unexpected trend changes to keep yearly multiplicative seasonality fitting. Here the seasonal component is an output of Fourier series fitting. Specific to this sample, for a better Prophet library fit for the multiplicative yearly seasonality model, a less flexible prior is required like the default value $Laplace(0, 0.05)$. Its decomposed components are visualized in Figure 11. So, adjustment of changepoint priors plays a critical role in the correct mapping of traffic amount between trend and seasonal components. A single changepoint in Figure 11 is located in a close location to the middle of the overall series and indicates the

slope change on-trend fits better when looked into decomposed components. But keeping lower or default priors tends to discard trend changes and makes the model less robust against real changepoints as analyzed in Section 4.1.2. An automated solution should be capable to distinguish multiplicative seasonality movements including aggressive ones from the level shifts caused by some other reasons automatically.

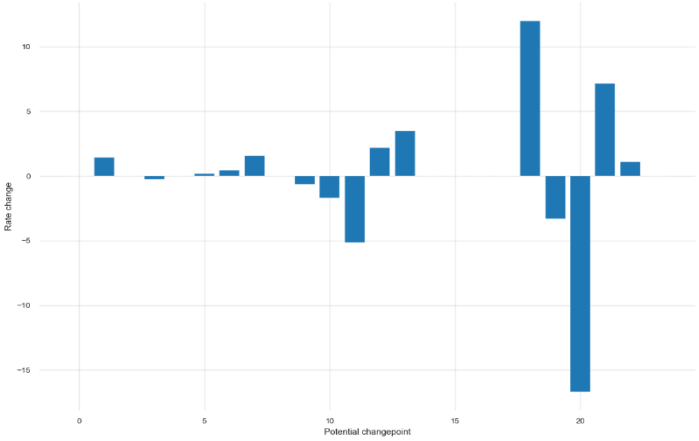

Figure 9. Ratio change distribution of flexible prior setup on multiplicative seasonal behavior

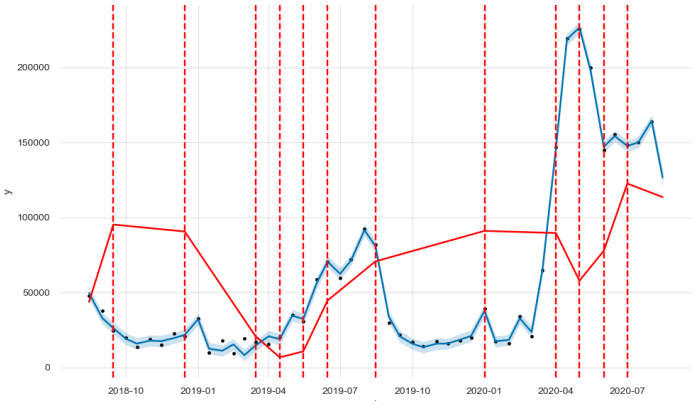

Figure 10. Changepoint locations of flexible prior setup on multiplicative seasonal behavior

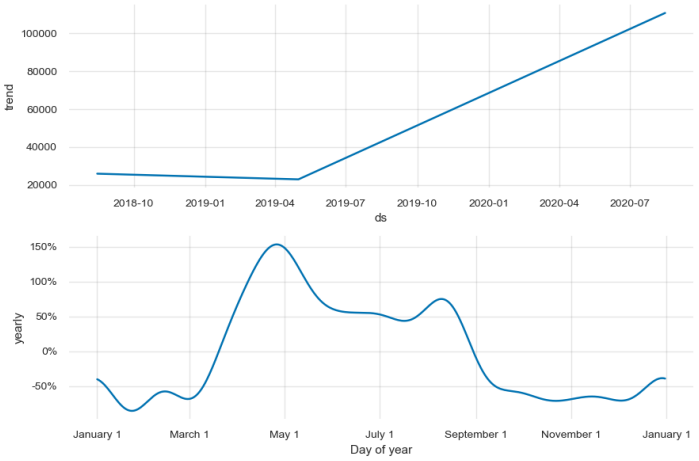

Figure 11. Components of less flexible prior setup on multiplicative seasonal behavior

Log-transformation of the original series decreases the distance between two years in terms of Jaccard distance for multiplicative seasonality base stations. This makes it straightforward to compare movement similarities among two years. As explained in Section 3.3, once log-transformation is applied to series and it is fitted with flexible before retrieving all possible change ratios, we get the distribution of ratios, visualized in Figure 12. The distribution of change ratios between two years is more similar in terms of amount and positions. It means lower Jaccard distances.

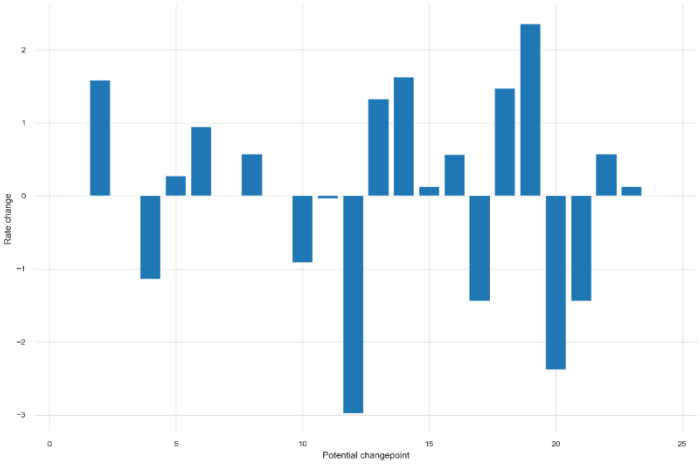

Figure 12. Components of less flexible prior setup on log-transformed traffic data of multiplicative seasonality

Algorithm 1 for changepoint controls explained in Section 3.1.1 returns false for this multiplicative seasonality base station since there is no significant difference between ratio changes of subsequent years. Both years have five significant ratio changes at similar amounts which are 2.7 and 2.3, respectively. Also, the similarity scores for monthly calculated SPI values are high. Jaccard distances which are explained with Algorithm 2 in Section 3.1.2 are calculated as 38%, 12%, and 38% for SPI change direction, change position, and amount for monthly SPI among two years. Hence, the subsequent controls for changepoints existence and change similarities based on Jaccard distance tell us, there exists seasonality but no changepoint for the specified base station. If changepoints exist, the algorithm would try to set a flexible trend curve to cover a level shift and subtracts it from the original series for correct seasonal decomposition. The specified base station is decomposed to saturating trend and seasonal components as the output of the STL decomposition part of the hybrid algorithm in Figure 14. The very recent values of components are considered which is 2020 in our case.

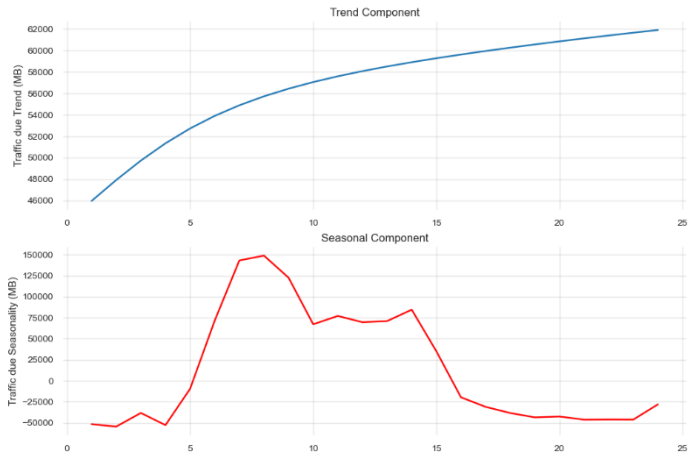

Figure 13. Components of our algorithm for multiplicative seasonality with saturating trend component

4.1.2. Level Shift Changepoints in History

Mobile networks are growing too fast, especially for the hotspot areas. So, newly deployed base stations or turned-off ones can cause dramatic trend shifts in neighbor base stations as expected by SMEs. There may be some other reasons like configuration changes of cells such as uptilt or downtilt actions for coverage optimization. So, when designing automated seasonal impact analysis, those types of level shifts should be considered and detected autonomously and distinguished from seasonal movements of traffic volume. A sample base station with the shifted level in terms of trend is visualized in Figure 14. Piece-wise Bayesian curve fitting with low changepoint prior (i.e., $Laplace(0, 0.05)$) could not be adequate to catch this changepoint as visualized in Figure 15. History of the specified base station includes strong level shift after July 2019. But since changepoint prior is set to a lower value, which means less flexible prior setup, piece-wise mapping of curve catches only a few changepoints. So changepoints are placed incorrectly around May and this caused to map traffic change into the seasonal component.

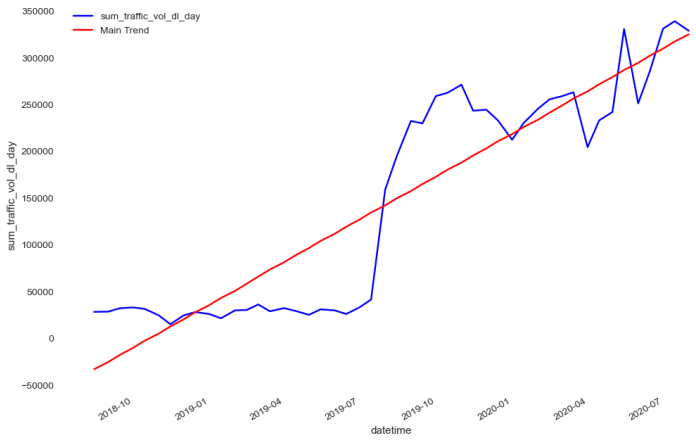

Figure 14. Base station with changepoint due to level shift

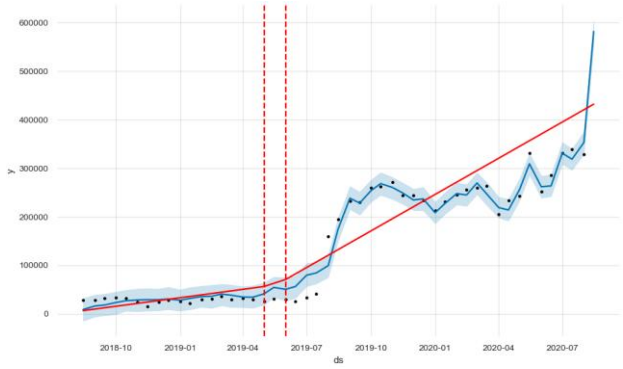

Figure 15. Changepoint locations of less flexible prior setup on multiplicative seasonal behavior

Similar decomposition output is also observed for STL based decomposition of series even if the series is log-transformed. As seen in Figure 16, when two years history is decomposed with STL (seasonal degree = 1, trend degree = 0), an abnormal change in traffic has explained with the incorrect seasonal component. It is not just STL that explains this change with the seasonal component so that in case of incorrect changepoint prior selection, Prophet library's seasonality component is also affected highly. So, it is not straightforward to have the correct changepoint prior when compared with Section 4.1.1 where a less flexible prior worked fine unlike in this case. Same prior failed as seen in Figure 15. For this special case, higher flexibility (i.e., $Laplace(0, 1.5)$) to identify changepoint location performs much better and maps traffic into trend component relatively more reasonable format as shown in Figure 17.

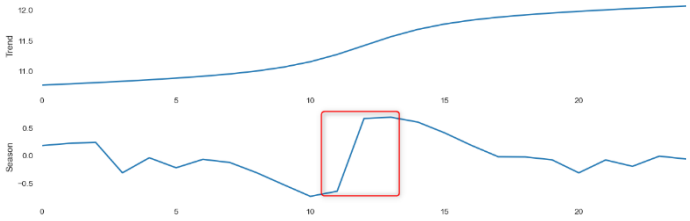

Figure 16. Misleading of STL decomposition with mapping of trend shift into seasonal component

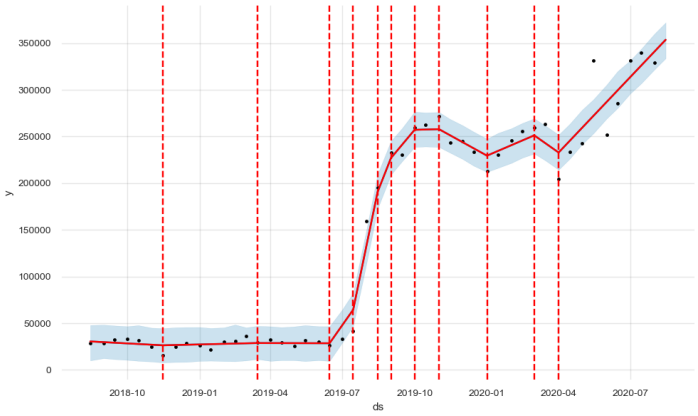

Figure 17. Changepoint locations of higher flexible prior setup on changepoint behavior

Our hybrid solution checks for the potential location of the changepoint and considers the correct section of history for decomposition. So, it takes the behavior of the latest year after changepoint as seen in Figure 18. Relatively smoother trend component with the levels of the year 2020 and seasonal component over it isis distilled successfully. Sure, the related base station has some repetitive 'up's and 'down's both in the years 2019 and 2020 but it is but the relative amount of these seasonal movements are weak next to the trend component, so base station labeled as non-seasonal characteristics. Because we label base stations as 'seasonal' if they have any seasonal period over trend component by 30% difference as explained in Section 3.

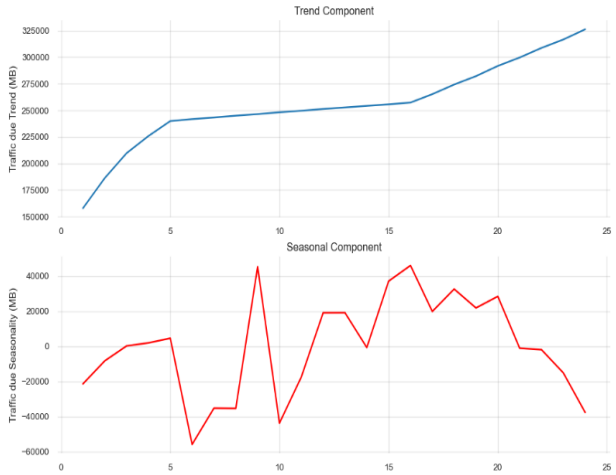

Figure 18. Components of our algorithm for changepoint due level shift

4.1.3. Trend Increase Changepoints in History

For some base stations, there is growing demand due to population increase, customer behavior changes, or some other reasons. The algorithm is required to catch these trend changes and adjust seasonal impact factors concerning these trend changes. As discussed in section 2, changepoints are not just because of level shifts. Non-stationary behavior of trend component with increasing or decreasing trend is also addressed as changepoints in literature. A sample trend change as exponential growth is visualized in Figure 19 where the trend is increasing with the new year.

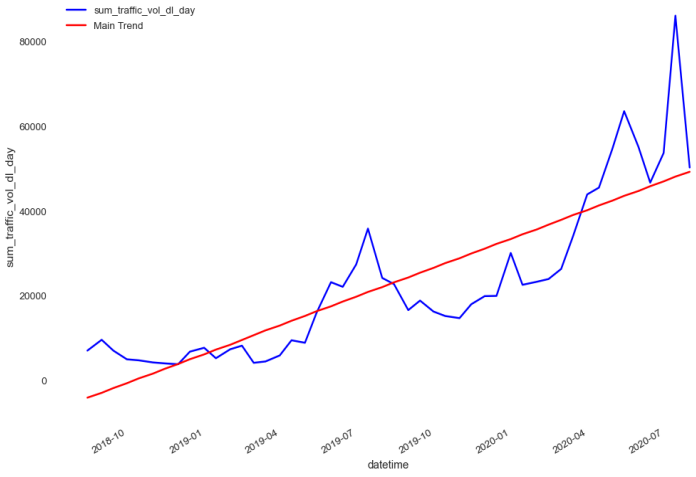

Figure 19. Base station with changepoint due to level shift

The algorithm detects a changepoint and tries to fit a piece-wise curve with a flexible changepoint prior. This curve is subtracted from the original traffic as shown in Figure 20. The remainder series is decomposed with STL to retrieve the final components of the algorithm in Figure 21. The seasonal impact for May, June, and July periods is higher and the month of July is the peak period for monthly data. So, any investment before month May is proposed for capacity planning suggestions.

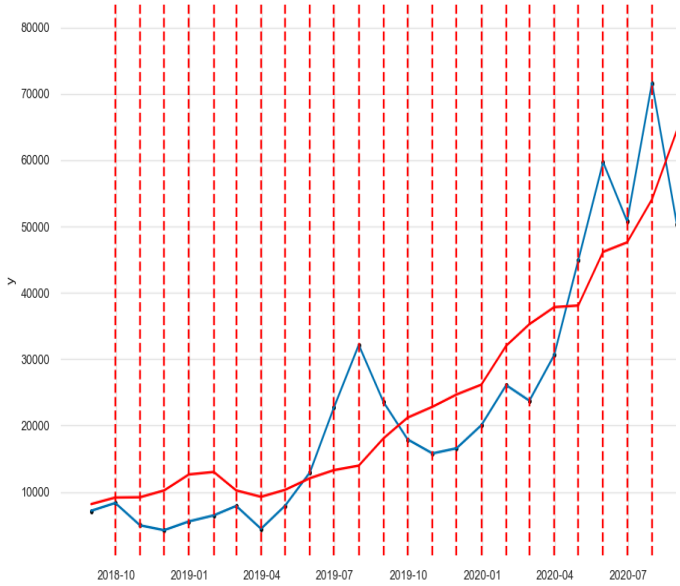

Figure 20. Changepoint locations of higher flexible prior with piece-wise fitting

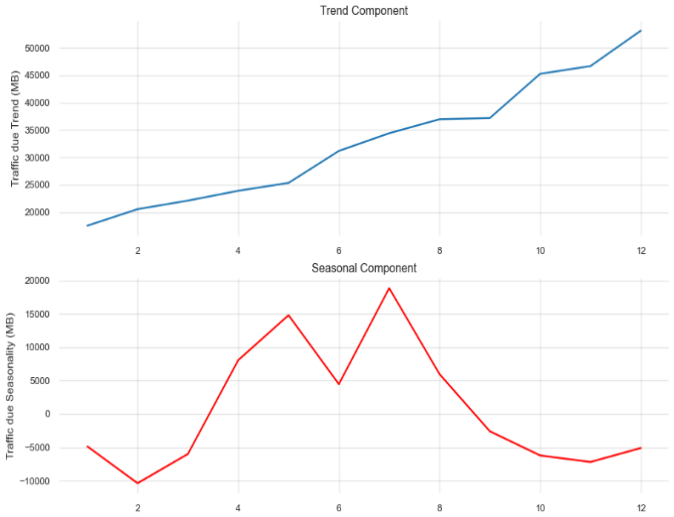

Figure 21. Components of our algorithm for a base station with changepoint due to the increasing trend

4.1.4. Weak Seasonality or Non-Seasonality

Each base station doesn't need to show strong seasonal characteristics. Some base stations do not have non-stationary behavior due to seasonality, so no seasonal pattern exists. Our algorithm is required to be aware of the seasonality over trend ratio for labeling the base station. Some base stations have a seasonal component that is similar to random walk or some others do not have strong peaks at seasons. We label those base stations as non-seasonal and do not consider their seasonal periods when evaluating future investment to overcome seasonal peak issues. For those regions, SMEs just consider changes in on-trend components. A sample non-seasonal, strong trend base station can be viewed in Figure 22.

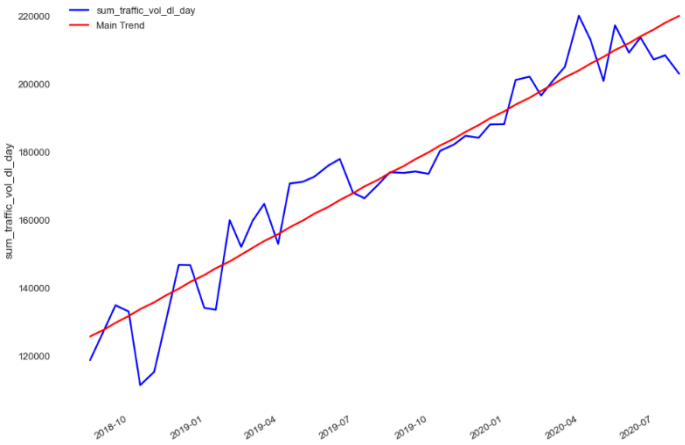

Figure 22. Base station with non-seasonal characteristics

The traffic characteristics in this category of base stations are defined with their trend and some residuals which cannot be explained with some seasonal patterns. If there exist some seasonal patterns, their impact over trend strength is found weak. Even if the visual output of the final algorithm has a seasonal component that is too low comparing to the trend component including residuals inside. The peak point is around 15 GB for the seasonal component comparing to 150 GB for the trend component. It is less than 10%. Also, since Jaccard distance measuring similarity and correlated movements are too high, this base station is labeled as non-seasonal by our algorithm as expected.

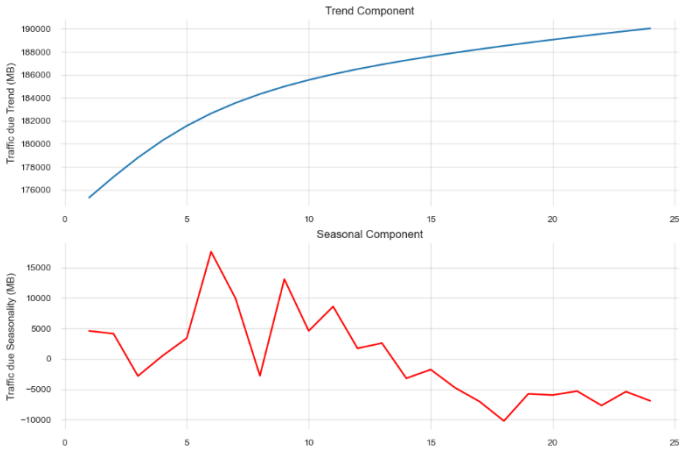

Figure 23. Components of our algorithm for a base station with non-seasonal characteristics

4.2. Accuracy Results and Discussions

In our study, we have 150 base stations labeled by SMEs and we use 100 of this set for accuracy testing. Another remaining batch of 50 base stations is used for exploratory analysis, parameters tunning for thresholds, and initial model tests. There are four main outputs of the algorithm such as,

- *Seasonal labeling:* category of the base station as seasonal or non-seasonal, in respect to seasonal pattern and its impact level. True if the base station is seasonal.

- *Power of seasonal impact:* SPI based relative impact score regarding the ratio of seasonal traffic amount to trend traffic amount

- *Existence of changepoint:* it is a value which is true if historical offline data of base station includes changepoints.

- *Duration of the season:* time in days during seasonal impact is over 30% continuously.

Our algorithm works on all test base stations and provides these four outputs. Two of them, seasonal labeling and the existence of changepoint are binary outputs. But the other two, power of seasonal impact and duration of the season are regressed values. To retrieve an overall accuracy score, we have turned these numeric outputs into categorical outputs by defining an acceptance criterion with SMEs' view. If the base station is correctly labeled for seasonality and changepoint status, it is then checked for seasonal impact amount and duration of the season as well in terms of Absolute Percentage Error (APE) as formulated given by (20). SPI is calculated above by considering the peak month of seasonal impact. There is also an actual value that is defined by SMEs per base station. The acceptable APE value for these two metrics is below 15%. Hence, it is satisfactory for planning engineers at network capacity investment decisions on time. In the case of retrieving 15% below the APE score for a base station in terms of SPI and duration, then both predictions are marked as True.

$$APE = \frac{|Calculated\ Value - Actual\ Value|}{Actual\ Value} \quad (20)$$

Each base station in test set is categorized by the algorithm, and they are labeled regarding to Table 2 truth table. Since we turn the problem into binary categorization, we get an overall accuracy metric with F1 score as formulated in Table 3. Not to miss on time investment decisions and cause decreased customer experience, how many real positives are detected by our algorithm is critical. This corresponds to 'recall' value. Moreover, not to cause unnecessary investment decisions which leads increased CAPEX, how many of our positive decisions is really positive is also important for us. This one corresponds to 'precision' value in Table 3. In order to have a balance between these two metrics, we evaluate our general success with F1 score which is harmonic mean of precision and recall. We give equal weight to both scores, so keep beta as 1.0 which is default value for F-measure.

Table 2. Explanation of truth

TP	True-Positive means from all four dimensions, the result is correct and base station is seasonal with correct decomposition.
TN	True-Negative means from all four dimensions, the result is correct and base station is non-seasonal.
FN	False-Negative means labeling an actual seasonal base station as non-seasonal.
FP	False-Positive means labeling an actual non-seasonal base station as seasonal.

Table 3. Metric formulas

Accuracy	(TP + TN) / (TP + FP + FN + TN)
Recall	TP / (TP + FN)
Precision	TP / (TP + FP)
F1 Score	2 x Precision x Recall / (Precision + Recall)

Distribution of SPI and seasonal period length in our test set is visualized in Figures 24 and 25, respectively. As aforementioned in Section 3.1, our threshold for seasonal labeling is 30% as the initial criterion provided by SMEs. Hence, distribution has a higher frequency below 50 percent in SPI distribution. Similarly shorter seasonal periods have more frequency below 30 days.

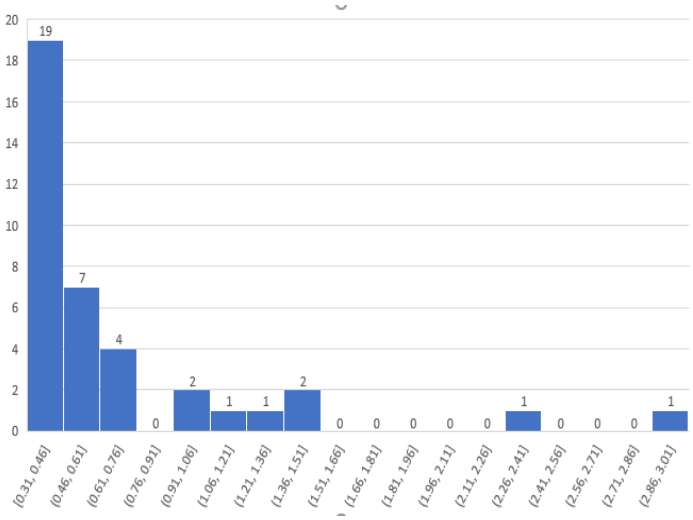

Figure 24. Distribution of test set SPI values output

The total number of labeling results is shown in Table 4 as a confusion matrix. Algorithm label 45 (True Positive) actual seasonal base stations correctly providing satisfactory period length calculation and seasonal impact values decomposition. The number of incorrect labeling or not-satisfactory decomposition is 8 base stations (False Negative) of them.

In these 8 FNs, 5 of them are due to incorrect decomposition of trend component slightly, whereas 3 of them are due to shorter period length calculations.

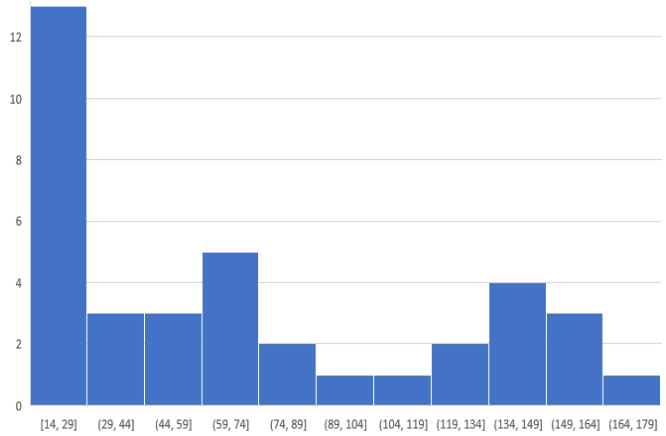

Figure 25. Distribution of test set seasonal period length

Here shorter means calculating period length more than 15% error. Our algorithm label 44 (True Negative) base stations correctly as non-seasonal. Due to some incorrect reactions to changepoints, SPI distribution similarity is not identified correctly. So, 3 of actual non-seasonal base stations are labeled as seasonal which is the number of false positives.

Table 4. Confusion matrix

	True Seasonal	True Non-Seasonal
Predicted Seasonal	45	3
Predicted Non-Seasonal	8	44

Overall accuracy results using precision, recall, and F1-score are listed in Table 5. We benchmark our algorithm with its underlying decomposition approaches as STL and Prophet library itself. We showed the improved accuracy of our algorithm on mobile network data comparing to these methods. Even our algorithm uses both these frameworks under the hood, by making some further auto-analysis over the changepoint possibility and their distribution, it understands critical changes in traffic history and selects the correct parameters and sections for improved decomposition performance. We retrieve 0.80 as an F1-score with our algorithm which is outperforming the average F1-score of STL and Prophet library approaches by 18.6%. As visualized in section 4.1, for historical data with changepoint inside, STL tends to explain these changes with a seasonal component which causes incorrect trend values for changepoint included base stations. Unlike STL, when Prophet library is executed with default parameters as automated changepoint detection capability it performs well but fails to cover drastic trend shifts and does not fit the trend component well enough. Prophet library is more successful at explaining multiplicative seasonality. Both solutions do not just consider afterward of changepoint for the final decision which is also another expectation from SMEs. They remain under the impact of series characteristics before changepoint time and preserving it as long-term memory. Base stations with these characteristics as explained in Section 4.1 are processed better with our algorithm, hence we get improved precision and recall scores in benchmarking as seen in Table 5.

Table 5. Accuracy results and benchmark

	Our Algorithm	STL	Prophet library
Precision	0.93	0.72	0.77
Recall	0.84	0.73	0.79
F1-Score	0.89	0.72	0.78

When we apply the final algorithm to the whole data set to categorize base stations we get the distribution in Figure 26. Total 31% of whole base stations are seasonal and 20% of base stations have changepoints in their history regarding our algorithm. The distribution of SPI values and seasonal period length is depicted in Figure 27 and Figure 28, respectively.

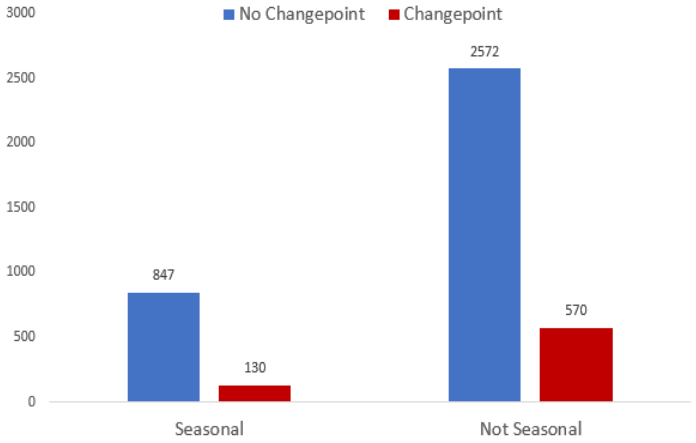

Figure 26. Distribution of labeled base stations of the whole dataset

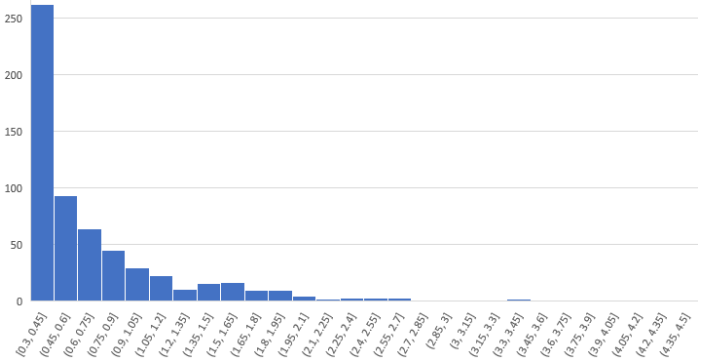

Figure 27. Distribution of whole dataset SPI values output

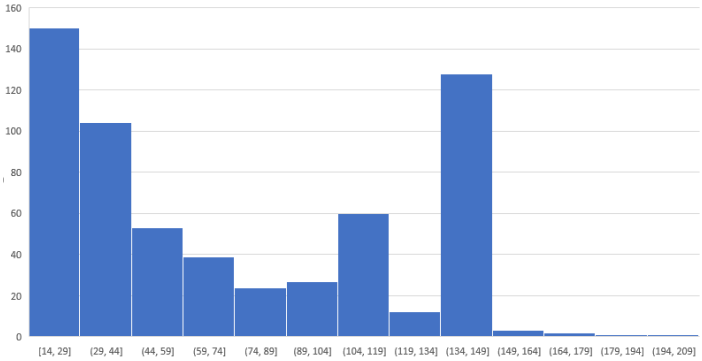

Figure 28. Distribution of whole dataset seasonal period length

5. Conclusions

In this study, an algorithm for seasonality detection and decomposition of mobile network base station traffic data is presented to extract seasonality impact on base stations by considering changepoint existence due to level shift or exponential trend increase. Automated capacity planning for

mobile networks requires long-term forecasting of traffic demand by using historical patterns. To decide the correct time of investment and correct capacity expansion size or improving the accuracy of forecasting algorithms with exogenous features, both seasonal decomposition and seasonal period identification improve decision accuracy. We designed a hybrid algorithm to calculate these features on live network data with improved accuracy which uses piecewise STL decomposition and Prophet library's regression with Laplace prior under the hood. Combining both methods with an awareness of their weak and strong parts and leveraging overall output with changepoint and similarity analysis help us to improve our accuracy around 18.6% comparing the average of single usage of these methods. We also provided some special cases that increase problem complexity and decrease decomposition accuracy. We visualized outputs of each method over struggling points on live network data. We completed all required preprocessing, data cleansing, and anomaly healing operations beforehand. Our analysis approaches traffic decomposition requirements for different purposes in a univariate, offline, parametric, and supervised way. For further research area, another solution which is unsupervised, online with limited history, non-parametric, multi-variate (ie. base station capacity limitations due to resource block unavailability or several connected UE counts) can be studied. Decomposing the seasonal impact of newly deployed base stations with low-quality history is also another further research area based on our discussions with subject matter experts.

6. Acknowledgement

A part of this work has been conducted under the frame of the Celtic-Next AI4Green project where efficient and risk-aware energy saving algorithms are studied in collaboration. Calculating high season start and end dates next to the multiplicative impact of seasons is significant for risk minimization of RAN energy saving algorithms since contemporary solutions rely on short-term predictions over limited history.

References

Aminikhanghahi, S. & Cook, D. J. (2017). A Survey of Methods for Time Series Change Point Detection. Knowledge and Information Systems, 51(2), 339–367.

Balke, N. S. (1993). Detecting Level Shifts in Time Series. Journal of Business & Economic Statistics, 11, 81–92

Basseville, M. & Nikiforov, I. (1993). Detection of Abrupt Change Theory and Application, Prentice-Hall, ISBN: 0-13-126780-9.

Burg, G. J. J. & Williams, C. K. I. (2020). An Evaluation of Change Point Detection Algorithms.

Chen, H. & Zhang, N. (2015). Graph-based change-point detection. Annals of Statistics, 43(1), 139–176.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. J. (1990). STL: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics, 6(1), 3–33.

Cortez, P., Rio, M., Rocha, M. & Sousa, P. (2006). Internet traffic forecasting using neural networks. Proceedings of IEEE International Conference on Neural Networks, 2635–2642.

Dagum, E. B., & Bianconcini, S. (2016). Seasonal adjustment methods and real time trend-cycle estimation in Statistics for Social and Behavioral Sciences. Springer.

Downey, A. B. (2008). A novel changepoint detection algorithm.

Erp S.V., Oberski D.L., Mulder J. (2019). Shrinkage priors for Bayesian penalized regression. Journal of Mathematical Psychology, 89, 31–50.

Gould, P. G, Koehler, A. B., Ord, J.K., Snyder, R. D., Hyndman R.J. & Vahid-Araghi, F. (2008). Forecasting time series with multiple seasonal patterns, European Journal of Operational Research, 191, 207–222.

GSMA Intelligence. (2020). Mobile Economy Research Report. Available: https://www.gsma.com/mobileeconomy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf, Accessed on: Mar. 21, 2021.

Harvey, A. C. & Shephard, N. (1993), Structural time series models in Handbook of Statistics, Elsevier.

Hyndman, R. J. & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, 27(3), 22.

Jaccard, P. (1912). The Distribution of the Flora in the Alphine Zone. New Phytologist, 11(2), 37–50.

Killick, R. & Eckley, I. A. (2014). changepoint: An R Package for Changepoint Analysis. Journal of Statistical Software, 58(3).

Lakshmanan, A. & Das, S. (2017). Two-stage models for forecasting time series with multiple seasonality.

Livera, A. M., Hyndman, R. J. & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association, 106(496), 1513–1527

Nikravesh, A. Y., Ajila, S. A., Lung, C.-H. & Ding, W. (2016). An Experimental Investigation of Mobile Network Traffic Prediction Accuracy. Services Transactions on Big Data, 3(1), 1–16.

Page, E. S. (1954). Continuous inspection schemes. Biometrika 41.1/2, 100–115.

Rosner, B. (1975). On the detection of many outliers. Technometrics, 17(2), 221–227.

Sciancalepore, V., Samdanis, K., Costa-Perez, X., Bega, D., Gramaglia, M. & Banchs, A. (2017). Mobile traffic forecasting for maximizing 5G network slicing resource utilization, Proceedings of IEEE International Conference on Computer Communications.

Scott, A.J. & Knott, M. (1974). A Cluster Analysis Method for Grouping Means in the Analysis of Variance. Biometrics, 30(3), 507–512.

Seabold, Skipper & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. Proceedings of the 9th Python in Science Conference.

Stan Development Team. (2020). Stan Modeling Language Users Guide and Reference Manual, 2.19.1. https://mc-stan.org

Taylor, S. J. & Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37–45.

Theodosiou M. (2011). Disaggregation & aggregation of time series components: A hybrid forecasting approach using generalized regression neural networks and the theta method. Neurocomputing, 74(6), 896–905.

Tikunov, D. & Nishimura, T. (2007). Traffic prediction for mobile network using Holt-Winter's exponential smoothing, Proceedingds of International Conference on Software, Telecommunications and Computer Networks, 310–314.

Yu, Y., Wang, J., Song, M. & Song, J. (2010). Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient. Proceedings of 2010 International Conference on Intelligent System Design and Engineering Application, 1(1), 980–983.