Leveraging Machine Learning for Network Traffic Forecasting

A Technical Paper prepared for SCTE by

Diane Prisca Onguetou, PhD
Independent Consultant
onguetou@umich.edu

Achintha Maddumabandara
Co-Author
Principal Data Analyst / Rogers Communications
2400 32 Ave NE, T2E 6T4, Calgary AB, Canada
achintha.maddumabandara@sjrb.ca

Jeffrey Lee
Co-Author
Manager, Data Acquisition and Reporting / Rogers Communications
2400 32 Ave NE, T2E 6T4, Calgary AB, Canada
jeffrey.lee@sjrb.ca

Table of Contents

Title	Page Number
1. Introduction	4
2. Case Studies	4
2.1. The Network Graph	4
2.2. Data Collection Over Time	6
2.3. Traffic Overview and Modelling	8
2.4. The Study	9
2.4.1. Main Objective	9
2.4.2. Accuracy	10
2.4.3. ML Pipeline	10
3. CAGR Approach to Network Traffic Forecasting	10
3.1. Global CAGR	10
3.2. Limitations of Global CAGR Approach	11
3.3. A Distributed CAGR Strategy	12
4. Exploring Machine Learning Time Series Models	14
4.1. Seasonal Decomposition of Time Series (STL)	14
4.2. Forecasting Models for Network Traffic Time Series	17
4.2.1. Holt Winters Exponential Smoothing	17
4.2.2. Seasonal Auto Regressive Integrated Moving Average (SARIMA)	18
4.3. Which Model is Best for Network Traffic Forecasting?	18
5. Key Insights	20
Abbreviations	21
Bibliography & References	21

List of Figures

Title	Page Number
Figure 1 – The Network Topology	5
Figure 2 – Distribution of Connectivity at a Hubsite	6
Figure 3 – Availability of Historical Data	7
Figure 4 – Number of Hubsites and Intraconnects (based on data collection)	8
Figure 5 – Evolution of Network Connectivity (based on data collection)	8
Figure 6 – Historical Traffic Evaluation for Several Links	9
Figure 7 – Global CAGR Estimate	11
Figure 8 – Global vs. Local CAGR	12
Figure 9 – Global vs Local CAGR Predictions for internal longhaul link	13
Figure 10 – Global vs Local CAGR Predictions for international link	13
Figure 11 – Global vs Local CAGR Predictions for internal metro link	13
Figure 12 – A Naïve Look at Total Traffic Periodicity	14
Figure 13 – Seasonal Decomposition for Total Traffic (additive)	15
Figure 14 – Seasonal Decomposition for the internal long haul link (additive)	16
Figure 15 – Seasonal Decomposition for the international link (multiplicative)	16
Figure 16 – Seasonal Decomposition for the internal metro link (multiplicative)	17
Figure 17 – Comparison of ML Model Predictions for internal long haul link	19

Figure 18 – Comparison of ML Model Predictions for international link 19

List of Tables

Title **Page Number**

Table 1 – CAGR Accuracy Checking [Total Traffic] 12

1. Introduction

Network traffic modelling and forecasting play a crucial role in maintaining network performance, optimizing resource allocation, and providing seamless user experience. Perhaps the most widely used method for network traffic forecasting is Compound Annual Growth Rate (CAGR). While CAGR can provide a view of network traffic growth, it does not provide the required context for informed decision making that can differentiate services from the competition. Adhoc analysis of network traffic models or conventional approaches may not capture the complex network behavior that is observed today. Machine learning (ML) network traffic models can be a powerful aid to align network behavior with organizational goals.

CAGR has limitations such as period choice sensitivity, network traffic volatility and seasonality. Incorrect traffic predictions increase the chance of suboptimal investments in capacity, reliability, and/or security. Network burst patterns based on high-profile events such as smartphone updates, major video game releases, or live global sporting events (e.g., the 2022 FIFA World Cup) may be overlooked. CAGR accuracy can be improved by calculating each network segment. In addition, ML network models can provide the additional context required to make informed decisions.

This paper proposes using ML to enable automated iterative calculations and model attributes such as trends and seasonality, failure events, subsequent interactions between the primary and failover links, and network burst patterns. This provides the additional context that is missing from CAGR alone to make the most informed business decisions.

This study considers a portion of a real Internet backbone. It analyzes traffic patterns within four consecutive years, using insights and findings to predict monthly network traffic in the fifth year. Section 2 describes the network under consideration and the data collection process. It discusses challenges incurred in this exercise and highlights influences that deteriorate data quality and the performance of any prediction. The section ends with a traffic modeling exercise and the presentation of accuracy metrics for future model evaluation. Section 3 revisits the traditional CAGR-based approach to network traffic forecasting. It exposes the limitations of the global CAGR strategy and discusses possible improvements for that method. Section 4 explores ML alternatives to network traffic forecasting. It shows how ML models can help to uncover seasonal traffic patterns and provide better forecasting. It emphasizes the importance of choosing the forecasting model appropriate to the data's nature, the presence or absence of seasonality, the prediction horizon, and the complexity of patterns in the traffic. This section also shows models in action and compares the performance of a few ML time series forecasting models when predicting traffic for the reference Internet long-haul. Section 5 summarizes critical intakes for effective and reliable forecasting and indicates future investigations. The paper ends with a list of abbreviations and references.

2. Case Studies

2.1. The Network Graph

The new insights and approaches developed in this paper are tested on a large service provider network with millions of users and a mix of almost all types of internet traffic. The network has hundreds of links with varied capacities, from multiple 10Gs for smaller hubsites to tens of 100Gs for backbone links. Link costs are also varied as some are on the service provider's own fiber infrastructure while others are leases from other providers. Figure 1 presents a logical view of the reference network, with three highlights for future illustrations throughout the paper. Highlighted links all originate in the same city. One represents an internal metro link between two hubsites of the same city. Another link pictures the connection of two cities through the national long haul. The third highlight is typical of international links.

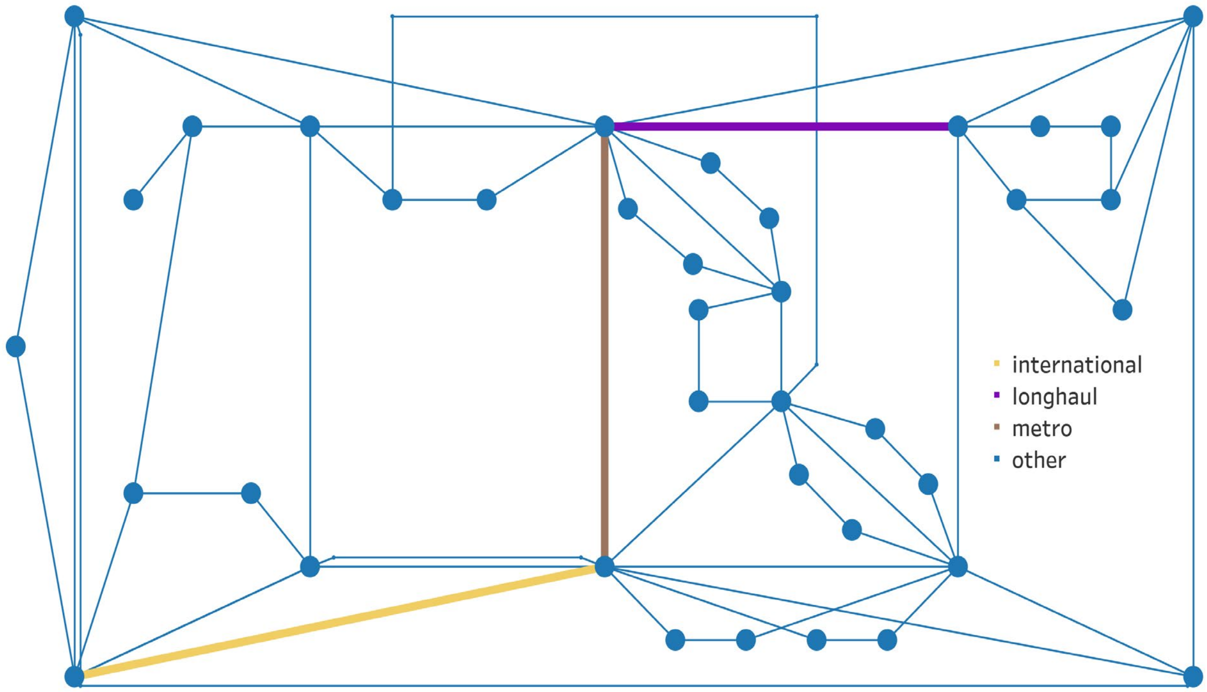


Figure 1 – The Network Topology

Regarding network connectivity, Figure 2 shows the connectivity distribution at a hubsite for the reference network. The connectivity or the degree at a hubsite is the number of links originating or terminating at the given hubsite. For example, the international hubsite degree is 7 while the national hubsite has connectivity 9 for the international link referenced in Figure 1. The histogram in Figure 2 indicates connectivity of 4, 5, or 8 for most hubsites. The graph under study is highly connected, with an average hubsite connectivity of 5.38.

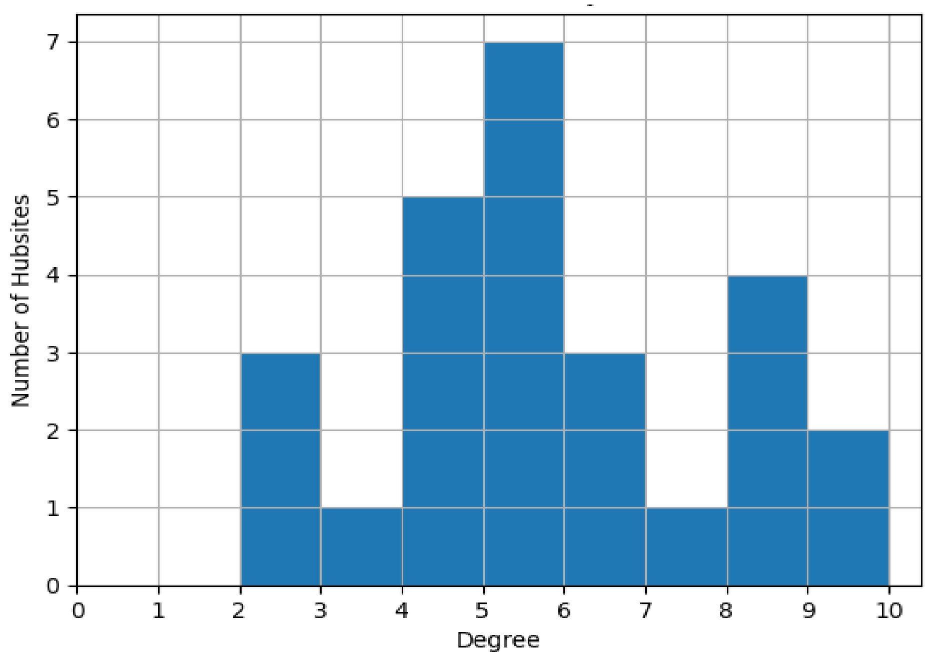


Figure 2 – Distribution of Connectivity at a Hubsite

2.2. Data Collection Over Time

Links are, in fact, bundle-intraconnects between hubsites. Data are measurements of upstream and downstream traffic in bits per second. Using the set of intraconnects from the above-described network, the heatmap in Figure 3 shows data availability as of June 30th, 2023. A dark shade indicates data records on the given date, while a lighter shade is for unavailable data.

Many links in Figure 3 show no data record for multiple reasons. Some hubsite devices are probably missing from the database due to human errors, and high volumes of historical data may have been purged to free up disk space. One data source from this study is an old legacy Network Management System (NMS) with limited storage. The NMS in question also requires manual device entry before being managed. The industry transition from NMS to powerful SDN controllers and orchestrators is recent. Another reason for missing data relates to network normal operations and changes. Network activities include the commissioning, decommissioning, and migration of routers. Changes also involve the turn up/down of new links and capacity augmentation on existing links with different identifiers, etc. Besides network device changes, technology evolves; some bundle intraconnects may have existed in another form (e.g., 1GE to 10GE to 100GE to bundle-ethernet of 100GE).

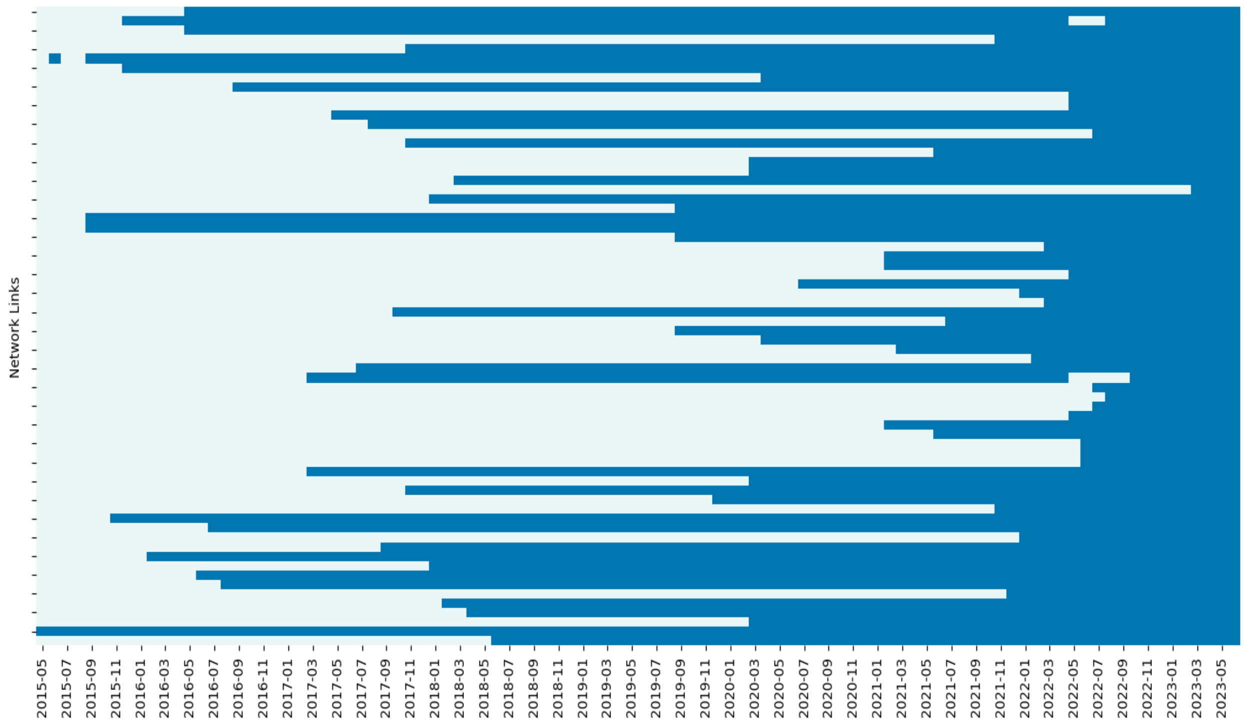


Figure 3 – Availability of Historical Data

Figure 4 and Figure 5 well illustrate legacy network tracking system inconsistency. It is highly improbable that the Internet backbone under examination had less than 30 nodes and interfaces before mid-2017. It is more realistic that either network elements were missing from the tracking system or experienced name changes over time. Considering this assumption, a reconciliation exercise might significantly improve the dataset quality. Reconciling data is arduous and time-consuming. It also requires some logic and automation to maintain the resulting dataset and future data collection clean for analysis.

Data acquisition is beyond this paper's scope. The study focuses on a subset of links with traffic traces for the previous five years as of June 30th, 2023. Those are about half of the links shown in Figure 3, including international, longhaul, and metro reference links in Figure 1, with seven years of historical data.

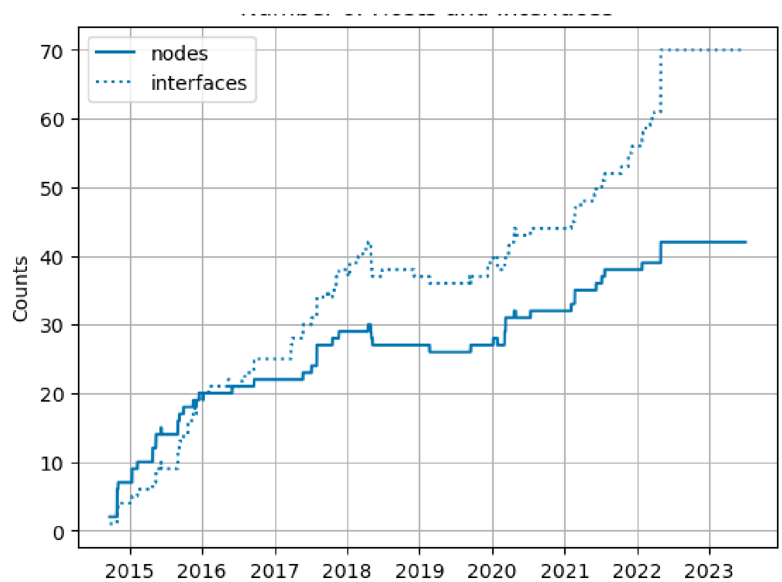


Figure 4 – Number of Hubsites and Intraconnects (based on data collection)

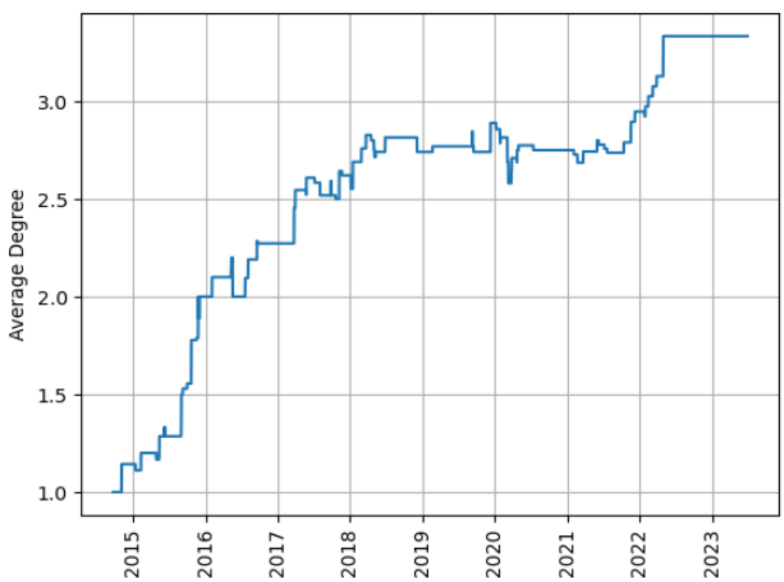


Figure 5 – Evolution of Network Connectivity (based on data collection)

2.3. Traffic Overview and Modelling

As previously stated, our study considers a subset of intraconnects with data from July 1st, 2018, to June 30th, 2023. Upstream and downstream traffic measurements are collected from each hubsite of the link. Links are bidirectional; the dataset is further simplified by approximating the link to the one intraconnect showing the highest data rate between the four traffic measurements for upstream and downstream. That maximum data rate pictures the worst-case scenario and thus is considered traffic usage over the simplified link.

On the other hand, the lowest granularity for traffic, common to all our data sources for this study, is daily aggregations. An ideal granularity is 5-minute samples, which is unrealistic to expect in 10+ years of storage-limited archives before adopting cloud computing. Regarding aggregation functions, one generation collector provided daily maximum, average, and several percentiles, while the other tool only provided maximum and average. Picking up daily maximums allowed consistency along the concatenated data set.

The daily maximums are further aggregated in monthly measurements since it is common to forecast network traffic for the medium to long term—a quarter, a year, and up to five years. In making monthly aggregates, it is ideal to consider the maximum daily 95th percentile (P95) or even take the compound P95 over all 5-minutes samples within given month. Depending on the application, this allows to control high profile and failure events causing traffic increases on (failover) links. Given the nature of the prior-described dataset for this study, reversing and taking the 95th percentile represents a decent monthly aggregate. Figure 6 shows the traffic trends for longhaul, international, and metro reference links. Note the traffic jump in mid-2020, maintained for an entire year on the three links, due to work-from-home during the COVID-19 pandemic.

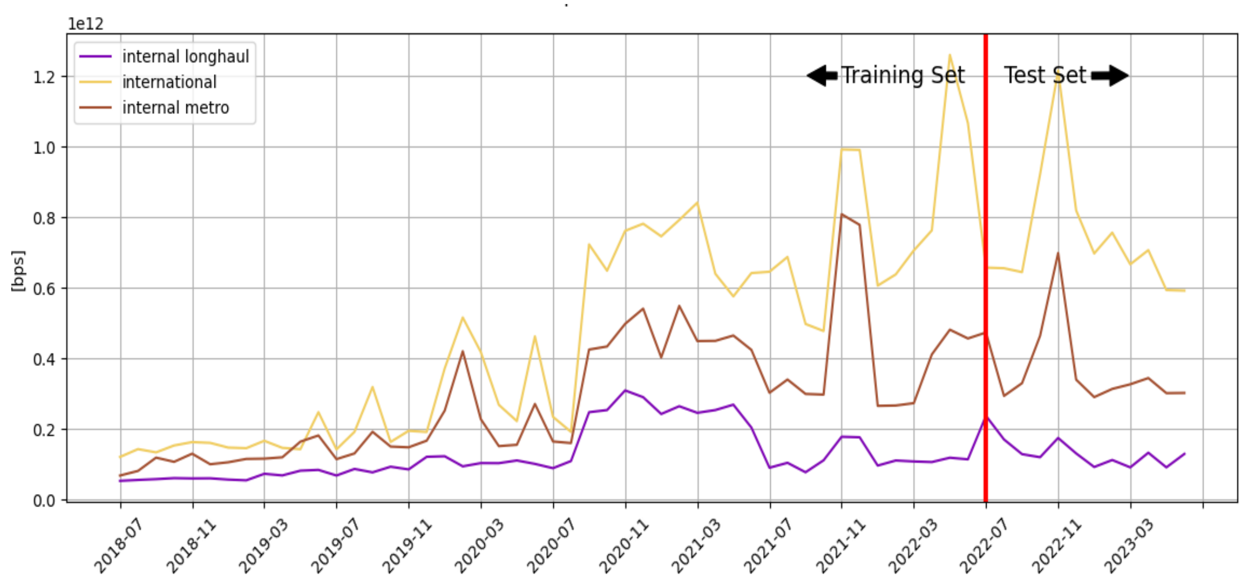


Figure 6 – Historical Traffic Evaluation for Several Links

2.4. The Study

2.4.1. *Main Objective*

Five years or 60 months of historical traffic measurements are available for the 30 selected links. The first four years or 48 months are used to train the models studying past traffic behavior and patterns. Insights gained from the observations support traffic predictions for the remaining 12 months. In other words, traffic measurements from the first four years make the training set, and the 5th year of data is the test set or comparison set. Goals include revisiting the traditional CAGR approach and discussing why ML is necessary, then building new ML-based forecasting models and conducting a comparative discussion.

2.4.2. Accuracy

Accuracy is critical in comparing traffic predictions with measurements in the 5th year. One metric is the root mean squared error (RMSE) which measures the average difference between a statistical model's predicted and actual values, as shown in the following equation. A good predictive model shows a RMSE much smaller than the mean and variance of the training set.

$$RMSE = \sqrt{\frac{\sum(actual\ value - predicted\ value)^2}{number\ of\ predictions}}$$

Although widely used to determine the accuracy of a given predictive model, it may be challenging to compare the performance of different models using RMSE.

2.4.3. ML Pipeline

All ML pipelines were implemented within Amazon Web Services (AWS) cloud computing platform. The authors used S3 for data storage, Athena for data access, and SageMaker to train and test ML models. scikit-learn, statsmodels, pmdarima, and TensorFlow Python ML modules were used for the implementation.

3. CAGR Approach to Network Traffic Forecasting

CAGR is one of the most popular forecasting methods to predict traffic in the cable industry. Total traffic across the entire network at a specific time makes a time series, a regression function is fitted to that time series, and a growth rate is derived to determine CAGR. The resulting CAGR coefficient is used anywhere in the network and whenever forecasting is needed.

3.1. Global CAGR

Simple CAGR implementations typically rely on either linear or exponential regressions. Figure 7 provides time series for the sum of traffic, i.e., blue curve, across the network—or more precisely, for the subset of 30 links under consideration. It is noticeable that fitting an exponential regression function to total traffic, in bit counts aggregates over months, is more appropriate than a linear regression line. The solid black curve corresponds to the exponential regression function that best fits the training set of data. Exponential regression equation is $y = b * a^x$, where regression coefficient a is the growth factor for the monthly periodicity under consideration. $CAGR = ((a + 1)^{12} - 1) * 100$ derives annual growth from the monthly growth rate.

Figure 7 – Global CAGR Estimate

3.2. Limitations of Global CAGR Approach

CAGR is straightforward to implement and is one of the best-known approaches to estimating compound traffic growth rates. However, there is much room for improvement when using the resulting growth factor for traffic forecasting. One drawback is the tendency to naively apply the estimated CAGR to the most recent measurement. For illustration, the solid gray line in Figure 7 shows predictions from applying the global CAGR to traffic measurement in June-2022, the most recent trace on hands. On the contrary, see how dotted gray lines adjust using the same growth factor but with the minimum and the peak measurements over the past twelve months (i.e., from July-2021 to June-2022). Traffic levels are too low or too high in seasons, and spending aligns between the regression and dotted prediction curves.

Table 1 also reveals poor accuracy for the CAGR approach. In all cases, RMSE is higher than the standard deviation although much smaller than the mean. Before applying the global CAGR to individual links, we can already see that its prediction accuracy is questionable. A good practice is tracking predictions over actual values to re-evaluate any forecasting model.

Table 1 – CAGR Accuracy Checking [Total Traffic]

Strategy for Predictions	[Mean, std]	RMSE
Exp. reg. curve	[5.06 Tbps, 2.05 Tbps]	3.95 Tbps
Most recent date		2.53 Tbps
Minimum over past 12 months		2.36 Tbps
Peak over past 12 months		5.01 Tbps

3.3. A Distributed CAGR Strategy

The other concern is considering the total-traffic CAGR as the growth factor of every link across the network. Figure 8 illustrates the problem: the x-axis corresponds to traffic peaks for individual links over the 48 months of training, and the y-axis gives CAGR in percentage. The red line is the global CAGR, as discussed in section 3.1. Scatter plots are CAGRs resulting from applying the exponential regression function to each link; 2/3rd are more than 10% above or below the global CAGR. The observation remains true in areas with the most concentration of links traffic or maximum traffic measurements, expected to influence total traffic trends greatly.

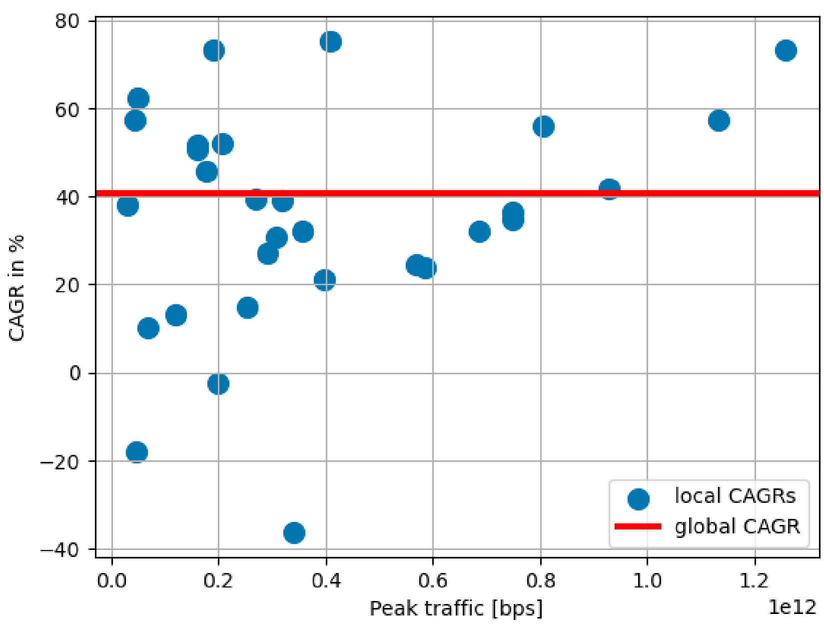


Figure 8 – Global vs. Local CAGR

Figure 9 is for the longhaul illustration, is for the international reference link, and Figure 11 is for the metro link use case. The three charts show predictions when applying the global CAGR in gray vs. the local CAGR in green to the prior-mentioned links. The black curve is for the exponential regression applied directly to the link instead of the total traffic. As before, solid curves are for cases when CAGR is applied to the most recent measurement, while dotted lines are for maximum and minimum measurements within the previous year. The actual traffic measurements match the colors for internal longhaul, international link, and internal metro in Figure 1 network topology. Comparing the green and gray lines illustrates the gap between local and global CAGR predictions. One might be closer or farther from the actual measurements, depending on how similar local and total traffic trends look alike.

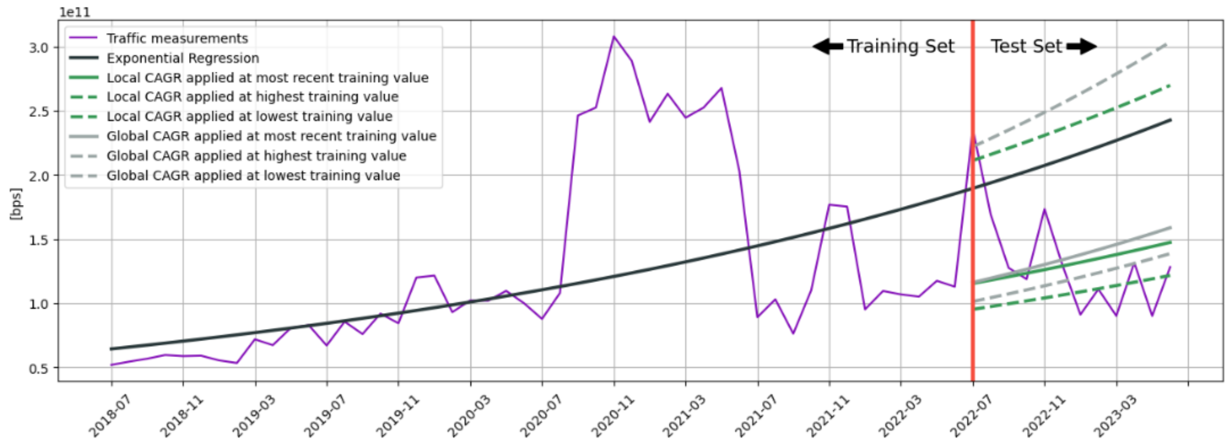


Figure 9 – Global vs Local CAGR Predictions for internal longhaul link

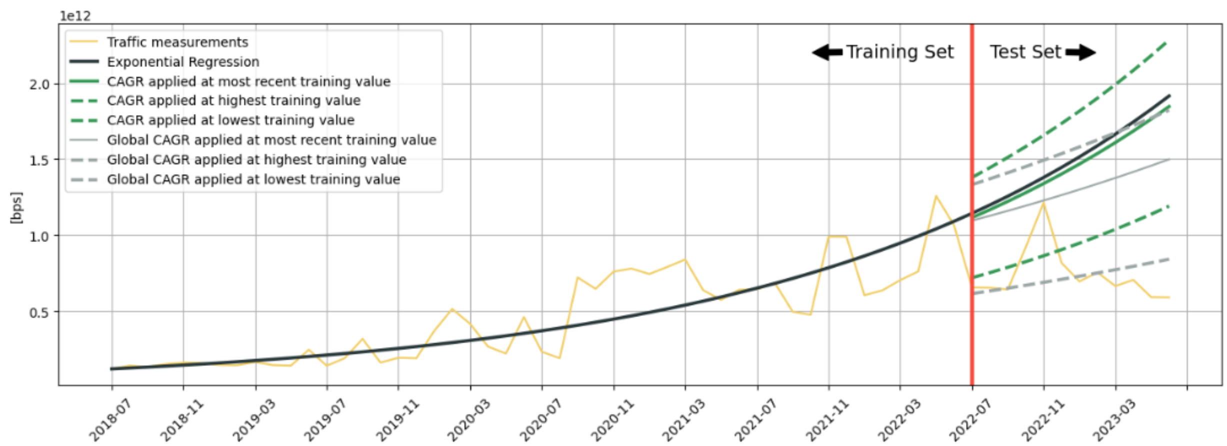


Figure 10 – Global vs Local CAGR Predictions for international link

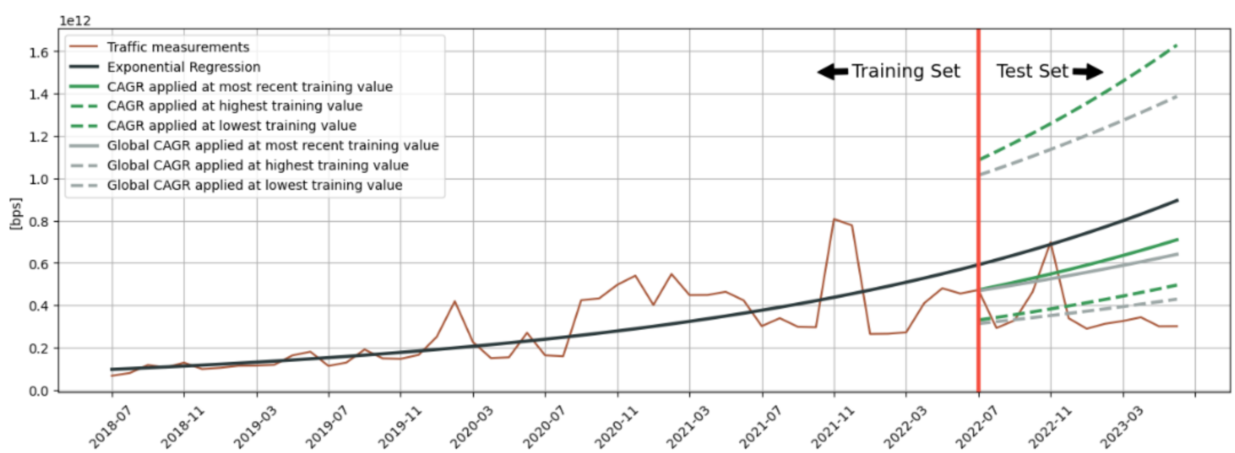


Figure 11 – Global vs Local CAGR Predictions for internal metro link

In summary, the simplicity of CAGR makes it appealing to cable operators. Trials in Section 3 show that a local CAGR, specific to its given link, performs much better than the total-traffic CAGR. The other intake is to define a logic behind the choice of reference traffic measurements to which the growth factor is applied; otherwise, extend regression curves to future times.

4. Exploring Machine Learning Time Series Models

The level of accuracy for CAGR-based predictions is still very high. The reason is that linear and exponential regressions supporting CAGR calculation focus on the trend only, while intuitions and other empiric observations suggest a certain periodicity in Internet traffic. Machine Learning offers a broad range of models, including time series models, specialized in analyzing and forecasting data collected over time, such as daily, weekly, monthly, or yearly observations.

A typical quick verification to check for periodicity in network traffic is to plot total traffic trends over months or weeks of a year for various years, over days of the months for multiple months, or even days of the week for different weeks. Figure 12 illustrates this principle for the network use case. Comparing yearly trends over months confirms slight seasonality in the data; November shows higher activities than other months, but nothing is clearly defined.

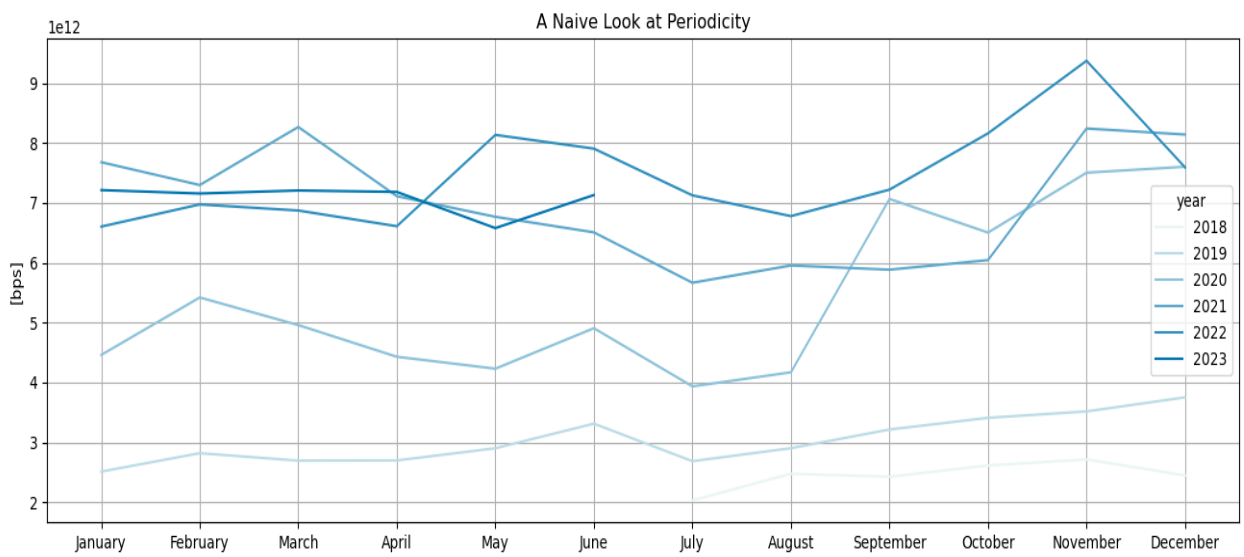


Figure 12 – A Naïve Look at Total Traffic Periodicity

4.1. Seasonal Decomposition of Time Series (STL)

Seasonal Decomposition Time Series (STL) is an ML technique that uses filters to handle seasonality and trend variations in a time series. STL separates trend, seasonal, and residual components from time series data. The trend component gives the overall movement over time, the long-term direction or pattern in the data. The seasonal component is the repeating and predictable patterns at regular intervals, such as daily, monthly, or yearly cycles. The residual component is random fluctuations or noise after removing the trend and seasonal components.

Figure 13, Figure 14, Figure 15 and Figure 16 show decomposed time series applying STL to total traffic and traffic traces from long haul, international, and internal metro sample links. There are two STL options: additive vs. multiplicative; Figure 13 and Figure 14 result from the additive model, while the other two come from the multiplicative version. In the additive case, adding trend, seasonal and residual components produce the actual observation; in the multiplicative model, the multiplication of the components produces the actual observation.

STL allows deep insights into underlying patterns and variations within the data. There is a clear yearly cycle in all illustrations; the low season is from July to August, and traffic starts ramping up in September, when students return to school, to reach its peak in November, maybe due to extensive online shopping around Back Friday. There are also local maximums in May and June that might be due to graduation or final games for Stanley Cup, NBA, and NFL sports. The international and metro links also show peaks in February and March; is this happening during the Super Bowl final and the release of new versions of smartphone OS and video game updates? Cable operators may want to confirm the events influencing traffic peaks by decomposing traffic time series at the day or the week granularity. Regarding trends, COVID-19 lockdowns significantly influence traffic trends, even a year after everything returned to normal. However, that lockdown period does not affect the periodicity component.

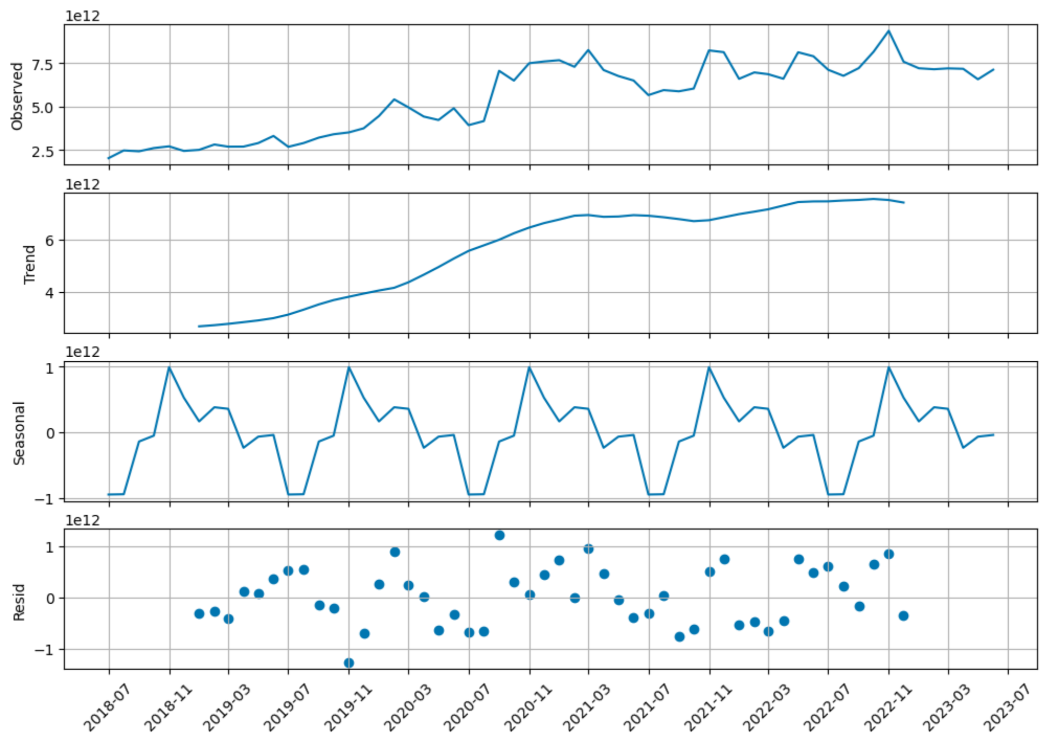


Figure 13 – Seasonal Decomposition for Total Traffic (additive)

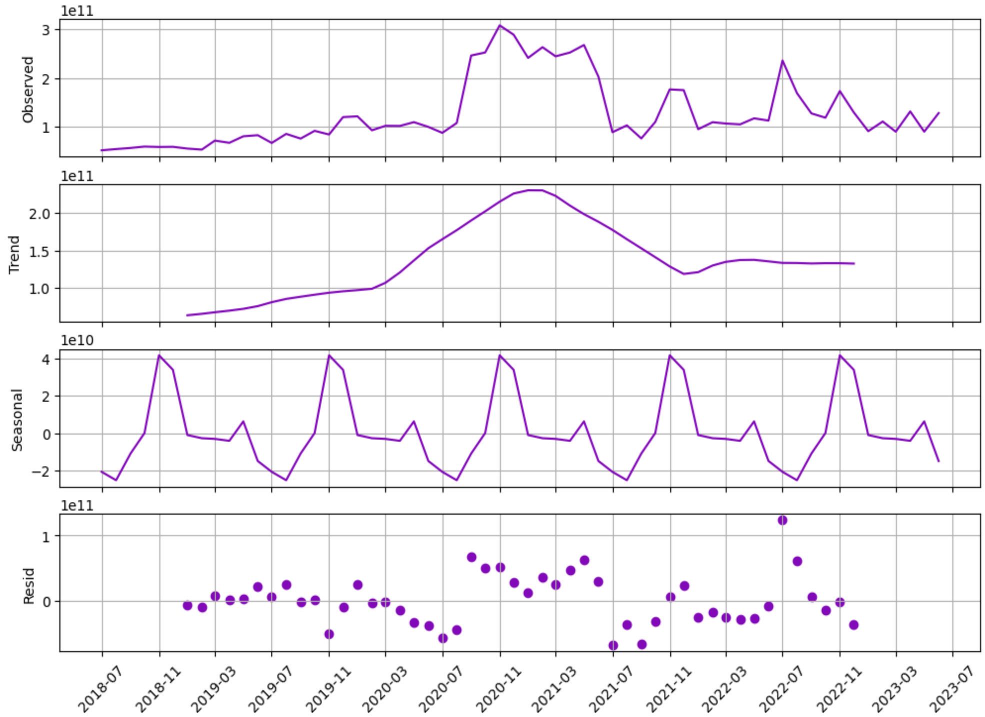


Figure 14 – Seasonal Decomposition for the internal long haul link (additive)

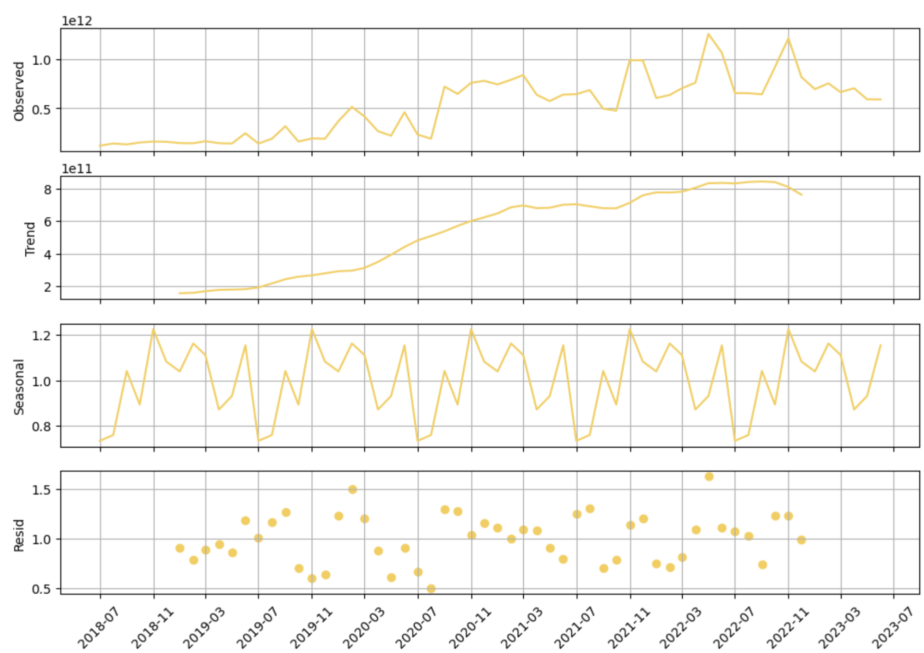


Figure 15 – Seasonal Decomposition for the international link (multiplicative)

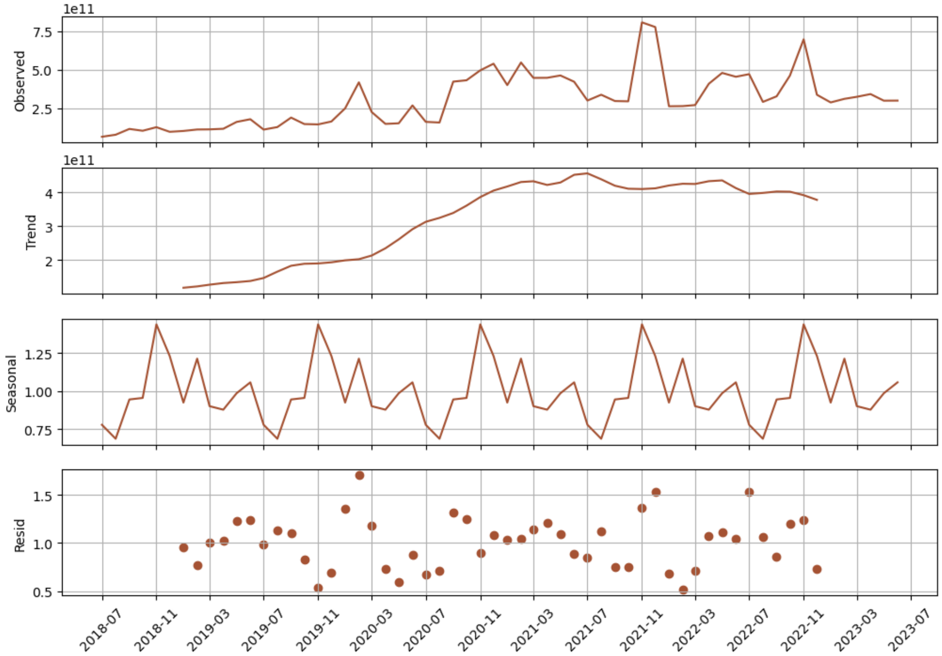


Figure 16 – Seasonal Decomposition for the internal metro link (multiplicative)

4.2. Forecasting Models for Network Traffic Time Series

As earlier stated, time series models consider the sequential nature of the data and aim to capture patterns, trends, and seasonality to make predictions about future values. STL does not directly predict the future but prepares for more accurate forecasts and predictions by identifying both trend and seasonality. There are various time series forecasting models, each with strengths and suitable applications. A few examples appropriate for network traffic forecasting are seasonal autoregressive integrated moving average (SARIMA), holt-winters exponential smoothing (ETS), Prophet from Facebook, long short-term memory (LSTM), and vector autoregression (VAR) extending autoregression to the analysis of interdependencies among multiple time series. The choice of the most appropriate model depends on the nature of the data, the presence of seasonality, the desired forecasting horizon, and the complexity of patterns in the time series.

4.2.1. Holt Winters Exponential Smoothing

Exponential smoothing (ETS) is a family of models using weighted averages of past observations to forecast future values, with different methods for handling trends and seasonality. Holt-Winters is a popular extension of ETS that consists of three components: level, trend, and seasonality. The level or smoothing component represents the time series' overall baseline or average value. Based on the current observation and a smoothing parameter alpha, the smoothing component updates at each time step. The trend component captures the time series' direction and rate of change. It is updated using a smoothing parameter (beta) that adjusts the trend over time. The seasonality component accounts for repeating patterns in the data; it updates with a smoothing parameter (gamma) that adapts to seasonal changes.

The data's presence and type of seasonality instill variations in forecasting with Holt-Winters. No seasonality Holt-Winters ETS is suitable for time series data without any seasonal patterns. Another variant is the additive seasonal model, appropriate when the seasonal fluctuations are consistent in amplitude and added to the trend and level components. The multiplicative seasonal variant is suitable for this traffic study, where the seasonal fluctuations increase and decrease with the level of the traffic time series. Python library ***statsmodels*** provides a framework for the Holt-Winters ETS multiplicative model, ready to be fitted in the data to generate predictions. Section 4.3 discusses Holt-Winters model performance.

4.2.2. Seasonal Auto Regressive Integrated Moving Average (SARIMA)

SARIMA extends the Autoregressive Integrated Moving Average (ARIMA) model and handles time series data with both non-seasonal and seasonal patterns, making it more suitable for datasets that exhibit recurring seasonal variations. ARIMA and SARIMA require making time series stationary before they can be applied. Trend and seasonality are removed from stationary time series to keep the mean, variance, and autocorrelation constant.

SARIMA model has two sets of hyper-parameters, (p, d, q) and (P, D, Q, s); where p is the order of the autoregressive (AR) component capturing the linear relationship between the current observation and its lagged values, d is the degree of non-seasonal differencing required to make the time series stationary, and q is the order of the moving average (MA) component capturing the linear relationship between current observation and the residual errors from past observations. P, D and Q are respectively SAR or seasonal AR, seasonal differencing, and SMA or seasonal MA; they are similar to p, d and q but for the seasonal part. The last parameter s gives the length of the seasonal cycle ($s = 12$, for the use case since data are monthly aggregations with yearly seasonality.)

Python library ***statsmodels*** provides a framework for SARIMA. If this simplifies the model fitting and predicting, it still requires setting and tuning hyper-parameters for prediction accuracy. Literature shares a simple strategy encapsulating SARIMA into a for loop, feeding it with a range of possibilities for each hyperparameter, and keeping the combination of hyperparameters with the least RMSE. Section 4.3 compares SARIMA performance with other time series models for network traffic forecasting.

4.3. Which Model is Best for Network Traffic Forecasting?

Figure 17, Figure 18 and **Error! Reference source not found.** give a closer look at the predictions for longhaul, international, and metro reference links. In all charts, exponential regression shows the worst performance, and ML time series techniques significantly improve predictions. SARIMA performance is inconsistent; the model reproduces previous traffic values after the MA window size. AR and MA hyperparameters require more tuning to perform. Holt-Winters ETS offers a good balance between performance and a simple implementation without further tuning.

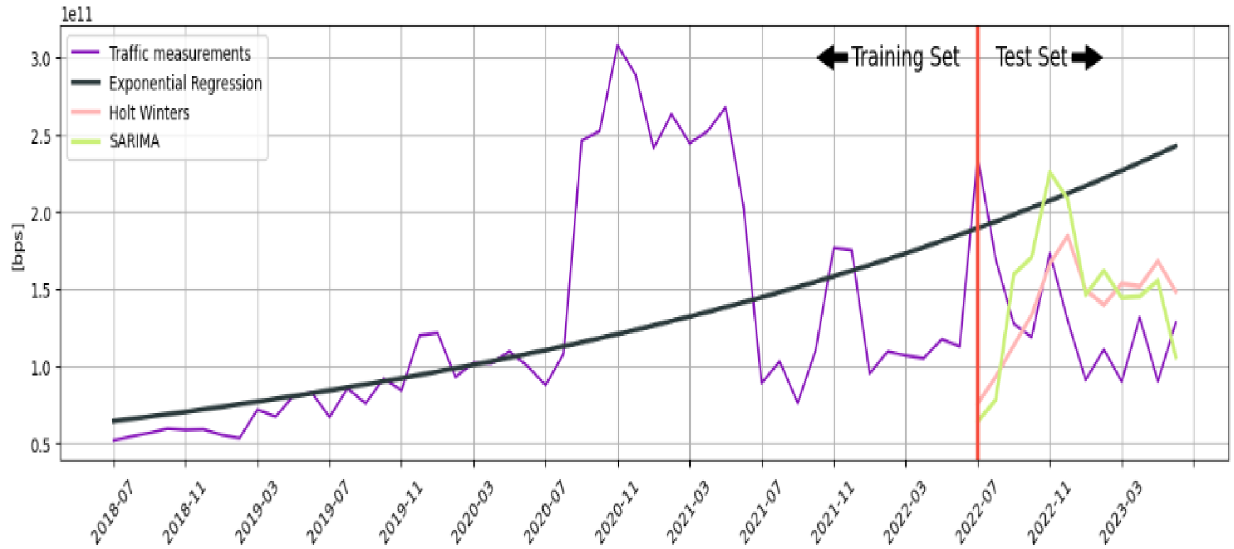


Figure 17 – Comparison of ML Model Predictions for internal long haul link

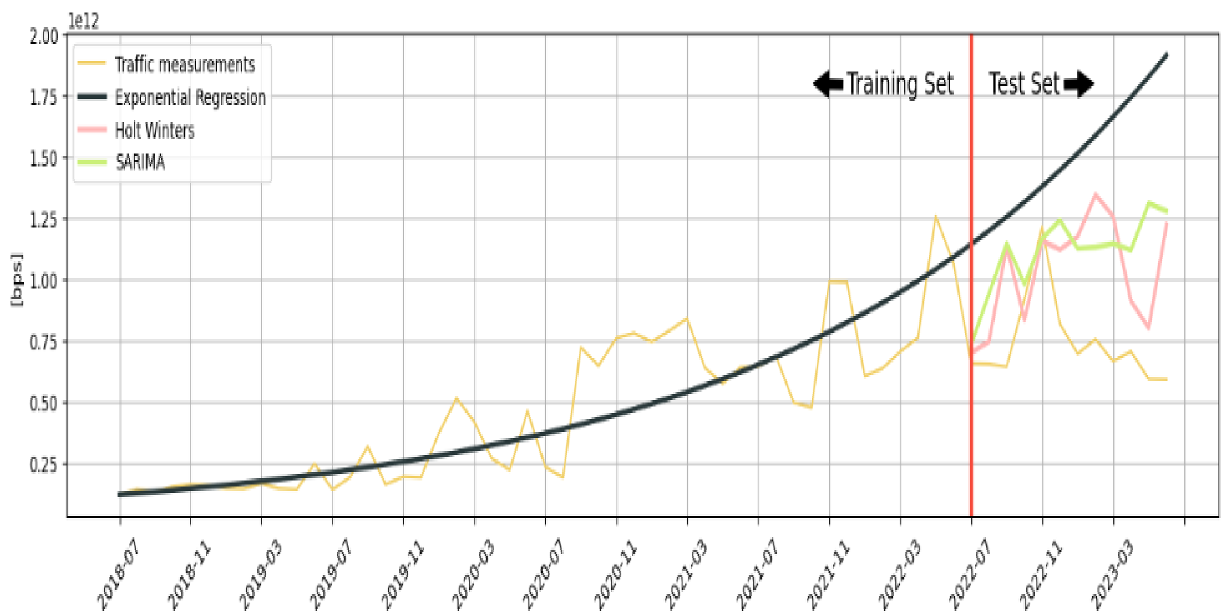
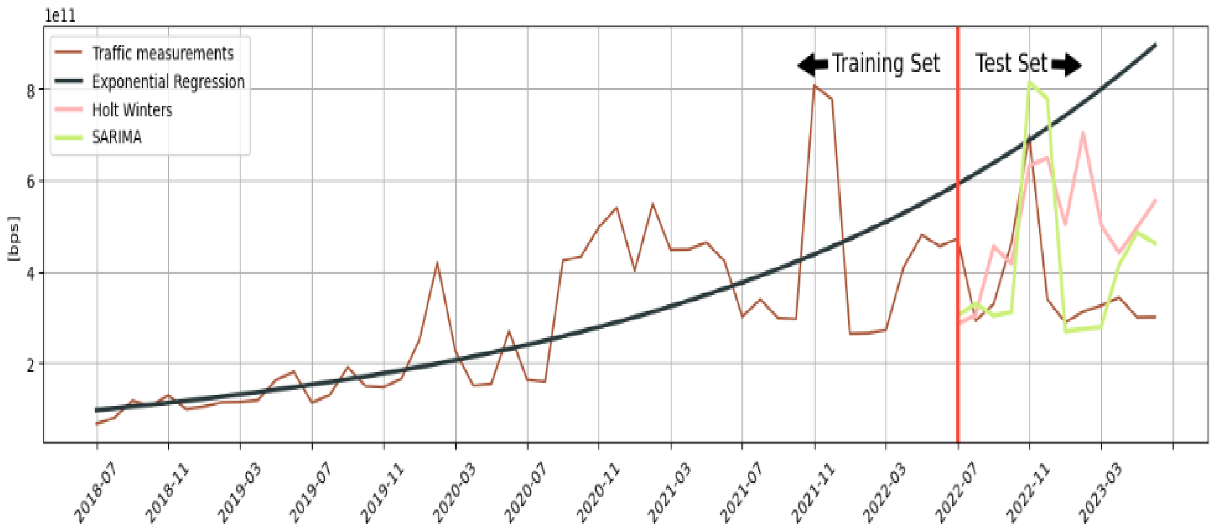


Figure 18 – Comparison of ML Model Predictions for international link



5. Key Insights

The following recommendations are based on our study forecasting traffic on a subset of 30 bundle-intraconnects from an internet backbone.

Most people will agree that the insights are only as good as the data used, so data cleaning is crucial to the success of any analysis. Reconciling missing and hidden information from the past can be time-consuming and frustrating, but the outcome is well worth the hassle. Part of clean data collection must also consider network changes, technology evolution, and network management and collection tool migrations. One may want to archive the lowest granular datasets and create suitable aggregations.

Network growth estimated for the entire network is not appropriate for the capacity planning of individual network links. This paper shows that different links have different growth rates, so applying a blanket growth rate for each link may result in inaccuracy in traffic forecasting. The authors recommend estimating the growth rate for individual links separately.

CAGR may be appropriate when determining the growth of the entire network, but it is less accurate for capacity planning. However, it is still widely used by cable operators due to its simplicity. This paper demonstrates how forecasting accuracy is impacted when applying CAGR on the most recent data sample. The authors encourage applying it on a window summary statistic instead.

CAGR's low performance in traffic forecasting starts with blindness to seasonality when calculating growth rates. ML can extract seasonality and trend, making it more appropriate for network traffic forecasting. No need to be intimidated by the term "Machine Learning." In the end, it is doing statistical analysis more efficiently. ML models are readily available in popular programming languages like Python and R or other proprietary programs.

When choosing the appropriate model, attention should be given to the nature of the data set. In this study, Holt-Winters ETS combines simplicity in understanding and implementation with efficiency in forecasting. SARIMA performs as well but requires careful tuning of hyperparameters.

Abbreviations

AR	autoregressive component
ARIMA	autoregressive moving average
AWS	Amazon web services
CAGR	compound annual growth rate
ETS	exponential smoothing
MA	moving average
MAPE	mean absolute percentage error
ML	machine learning
NMS	network management system
RMSE	root mean squared error
RNN	recurrent neural network
SAR	seasonal autoregressive component
SARIMA	seasonal autoregressive moving average
SAM	seasonal moving average
STL	seasonal decomposition of time series
SDN	software-defined networking
VAR	vector autoregression
WFH	working from home

Bibliography & References

Norwell, MA: Kluwer Academic Publishers, 2001: *Principles of Forecasting: A Handbook for Researchers and Practitioners*; J. Scott Armstrong

Texts (2nd ed): *Forecasting Principles and Practice*; Rob J Hyndman and George Athanasopoulos

O'Reilly September 2006: *Introduction to Machine Learning*; Andreas C. Muller; Sarah Guido

O'Reilly March 2018: *Machine Learning with Python Cookbook*; Chris Albon

