Peak Volume Prediction via Time Series Decomposition

by

Tucker Reed Stewart

Supervised by Juhua Hu

A senior thesis submitted in partial fulfillment of the departmental honors requirements for the degree of

Bachelor of Science
Computer Science & Systems
University of Washington Tacoma

June 2020

Presentation of work given on 06/05/2020

The student has satisfactorily completed the Senior Thesis, presentation, and senior elective course requirements for CSS Departmental Honors.

Faculty advisor: ______________ Date 06/11/2020

CSS Program Chair: Donald Ch______________ Date June 12, 2020

Peak Volume Prediction via Time Series Decomposition

Tucker Stewart (trstew@uw.edu)
Supervised by Dr. Juhua Hu, Dr. Anderson Nascimento, and Bin Yu (Infoblox)

***Abstract*—For network administration and maintenance, it is critical to anticipate when networks will receive peak volumes of traffic so that adequate resources can be allocated to service requests made to servers. In the event that sufficient resources are not allocated to servers, they can become prone to failure and security breaches. However, popular forecasting models such as Autoregressive Integrated Moving Average (ARMIA) and Recurrent Neural Networks (RNN) forecast time series data generally, thus lack in predicting peak volumes in the series. In this thesis, we aim to study how time series decomposition can be used to improve prediction when peak volumes occur in time series. More than often, time series are a combination of different features, which may include but are not limited to 1) Trend, the general movement of the traffic volume, 2) Seasonality, the patterns repeated over some time periods (e.g. daily and monthly), and 3) Noise, the random changes in the data. Considering that the fluctuation of seasonality can be harmful for trend prediction, we propose applying the Fourier Transform to extract seasonalities and study how forecasting these components independently can be used to improve both the general time series forecasting and the peak volume prediction.**

***Index Terms*—Network Traffic, Peak Volume Prediction, Time Series Decomposition, Fourier Transform**

I. Introduction

Many of previous works done for time series forecasting have focused on general trend prediction [1–4]. While these kind of forecasting models have been sufficient for most problems where entities are forecasting a target variable for any desirable point in time generally, these models lack in forecasting peak values in the data and often under predict these values. For some entities, this task is of even greater significance; namely network administrators of Domain Name System (DNS) and Dynamic Hosting Configuration Protocol (DHCP) servers.

For network administration and maintenance, it is critical to anticipate when networks will receive peak volumes of traffic so that adequate resources can be allocated to service requests made to servers. Networks receive spikes in traffic for a number of reasons that are difficult to anticipate. Demand from customers for access to these servers and networks can see a rapid rise thus traffic on the network increases. However, requests made to these servers can also be malicious in nature as these peaks in traffic volume can also be the consequences of malicious actors. One such way is a DNS Denial of Service (DDoS) attack. This is where a malicious actor hijacks the machines of benign users to create a bot net that then generates DNS queries towards the victim DNS server to overload it [5–7]. Despite the reason that network traffic receives these sudden increases in flow, the risks in not allocating enough resources to service these requests are the same. When resources are not sufficiently allocated to networks, they become prone to failure, render the server unresponsive to customers, and create a potential for security breaches. For this reason, the consequences of under predicting these peak values are much greater than over predicting.

However, popular forecasting models such as Autoregressive Integrated Moving Average (ARIMA) and Recurrent Neural Networks (RNN) have been focused on forecasting time series data generally, thus lack in predicting peak volumes in the series [1–4, 8–10]. ARIMA is a statistical model that forecasts values as a linear combination of previously observed values. Among many of ARIMA's strengths, it is great for modelling the trend and seasonality of a time series [11–13] but lacks in forecasting the extrema within the data. This is due to network traffic being non-linear and peak values are stochastic events that are not captured in the key systematic components of the data [8]. Perhaps by isolating each component of the time series, such as the seasonality, we can get better performance towards peak prediction.

Therefore in this thesis, we aim to study how time series decomposition can be used to improve prediction when peak volumes occur in time series. More than often, time series are a combination of different features, which may include but are not limited to 1) Trend, the general movement of the traffic volume, 2) Seasonality, the patterns repeated over some time periods (e.g. daily and monthly), and 3) Noise, the random changes in the data [14]. Considering that the fluctuation of seasonality can be harmful for trend prediction, we propose to extract seasonalities and study how forecasting these components independently can be used to improve both the general time series forecasting and the peak volume prediction.

To test the efficacy of time series decomposition for general and peak prediction, we propose a framework that uses signal transformation techniques to decompose the time series. Here, the Fourier transform is used to extract sinusoidal seasonal components from the trend and noise components. Then the trend plus noise component of the network traffic data are handled together and a model, such as ARIMA or Neural Networks, can be fitted to the data to forecast future observations. We applied this decomposition framework to a Unresolved DNS Queries data set provided by Infoblox, our synthetic data set idealized for the Fourier Transform, and electricity consumption as another real data set to test our methodology. This empirical study demonstrates an increase in performance for peak prediction using Fourier Decomposition.

II. Related Work

A. General Forecasting Models

Most of the work done for modelling the flow of traffic interacting with a network has been focused on general trend prediction, that is, minimizing the error for all values rather than just peak values. Popular models that have been successful in modelling time series can be divided into two categories: linear statistical models and non-linear models.

Many entities have had success in applying linear models such as ARIMA, Seasonal ARIMA (SARIMA), and other auto regressive models [1, 4, 9]. These models were one of the first to become popular because they are relatively easy statistical approaches to implement. Compared to other modelling techniques, they are relatively simple in their architecture and requires less data to train. While linear models have been successful in modelling network traffic effectively, due to their auto regressive nature in forecasting future observations as a linear combination of previous observations, they fail to represent the stochastic and non-linear dynamics of network traffic, namely the peak volumes of queries [9]. For this reason, auto regressive models and other linear models are insufficient in capturing peak information.

To address the non-linear nature of network traffic, Neural Networks (NN) have been used as a suitable alternative to forecasting network traffic. In studies comparing ARIMA and variants of NN, NN has been showing either marginal or considerable improvement to ARIMA for forecasting network traffic. There are, however, several different types of NNs that have been implemented in forecasting network traffic. The majority of architectures that have been applied to network traffic can be divided into two main families; Feed Forward Neural Networks containing ANNs and Convolutional Neural Networks (CNN) [3], and Recurrent Neural Networks such as Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU) [8].

In a study comparing the performance of ARIMA and ANN on WiMAX wireless network traffic, Stolojescu found that ANN was able to achieve better performance for forecasting small future time intervals compared to ARIMA [2].

RNNs may be preferable due to their ability to memorize information about previously observed data. This is important for time series forecast due to their temporal aspect and each observation is dependent on previous observations. This has made RNN a prime candidate for forecasting network traffic. Fu et al. [8] was able to achieve slight decreases in the MSE and Mean Absolute Error (MAE) using LSTM and GRU over ARIMA . RNNs were not used in this empirical study as they demand vast quantities of data to be effective that not all data sets, namely the Unresolved DNS Queries data set, could provide.

B. Time Series Decomposition

Time Series Decomposition bases itself on the concept that time series are comprised of several different components either additively or multiplicatively. The most prominently identified components in time series decomposition include, but are not limited to, the Trend, Seasonality, and the Residual. The Trend is the general movement the series follows. Over time, does the data tend to increase or decrease overall and at what rate? Next the data can have multiple seasonal components which are the patterns that repeat for some time interval; namely daily, weekly, monthly, annually, or following with the seasons. Lastly, there is the residual, also known as the noise or error within the data. These are the random changes that cause the data to deviate from the usual patterns. Decomposition is the process in which we separate the time series into smaller components that contribute to the overall result [14]. The motivation behind doing so is that by decomposing the series into individual pieces they can then be isolated and have a model fit to identify that particular pattern to improve performance of the overall forecasting model [4].

One method for decomposing a time series is to apply signal processing transformations to the series to transform the series into another base. Then, the information presented in the new base can be used to extract the components from the original series. Fourier Transform is one of these methods. Concretely, Fourier Transform can be used to transform a series from the time domain to the frequency domain. The series is then represented as the sum of sinusoids (i.e., sine and cosine waves). This is particularly useful for identifying and extracting seasonal components from the times series. Lewis et al. [15] found that they were able to accurately forecast the volume of call received by a call centre by applying the Fourier Transform and forecasting in the frequency domain rather than the time domain. However, not much work has been done to apply the Fourier Transform on network traffic.

III. The Proposed Method

Given a univariate time series of the form

$$X = \{\cdots, x_{t-1}, x_t, x_{t+1}, \cdots\}$$

where x_t is the quantity of the measured variable (e.g., the traffic volume) at time t. The process of forecasting the traffic volume of a network at time $t+1$ will receive can be denoted as

$$x_{t+1} = F(x_t, x_{t-1}, \cdots, x_{t-w+1}) \tag{1}$$

where F is a general time-series forecasting algorithm and w is the window size that is the number of past observations used for forecasting.

Instead of learning F directly from the observed time series, We apply the Fast Fourier Transform [16] to transform X from the time domain into the frequency domain. In the frequency domain, the time series is represented as the sum of complex sinusoids, sine and cosine waves, which are good candidates for the seasonal components of the data. Thereafter, the seasonal components than may be harmful for peak prediction, can be separated from the original time series X.

For the remaining components (i.e., trend + noise) of X denoted as

$$X' = \{\cdots, x'_{t-1}, x'_t, x'_{t+1}, \cdots\}$$

we aim to learn the prediction function as

$$x'_{t+1} = F(x'_t, x'_{t-1}, \cdots, x'_{t-w+1}) \quad (2)$$

Before fitting the model, we prepare the data X' by scaling and normalization. Inspired by a previous work [10], we also apply the local normalization to better capture the peak information. However, before the local normalization can be applied, the time series X' needs to be scaled to [1, 2] using Eqn. 3.

$$s_t = \frac{x'_t - X'_{min}}{X'_{max} - X'_{min}} + 1 \quad (3)$$

This scaling step is important considering the special property of local normalization in Eqn. 4. Specifically, we need to ensure that the data does not have any values between $[0, 1)$ since local normalization divides the current observation by its previous observation and values in $[0, 1)$ can greatly distort the scale. While local normalization tends to perform greater in terms of peak prediction, it can also suppress seasonality that exists within the time series. Fortunately, the seasonal components have been removed and will be forecast independently in our proposal.

For local normalization, each data point is normalized by the previously observed value, such that the normalized times series,

$$L = \{\cdots, l_{t-1}, l_t, l_{t+1}, \cdots\}$$

is calculated by

$$l_t = \frac{s_t - s_{t-1}}{s_{t-1}} \quad (4)$$

Finally, the data L is scaled into the range of [-1, 1] using Eqn. 5 to ensure that the scale of the values does not affect the model estimation.

$$y_t = \frac{l_t}{|L_{max}|} \quad (5)$$

In summary, to forecast the future volumes of x'_{t+1} in Eqn. 2, we will learn a prediction model as

$$y_{t+1} = F(y_t, y_{t-1}, \cdots, y_{t-w+1}) \quad (6)$$

Then, y_{t+1} will be de-normalized to forecast x'_{t+1}.

In the following subsection, we illustrate the main procedure of the proposed method using the synthetic data.

A. Peak Prediction via Fourier Decomposition (PPFD)

The Fast Fourier Transform is an algorithm that transforms a signal, a time series in our setting, from the time domain into the frequency domain, i.e. frequency spectrum. The frequency domain of a time series represents the set of sinusoids who sum to the original time series in the time domain. These sinusoids are good candidates for the seasonal components for a time series. However, it is only the high amplitude sinusoids that are likely candidates for being seasonal components as low amplitude, high frequency components are likely noisy data.

Therefore, we propose to extract only the first c highest amplitude sinusoids excluding the zero frequency sinusoid. These sinusoids with a given frequency, amplitude, and phase, can easily be calculated as cosines for the desired time interval to forecast. Then, the remaining components, the Trend and Noise of the time series, can be transformed back into the time domain and forecast using another forecasting model F such as ARIMA. Our proposed peak prediction procedures through Fourier decomposition are as follows.

1) Apply the Fast Fourier Transform (FFT) on the time series X to get the frequency spectrum as in Fig. 1. It should be noted that the zero frequency component is not a seasonal component and the seasonailities can be emphasized by removing the zero component as in Fig. 2. The synthetic data was generated by adding three seasonal components (more details can be found in Section IV) and the three seasonal components can be easily observed in Fig. 2.

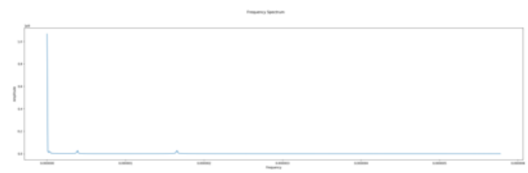

Fig. 1: Frequency domain of the synthetic data

Fig. 2: Frequency domain, with the zero frequency component removed, of the synthetic data.

2) Filter out the first c highest amplitude components by setting them to zero as done in Fig. 3 and store them.

Fig. 3: Frequency domain of the synthetic data with the three highest amplitude sinusoids removed

3) Forecast each of the individual seasonal components that were extracted from the frequency spectrum.
   a) For each of the seasonal components, compute the amplitude from the complex numbers of the frequency spectrum and the phase shift.
   b) With the amplitude, frequency, and phase shift, extrapolate each seasonal component as a cosine wave over the forecasting time interval as in Fig. 4.
   c) Sum the seasonal components together to get the combined seasonality as in Fig. 5.

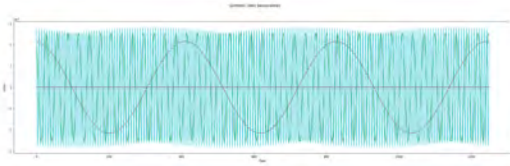

Fig. 4: The three sinusoids extracted from the synthetic data set calculated as cosine waves.

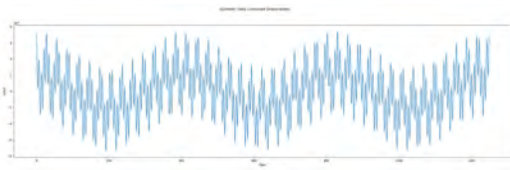

Fig. 5: Sum of the three extracted sinusoids from the synthetic data

4) Run inverse FFT on the filtered frequency spectrum to get the time series X' with the seasonal components removed.
5) Scale the filtered time series to range of [1, 2] with Eqn. 3, apply local normalization with Eqn. 4, and then scale the time series again to [-1, 1] using Eqn. 5.
6) Fit a forecasting model F on Y as in Eqn. 6 and use F to forecast the remaining components.
7) Add the predicted seasonal components from Step 3 and the remaining components from Step 6 together to get the total forecast.

IV. Experiment

To demonstrate the effectiveness of our proposed forecasting method, we evaluate it on both synthetic and real data sets. Each of these data sets is a univariate time series aggregated at different time intervals. The statistics of each data set are summarized in Table I.

A. Setup

As a baseline for the results of our proposed methodology, we apply ARIMA, ANN, and Fourier Forecasting independently from ARIMA and ANN.

1) ANN
2) ARIMA with the optimal parameters selected.
3) Fourier Forecasting where all sinusoids are forecasted and summed together.

In comparison to these baseline models, we will see the effectiveness our proposed framework through the following experiments.

1) PPFD using ANN to forecast the non-seasonal components.
2) PPFD using ARIMA to forecast the non-seasonal components.

For all of these experiments, only two general time-series forecasting models are used, that is, ANN and ARIMA. These models are setup for each experiment as follows.

1) ANN: The architecture of the model consists of a single input layer, one hidden layer made up of five sigmoid units, and an output layer with a single linear unit. The size of the input layer will vary depending on what kind of seasonality we expect the data to exhibit. For the Unresolved DNS queries and our synthetic data set, the input layer will have seven units to capture the weekly seasonality.
2) ARIMA: ARIMA requires the parameters (P, D, Q). All data sets required one level of differencing so D is always set to 1. The number of parameters, P and Q, vary for the data set so the best parameters are selected for each data set.

B. Evaluation Metrics

RMSE is the standard evaluation metric for regression problems such as time series forecasting. While this evaluation metric is sufficient for general prediction performance, it weights under prediction equally to over prediction and thus is not well suited for peak prediction. Concretely, the Mean Squared Error calculated the mean of squared errors as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2 \tag{7}$$

where x_i is observed value and $\hat{x}_i$ is the forecasted value provided by the model.

Bin et al. [10] proposed a new cost-adaptive loss function that weights under prediction greater than over prediction called the Weighted Sign Error (WSE). Then, to penalize under prediction more heavily than over prediction, the loss function is adapted to consider the sign of the error for each observation, resulting in the following function as

$$WSE = \frac{1}{N}\sum_{i=1}^{N}\alpha^{\frac{1+sign(\hat{x}_i - x_i)}{2}}(\hat{x}_i - x_i)^2 \tag{8}$$

$\alpha \in [0, 1]$ is a weighted coefficient that determines the weight of over prediction. For our experiments, α is set to 0.2, and $sign()$ is a function that returns a numerical value based on the sign of the error. Therefore, $sign(\hat{x}_i - x_i)$ returns 1 if $\hat{x}_i \geq x_i$ and -1 otherwise.

To evaluate the performance of each model, we use Root Mean Squared Error (RMSE) and Root Weighted Sign Error (RWSE) defined as follows.

$$RMSE = \sqrt{MSE} \tag{9}$$

$$RWSE = \sqrt{WSE} \tag{10}$$

We apply both of these evaluation methods on the entire validation set to evaluate both the general prediction and peak prediction as in [10]. In addition to these evaluation metrics, we use a statistical method to identify the peaks in the time series (i.e., *Find Peaks* function in Scipy's Signal module) and track the number of values in the validation data set that are under predicted and over predicted, whose RMSE and RWSE are also separately calculated to better capture the performance on peak predictions.

Data Set	Data Points	Mean	Min	Median	Max
Synthetic	7500	1375816074.4	824704284.8	1375148204.1	1944188859.3
Unresolved DNS Queries	968	105400760.9	46	98374184	294870823
Unresolved DNS Queries (Linear Interpolation)	1040	100498446.9	46	94268425	294870823
Electricity Consumption	32588	1.07789	0.0	0.78252	6.56053
Electricity Consumption (Linear Interpolation)	32164	1.09039	0.0296	0.79092	6.56053

TABLE I: Statistics of the Data Sets

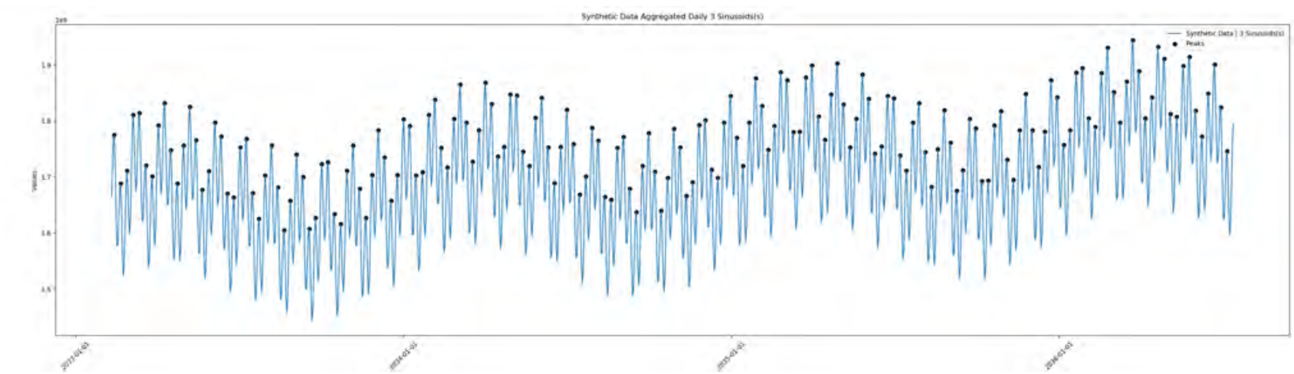

Fig. 6: Synthetic Data Set Aggregated Daily. (Last fold from the cross validation)

Model	c	RMSE	RWSE	Peak RMSE	Peak RWSE	Under Predicted	Over Predicted
Baseline Models							
ANN		0.02442	0.01908	0.01007	0.00994	630	261
ARIMA		0.02395	0.01874	0.00913	0.00913	883	8
FOURIER		0.03366	0.02597	0.01276	0.01220	723	168
Fourier Decomposition							
PPFD with ANN	1	0.02242	0.01755	0.00958	0.00944	639	252
	2	**0.02226**	**0.01741**	0.00921	0.00853	561	330
	3	0.02397	0.01877	**0.00873**	**0.00659**	**447**	**444**
PPFD with ARIMA	1	0.02961	0.02091	0.01805	0.01314	782	109
	2	0.02873	0.01975	0.01937	0.01266	654	237
	3	0.02992	0.02014	0.01940	0.01513	449	442

TABLE II: Performance Comparison on Synthetic Data

To test the efficacy of each model, we need to use some form of cross validation. Since time series data are ordered and there is essential information in the time lags between data points, shuffling the data for k-fold cross validation is not desirable. We used time series cross validation, also known as forward chaining cross validation [17]. The number of rounds used in our experiments is five. All evaluation values are aggregated across all five folds and the average of each error, across the five folds is recorded.

V. Data Description and Results

A. Synthetic Data Set

We generated a synthetic data set free of noise, to test our proposed PPFD framework on data that has very clear sinusoidal seasonal components. This data is generated by calculating a linear trend and adding sine waves of a given amplitude and periodicity. To match with the Unresolved DNS query data set, this data set is sampled daily. The trend is calculated by the function $y_t = m * x_t + b$ where the slope, m, is 100,000 and the y-intercept, b, is 1,000,000,000. Then for the seasonality, we added three seasonal components. Since the Fourier Transform represents the frequencies as complex sinusoids, we generated the seasonal components as sine waves for the periods of weekly, monthly, and yearly and with amplitudes of 80,000,000, 72,000,000, and 56,000,000 respectively. These components are then added together to get the resulting synthetic data set depicted in Fig. 6.

Table II shows the prediction results of different methods, where c indicates the number of highest seasonal components extracted. for the baseline FOURIER model, all the sinusoids are forecasted giving us a c of $\lceil \frac{N}{2} \rceil$, where N is the number of observations in the time series. First, it can be easily observed that PPFD with ANN can not only improve forecasting performance of the peak volumes but also that of the general prediction, which demonstrates that seasonal fluctuations can be harmful not just for the peak predictions but also for the general predictions. Second, with the help of Fourier decomposition that the exact number (i.e., $c = 3$) of seasonal components are extracted, the performance on peaks only (i.e., Peak RMSE, Peak RWSE) is significantly improved and the number of under predicted peaks is significantly decreased. This can also be observed in Fig. 7, where one example in red rectangular of peak volume that is under predicted by the baseline of ANN, is well predicted by PPFD with

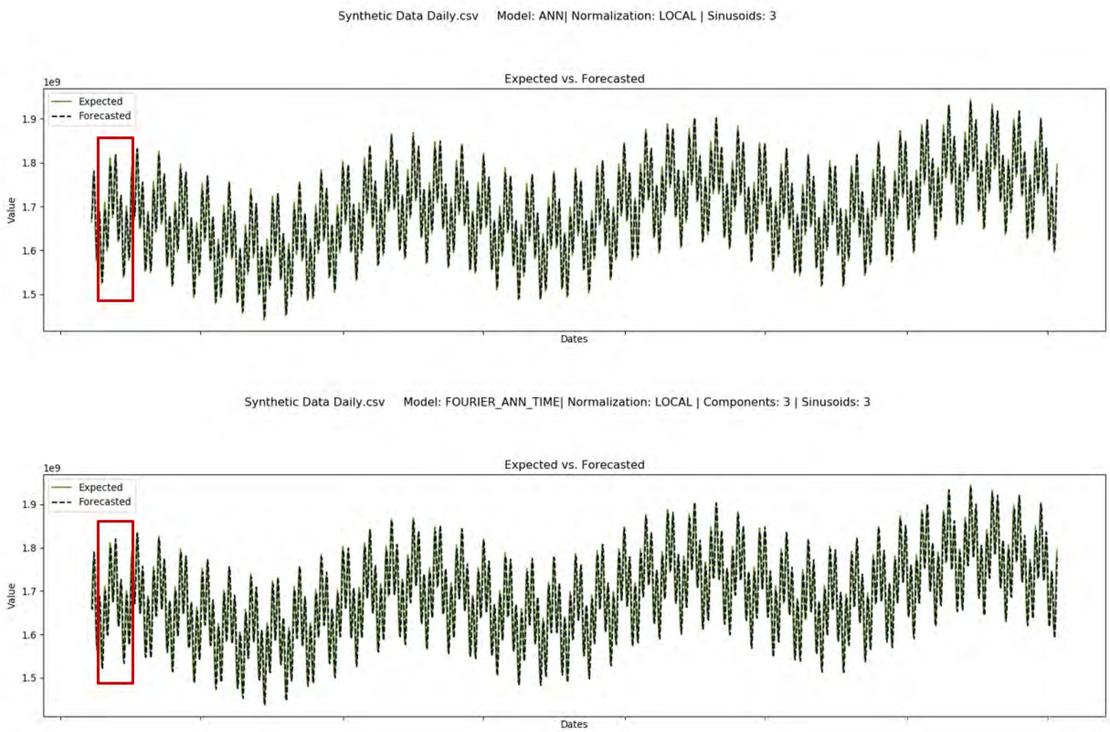

Fig. 7: Expected vs Forecasted: ANN (Top) vs. PPFD with ANN ($c = 3$) (Bottom) on Synthetic Data.

Fig. 8: Daily Unresolved DNS Queries after applying Linear Interpolation

ANN. Therefore, for time series with clear seasonality and no noise, forecasting seasons and trend separately can help in both general prediction and peak prediction.

B. Unresolved DNS Query Traffic

The DNS traffic data is aggregated daily over three years. The DNS traffic does not exhibit any obvious seasonal components and represent a wide range of values. In one day, a network could receive as few as 46 DNS queries to as many as almost three-hundred million. Hence, allocating a static amount of network resources that is sufficient to service the maximum observed number of unresolved DNS queries is wasteful. This is why, it is essential to predict these peak volumes and then allocate a sufficient number of resources only when it is necessary. Linear Interpolation is applied to fill in any gaps within the data set as described in Table I and in Fig. 8.

Even though the Unresolved DNS queries data does not exhibit any obvious seasonal patterns with the data, PPFD did improve the performance for peak values and in some circumstances all values as demonstrated by the results in Table III. Looking at Fig. 9, there are three prominent peaks, encapsulated by the red boxes, in the DNS query data set that are now being over predicted or predicted closer to the expected series rather than under predicted by the baseline. Due to the non-obvious seasonal patterns, PPFD with larger values of c generally increases the number of peaks that can be over predicted rather than under predicted. This demonstrates

Model	c	RMSE	RWSE	Peak RMSE	Peak RWSE	Under Predicted	Over Predicted
Baseline Models							
ANN		0.17796	**0.13408**	0.23630	0.23602	105	5
ARIMA		0.18932	0.13761	0.23634	0.23588	101	9
FOURIER		0.18923	0.13790	0.23430	0.23422	105	5
Fourier Decomposition							
PPFD with ANN	3	0.18004	0.13416	0.23481	0.23429	105	5
	5	0.17896	0.13463	0.23344	0.23330	102	8
	7	**0.17649**	0.13433	0.23385	0.23376	100	10
	10	0.18272	0.13757	0.24157	0.24150	105	5
PPFD with ARIMA	3	0.19745	0.13973	0.23294	0.23160	96	14
	5	0.19616	0.13978	0.23360	0.23238	97	13
	7	0.19636	0.14014	0.23353	0.23218	96	14
	10	0.19593	0.13999	**0.23176**	**0.23080**	**96**	**14**

TABLE III: Performance Comparison on Unresolved DNS Queries with Linear Interpolation

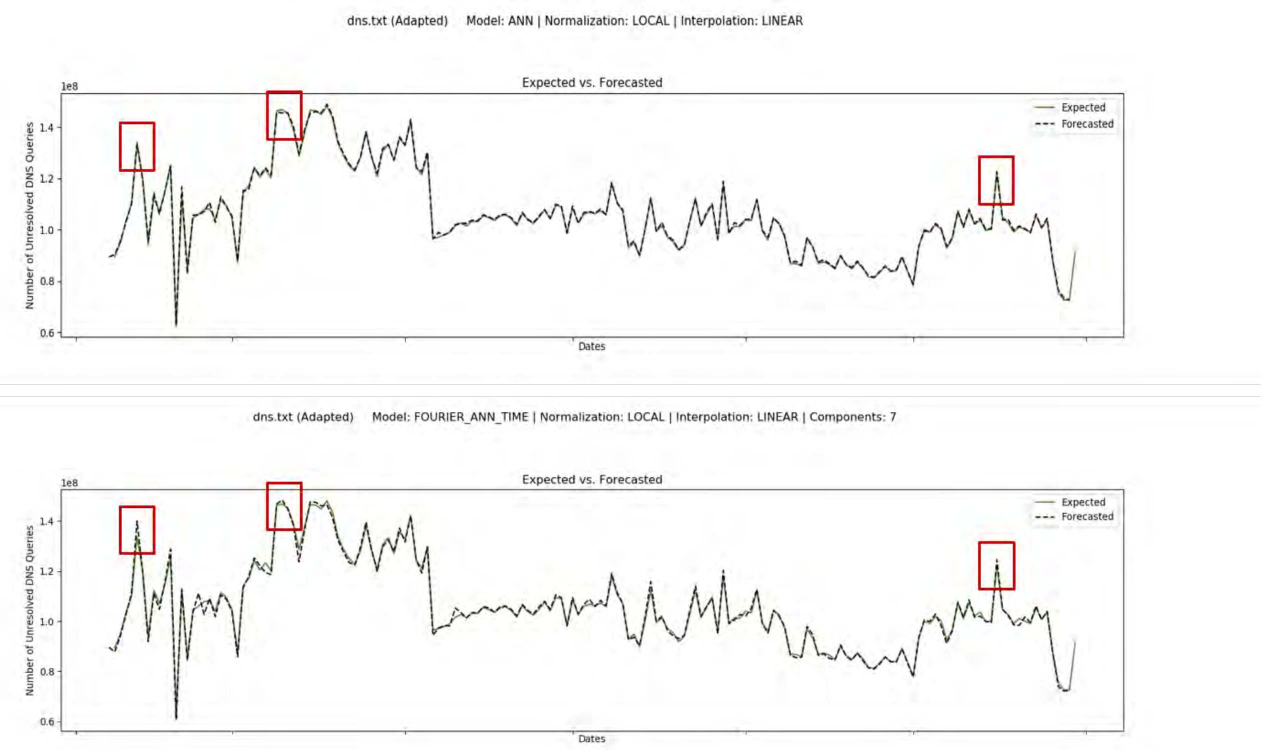

Fig. 9: Expected vs Forecasted: ANN (Top) vs. PPFD with ANN ($c = 7$) (Bottom) on Unresolved DNS Queries. (The time lag of one day was reduced for the sake of this visualization only.)

that the proposed PPFD is able to help predict both peak volumes and general volumes in real tasks, even when the seasonality of the data is not that obvious.

C. Electricity Consumption

Lastly, we include an experiment on a real data with more obvious seasonality to test the efficacy of the proposed framework, which is the electricity consumption aggregated hourly. This data set contains large segments of missing data points that are set to 0. There is a gap of 119 continuous missing data points starting on the August 17^{th}, 2010 at 10pm with another gap of 87 data points after that. Therefore, all data after August 17^{th}, 2010 at 10pm is thus removed and then Linear Interpolation is applied as shown in Table I and Fig. 10. Table IV provides similar findings, which further demonstrates that our proposed method, PPFD, is beneficial for both peak prediction and general prediction.

VI. Conclusion

For entities such as Infoblox, that provide services such as DNS, DHCP, IP address management; it is crucial to anticipate when these services will receive their peak volumes of requests. In this work, we have found that time series decomposition, particularly Fourier Forecasting, has improved the performance of the models that have been applied to network traffic in forecasting the peak values and in some circumstances improved the performance for general prediction of all values.

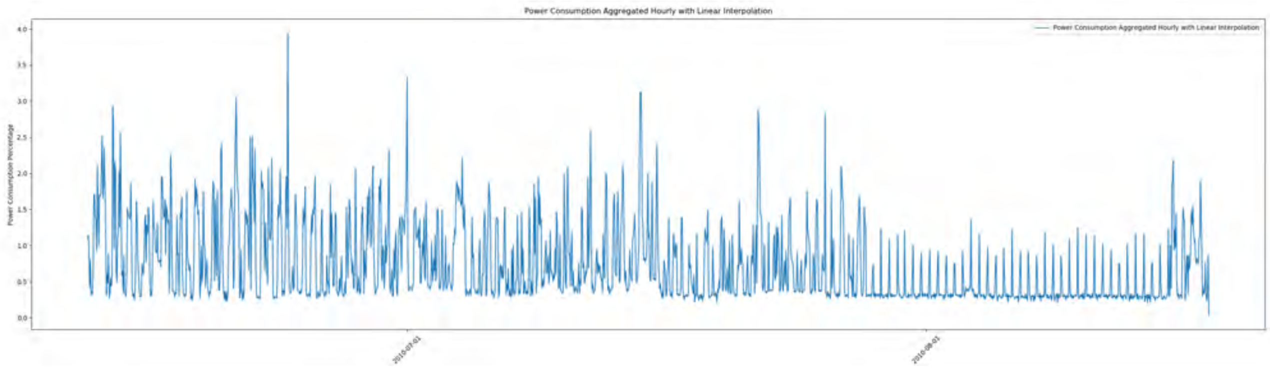

Fig. 10: Hourly electricity consumption data after applying Linear Interpolation

Model	c	RMSE	RWSE	Peak RMSE	Peak RWSE	Under Predicted	Over Predicted
Baseline Models							
ANN		0.61432	0.48528	0.88223	0.88217	3599	120
ARIMA		0.66396	0.49462	0.86497	0.86390	3508	211
FOURIER		0.62570	0.48862	0.88123	0.88113	3556	163
Fourier Decomposition							
PPFD with ANN	3	**0.61333**	**0.48338**	0.87782	0.87776	3523	196
	5	0.61652	0.48544	0.87876	0.87868	3510	209
	7	0.61495	0.48621	0.88115	0.88104	3496	223
	10	0.61846	0.48619	0.87673	0.87662	3503	216
PPFD with ARIMA	3	0.65982	0.49305	0.86716	0.86639	3480	239
	5	0.66156	0.49391	0.86633	0.86543	3481	238
	7	0.66307	0.49481	**0.86484**	**0.86377**	**3457**	**262**
	10	0.66459	0.49573	0.86622	0.86511	3457	262

TABLE IV: Performance Comparison on Electricity Consumption Data with Linear Interpolation

In future work, we want to study the effects that missing data has on Fourier Forecasting to assess the robustness of this procedure against missing data. In addition to this investigation, we want to add to this work by testing the effects of our proposed framework on more complex Neural Networks such as CNN, LSTM, GRU.

References

[1] P. K. Hoong, I. K. T. Tan, and C. Y. Keong, "Bittorrent network traffic forecasting with arma," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 4, no. 4, pp. 143–156, Jul 2012.

[2] C. Stolojescu-Crisan, "Data mining based wireless network traffic forecasting," in *10th International Symposium on Electronics and Telecommunications*, Nov 2012, pp. 115–118.

[3] A. Mozo, B. Ordozgoiti, and S. Gómez-Canaval, "Forecasting short-term data center network traffic load with convolutional neural networks," *PLoS ONE*, vol. 13, no. 2, p. e0191939, Feb 2018.

[4] Purnawansyah, Haviluddin, R. Alfred, and A. F. O. Gaffar, "Network traffic time series performance analysis using statistical methods," *Knowledge Engineering and Data Science (KEDS)*, vol. 1, no. 1, pp. 1–7, Jan 2018.

[5] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, "Dns amplification attack revisited," *Computers & Security*, vol. 39, p. 475(11), 2013.

[6] Z. Nafarieh, E. Mahdipur, and H. H. H. S. Javadi, "Detecting active bot networks based on dns traffic analysis," *Journal of Advances in Computer Engineering and Technology*, vol. 5, no. 3, pp. 129–142, 2019.

[7] D.-T. Truong and G. Cheng, "Detecting domain-flux botnet based on dns traffic features in managed network," *Security and Communication Networks*, vol. 9, no. 14, pp. 2338–2347, 2016.

[8] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," Nov 2016, pp. 324–328.

[9] D. Ergenç and E. Onur, "On network traffic forecasting using autoregressive models," Dec 2019.

[10] B. Yu, G. Graciani, A. Nascimento, and J. Hu, "Cost-adaptive neural networks for peak volume prediction," in *n Proceedings of Beijing '19: ACM International Conference on Information and Knowledge Management*, 2019.

[11] C. Chatfield and H. Xing, *The Analysis of Time Series: An Introduction with R*, 7th ed. Chapman and Hall/CRC, 2019.

[12] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and its Applications with R Examples*, 4th ed. Springer, 2017.

[13] I. Soyiri and D. Reidpath, "Evolving forecasting classifications and applications in health forecasting," *International Journal of General Medicine*, vol. 5, pp. 381–389, 2012.

[14] E. M. Pickering, M. A. Hossain, R. H. French, and A. R. Abramson, "Building electricity consumption: Data analytics of building operations with classical time series decomposition and case based subsetting," *Energy & Buildings*, vol. 177, pp. 184–196, Oct 2018.

[15] B. Lewis, R. Herbert, and R. Bell, "The application of fourier analysis to forecasting the inbound call time series of a call centre," in *Proceedings of the International Congress on Modeling and Simulation MODSIM03*, Townsville, Australia, 2003, pp. 1281–1286.

[16] O. Christensen, *The fourier transform*. Springer International Publishing, 2010, no. 9780817649791.

[17] J. P. Hermias, K. Teknomo, and J. C. N. Monje, "Short-term stochastic load forecasting using autoregressive integrated moving average models and hidden markov model," in *2017 International Conference on Information and Communication Technologies (ICICT)*, vol. 2017-. IEEE, 2017, pp. 131–137.