Received 6 May 2020; revised 30 May 2020; accepted 1 June 2020. Date of publication 4 June 2020; date of current version 19 June 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.3000059

Wireless Traffic Usage Forecasting Using Real Enterprise Network Data: Analysis and Methods

SU P. SONE, JANNE J. LEHTOMÄKI (Member, IEEE), AND ZAHEER KHAN (Member, IEEE)

Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland

CORRESPONDING AUTHOR: S. P. SONE (e-mail: sone.supyae@oulu.fi)

This work was supported in part by the Infotech Oulu through the framework of digital solutions in sensing and interactions, and in part by the Academy of Finland 6Genesis Flagship under Grant 318927.

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

ABSTRACT Wireless traffic usage forecasting methods can help to facilitate proactive resource allocation solutions in cloud managed wireless networks. In this paper, we present temporal and spatial analysis of network traffic using real traffic data of an enterprise network comprising 470 access points (APs). We classify and separate APs into different groups according to their traffic usage patterns. We study various statistical properties of traffic data, such as auto-correlations and cross-correlations within and across different groups of APs. Our analysis shows that the group of APs with high traffic utilization have strong seasonality patterns. However, there are also APs with no such seasonal patterns. We also study the relation between number of connected users and traffic generated, and show that more connected users do not always mean more traffic data, and vice versa. We use Holt-Winters, seasonal auto-regressive integrated moving average (SARIMA), long short-term memory (LSTM), gated recurrent unit (GRU) and convolutional neural network (CNN) methods for forecasting traffic usage. Our results show that there is no single universal best method that can forecast traffic usage of every AP in an enterprise wireless network. The combined models such as CNN-LSTM and CNN-GRU are also used for spatio-temporal forecasting of a single AP traffic usage. The results show that considering spatial dependencies of neighboring APs can improve the forecasting performance of a single AP if it has significant spatial correlations.

INDEX TERMS 5G, CNN, CNN-GRU, CNN-LSTM, forecasting, GRU, holt-winters, LSTM, neural network, real network data, SARIMA, spatio-temporal, temporal, time series analysis, WLAN.

I. INTRODUCTION

DIFFERENT from the previous generation of wireless networks, fifth generation (5G) and beyond wireless networks are expected to provide wireless connectivity to billions of devices and they would be required not only to have improvements in data speed but also to incorporate support for ultra-reliable low latency communication (URLLC). To achieve this, more advanced and efficient network infrastructures as well as new network function modules which can perform data analytics and forecasting on key performance indicators (KPIs) to facilitate proactive network resource allocation in a short timescale [1] will be required. The 3rd Generation Partnership Project (3GPP) has recently introduced a data driven automated centralized framework called network data analytics function (NWDAF) [2] to handle real-time data analytics and forecasting with huge amount of data. NWDAF is expected to make use of any real-time data from different resources in the 5G core network for assisting traffic routing, background data transfer and network performance predictions [3] so that data analytics and predictions in NWDAF become important. Consequently, not only cellular networks but also enterprise networks are being extended using data analytics modules like NWDAF for better wireless connectivity. For example, Cisco Miraki [4] is using dedicated data analytics modules which collect data to perform better resource allocation decisions.

Enterprise wireless network analytics requires that data relating to various network parameters, such as traffic utilization and number of connected users at an access point (AP), are collected and examined over time and space. Forecasting the total traffic utilization of the entire network from the network's perspective and forecasting traffic utilization of each AP of an enterprise network are both extremely helpful in network management and proactive resource allocation. In general, time series temporal analysis and forecasting methods can be applied on wireless network traffic data which is one of the main network parameters. Moreover, spatial dependencies of neighboring APs also have influence on forecasting wireless network traffic time series of an AP. Therefore, spatial analysis and spatio-temporal forecasting methods which make use of spatial dependencies of the neighboring APs to forecast the temporal traffic usage of a target AP should be also examined. Time series analysis methods exploit the property that network traffic data points taken over time may have some internal structures which include stationarity/non-stationarity of time series, auto-correlations, trend or seasonal variations. Spatial analysis tells how the neighboring APs are spatially correlated to a target AP. By taking into account the internal structures discovered by temporal and spatial analysis, one can use an appropriate temporal or spatio-temporal forecasting model to make predictions about behavior of network traffic usage which in turn will be useful for proactive resource allocation. For example, a good prediction about how much network traffic will be utilized at a certain AP (or certain group of APs) within short time period can help in proactively allocating appropriate resources at those APs.

Broadly speaking, various time series temporal forecasting methods can be divided into two major groups: statistical methods including exponential smoothing (ES), auto-regressive integrated moving average (ARIMA) and the theta model [5], and machine learning methods such as recurrent neural network based long short-term memory (LSTM), gated recurrent unit (GRU), convolutional neural network (CNN) and support vector regression (SVR) [6]. The past decade of research in the area has shown that there is not one temporal forecasting method that fits all types of time series data. Instead, in general, the detailed temporal analysis and appropriate forecasting method selection which is function of the input time series traffic data are required. For example, if the analysis of input data shows strong seasonality and trend, the statistical methods have been shown to give better results than machine learning [7]. On the other hand, machine learning methods have shown the ability to handle complex non-linear patterns and rapid changes [8]. For spatio-temporal forecasting, combination of recent famous machine learning methods such as auto-encoder (AE) with LSTM and CNN with LSTM are mostly used to exploit the benefit from spatial dependencies of the APs in the network [9].

In this paper, we perform detailed temporal analysis and perform forecasting of traffic utilization in a real enterprise network using both statistical and machine learning methods. We also perform spatial analysis for spatio-temporal forecasting using combined machine learning methods. To the best of our knowledge, this is the first time both temporal and spatial analysis for forecasting wireless traffic data of a real enterprise network is performed and the famous statistical methods and state-of-the-art machine learning methods are compared. The main contributions of this paper are:

1) To perform traffic usage forecasting that can be used at a resource controller of an enterprise network for proactive resource allocation, we study temporal and spatial dependencies of traffic usage data collected over a period of more than a month from 470 APs deployed in the University of Oulu.
2) For the ease of analysis and forecasting, traffic time series of APs are separated and classified into different groups based on their traffic usage patterns. We also present time series temporal analysis results such as auto-correlations and cross-correlations within and across different groups of APs. Moreover, we also examine the time series representing the number of connected users to see the relation between the number of connected users to various APs and their traffic usage.
3) We perform correlation-based spatial analysis with two different methods, Pearson spatial correlation and Moran's I spatial auto-correlation, for a target AP with its neighboring APs to be able to utilize spatial dependencies in spatio-temporal forecasting.
4) By utilizing the time series analysis results, we compare and evaluate temporal forecasting performance for traffic utilization of highly utilized APs and the entire network using five different methods: 1) Holt-Winters which is a smoothing based method, 2) Seasonal ARIMA (SARIMA) which is a regression based method, 3) LSTM which is a recurrent neural network based method, 4) GRU which is also a recurrent neural network but simpler and faster than LSTM, and 5) CNN which is a type of deep neural network with convolution and fully connected layers.
5) We also compare and evaluate spatio-temporal forecasting performances for traffic usage of a target AP by using two different methods: 1) CNN-LSTM (combination of CNN and LSTM), and 2) CNN-GRU (combination of CNN and GRU). In addition, we examine the computational complexity involved in each of the compared temporal and spatio-temporal forecasting methods in our work.

The rest of the paper is organized as follows. Section II provides the previous literature related to traffic time series analysis and forecasting. The overview of the system model is introduced in Section III. Section IV presents the basic idea of different forecasting methods and performance metrics used in this paper. The collected data set description and detailed explanations of time series temporal analysis and spatial analysis for our enterprise network traffic can be seen in Sections V and VI, respectively. The evaluations and comparisons of the forecasting performances can also be seen in Section VII. Finally, we conclude the paper in Section VIII and also present some directions for future research.

II. RELATED LITERATURE

Data analysis and data mining in NWDAF has a big role in next generation networks. The details of the network information that NWDAF needs to collect for data analysis and predictions to optimize the network, such as amount of traffic volume, number of connected devices, locations and so on, are stated in [3]. In general, most of the collected network information such as traffic utilization and channel utilization are in the form of time series [10]. However, [11] stated that due to highly dynamic traffic utilization and evolving distribution properties of the time series, traditional temporal analysis such as removing trends and differencing time series as pre-processing stage for time series predictions are not suitable for wireless networks. Instead, the wireless home network traffic time series in [11] are characterized by using correlation-based similarity. Moreover, studying temporal behaviors and patterns of the time series using correlation functions is common in wireless time series analysis as we used in our work, for example [12], [13]. In addition, [14], [15] and [16] showed that spatial dependencies of neighboring cells or base stations also have influence on forecasting cellular traffic time series of a target cell or a base station. For this reason, we also did spatial analysis for the APs deployed in a specific area of the University. In [14], [15], Pearson spatial correlation is used for spatial analysis and in [16] Moran's I spatial autocorrelation is used.

There is abundant literature on analyzing and forecasting traffic time series of cellular networks with both statistical and machine learning methods. The cellular radio traffic time series of a particular cell is predicted for one week in the future with Holt-Winters exponential smoothing method in [17]. LSTM and ARIMA are utilized to predict the base station traffic in [18] and the aggregated network traffic in [19]. Reference [20] also demonstrated high performance of recurrent neural networks such as LSTM and GRU compared to ARIMA for network traffic prediction. In [18], [19] and [20], LSTM performed higher than other methods. The research in [21] presented performance comparisons of LSTM and GRU, and showed that one method can outperform another depending on input sequences, hence, there is no clear winner between LSTM and GRU in time series prediction. Moreover, [22] stated that a CNN model can be applied to time series forecasting problems as it is expected to be good on some noisy series due to its layered structure.

Most of the recent researches such as [14] and [23], focused on both temporal and spatial cellular traffic analysis followed by prediction with ARIMA and Holt-Winters for temporal forecasting as well as combination of CNN and LSTM for spatio-temporal forecasting with commonly used metrics, mean absolute error (MAE), root mean square error (RMSE) and normalized RMSE (NRMSE). Moreover, another temporal and spatio-temporal forecasting without grid-based region partitioning of the cellular traffic data for each of total 5929 cell towers in a major city of China can be found in [16]. Reference [15] proposed a strategy by combining auto-encoder and LSTM for spatio-temporal prediction of cellular network traffic. However, [9] stated that auto-encoder may fail to learn the fully characterized features for spatial dependencies between neighboring cells so that they established CNN based framework while [23] claimed that the state-of-the-art method, CNN-LSTM, outperformed the CNN and LSTM for spatio-temporal mobile traffic forecasting. All these recent works proved that spatio-temporal forecasting improves the forecasting performance for a single AP. Therefore, we used CNN-LSTM and also established CNN-GRU as an alternative to CNN-LSTM to forecast traffic usage of a target AP in our work.

The recent work [24] studied the use of deep learning techniques for classification of mobile encrypted traffic. The work in [25] utilized 1D-CNN and GRU as multimodel deep learning and proposed the novel framework called MIMETIC to classify mobile encrypted traffic data. Different from [24] and [25], our work focused on temporal and spatio-temporal forecasting of traffic usage of an enterprise network. For our spatio-temporal forecasting, 2D-CNN is used to extract spatial dependencies and LSTM or GRU is used to learn the temporal relations in CNN-LSTM or CNN-GRU where CNN and LSTM or GRU combination is in cascade form which is different from [25]. Moreover, [26] proposed a mobile traffic super-resolution technique to make fine grained information from low resolution measurements by inspiring image processing model which is the combination of Zipper Network (ZipNet) and Generative Adversarial neural network (GAN). The proposed model is also able to capture spatio-temporal relations between traffic volume snapshots with low resolution and the corresponding usage at specific area level with high resolution. Its dataset dealt with cellular traffic in licensed spectrum in the form of image and focused on image processing techniques to capture both spatial and temporal relations of the traffic. However, our work focuses on dataset of an enterprise wireless network using unlicensed spectrum. Moreover, our datasets are in the form of time series representing traffic utilization, and number of connected users to an AP at a given time interval.

In addition, the cellular traffic time series used in [16] shows weak seasonality and similarities between weekdays and weekends which is different from the cellular traffic series used in [14], [15] and [23] telling that cellular traffic time series analysis and forecasting results can vary for different networks. The extended version of ARIMA which is called seasonal autoregressive conditional heteroskedasticity (ARCH) based model is used to forecast the traffic time series of an enterprise network in [27]. Only the total traffic of the network time series is used in [27] and the data series is separated into weekdays and weekends. It also stated that the traffic time series exhibits non-stationarity with sometimes chaotic behavior. Our collected data series also shows the daily patterns in weekdays and some sporadic patterns in weekends. Due to this reason, we also separated the collected data into weekdays and weekends as well as we classified into different groups to separate time series

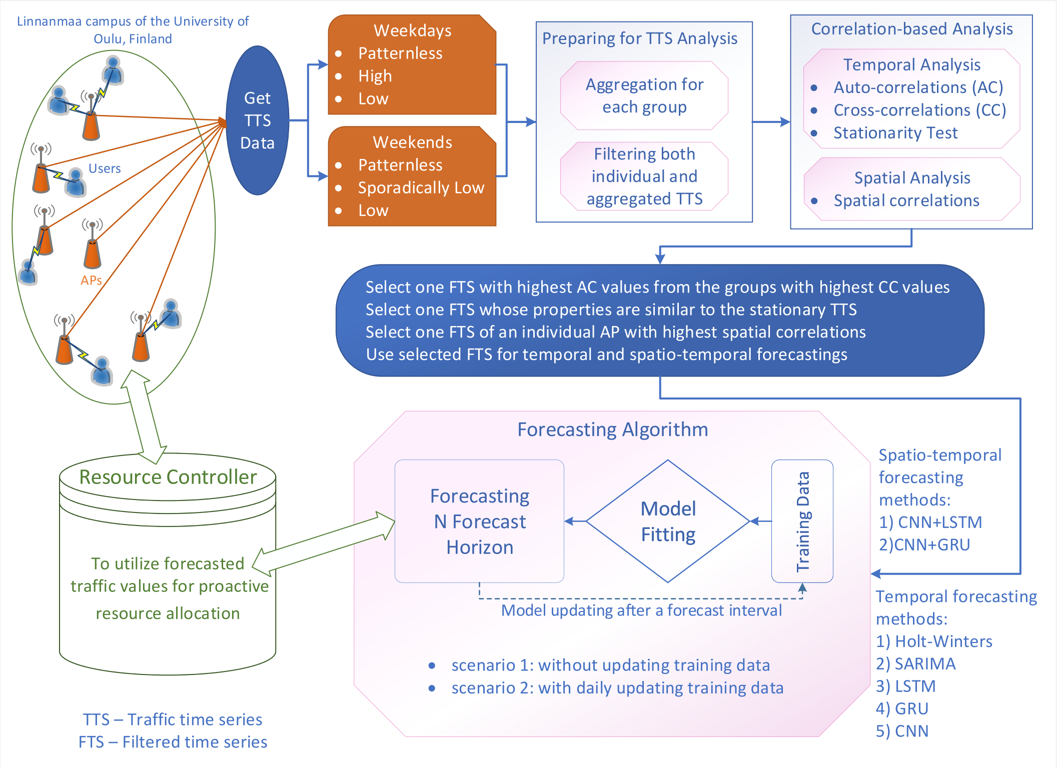
Linnanmaa campus of the University of Oulu, Finland
Users
APs
Get TTS Data
Weekdays
Patternless
High
Low
Weekends
Patternless
Sporadically Low
Low
Preparing for TTS Analysis
Aggregation for each group
Filtering both individual and aggregated TTS
Correlation-based Analysis
Temporal Analysis
Auto-correlations (AC)
Cross-correlations (CC)
Stationarity Test
Spatial Analysis
Spatial correlations
Select one FTS with highest AC values from the groups with highest CC values
Select one FTS whose properties are similar to the stationary TTS
Select one FTS of an individual AP with highest spatial correlations
Use selected FTS for temporal and spatio-temporal forecastings
Resource Controller
To utilize forecasted traffic values for proactive resource allocation
Forecasting Algorithm
Forecasting N Forecast Horizon
Model Fitting
Training Data
Model updating after a forecast interval
scenario 1: without updating training data
scenario 2: with daily updating training data
Spatio-temporal forecasting methods:
1) CNN+LSTM
2)CNN+GRU
Temporal forecasting methods:
1) Holt-Winters
2) SARIMA
3) LSTM
4) GRU
5) CNN
TTS – Traffic time series
FTS – Filtered time series

FIGURE 1. Diagram of the system model.

with chaotic behaviors. However, forecasting in [27] is only for one-step ahead and no detailed analysis of the network data is presented. Despite of having various researches of traffic temporal and spatial analysis for forecasting traffic usage of cellular and other networks, there is lack of literature on research presenting temporal and spatial detailed analysis followed by forecasting traffic usage for an enterprise network. Therefore, we performed temporal and spatial detailed analysis as well as temporal and spatio-temporal forecasting traffic usage of a certain AP (or certain group of APs) in a real enterprise network.

III. OVERVIEW OF THE SYSTEM MODEL

Forecasting the total traffic usage of the entire network and forecasting traffic usage of each AP can be extremely helpful for proactive resource allocation in network management. However, an enterprise network is completely different from the cellular network so that we performed temporal and spatial analysis for forecasting traffic data of a real enterprise network and compared famous statistical methods and state-of-the-art machine learning methods. As shown in Fig. 1, network traffic data from all of the APs around the campus of the university are collected. The collected traffic usages of weekdays and the weekends exhibit different patterns due to the nature of studied enterprise network (university campus) where majority of the wireless data will be consumed during weekdays. When weekend time series data is significantly different from weekdays, it is standard forecasting approach to separate weekdays and weekends, see for example [11], [27]. Therefore, we have separated the weekdays and weekends time series data to make the analysis simpler and coherent. Moreover, as there are some APs which exhibit different behavior than the stated weekday and weekend patterns, for ease of data analysis, we classified APs based on their traffic utilization characteristics into different groups.

Another interesting observation about the network time series data is that while the traffic of each individual AP can have high variability, the aggregation of time series of different APs within each classification group shows less variability. This means that modeling individual AP data can be more difficult as compared to the aggregated data. Therefore, we use both individual and aggregated raw data in our analysis. Moreover, as the transmitted traffic time series show high variability and some data points can be considered as the outliers of the main data pattern, certain filtering should be performed in order to get smooth values. Among different types of series smoothing, median filter is mostly used to obtain the important pattern of the time series preserving the edges which are not outliers [28]. Hence, along with raw data, filtered time series data are also used in our analysis. Then, stationarity test and correlation-based analysis for both temporal and spatial dependencies are performed. From the results of time series analysis, time series with the most interesting characteristics are selected for further forecasting step. After selecting appropriate time series, samples of the time series are divided into training

TABLE 1. The important factors and their challenges of forecasting wireless network traffic time series.

Factors	Challenges
Training Sample Size	It is almost impossible to say how much is enough sample size to train the model in wireless traffic forecasting.The amount of required training data depends on many different aspects of the forecasting model and the considered wireless network traffic time series.
Training Frequency	Since data of wireless network traffic time series are time-varying and can be non-stationary, the statistical properties of the data distribution can change as new data come in. For this reason, time series forecasting models are required to retrain after some time so that what the training frequency should be set to get the optimal forecasting performance becomes an interesting question.
Multiple Seasonality	Most of the wireless network traffic time series including our collected data series have multiple seasonalities such as daily pattern, weekly pattern and so on. Multiple seasonalities can lead to complexities in the forecasting model and and how to handle these appropriately is still a challenging research issue.
Irregular Changes	For wireless network traffic time series, sudden changes that models can not predict can happen frequently due to the uncertainty in number of users or the uncertainty in utilized data per person.
Data Processing	To enhance the performance of forecasting models, before their training, the time series data should be pre-processed using techniques such as aggregating, filtering and preparing appropriate training dataset. All these steps require selection of right parameters to get better forecasting performance which can be challenging.

data and testing data to be utilized by different forecasting models. Two important features relating to forecasting problems are forecast interval (FI) and forecast horizon (FH). The frequency with which new forecasts are prepared is called forecast interval. The forecast horizon is the number of future time periods for which forecasts must be produced.

For time series forecasting, we defined $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ as the input traffic series, $\mathbf{y} = \{y_1, y_2, \ldots, y_T\}$ as corresponding response traffic values of $\mathbf{x}$, for example, if x_i is the current traffic usage, y_i will be the traffic usage of next day at the same time for 1 day ahead prediction. We also defined $\hat{\mathbf{x}} = \{x_{T+1}, x_{T+2}, \ldots, x_N\}$ as the testing data, where T is the number of samples included in the training dataset and N is the total number of sample in the time series. In general, time series forecasting can be mathematically described as

$$\begin{aligned} &Training, \ \mathcal{F} : \mathbf{x}, x_i \in \mathbb{R} \rightarrow \mathbf{y}, y_i \in \mathbb{R} \\ &Prediction, \ \mathcal{F} : \hat{\mathbf{x}}, \hat{x}_i \in \mathbb{R} \rightarrow \mathbf{p}, p_i \in \mathbb{R} \end{aligned} \tag{1}$$

where function $\mathcal{F}$, which consists of weights, bias or residual error matrices, represents the relations between input and true response pairs of training dataset $(\mathbf{x}, \mathbf{y})$. Function $\mathcal{F}$ is optimized during a model training phase and the forecasted time series $\mathbf{p}$ is executed in a model prediction phase by applying optimized function $\mathcal{F}$ on the testing data $\hat{\mathbf{x}}$. In our case, we assume that a forecasted time series of one FI is $\mathbf{p} = \{p_{T+1}, p_{T+2}, \ldots, p_{T+H}\}$ and H is the number of forecasted samples to be produced within that FI. Let D be the number of samples for one day period. In first scenario called 5-day FH with only one FI, we forecasted traffic data for 5 days ahead continuously, where $H = 5 \times D$ without updating training dataset by assuming network does not change for a certain period. In next scenario called 1-day FH with 5 FIs, we forecasted traffic data for one day at a time (in each FI) to consider recent changes of network traffic by updating training dataset daily. First, the initial training dataset $(\mathbf{x}, \mathbf{y})$ is used to produce forecasted series $\mathbf{p} = \{p_{T+1}, p_{T+2}, \ldots, p_{T+D}\}$ within first FI, then, $(\mathbf{x}, \mathbf{y})$ is updated at the end of that FI as $(\mathbf{x}, \mathbf{y}) = \{(x_1, y_1), \ldots, (x_{T+1}, y_{T+1}), \ldots, (x_{T+D}, y_{T+D})\}$ where (x_{T+i}, y_{T+i}) is the $(T+i)^{th}$ pair of input and true response values. Then, the updated training dataset after each FI is used to retrain the model to produce forecasted series for next FI in 1-day FH scenario.

Time series forecasting itself has its own challenges since data points are time-varying and most of them are non-stationary. For particular wireless network traffic time series, the challenges are more specific. In Table 1, we present important factors and their challenges to consider before forecasting wireless network traffic time series. To consider these challenges, we performed temporal forecasting with two training dataset updating scenarios to study the effect of different training frequency. We also addressed the problem of having multiple seasonalities by separating weekdays and weekends which is also relevant for an enterprise network traffic data. Moreover, we presented the performance of forecasting time series with unusual fluctuations using machine learning methods which are famous for handling irregular changes. For data processing, the detailed analysis of time series are explained in Section V.

IV. FORECASTING METHODS AND PERFORMANCE METRICS

A. REASONS OF CHOOSING AND COMPARING THE PRESENTED FORECASTING METHODS

As wireless traffic time series often exhibit seasonal patterns, one classical approach that is available for forecasting such data is Holt-Winters which is also known as triple exponential smoothing method. In [16], the results also showed that Holt-Winters outperformed the state-of-the-art machine learning methods and gave the best MAE for cellular traffic temporal forecasting. Seasonal ARIMA (SARIMA) which is an extension of ARIMA supports the direct modeling of the seasonal component and it can describe non-stationary time series satisfactorily by neglecting the random fluctuations so that the underlying pattern of the time series can be seen clearly. Therefore, SARIMA is suitable to use for forecasting wireless traffic data. Moreover, the recurrent neural networks (RNNs), such as LSTM and GRU, are famous and well-suited to forecasting wireless traffic time series due to their advantage of preventing gradient vanishing problem. LSTM can remember the long time information (long-term memory) as in RNN and it can also learn how much information to keep from the present state (short-term memory). Research shows LSTM to be one of the most powerful tool in wireless traffic time series forecasting [29] and the literature mentioned in Section II, make a strong case to use it for forecasting wireless traffic data in our work. GRU controls the flow of information like in LSTM but without having a memory unit so that it has less gating units than in LSTM. When difference in forecasting performances of LSTM and GRU is insignificant, GRU has the advantage of lower computational complexity [21]. Therefore, in this paper, GRU is used and compared with other forecasting methods to forecast wireless traffic time series.

CNNs are famous in image processing [30] and also used in network traffic time series forecasting [31] due to their abilities of capturing the temporal dependencies in the dataset through the operations of relevant filters. Moreover, [22] also stated that a CNN model can be applied to temporal time series forecasting problems to learn the filters which are able to recognize specific patterns in the input data and use them to forecast the future values. It is also expected to be good on some noisy series due to its layered structure. These reasons gave the strong support to choose Holt-Winters, SARIMA, LSTM, GRU and CNN methods to compare and evaluate for temporal forecasting in our work. Among different combination of machine learning methods for spatio-temporal forecasting, auto-encoder with LSTM and CNN with LSTM combinations are widely used. As it is mentioned in Section II, auto-encoder cannot guarantee to learn the important features completely but it is not the case in CNN [15]. On the other hand, CNN-LSTM, which has been successfully used in activity recognition in videos, also outperformed the other spatio-temporal methods in cellular traffic forecasting [23]. For these reasons, we selected CNN-LSTM to forecast wireless traffic usage of a target AP in our work. We also established CNN-GRU to compare with CNN-LSTM for spatio-temporal traffic forecasting since GRU is faster and simpler than LSTM. As a summary, the different methods we used for temporal forecasting of our network traffic time series are: Holt-Winters, SARIMA, LSTM, GRU and CNN. For spatio-temporal forecasting, we used CNN-LSTM and CNN-GRU.

B. HOLT-WINTERS

Holt-Winters is used to forecast the time series by assigning exponentially decreasing weights and values on the old data. Based on seasonality, Holt-Winters has two type of models, additive model and multiplicative model. However, we used only additive model as wireless mobile traffic data series are more compatible with additive model [32]. Let l_t, b_t and s_t be the sequences of level, trend and seasonal factors, respectively. The closed form expressions used in the method to forecast the traffic value p_{t+m} of m^{th} time instance ahead with seasonal length L are expressed as in [33]

$$\begin{aligned}\text{Level}:\ l_t &= \alpha(x_t - s_{t-L}) + (1-\alpha)(l_{t-1} + b_{t-1})\\ \text{Trend}:\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}\\ \text{Season}:\ s_t &= \gamma(x_t - l_t) + (1-\gamma)s_{t-L}\\ \text{Forecast}:\ p_{t+m} &= l_t + mb_t + s_{t-L+1+(m-1)modL}\end{aligned} \tag{2}$$

where α, β and γ are level smoothing factor, trend smoothing factor and seasonal smoothing factor, respectively, which are optimized as usual to fit the training samples. The initial values such as l_0, b_0 and s_0 are calculated and assigned according to given time series as in [33].

C. SARIMA

SARIMA has AR term, MA term and integrated (I) term to fit the seasonal data as well as possible. It can be expressed as ARIMA$(p, d, q)(P, D, Q)s$, where p is the number of AR terms, d is the number of difference, q is the number of MA terms, P is the number of seasonal AR terms, D is the number of seasonal difference, Q is the number of seasonal MA terms and s is seasonal period of time series. SARIMA model can be trained to fit the data by adjusting the above parameters [34]. The prediction algorithm for traffic value x_t and addictive white noise $w_t \sim \mathcal{N}(0, \sigma^2)$ at time t is as follows [35]:

$$\begin{aligned}\text{AR Term}:\ \phi(B) &= 1 - \phi_1 B - \cdots - \phi_p B^p\\ \text{Seasonal AR Term}:\ \Phi\big(B^s\big) &= 1 - \Phi_1 B^s - \cdots - \Phi_p B^{sp}\\ \text{MA Term}:\ \theta(B) &= 1 + \theta_1 B + \cdots + \theta_q B^q\\ \text{Seasonal MA Term}:\ \Theta\big(B^s\big) &= 1 + \Theta_1 B^s + \cdots + \Theta_q B^{sq}\\ \text{Forecast}:\ \phi^*(B)\Phi^*\big(B^s\big)x_t &= \theta(B)\Theta\big(B^s\big)w_t\end{aligned} \tag{3}$$

where $\phi^*(B) = \phi(B)(1-B)^d$ and $\Phi^*(B^s) = \Phi(B^s)(1-B^s)^D$ are the backward shift operators of the algorithm.

Initiating an appropriate SARIMA model based on the auto-correlation function (ACF) and Partial ACF (PACF) is common but it can also be inconclusive and misleading according to an example in [33]. In general, [36] suggested to initiate with the model whose order of AR-MA-I are ARIMA$(0, 1, q)(0, 1, 1)_s$, where q can be 1 or 2 and s is seasonal period, for a time series with strong seasonal pattern. Reference [36] stated that ARIMA$(0, 1, 1)(0, 1, 1)_s$ is the most commonly used SARIMA model which also is essentially a seasonal exponential smoothing model. According to correlation-based analysis results, s is assigned as 144. However, we chose ARIMA$(0, 1, 2)(0, 1, 1)_{144}$ model which gave optimal performance for all of our time series data.

D. LSTM

LSTM is a variation of RNN and it is a type of artificial neural network whose one application is to recognize patterns in time series data. The main part of a LSTM network is called a cell which consists of 3 main regulation structures

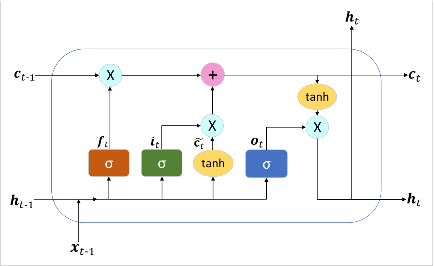

FIGURE 2. A cell structure of a LSTM network.

to control the amount of information which are called input gate, forget gate and output gate [37]. The gates of a cell in LSTM are in the form of sigmoid activation functions whose outputs are between 0 to 1, where only 0 indicates that nothing can pass through the gate, and 1 indicates that everything can pass through the gate. Let $\mathbf{x}_t$ be the input time series for an LSTM at time t, the closed form expressions for a LSTM unit can be written as follows [38]:

$$\begin{aligned} \text{Input gate}: \mathbf{i}_t &= \sigma(\mathbf{w}_{ih}\mathbf{h}_{t-1} + \mathbf{w}_{ix}\mathbf{x}_t + \mathbf{b}_i) \\ \text{Forget gate}: \mathbf{f}_t &= \sigma\left(\mathbf{w}_{fh}\mathbf{h}_{t-1} + \mathbf{w}_{fx}\mathbf{x}_t + \mathbf{b}_f\right) \\ \text{Output gate}: \mathbf{o}_t &= \sigma(\mathbf{w}_{oh}\mathbf{h}_{t-1} + \mathbf{w}_{ox}\mathbf{x}_t + \mathbf{b}_o) \end{aligned} \tag{4}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are the gate parameters, $\mathbf{w}_{ih}$, $\mathbf{w}_{ix}$, $\mathbf{w}_{fh}$, $\mathbf{w}_{fx}$, $\mathbf{w}_{oh}$, $\mathbf{w}_{ox}$ are the weight vectors for the corresponding input time series, and $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o$ are biases for input gate, forget gate and output gate, respectively. σ represents the sigmoid activation function of the gate and $\mathbf{h}_{t-1}$ is the output series of previous LSTM block. After computing the states of the gates, the cell of a LSTM network computes the candidate cell state ($\tilde{\mathbf{c}}_t$), the current cell state ($\mathbf{c}_t$) and the final output ($\mathbf{h}_t$) as follows:

$$\begin{aligned} \text{Candidate cell state}: \tilde{\mathbf{c}}_t &= \tanh(\mathbf{w}_{ch}\mathbf{h}_{t-1} + \mathbf{w}_{cx}\mathbf{x}_t + \mathbf{b}_c) \\ \text{Cell state}: \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\ \text{Final output}: \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \end{aligned} \tag{5}$$

where tanh is the hyperbolic tangent and $\odot$ means an element-wise multiplication. $\mathbf{w}_{ch}$, $\mathbf{w}_{cx}$ and $\mathbf{b}_c$ are the weights and bias of the cell. The gates and the cell structure can be seen in Fig. 2.

For neural network-based machine learning methods, deep and narrow networks can create more complex feature representations of the current input than shallow and wide networks [39]. However, stacking many layers does not always help for time series forecasting [40]. The optimal number of layers also depends on the data, for example, the LSTM performance comparison of different layers and nodes for filtered aggregated time series of total 470 APs which can be seen in Table 2. For model selection, K-fold cross-validation approach, in which data is randomly divided into K equal parts, is common for machine learning models. However, it does not work in the case of time series forecasting, since it ignores the temporal dependency of the time series [41]. Therefore, we used time series cross-validation method called rolling origin evaluation where $n-1$ chronological windows are used for training and n^{th} window is used as validation [42]. The collected data set is split into 5 windows since we are considering our time series data in daily periods and the hyperparameters which gave the optimal average result of all windows are selected for machine learning forecasting models.

TABLE 2. Performance comparison of LSTM with different layers and hidden units for filtered aggregated time series of total 470 APs for 5-day FH.

No. of layers	No. of nodes	R^2 score	Time complexity
1	32	0.8416	78.7287
1	64	0.8502	88.3561
2	32	0.8526	85.4123
2	64	0.8498	122.9440
3	32	0.8426	108.6452
3	64	0.8513	207.6043

We first started with commonly used parameters for time series forecasting from [41] and figured out the optimal values for our time series by using time series cross-validation approach. We used 2-layer LSTM each layer with 32 nodes (memory cell size) for filtered aggregated time series of total 470 APs and 64 nodes (memory cell size) for filtered aggregated time series of High group. Each LSTM layer is followed by dropout layer with probability 0.5 to prevent overfitting. Then, one dense layer is added at the end. Among different optimisers for LSTM model training, Adam optimiser is selected in our work since it can converge faster than other optimisers [14].

E. GRU

GRU is introduced in [43]. Like in LSTM, GRU has two gates called reset gate and update gate using sigmoid activation functions. After the gating operations, current memory, $\tilde{\mathbf{h}}_t$ and final output, $\mathbf{h}_t$ are calculated since GRU does not have a separate memory cell, which is in LSTM, to compute the cell states. Let $\mathbf{x}_t$ be the input time series for GRU at time t, the closed form expressions for a GRU unit can be written as follows [44]:

$$\begin{aligned} \text{Reset gate}: \mathbf{r}_t &= \sigma(\mathbf{w}_{rh}\mathbf{h}_{t-1} + \mathbf{w}_{rx}\mathbf{x}_t + \mathbf{b}_r) \\ \text{Update gate}: \mathbf{z}_t &= \sigma(\mathbf{w}_{zh}\mathbf{h}_{t-1} + \mathbf{w}_{zx}\mathbf{x}_t + \mathbf{b}_z) \\ \text{Memory}: \tilde{\mathbf{h}}_t &= \tanh(\mathbf{w}_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{w}_{hx}\mathbf{x}_t + \mathbf{b}_h) \\ \text{Final output}: \mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \end{aligned} \tag{6}$$

where, $\mathbf{r}_t$ and $\mathbf{z}_t$ are the gate parameters, σ represents the sigmoid activation function of the gate, $\mathbf{h}_{t-1}$ is the output series of previous GRU block and $\odot$ represents an element-wise multiplication. $\mathbf{w}_{rh}, \mathbf{w}_{rx}, \mathbf{w}_{zh}, \mathbf{w}_{zx}, \mathbf{w}_{hh}, \mathbf{w}_{hx}$ are the weight vectors for corresponding input time series and $\mathbf{b}_r, \mathbf{b}_z, \mathbf{b}_h$ are biases for corresponding activation functions. The gates and structure of GRU can be seen in Fig. 3. Hyperparameters of GRU model are same as in the case of LSTM since we would like to compare their performance on the same ground for our time series.

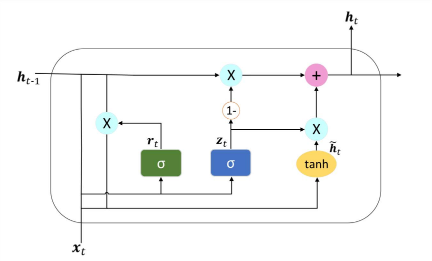

FIGURE 3. A block structure of a GRU network.

TABLE 3. Performance comparison of CNN with different activation functions for filtered aggregated time series of total 470 APs for 5-day FH.

Type of activation function	R^2 score	Time Complexity
Sigmoid, $a(x) = \frac{1}{1+e^{-x}}$	0.8243	70.7713
Tanh, $a(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	0.8218	73.0363
Relu, $a(x) = max(0, x)$	0.8245	61.8379
Leaky Relus, $a(s) = \begin{cases} x, & (x > 0) \\ kx, & \text{(otherwise)} \end{cases}$	0.8240	65.4821
Softplus, $a(x) = ln(1 + e^x)$	0.8152	104.9947

F. CNN

A common CNN consists of convolution layers which extract the high-level features from data by convoluting data with a filter, pooling layers which reduce the size of convoluted features to decrease the computational requirements and also help in extracting dominant features, and fully connected layers which have connections to all activations in previous layers to learn non-linearities of the high-level features from convolution layers. With this structure, a CNN model can be applied to time series forecasting problems to learn the filters which are able to recognize specific patterns in the input data and use them to forecast the future values [22]. Let $\mathbf{x}_t$ be the input time series at time t, the closed form expressions for a simple CNN can be written as follows [45]:

$$\begin{aligned} \text{Convolution Layer} : \mathbf{z}_{t1} &= \mathbf{x}_t \circledast \mathbf{F} \\ \text{Activation Layer} : \mathbf{A}_{t1} &= a(\mathbf{z}_{t1}) \\ \text{Fully Connected Layer} : \mathbf{z}_{t2} &= \mathbf{w}_t{}^{\mathsf{T}} \mathbf{A}_{t1} + \mathbf{b} \\ \text{Output} : \mathbf{o}_t &= a(\mathbf{z}_{t2}) \end{aligned} \tag{7}$$

where $\mathbf{F}$ is the filter in a convolution layer, $a()$ is the activation function, $\mathbf{w}_t$ is a weight matrix, $\mathbf{b}$ is a bias matrix and $\mathbf{z}_{t1}, \mathbf{A}_{t1}, \mathbf{z}_{t2}, \mathbf{o}_t$ are the outputs of their respective layers. First, we used time series cross-validation methods as mentioned in Section IV-D for choosing optimal hyperparameters for CNN. We also tried with filter size 30 since it gave optimal result for network traffic time series forecasting in [31]. However, the validation performance differences between all combinations of number of layers {1, 2, 3} and filters {30, 32, 64} are insignificant for our training dataset. Therefore, we tested all models with testing dataset and they have insignificant performance differences also for the testing dataset. Nevertheless, the model with 2-layer 2D-CNN and 30 filters gave slightly better result than other combinations. Despite of having insignificant performance differences between the models for our traffic time series, the impact of increasing hidden layers is significant in training time complexity. Additionally, if the number of hidden layer is increased, the model capacity, which is the ability to learn more complex features during training phase, becomes higher and it requires huge amount of training data. Hence, without a huge training dataset, increasing hidden layers can overfit the training data. Also, the stride size of pooling layer should be small since the main purpose of pooling layer is to reduce the size of data in CNN [45].

One important component for CNN is the activation function used in the network which does the non-linear transformation. Sigmoid function and tanh function, the updated version of sigmoid function, were widely used during the early age of CNN. However, they have the major drawback of losing information due to gradient vanishing problem. Relu function is trendy and most widely used due to its advantages such as sparse activation, better gradient propagation, scale-invariant and efficient computation. One main shortcoming of Relu function is that the derivative will be always zero when input is negative, hence, the gradient of loss function during network training becomes zero and it can affect the final result. One variant of Relu which is called leaky Relus function can overcome this shortcoming. Beside, softplus function is also famous for overcoming drawbacks of Relu function but it has higher computational complexity than Relu function [46]. Therefore, we also tested our traffic time series with five commonly used activation functions in the model with 2-layer 2D-CNN and 30 filters. As example results in Table 3 show, Relu activation function is the best for our traffic time series in terms of performance and time complexity. Moreover, reference [31] also used the same activation function so that we adopted 2-layer 2D-CNN each with 30 filters and ReLu activation followed by max-pooling layer with unit stride and dense layer at the end for our traffic time series. The same CNN model is used for all types of time series in our work, since tuning hyperparameters of CNN did not help to improve performance significantly for our traffic time series.

G. CNN-LSTM AND CNN-GRU

In wireless network, spatial dependencies of neighboring APs also have influence on forecasting traffic usage of an AP so that the machine learning models with combination of CNN and LSTM, which has been successfully used in activity recognition in videos, are used as spatio-temporal forecasting [23]. In general, 2D-CNN can extract the features of spatial dependencies from the input data and LSTM can learn the temporal relations of the series. To select the optimal hyperparameters for CNN-LSTM, we first adopted the model from [23] for initial hyperparameters and figured out the optimal hyperparameters as in LSTM. The model consists of 2-layer 2D-CNN, each with 32 filters and ReLu

Algorithm 1: Temporal Forecasting With LSTM/ GRU/ CNN

```
Load Data: Select one FTS with highest AC values
from the groups with highest CC values (or) one FTS whose
properties are similar to a stationary series.
Normalization: $x_i \leftarrow \frac{s_i-\mu}{\sqrt{\sigma^2}}, i = 1, 2, ..., N$
Divide Data: $\mathbf{D}_{tr} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_b, \mathbf{y}_b)\}$
    $\mathbf{D}_{te} = \{(\mathbf{x}_{b+1}, \mathbf{y}_{b+1}), (\mathbf{x}_{b+2}, \mathbf{y}_{b+2}), ..., (\mathbf{x}_m, \mathbf{y}_m)\}$
        // $b = \frac{T}{batch\_size}, m = \frac{N}{batch\_size}$
Define: Model = Sequential() // for time series
Set: layer.LSTM(nodes, activation, dropout) or GRU() or
     CNN()
     layer.MaxPooling(stride_size) // only for CNN
     layer.LSTM(nodes, activation, dropout) or GRU()
     or CNN()
     layer.MaxPooling(stride_size) // only for CNN
     layer.Dense(1) // for traffic value
     Model.compile(optimizer='adam',loss='mae')
for $i$ = 1 to No. of forecast intervals (FIs) do
    Train Model: $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_b\} \leftarrow$ Model($\mathbf{D}_{tr}$)
                 optimize $L_{mae} = \frac{1}{T}\sum_{i=1}^{T} |y_i - z_i|$
                 update weights
    Predict Model($i^{th}$ FI): $\{p_{T+1}, p_{T+2}, ..., p_{T+H}\} \leftarrow$
                 Model($\mathbf{D}_{te}$($i^{th}$ FI))
    Update Training Data: $\mathbf{D}_{tr} \leftarrow$ add $\mathbf{D}_{te}$($i^{th}$ FI)
                 into $\mathbf{D}_{tr}$
Denormalization: $f_i \leftarrow \{p_i * \sqrt{\sigma^2}\} + \mu$,
                 $i = T+1, T+2, ..., N$
Evaluate: $MAE = \frac{1}{N-T}\sum_{i=T+1}^{N} |s_i - f_i|$
    // also evaluate using 1)RMSE,
    2)NRMSE and 3)$R^2$ score
```

Algorithm 2: Spatio-Temporal Forecasting With CNN-LSTM / CNN-GRU

```
Load Data: Select FTS of a single AP with highest
spatial correlations, FTS of its neighbor APs (to use as
features in $\mathbf{x}_i$ of $\mathbf{D}_{tr}$ and $\mathbf{D}_{te}$).
Normalization: $x_i \leftarrow \frac{s_i-\mu}{\sqrt{\sigma^2}}, i = 1, 2, ..., N$
        // for each FTS
Divide Data: $\mathbf{D}_{tr} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_b, \mathbf{y}_b)\}$
    $\mathbf{D}_{te} = \{(\mathbf{x}_{b+1}, \mathbf{y}_{b+1}), (\mathbf{x}_{b+2}, \mathbf{y}_{b+2}), ..., (\mathbf{x}_m, \mathbf{y}_m)\}$
        // $b = \frac{T}{batch\_size}, m = \frac{N}{batch\_size}$
Define: Model = Sequential() // for time series
Set: layer.CNN(nodes, activation)
     layer.MaxPooling(stride_size)
     layer.CNN(nodes, activation)
     layer.MaxPooling(stride_size)
     layer.LSTM(nodes, activation,dropout) or GRU()
     layer.LSTM(nodes, activation,dropout) or GRU()
     layer.Dense(1) // for traffic value
     Model.compile(optimizer='adam',loss='mae')
for $i$ = 1 to No. of forecast intervals (FIs) do
    Train Model: $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_b\} \leftarrow$ Model($\mathbf{D}_{tr}$)
                 optimize $L_{mae} = \frac{1}{T}\sum_{i=1}^{T} |y_i - z_i|$
                 update weights
    Predict Model($i^{th}$ FI): $\{p_{T+1}, p_{T+2}, ..., p_{T+H}\} \leftarrow$
                 Model($\mathbf{D}_{te}$($i^{th}$ FI))
    Update Training Data: $\mathbf{D}_{tr} \leftarrow$ add $\mathbf{D}_{te}$($i^{th}$ FI)
                 into $\mathbf{D}_{tr}$
Denormalization: $f_i \leftarrow \{p_i * \sqrt{\sigma^2}\} + \mu$,
                 $i = T+1, T+2, ..., N$
Evaluate: $MAE = \frac{1}{N-T}\sum_{i=T+1}^{N} |s_i - f_i|$
    // also evaluate using 1)RMSE,
    2)NRMSE and 3)$R^2$ score
```

activation. First layer is followed by max-pooling layer with stride size 2 and second layer is followed by max-pooling layer with unit stride. Then, the output of 2D-CNN is passed into LSTM with 64 nodes each followed by dropout layer with probability 0.8. We also established CNN-GRU as an alternative to CNN-LSTM for spatio-temporal traffic forecasting. The same hyperparameters are used in CNN-GRU as in CNN-LSTM.

Let's s_i denote the i^{th} sample, N is the total number of samples in a selected time series used for forecasting and T is the number of samples in training dataset. The overview algorithms of temporal and spatio-temporal forecasting with machine learning methods can be seen in Algorithm 1 and Algorithm 2, respectively. Before forecasting, auto-correlations (AC), cross-correlations (CC) and stationarities of aggregated or individual filtered time series (FTS) as well as spatial correlations of individual FTS of APs from a focus area are evaluated as temporal and spatial analysis. According to these analysis results, specific FTS are selected for further temporal or spatio-temporal forecasting as in Algorithms 1 and 2.

H. PERFORMANCE METRICS

To evaluate and compare both temporal and spatio-temporal forecasting methods, we need to introduce the performance metrics that can measure their accuracy. There are many different performance metrics to evaluate the time series forecasting methods. Among them, the most common and widely used metric is called root mean square error (RMSE) [42] which is given as

$$RMSE = \sqrt{\frac{1}{M}\sum_{t=1}^{M}(x_t - p_t)^2} \tag{8}$$

where M is the total number of samples in a predicted period, x_t is true value and p_t is predicted value. However, the disadvantage of RMSE is that it gives more weight to larger errors. One way to reduce the effect of a few outliers in traffic data is to remove the square term and using mean absolute error (MAE) [42] which is expressed as

$$MAE = \frac{1}{M}\sum_{t=1}^{M}|x_t - p_t| \tag{9}$$

The above metrics are not normalized and the performance thresholds are varying according to the true values. One solution to overcome this problem is to normalize RMSE. Since there is no consistent definition of normalized RMSE (NRMSE) [47] in the literature, we use difference between maximum and minimum to normalize RMSE to obtain

NRMSE which is defined as

$$NRMSE = \frac{RMSE}{max(\mathbf{x}) - min(\mathbf{x})} \tag{10}$$

where RMSE is root mean square error defined as in (8), $max(\mathbf{x})$ is the maximum value and $min(\mathbf{x})$ is the minimum value of true data series of $\mathbf{x}$. Another way to compare the performances using normalized standard metric is coefficient of determination which is also known as R^2 score [48]. It is calculated as follows:

$$\text{Total Sum} : S_{tot} = \sum_{t=1}^{M} (x_t - \bar{x})^2$$

$$\text{Sum of squared residuals} : S_{res} = \sum_{t=1}^{M} (x_t - p_t)^2$$

$$\text{Coefficient of determination} : R^2 = 1 - \frac{S_{res}}{S_{tot}} \tag{11}$$

where $\bar{x}$ is the sample mean of the true data. In this paper, we use all RMSE, MAE, NRMSE and R^2 score to compare the performance of different time series prediction algorithms.

V. TEMPORAL ANALYSIS OF NETWORK DATA

A. DESCRIPTION OF COLLECTED REAL NETWORK DATA

We collected the real measurement data from a total of 470 APs deployed in the Linnanmaa campus of the University of Oulu, Finland. The dataset contains measurements of received traffic data (in bytes), transmitted traffic data (in bytes) and number of users in the form of time series as well as name of the locations and ID numbers for a total 470 APs. Each measurement provides a data point at every 10-minute interval between January 5, 2019 and February 8, 2019, hence, each time series has 5040 observations. Among received traffic data at an AP, which is also known as uplink data, and transmitted traffic data from an AP, which is called downlink data, the transmitted traffic data dominates significantly over the received data at an AP. Due to this reason, we mostly focus on the transmitted traffic data. We also focus on the connected number of users to an AP.

B. GENERAL REPETITIVE PATTERNS OF TRAFFIC USAGE

The collected traffic time series data over the period of one month shows that heavy traffic usage of several APs can be seen during office/university hours within weekdays and sporadically light traffic usage can be seen in weekends as shown in Fig. 4. The group classification for weekdays and weekends can be seen in Fig. 1. For both weekdays and weekends, we use the group called "Patternless" whose members were identified by correlation-based analysis. The detailed correlation-based analysis of both weekdays and weekends for different groups is presented in Section V-D and V-E. For weekdays, APs with random traffic patterns which have strong AC with previous lags at first and followed by a linear fall off are classified into Patternless group since these APs have the AC characteristics of a random walk time series with complete random movement which are unpredictable [12]. APs with significant traffic usage but no significant correlations for their traffic time series which is also a characteristic of randomness in the time series [12] are also included in Patternless group.

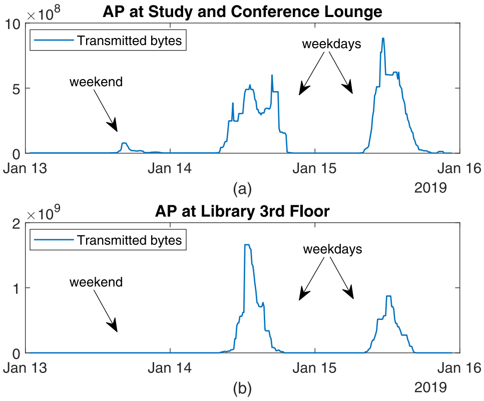

FIGURE 4. Transmitted traffic for Sunday (Jan 13), Monday (Jan 14) and Tuesday (Jan 15).

APs with repetitive daily patterns during weekdays are classified into High and Low groups depending on their mean traffic level. We used Otsu's algorithm which is well known for being used in image processing [49] and is also being used for wireless communication data thresholding [50]. In our work, Otsu's method is used with three thresholds for automatic mean level thresholding to separate APs with seasonality into High and Low groups using the lowest threshold. Then, we selected APs whose rounded means are above or equal to the highest threshold as the representative APs of High group with very high traffic usage. In general, we would like to perform traffic usage forecasting at resource controller of the network for proactive resource allocation which is mainly required for APs with very high traffic usage in an enterprise network. For weekends, some APs do not have significant traffic usage over the whole weekends and most APs do not have correlation between Saturday and Sunday. Therefore, APs with almost no transmission over the weekends are in Almost No data group. APs with sporadic transmission patterns in which there is increasing traffic from 8am onward and then decreasing over the evening like in weekdays on some weekend days while there is no traffic utilization on other weekend days go to Sporadically Low group. Patternless group is for APs with random traffic utilization over the weekends (similar to weekdays Patternless group).

C. TIME SERIES SMOOTHING

As the collected data is huge, we select the same number of representative APs (as in High group) with highest mean traffic usages from each classification group of APs. We consider individual APs and aggregated data (from the representative APs) of each group for later correlation-based

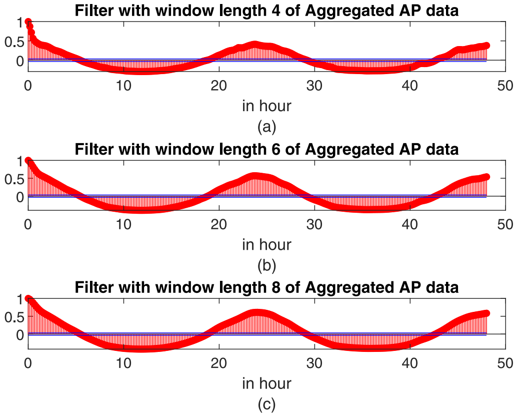

FIGURE 5. **Auto-correlation functions of time series data with various filter length and original data of an APs in Low group of weekdays.**

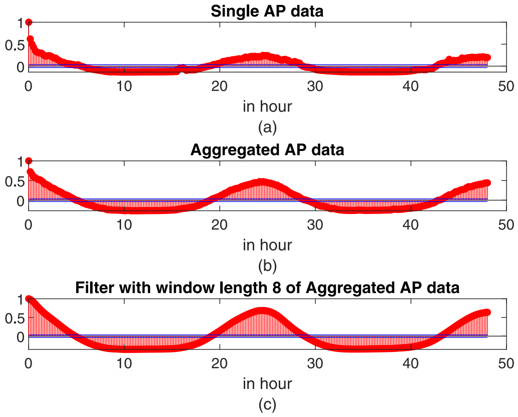

FIGURE 6. **Auto-correlation functions of a single AP, and aggregated data of APs with and without filter in High group of weekdays.**

analysis. Median filter with various window lengths is also used for time series smoothing. The important part of applying median filter is selecting an appropriate window length. For example, median filtering with window length 8 means taking the median value of 1 hour 20 minutes duration at every time step in the time series. We observed that the median filter with window length 4 gives high correlation values for individual APs. For aggregated time series of each group, median filter with window length 8 gives the smoothest pattern and high correlation values as in Fig. 6(c). Example AC results of filtering the aggregated traffic series of Low group for weekdays with different window length are shown in Fig. 5 and they show that time series traffic data are slightly sensitive to time aggregations and filtering. Moreover, it is also observed that when we aggregated the time series of all 470 APs to see the behavior of traffic utilization from network's perspective, the resultant aggregated time series became less noisy and applying filtering to it further smoothed the time series data. As our goal for correlation-based time series analysis is to study correlation properties of individual APs and APs in classified groups, hence, results relating to the aggregated time series of all 470 APs are not presented in correlation-based analysis.

D. AUTO-CORRELATION

To analyze the patterns in time series data, it is typical to use AC and CC metrics to find the similarities, seasonality and differences of the time series of APs. AC is evaluated to see how strong the connections between the values of a single time series at different delays are. It can provide information about any repeating patterns and predictive power of the time series [12]. We next present results relating to the AC of the collected time series data for various APs of different groups. Let's denote the two different time series of single APs with x and y, and the two different aggregated time series with X and Y, where x_t, y_t, X_t and Y_t represent the traffic value at time t of each series. The AC function denoted as $\alpha_{xx}(k)$ used for time series analysis at time lag k is then given by [51]

$$\alpha_{xx}(k) = \frac{c_{xx}(k)}{c_{xx}(0)} \tag{12}$$

where $c_{xx}(k)$ is the auto-covariance function at time lag k and is given as

$$c_{xx}(k) = \frac{1}{T}\sum_{t=1}^{T-k}(x_t - \bar{x})(x_{t+k} - \bar{x}), \tag{13}$$

$c_{xx}(0)$ is the sample variance and is given as

$$c_{xx}(0) = \frac{1}{T}\sum_{t=1}^{T}(x_t - \bar{x})^2, \tag{14}$$

and T is the number of time series samples and $\bar{x}$ is the sample mean.

For weekdays of a representative AP in High group, the AC of traffic values at next 10 minutes, 1 hour and 24 hours are 62%, 31% and 22%, respectively. Our results in Fig. 6(a) show this AC relationship for an AP belonging to High group. The AC of a representative AP from Low group also has the same behavior as in High group but with much lower AC values of 15%, 7%, and 4% at next 10 minutes, 1 hour, and 24 hours, respectively. The AC of aggregated traffic values in High and Low groups show significantly higher values as the AC even at the next 24 hours are 45% and 17%, respectively. The AC for aggregated traffic of High group can be seen in Fig. 6(b). The result tells that aggregated traffic of APs in High group can be predicted using seasonality/daily pattern based methods for next day with higher accuracy than APs in Low group.

The traffic of individual APs in Patternless group exhibit randomness, hence, some APs show the AC characteristic of a random walk time series and some APs show no significant correlations. When the traffic of individual APs in this group is aggregated, the AC of aggregated time series shows linear decrease with some fluctuations within 24 hours and then shows small increase in AC values at 24 and 48 hours with overall decreasing AC values. This means that the aggregated

TABLE 4. The cross-correlations between different single APs within and across the groups considering only weekdays.

	Patternless			High			Low		
	1 hr	6 hr	24 hr	1 hr	6 hr	24 hr	1 hr	6 hr	24 hr
Patternless	-0.015	-0.015	-0.001	0.025	-0.01	0.01	0.04	0.05	0.02
High	0.003	-0.023	0.032	0.24	-0.07	0.28	0.23	-0.04	0.26
Low	0.02	0.0008	0.033	0.29	-0.08	0.31	0.028	-0.02	0.034

traffic shows very weak seasonality at 24 hours which gets even weaker at the 48 hours interval but with overall random walk time series characteristics. This behavior is significantly different from the High and Low groups as the seasonality in these two groups is almost constant over several multiples of 24 hours (24 hours, 48 hours, etc.) and is high compared to the Patternless group.

There is very low traffic utilization over the weekend as compared to the weekdays. As evaluating AC of each individual APs for weekends is not insightful, we only present results relating to aggregated time series for the weekends. The aggregated data for Patternless group shows linearly decreasing AC values over the period of 48 hours with the characteristics of random walk model. The AC does not increase at 24 and 48 hours indicating that even the weak seasonality is not present which is as expected. For the sporadically low group, the AC shows linear decrease in data for the 8 hours interval which is followed by small negative AC values which indicate the behavior that traffic is utilized during day and then almost no traffic utilization at night. The sporadically low group also shows very little seasonality at 24 hours interval such as 40% at next 10 minutes, 15% at next 1 hour and 5% at next 24 hours. For APs with almost no data transmission during weekends, it is not insightful to evaluate their correlations since they do not even have significant transmission.

E. CROSS-CORRELATION

Correlation function helps to describe the evaluation of the process through the time and it is often called an analysis in the time domain to detect the non-randomness in data. In general, CC is evaluated to see how the values of different time series are connected. In our work, CC is used to measure similarity between traffic time series of different APs or different groups of APs as a function of the time lag of one relative to the other. The CC at time lag k between two different time series x and y is

$$\psi_{xy}(k) = \frac{c_{xy}(k)}{\sqrt{c_{xx}(0)}\sqrt{c_{yy}(0)}} \tag{15}$$

where $c_{xy}(k)$ is the cross-covariance between time series x, y and is given by

$$c_{xy}(k) = \frac{1}{T}\sum_{t=1}^{T-k}(x_t - \bar{x})(y_{t+k} - \bar{y}), \tag{16}$$

$\sqrt{c_{xx}(0)}$ and $\sqrt{c_{yy}(0)}$ are the sample standard deviations of the series, T is the number of time series samples, and $\bar{x}$ and $\bar{y}$ are the sample means.

For weekdays of individual APs, the CC between two different representative APs within the same group and the CC between a representative AP of each group across the different groups can be seen in Table 4. According to the Table 4, there exists very small negative CC between the data points of a single AP from each group at 6 hour difference. The CC between two representative APs within High group is significant at 1 hour and 24 hours lags. Moreover, High and Low groups are significantly correlated to each other, telling that they have the similar characteristics. The rest of the cases show no significant correlations. For High group, all of the CCs within and across the groups have consistent CC behavior such that for 1 hour and 24 hours are positively correlated and for 6 hour is negatively correlated unlike in Patternless group with CC values which are mostly close to zero with random fluctuations. The results from Table 4 are based on a particular AP that we selected from the different groups. Unlike in AC, the CC at different time lags are from different APs so that the results might be biased according to our choice of APs. We can also find the average of all four representative APs in a group to reduce the bias.

The average CC for APs within a group can be simply calculated as

$$\hat{\psi}_{xy}(k) = \frac{\sum_{i=1}^{M}\psi_{xy}^{i}(k)}{M} \tag{17}$$

where $M = \binom{P}{2}$ represents the number of 2-combinations of APs from the considered group with no repetition, P is the number of APs in one group, and $\psi_{xy}^{i}(k)$ represents the CC of the i^{th} combination within the set, x and y are the different time series from the same group. The average CC for APs across the groups can be defined similarly as

$$\hat{\psi}_{xy}(k) = \frac{\sum_{i=1}^{N}\psi_{xy}^{i}(k)}{N}, x \in \mathcal{G}_1, y \in \mathcal{G}_2 \tag{18}$$

where $N = P^2$ represents the number of 2-combinations between APs from two different groups $\mathcal{G}_1$ and $\mathcal{G}_2$. The average CC within the same group and CC across the different groups can be seen in Table 5. According to the Table 5 for weekdays, only APs in High group show the significant CC at 1 hour and 24 hours lags. For all other cases, there is very little or no correlations. By comparing Table 5 and Table 4, we see that main difference coming from taking into account all combinations within/across the groups is that averaging high and low groups results in much less correlated values than using only single representative APs.

For the aggregated time series, the CC, $\tilde{\psi}_{XY}(k)$, can be defined similarly as for a single AP. The aggregated time series of a group can be expressed as $X = \sum_{i=1}^{P} x^i$ and

TABLE 5. The average cross-correlations within and across the groups considering only weekdays.

	Patternless			High			Low		
	1 hr	6 hr	24 hr	1 hr	6 hr	24 hr	1 hr	6 hr	24 hr
Patternless	0.04	0.03	0.02	-0.002	-0.003	-0.001	-0.001	-0.007	-0.008
High	0.003	0.022	0.005	0.21	-0.04	0.22	0.095	-0.024	0.099
Low	-0.005	0.005	-0.0001	0.11	-0.015	0.09	0.04	-0.009	0.04

TABLE 6. The cross-correlations of aggregated series itself and across the groups considering only weekdays.

	Patternless			High			Low		
	1 hr	6 hr	24 hr	1 hr	6 hr	24 hr	1 hr	6 hr	24 hr
Patternless	0.65	0.44	0.24	-0.06	-0.01	-0.06	-0.05	-0.02	-0.05
High	-0.02	0.11	-0.04	0.52	-0.10	0.45	0.25	-0.04	0.25
Low	-0.05	0.03	-0.004	0.27	-0.05	0.27	0.16	-0.02	0.17

the CC of aggregated time series across the groups can be calculated using (15) by only substituting $x = X$ and $y = Y$, where X and Y are aggregated time series of different groups. The CC of the aggregated time series within the same group, which is in fact the AC of the aggregated time series of a group, and across the groups are shown in Table 6 for weekdays. Since the CC within the same group is exactly the same as AC, the behavior of correlations of each group are same as in Section V-D. The strong correlations between High and Low groups can be seen in Table 6 also. By comparing Table 6 and Table 5, we see that the correlation between Patternless group and itself has significantly increased with linearly decreasing correlations due to its random walk properties.

Since time series data of each AP does not have any consistent pattern during weekends, the CC between different APs within the same group, and the CC between representative APs of each group across the different groups are not significant. The average CCs of within and across the groups of weekends are not significant also. Even the CCs of aggregated time series within and across the groups are not significant unlike in AC, meaning that one AP can not be predicted based on another AP data for weekends since they do not have any correlated information.

F. TIME SERIES STATIONARY TEST

For time series prediction, stationarity of a time series is important since it can strongly influence the behaviour and properties of a time series [42]. In general, stationary means the statistical properties of a time series do not change over time and there are several different definitions of stationarity (e.g., strong stationarity and weak stationarity). The previous literature [11] stated that wireless traffic time series are not stationary both in general sense and in wide sense as their distribution characteristics change over time. Therefore, we tested our own time series data for stationarity using unit root tests such as Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests [52]. We applied both unit root tests on our data in two different ways. First, the short time periods of a time series is tested by using overlapping sliding window protocol with specific time frame. Then, the most frequently occurred outcome case is taken as the result for short time period of the whole time series, hence, the collected time series are analyzed and tested to use for short-term predictions in the future. According to the results, most of the 10-hour periods of the time series of all APs in each group are absent of unit roots for both weekdays and weekends. Second, the complete time series is tested for unit roots in order to know the behavior of the long-term traffic. Accordingly, most of the APs in Patternless groups of both weekdays and weekends have unit roots and all of the APs in other groups are absent of unit roots in the whole time series.

On the other hand, absence of unit root in a time series does not grantee for stationarity [53] so we also tested for stationarity by checking whether mean and variance are constant over time, and covariance between two time instants depends only on the time difference between the two time instants and not the actual time at which the covariance is computed [54]. We tested for individual APs time series, aggregated time series of each group and an aggregated time series of all 470 APs with 4 hours, 8 hours, 16 hours and 24 hours time intervals. None of the scenarios above have shown constant mean and variance. However, means and variances of an aggregated time series of all 470 APs is observed to be flatter than other time series.

For covariance test, let $K_{xx}(t_1, t_2)$ be the auto-covariance between windows t_1 and t_2 of a time series x, and $K_{xx}(\tau)$ be the auto-covariance of τ^{th} window of a time series x, where $t_2 - t_1 = \tau$. We observe that none of the selected windows of the time series satisfy the definition of weak stationarity. However, Fig. 7 shows that when the traffic of more number of APs is aggregated, there is more similarity between $K_{xx}(t_1, t_2)$, $K_{xx}(t_3, t_4)$ and $K_{xx}(\tau)$, where $t_2 - t_1 = t_4 - t_3 = \tau$. This can be observed in Fig. 7(c). In summary, the raw time series data are not stationary but when they are aggregated, the results show similarity to the stationary time series properties.

G. USERS AND TRAFFIC RELATION

Next, we look into the relation between number of connected users and generated traffic. For weekdays, especially in High and Low groups, the number of connected users time series shows similar pattern to traffic time series. However, Fig. 8

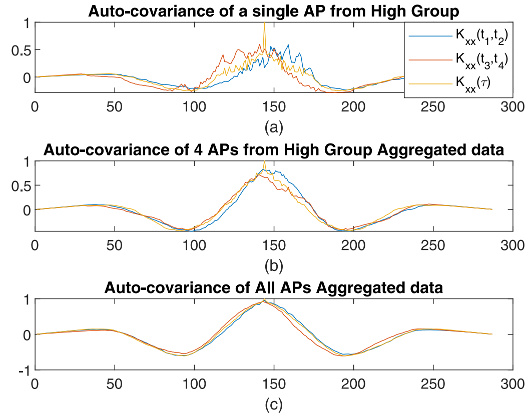

FIGURE 7. Auto-covariance of non-overlapping sliding windows with 24 hours time interval.

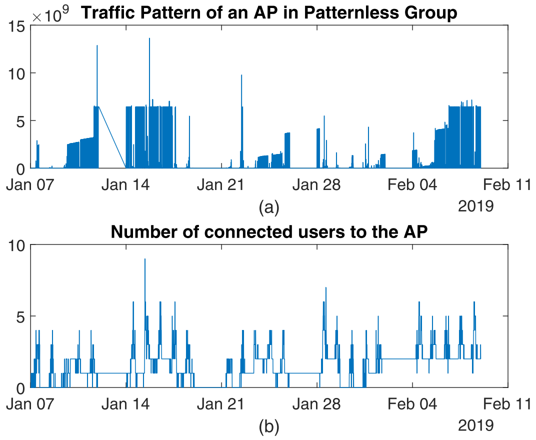

FIGURE 8. Pattern of traffic and number of users in one representative AP of Patternless group for weekdays.

shows that the patterns are not similar between the connected user time series and traffic time series for the Patternless group. The works in [55] and [56] have observed the similar behavior and the reason for it which can be explained as follows. In Fig. 8, a small number of users with heavy data usage contribute the majority of traffic in APs so that data usage pattern for users are highly uneven. We also found that mostly the number of users that contribute to the most traffic is much lower than the total number of users connected to that AP at a given time.

Moreover, the AC of connected users time series and traffic time series of a single AP are compared. Fig. 9 shows that the number of connected users at different time lags are more correlated and hence more predictable than traffic data of the AP for weekdays. Unlike in traffic time series analysis, even for weekends, the connected users time series shows some correlations. AC of users are more widely spread than in traffic correlations as in Fig. 9. To evaluate the CC between user and traffic, sample Pearson correlation coefficients are calculated by setting the number of connected users time series as the variables which is causing the changes in traffic time series variables. The sample Pearson correlation between two different variables is

$$r_{uw} = \frac{\sum_{t=1}^{T}(u_t - \bar{u})(w_t - \bar{w})}{\sqrt{\sum_{t=1}^{T}(u_t - \bar{u})^2}\sqrt{\sum_{t=1}^{T}(w_t - \bar{w})^2} \qquad (19)$$

where T is the number of points in one time series data, u_t, w_t are the sample points at time t of different time series and $\bar{u}$, $\bar{w}$ are the sample means. Pearson's correlation coefficients for each representative AP of each group during weekdays and during weekends can be seen in Tables 7 and 8, respectively. Despite of having highly uneven data usage pattern for the connected users, it can be seen that APs in High and Low groups have high Pearson correlation coefficients during weekdays. Moreover, different from CC values between traffic time series for weekends in Section V-E, it can be seen that there are higher Pearson correlation coefficients values for all representative APs of Sporadically Low group in weekends.

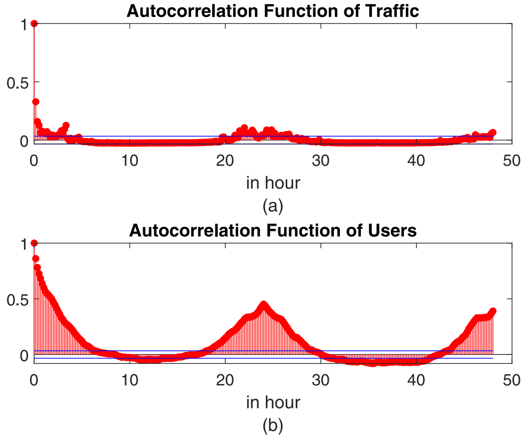

FIGURE 9. Auto-correlations of traffic and number of users in one representative AP of Low group for weekdays.

TABLE 7. The Pearson correlation between traffic and user variables for weekdays.

	Patternless	High	Low
AP1	0.056	0.66	0.45
AP2	0.084	0.50	0.14
AP3	0.050	0.50	0.29
AP4	-0.078	0.34	0.12

TABLE 8. The Pearson correlation between traffic and user variables for weekends.

	Patternless	Sporadically Low	Almost No
AP1	0.071	0.329	-0.0008
AP2	0.773	0.151	0.210
AP3	0.114	0.281	0.054
AP4	0	0.207	0.141

VI. SPATIAL ANALYSIS OF NETWORK DATA

An enterprise network is completely different from the cellular network due to the facts: 1) some APs in an enterprise network have their own purpose and do not correlate with other surrounding APs and 2) traffic usage of APs from Patternless group can not be predicted and considering spatial dependencies of these APs is not useful also. According

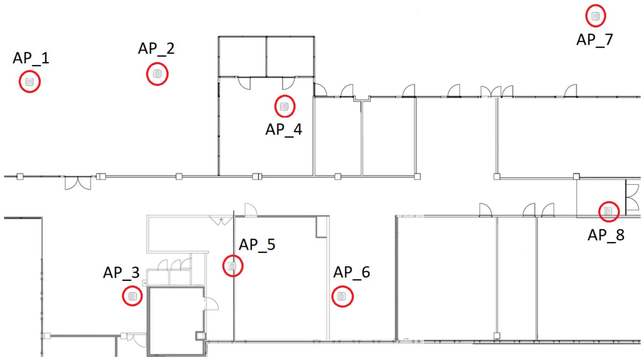

FIGURE 10. Locations of APs at study and conference lounge area.

TABLE 9. Pearson spatial correlation between a target AP of study and conference lounge, AP(1), and its neighboring APs.

AP(1)	AP(2)	AP(4)	AP(7)
1	0.1838	0.5441	0.4267
AP(3)	AP(5)	AP(6)	AP(8)
0.4227	0.3497	0.4807	0.3882

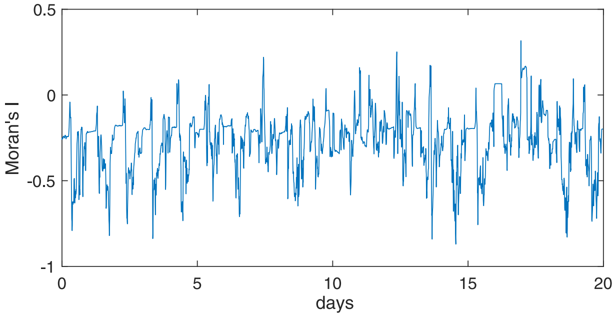

FIGURE 11. Moran's I spatial auto-correlation of APs in study and conference lounge area for 20 days.

to above temporal analysis results, proactive resource allocation is essential for APs with high utilization deployed at the area with high user density during weekdays. Therefore, we selected the most crowded area of the University, study and conference lounge, where total 8 APs with high traffic utilization are located as in Fig. 10. Since we are focusing on the spatial analysis for each AP located at the focus area of the University considering spatial dependencies of its neighboring APs, we used only filtered traffic utilization values of the APs as time series smoothing method without aggregating any traffic time series of APs.

As in time series temporal analysis, correlation-based methods are also used for spatial dependencies analysis. We examined spatial dependencies with two different widely used metrics: Pearson correlation [14], [15] and Moran's I auto-correlation [16]. The spatial correlation between a target AP, AP$_u$, and its neighboring AP, AP$_w$, can be measured by Pearson correlation which is the same as in equation (19). Moran's I measures the overall spatial auto-correlation between APs of a focus area. It is evaluated to see how APs in the focus area are similar to or different from their surrounding APs. In general, let $x_t(i)$ denote the traffic value of an i^{th} AP at time t, the Moran's I is given by

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}} \times \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}(x_t(i)-\bar{x}_t)(x_t(j)-\bar{x}_t)}{\sum_{i=1}^{n}(x_t(i)-\bar{x}_t)^2} \tag{20}$$

where W is the binary weight matrix and an individual entry, W_{ij}, is 1 when i^{th} and j^{th} APs are adjacent and 0 if they are not, n is the number of APs located within the focus area and $\bar{x}_t$ is the mean traffic value of all associated APs at time t.

First, we examined the Pearson spatial correlations for each AP between it and its neighboring APs. From results for each AP, we selected only one AP with maximum correlations to all of its neighboring APs and with high traffic utilization as a target AP whose spatial correlations are given in Table 9. APs are arranged sequentially based on their distance from a target AP and placed in Table 9 with respect to their actual locations on map. Despite of having close distance, AP(1) is not significantly correlated with AP(2). However, AP(1) and AP(4) are highly correlated. Moreover, AP(2), AP(5) and AP(8) do not have any significant spatial correlation with other APs while AP(4) is also highly correlated with AP(6) and AP(7).

In general, Pearson spatial correlation is not sufficient for spatial analysis so that we also examined Moran's I spatial auto-correlation of a focus area. Fig. 11 shows Moran's I values of APs in our focus area for 20 days. The mean Moran's I value is -0.2683 with negative spatial auto-correlation which means adjacent APs do not have similar behaviors but some distant neighboring APs have. The above Pearson spatial correlations also support the results of Moran's I being negative.

VII. NETWORK TIME SERIES DATA FORECASTING

In an enterprise network, the majority of the wireless data is consumed during weekdays so that proactive resource allocation and network management are mainly required for weekdays. Therefore, we next utilize the analyzed time series of weekdays from the previous sections which are already classified and smoothed to optimize the performance of forecasting methods used in our work. By using temporal forecasting, filtered aggregated time series of total 470 APs which has the most similar properties of a stationary series is used to predict the total traffic usage of the network. In addition, we knew that traffic usage of APs in Patternless group are not predictable and APs in High and Low groups have similar traffic usage patterns with only difference in mean levels as their CC values are high. Moreover, predicting high traffic usage are typically important for network management and proactive resource allocation. Hence, we focused on aggregated and filtered time series of High group representing the traffic usage of a group of APs, which also has the highest AC values, by using temporal forecasting. For a single AP prediction, we focused on the traffic usage of a

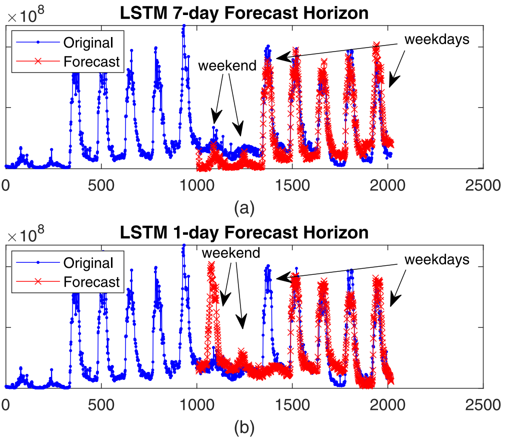

FIGURE 12. Temporal forecasting with LSTM Algorithm for filtered aggregated time series of total 470 APs including weekends.

TABLE 10. Comparison for the aggregated time series of total 470 APs for 5-day FH.

	RMSE	MAE	NRMSE	R^2 score
Holt-Winters	2.41×10^7	1.95×10^7	0.0701	0.8230
SARIMA	3.03×10^7	2.49×10^7	0.0882	0.7203
LSTM	$\mathbf{2.20 \times 10^7}$	$\mathbf{1.67 \times 10^7}$	**0.0639**	**0.8526**
GRU	2.24×10^7	1.87×10^7	0.0652	0.8464
CNN	2.42×10^7	1.98×10^7	0.0704	0.8217

target AP with the highest spatial correlations from a specific area by using both temporal and spatio-temporal forecasting in this paper.

A. PERFORMANCE COMPARISON FOR TEMPORAL FORECASTING METHODS

Before presenting the forecasting performances of classified and analyzed time series, we would like to prove that weekdays, weekends separation of traffic time series is a better approach to gain good forecasting results for an enterprise network. The temporal forecasting performances of LSTM with 7-day FH and 1-day FH scenarios applied on filtered aggregated time series of total 470 APs without separating weekdays and weekends can be seen in Fig. 12. We can see that LSTM is predicting reasonably well when using 7-day FH although the forecasted results for weekends are not very good. However, its performance degrades significantly when using 1-day FH. Moreover, we tried with Holt-Winters double seasonal forecasting method [57] by considering both daily seasonality and weekly seasonality. However, forecasting performance for weekends was also significantly degraded.

For the filtered aggregated time series of total 470 APs, LSTM gives the best result in every performance metric for 5-day FH as shown in Table 10, while Holt-Winters has the best performance for every metric in 1-day FH which can be seen in Table 11. The forecasting performances of Holt-Winters and GRU algorithms for filtered aggregated time series of total 470 APs can be seen in Fig. 13 and Fig. 14, respectively. According to Table 10, most of the machine learning methods have better performances than statistical methods for 5-day FH. However, according to Table 11, statistical methods give better results for 1-day FH which is completely opposite of the previous situation. At the same time, the performance comparisons of filtered aggregated time series of representative APs of High group for 5-day FH and 1-day FH scenarios can be seen in Table 12 and Table 13, respectively. For High group time series in 5-day FH, Holt-Winters performs the best in RMSE, NRMSE and R^2 score while SARIMA gives the best result of MAE. However, the MAE difference between Holt-Winters and SARIMA is not significant so that we can assume Holt-Winters performs the best for 5-day FH in overall. For 1-day FH, SARIMA gives the best results in every metric for the filtered aggregated time series of representative APs of High group.

TABLE 11. Comparison for the aggregated time series of total 470 APs for 1-day FH.

	RMSE	MAE	NRMSE	R^2 score
Holt-Winters	$\mathbf{1.97 \times 10^7}$	$\mathbf{1.41 \times 10^7}$	**0.0575**	**0.8811**
SARIMA	1.99×10^7	1.49×10^7	0.0581	0.8786
LSTM	1.99×10^7	1.50×10^7	0.0592	0.8750
GRU	2.15×10^7	1.55×10^7	0.0625	0.8589
CNN	2.23×10^7	1.67×10^7	0.0650	0.8478

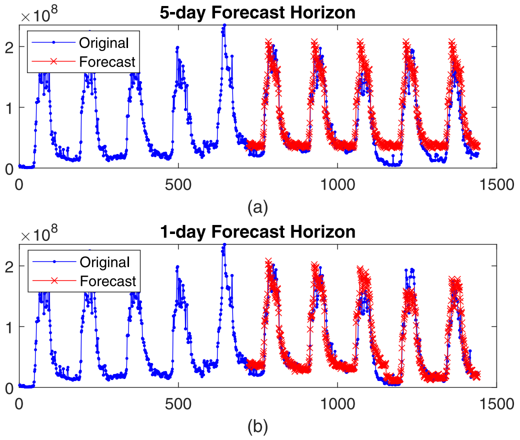

FIGURE 13. Forecasting with Holt-Winters Algorithm for filtered aggregated time series of total 470 APs.

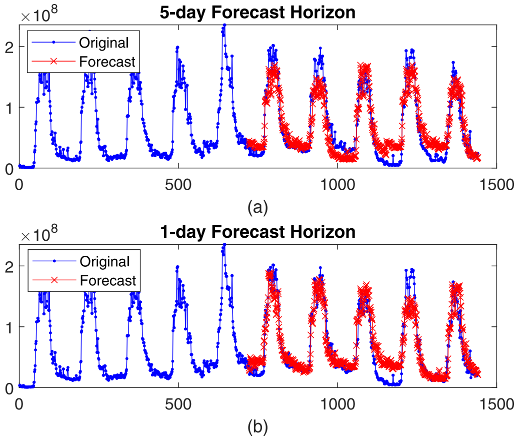

FIGURE 14. Forecasting with GRU Algorithm for filtered aggregated time series of total 470 APs.

TABLE 12. Comparison for the aggregated time series of representative APs of high group for 5-day FH.

	RMSE	MAE	NRMSE	R^2 score
Holt-Winters	$\mathbf{7.70 \times 10^8}$	3.74×10^8	**0.0781**	**0.6360**
SARIMA	8.24×10^8	$\mathbf{3.68 \times 10^8}$	0.0827	0.5826
LSTM	7.94×10^8	3.88×10^8	0.0806	0.6122
GRU	8.28×10^8	4.21×10^8	0.0840	0.5782
CNN	8.29×10^8	4.04×10^8	0.0842	0.5773

TABLE 13. Comparison for the aggregated time series of representative APs of high group with for 1-day FH.

	RMSE	MAE	NRMSE	R^2 score
Holt-Winters	8.06×10^8	4.54×10^8	0.0818	0.6007
SARIMA	$\mathbf{7.95 \times 10^8}$	$\mathbf{3.56 \times 10^8}$	**0.0807**	**0.6116**
LSTM	8.43×10^8	4.46×10^8	0.0855	0.5636
GRU	8.23×10^8	3.67×10^8	0.0835	0.5836
CNN	9.00×10^8	4.51×10^8	0.0914	0.5019

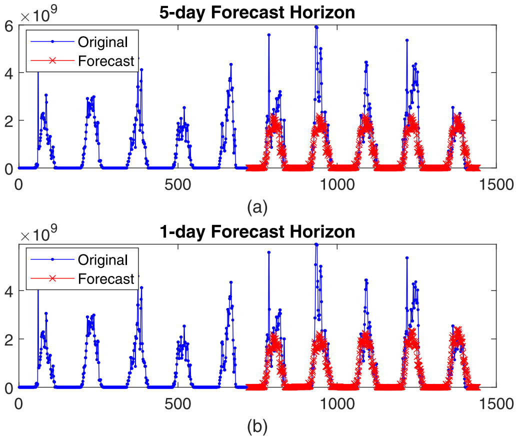

FIGURE 15. Forecasting with SARIMA Algorithm for filtered aggregated time series of representative APs of High group.

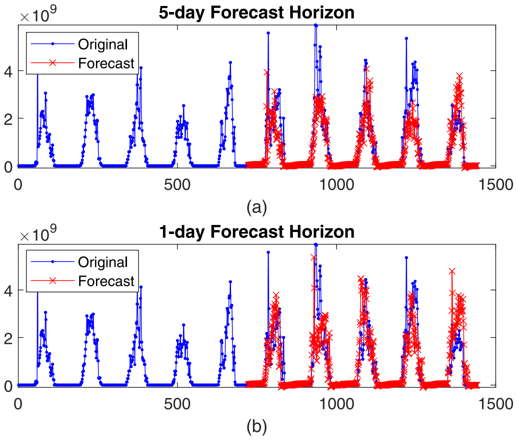

FIGURE 16. Forecasting with LSTM Algorithm for filtered aggregated time series of representative APs of High group.

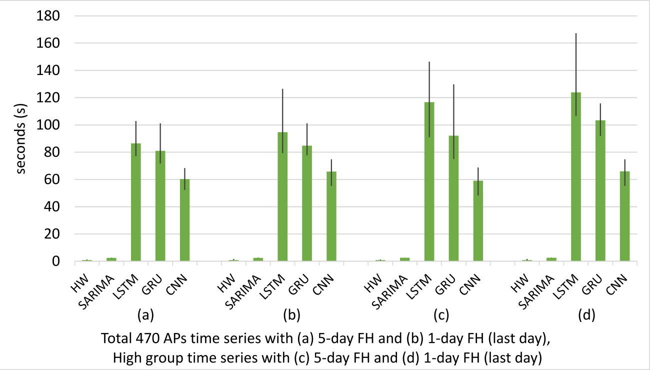

FIGURE 17. The computational time comparison of both statistical and machine learning methods for different scenarios.

Moreover, the forecasting performances of SARIMA and LSTM methods tested on filtered aggregated time series of representative APs of High group with both scenarios are shown in Fig. 15 and Fig. 16, respectively. According to Table 12, machine learning methods, such as LSTM, GRU and CNN, have lower performance compared to statistical methods since they are sensitive to the unusual traffic fluctuations as shown in Fig. 16(a). Holt-Winters and SARIMA give the good results by forecasting only the main load of data assuming the unusual fluctuations as outliers as shown in Fig. 15(a). For 1-day FH, machine learning methods failed to forecast the fluctuations in daily data pattern same as before which can be seen in Fig. 16(b) for LSTM. On the other hand, Holt-Winters started trying to forecast the unusual fluctuations based on daily updated new training dataset so that it also has lower performance. Only SARIMA which considers the unusual fluctuations as outliers even in the daily updating case has the best performance for filtered aggregated time series of representative APs of High group as shown in Fig. 15(b). Although LSTM gives better results than GRU in most of the cases, performance difference between LSTM and GRU is insignificant while GRU has the advantage of being simpler and faster than LSTM which can be seen in time complexity results, specifically in Fig. 17 of Section VII-B. In addition, CNN does not perform as well as other methods for temporal forecasting of our time series. However, it has the advantage of being able to handle spatial dependencies of the time series of APs which is very useful in spatio-temporal forecasting done in Section VII-C.

The accuracy differences between two scenarios are also significant in filtered aggregated time series of total 470 APs. Every forecasting method gives the better accuracy values in all metrics for 1-day FH compared to 5-day FH. This indicates that considering daily traffic changes and updating training data help to increase the performances of the forecasting algorithms, as an example, difference between two scenarios of Holt-Winters forecasting method can be seen in Fig. 13. However, for filtered aggregated time series of representative APs of High group, all of the forecasting algorithms, except SARIMA, get worse performances in 1-day FH than in 5-day FH due to their sensitivity to outliers in the daily updated training dataset. Only with SARIMA, the daily updating training data helps to increase performance of the forecasting algorithms.

TABLE 14. Comparison of training time complexity of different models.

Holt-Winters [59]	$O(N_{tr})$
SARIMA [60]	$O(m^3 N_{tr})$
LSTM [61], [62]	$O((\sum_{j=1}^{L} 4N_{ij}H_j+4H_j^2+3H_j+N_{oj}H_j)T_{tr})$
GRU	$O((\sum_{j=1}^{L} 3N_{ij}H_j+3H_j^2+2H_j+N_{oj}H_j)T_{tr})$
CNN [63]	$O(((\sum_{j=1}^{L} M_j N_j m_j n_j F_j)/F_p)T_{tr})$

B. COMPUTATIONAL COMPLEXITY OF TEMPORAL FORECASTING METHODS

In addition to temporal forecasting performance of the models, computational complexity is also considered as an important factor for model comparison. Computational complexity of a time series forecasting model is mainly based on the number of input variables and the hyperparameters determining the complexity of the model [8]. In [24] and [25], the number of trainable parameters and the Run-time Per-Epoch (RTPE) based on ten-fold cross-validation reporting both mean and variance of the performance are presented as computational complexity of the deep learning models. However, the nature of the statistical and machine learning model algorithms are different, such as RTPE can only be used for machine learning models, and hyperparameters can also vary with different time series even for the same method. For these reasons, references [7] and [58] determined computational complexity by presenting running time of the forecasting models as time complexity (TC) in seconds.

Therefore, we presented TC of different methods in two different approaches: a) empirical measurement for a complete running time of a model including training and forecasting processes, and b) theoretical expression of training TC of a model using big-O notation. For empirical TC measurement, we defined the computational complexity of our forecasting models as mean running time needed to train a given model and forecast the desired *N* FH of 10 iterations, which resemble the range of iteration for averaging computational TC in [25]. Empirical time complexities of the models are estimated on the same hardware architecture with following specifications: Intel Core i5-8250U CPU @1.6 GHz, 8.00 GHz RAM, x64 based processor in the same load condition without any background processing. The time complexities of both statistical and machine learning models are presented in Fig. 17.

The theoretical expressions of training (data fitting) TC for our models, according to [59]–[63], are presented in Table 14, where N_{tr} is the number of training samples, m is the order of SARIMA such as $m = \max(p, q+1, P, Q+1)$, N_{ij} is the number of input samples of j^{th} layer, H_j is the number of hidden units of j^{th} layer, N_{oj} is the number of output samples of j^{th} layer, T_{tr} is the number of time steps to train all of the training samples, L is the number of layers, input samples size $(M \times N)$ of CNN, F_j is the number of filters with filter size $(m \times n)$ of j^{th} layer and F_p is the filter size of max pooling layer in CNN. We assumed the theoretical TC of GRU as in Table 14 based on the theoretical expression of LSTM according to equations (4), (5), (6) and [61]. Moreover, theoretical TC for CNN model with one input channel [63] is also expressed in Table 14.

As expected, the results in Fig. 17 show that LSTM has the highest TC in all scenarios. Holt-Winters and SARIMA have very low complexity compared to neural network-based models. For LSTM and GRU, TC is increasing with different scenarios. This is due to two reasons: 1) the results in Fig. 17 for 1-day FH cases are for the last FI (last day) prediction which has updated training dataset with higher number of samples, and 2) the optimal number of hidden units for LSTM/GRU are higher for filtered aggregated time series of High group. However, there is not much difference in TC of CNN for both types of time series since we are using the same hyperparameters in all cases except having more samples in training dataset for the last FI of 1-day FH cases. In Table 14, the theoretical time complexities of LSTM and GRU are also obviously higher than HW and SARIMA. Moreover, despite of having similar architectures, the theoretical TC of LSTM is higher than GRU since GRU has less gating units and it is simpler than LSTM. In general, the theoretical TC of CNN mainly depends on the input samples size and filter size. Hence, with the hyperparameter values of CNN used in our work, its theoretical TC is lower than LSTM and GRU.

C. PERFORMANCE COMPARISON FOR SPATIO-TEMPORAL FORECASTING METHODS

A target AP with high utilization from a focus area has the significant spatial correlations to its neighboring APs according to spatial analysis results and is included in High group so that it has the similar pattern of filtered aggregated time series of representative APs of High group. Hence, we only focused on spatio-temporal forecasting with 1-day FH scenario in which machine learning methods did not outperform the statistical methods in temporal forecasting. Before training the model, we needed to prepare the input data into the grids for 2D-CNN to be able to extract the spatial dependencies. As in [23], we prepared a data patch in the form of (2x4) grids for total 8 APs which has the same placements as their actual locations on map. One data patch for one time instant is applied for spatial feature extraction, then, the model is trained and updated in the same way as in 1-day FH scenario of temporal forecasting.

It can be seen from Table 15 that one of the spatio-temporal forecasting methods considering spatial dependencies of its neighboring APs, CNN-LSTM, has the highest performance and the highest TC for forecasting traffic usage of a target AP. The first 4 methods are only temporal forecasting without considering spatial dependencies for filtered time series of a target APs with no aggregation of different APs. CNN is excluded in this case due to its low performance for temporal forecasting. However, when CNN is applied to extract spatial features of the neighboring APs and combined with LSTM or GRU, the combined spatio-temporal models

TABLE 15. Comparison for the target AP of study and conference lounge for 1-day FH.

	RMSE	MAE	NRMSE	R^2 score	TC(s)
Holt-Winters	1.50×10^8	1.07×10^8	0.0606	0.6630	0.921
SARIMA	1.60×10^8	1.01×10^8	0.0635	0.6133	2.432
LSTM	1.80×10^8	1.10×10^8	0.0716	0.5094	95.506
GRU	1.70×10^8	1.02×10^8	0.0704	0.5394	86.024
CNN-LSTM	$\mathbf{1.36\times10^8}$	$\mathbf{0.90\times10^8}$	**0.0542**	**0.7207**	299.645
CNN-GRU	1.38×10^8	0.92×10^8	0.05481	0.7149	261.63

outperformed all of the temporal forecasting models. On the other hand, GRU outperformed LSTM in temporal forecasting for 1-day FH scenario as in Table 13 while CNN-LSTM outperformed CNN-GRU in spatio-temporal forecasting for 1-day FH scenario. However, CNN-GRU has the advantage of lower computation complexity than CNN-LSTM. The empirical TC of the models for 1-day FH (last day) scenario are also presented in Table 15.

VIII. CONCLUSION AND FUTURE WORK

The analysis and forecasting results from our work can be helpful to perform traffic usage forecasting of an individual AP (or a group of APs) and of the entire network at a resource controller for network management and proactive resource allocation in an enterprise network. In this paper, we have used real network traffic data of an enterprise wireless network comprising 470 APs. The collected dataset is separated and classified into different groups. In temporal analysis, AC tell that only High and Low groups of weekdays data have long-term predictive pattern and CC show that only High and Low groups have strong similarities within and across the groups. Our temporal and spatial analysis tell us the different behaviors and characteristics in the real traffic utilization of APs from an enterprise wireless network.

We compare five temporal forecasting methods with two different scenarios, which are 5-day FH and 1-day FH. According to the results explained in Section VII-A, Holt-Winters is the best for smoother series in short-term forecasting and good for the series with unusual fluctuations in long-term forecasting. SARIMA works well for spiky series in short-term forecasting due to its insensitivity of outliers from updated training dataset. LSTM is good to forecast smoother series which do not have much noise (fluctuations) for long-term FH since LSTM can keep the important data and forget the unnecessary data from the long-term training data [38]. GRU has the better performance than LSTM in short-term forecasting of spiky series with advantage of having low computational complexity. CNN is better than SARIMA in long-term forecasting of smoother series. However, CNN would not be the good option to use for temporal forecasting wireless traffic time series compared to the other machine learning methods. The performance of different time series temporal forecasting methods tested on the real network data explains that there is no universally best temporal forecasting method for traffic time series in an enterprise wireless network. We also examine two spatio-temporal forecasting methods and compare with four temporal forecasting methods for a target AP. The performance of spatio-temporal forecasting methods explains that considering spatial dependencies of the neighboring APs helps to improve the forecasting performance of a single AP.

As the future direction of our work, mutual information (MI) which requires calculation of joint probability density function (PDF) can be used for time series data temporal analysis instead of correlation functions. Although correlation functions are commonly used for wireless traffic time series [11], [13], MI is stated as a better quantity to measure the dependence between two quantities since the correlation function measures only linear dependence while MI measures the general dependence [64]. On the other hand, despite being able to describe both linear and nonlinear dependence of the data, MI is difficult to calculate since it is still challenging to compute joint PDF of the data unless they are jointly normal and results are prone to under/over-estimation bias with no clear interpretation [65]. However, utilizing MI instead of correlation functions would lead to a more comprehensive analysis.

REFERENCES

[1] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, and X. Du, "From IoT to 5G I-IoT: The next generation IoT-based intelligent algorithms and 5G technologies," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 114–120, Oct. 2018.

[2] *System Architecture for the 5G System; Stage 2 (Release 15), Version 2.0.1*, 3GPP Standard TS 23.501, 2017.

[3] *Study of Enablers for Network Automation for 5G (Release 16), Version 16.0.0*, 3GPP Standard TS 23.791, 2018.

[4] *Meraki Cloud Controller Product Manual*, Cisco, San Jose, CA, USA, Mar. 2011.

[5] M. Pawlikowski and A. Chorowska, "Weighted ensemble of statistical models," *Int. J. Forecasting*, vol. 36, no. 1, pp. 93–97, Jan. 2019.

[6] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimization for mobile networks toward 5G," *IEEE Netw.*, vol. 30, no. 1, pp. 44–51, Jan./Feb. 2016.

[7] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLoS ONE*, vol. 13, no. 3, Mar. 2018, Art. no. e0194889.

[8] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econ. Rev.*, vol. 29, pp. 594–621, Aug. 2010.

[9] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656–1659, Aug. 2018.

[10] A. Hanbanchong and K. Piromsopa, "SARIMA based network bandwidth anomaly detection," in *Proc. IEEE 9th Int. Conf. Comput. Sci. Softw. Eng. (JCSSE)*, May 2012, pp. 104–108.

[11] K. Mirylenka, V. Christophides, T. Palpanas, I. Pefkianakis, and M. May, "Characterizing home device usage from wireless traffic time series," in *Proc. 19th Int. Conf. Extending Database Technol. (EDBT)*, Jun. 2016, pp. 539–550.

[12] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*. New York, NY, USA: Springer, Apr. 2017.

[13] D. Miao, X. Qin, and W. Wang, "The periodic data traffic modeling based on multiplicative seasonal ARIMA model," in *Proc. 6th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2014, pp. 1–5.

[14] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2018, pp. 231–240.

[15] J. Wang *et al.*, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[16] X. Wang *et al.*, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2190–2202, Oct. 2018.

[17] D. Tikunov and T. Nishimura, "Traffic prediction for mobile network using Holt–Winters exponential smoothing," in *Proc. IEEE 15th Int. Conf. Softw. Telecommun. Comput. Netw.*, 2007, pp. 1–5.

[18] Q. Duan, X. Wei, Y. Gao, and F. Zhou, "Base station traffic prediction based on STL-LSTM networks," in *Proc. IEEE 24th Asia–Pac. Conf. Commun. (APCC)*, 2018, pp. 407–412.

[19] A. Lazaris and V. K. Prasanna, "Deep learning models for aggregated network traffic prediction," in *Proc. IEEE 15th Int. Conf. Netw. Service Manag. (CNSM)*, 2019, pp. 1–5.

[20] N. Ramakrishnan and T. Soni, "Network traffic prediction using recurrent neural networks," in *Proc. IEEE 17th Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2018, pp. 187–193.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," Dec. 2014. [Online]. Available: arXiv:1412.3555.

[22] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," Mar. 2017. [Online]. Available: arxiv:1703.04691.

[23] C. W. Huang, C. T. Chiang, and Q. Li, "A study of deep learning networks on mobile traffic forecasting," in *Proc. IEEE 28th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.

[24] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 445–458, Jun. 2019.

[25] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, "MIMETIC: Mobile encrypted traffic classification using multimodal deep learning," *Comput. Netw.*, vol. 165, Oct. 2019, Art. no. 106944.

[26] C. Zhang, X. Ouyang, and P. Patras, "ZipNet-GAN: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, 2017, pp. 363–375.

[27] B. Krithikaivasan, Y. Zeng, K. Deka, and D. Medhi, "ARCH-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 683–696, Jun. 2007.

[28] M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics Volume 3 of Kendall's Advanced Theory of Statistics*. Hoboken, NJ, USA: Wiley, Dec. 1945.

[29] I. Alawe, A. Ksentini, Y. Hadjadj, and P. Bertin, "Improving traffic forecasting for 5G core network scalability: A machine learning approach," *IEEE Netw.*, vol. 32, no. 6, pp. 42–49, Nov. 2018.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

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

[32] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[33] S. Wheelwright, S. Makridakis, and R. J. Hyndman, *Forecasting: Methods and Applications Volume 2 of Business Forecasting*. Hoboken, NJ, USA: Wiley, 1998.

[34] J. Shi, G. He, and X. Liu, "Anomaly detection for key performance indicators through machine learning," in *Proc. IEEE Int. Conf. Netw. Infrastructure Digit. Content (IC-NIDC)*, Aug. 2018, pp. 1–5.

[35] F. F. Nobre, A. B. S. Monteiro, P. R. Telles, and G. D. Williamson, "Dynamic linear model and SARIMA: A comparison of their forecasting performance in epidemiology," *Stat. Med.*, vol. 20, pp. 3051–3069, Oct. 2001.

[36] R. Nau, *ARIMA Models for Time Series Forecasting: Slides on Seasonal and Nonseasonal ARIMA Models*, Duke Univ., Durham, NC, USA, 2014. [Online]. Available: https://people.duke.edu/∼rnau/Slides_on_ARIMA_models–Robert_Nau.pdf.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Dec. 1997.

[38] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2017.

[39] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2014, pp. 2342–2350.

[40] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018. [Online]. Available: arXiv:abs/1801.02143.

[41] J. Brownlee, *Deep Learning for Time Series Forecasting: Predict the Future With MLPs, CNNs and LSTMs in Python*, Mach. Learn. Mastery, Melbourne, VIC, Australia, 2018.

[42] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice Volume 2 of Business & Economics*. London, U.K.: OTexts, May 2018.

[43] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[44] R. Dey and F. M. Salemt, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, 2017, pp. 1597–1600.

[45] S. Savarese, A. Sadeghian, K. Fang, D. Xu, F. Xia, and J. Gao, *Computer Vision, From 3D Reconstruction to Recognition*, Standford Univ., Standford, CA, USA, 2018. [Online]. Available: https://https://web.stanford.edu/class/cs231a/lectures/intro_cnn.pdf

[46] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Appl. Sci.*, vol. 10, no. 5, p. 1897, 2020.

[47] M. V. Shcherbakov, A. Brebels, N. L. Shcherbakova, A. P. Tyukov, T. A. Janovsky, and V. A. Kamaev, "A survey of forecast error measures," *World Appl. Sci. J.*, vol. 24, pp. 171–176, Jan. 2013.

[48] S. P. Washington, M. G. Karlaftis, and F. Mannering, *Statistical and Econometric Methods for Transportation Data Analysis*. London, U.K.: Chapman and Hall, Dec. 2010.

[49] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[50] D. Datla, A. M. Wyglinski, and G. J. Minden, "A statistical approach to spectrum measurement processing," *Inf. Telecommun. Technol.*, to be published.

[51] D. C. Montgomery, C. L. Jennings, and M. Kulahci, "An introduction to time series analysis and forecasting," in *Wiley Series in Probability and Statistics*, vol. 2. Hoboken, NJ, USA: Wiley, Mar. 2008.

[52] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *J. Econometrics*, vol. 54, pp. 159–178, Jan. 1992.

[53] J. Davidson, "Establishing conditions for the functional central limit theorem in nonlinear and semiparametric time series processes," *J. Econometrics*, vol. 106, pp. 243–269, Feb. 2002.

[54] N. Damodar *et al.*, *Basic Econometrics Volume 2 of Econometrics*. New York, NY, USA: Mc-Graw Hill, 2004.

[55] J. Yang, Y. Qiao, X. Zhang, H. He, F. Liu, and G. Cheng, "Characterizing user behavior in mobile Internet," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 95–106, Mar. 2015.

[56] Y. Jin *et al.*, "Characterizing data usage patterns in a large cellular network," in *Proc. ACM SIGCOMM Workshop Cellular Netw. Oper. Challenges Future Design*, Aug. 2012, pp. 7–12.

[57] J. W. Taylor, "Triple seasonal methods for short-term electricity demand forecasting," *Eur. J. Oper. Res.*, vol. 204, no. 1, pp. 139–152, 2010.

[58] A. Q. Munir and R. Wardoyo, "Comparison analysis of time series data algorithm complexity for forecasting of dengue fever occurrences," *Int. J. Adv. Res. Comput. Sci.*, vol. 6, no. 7, pp. 1–7, 2015.

[59] M. V. de Assis, L. F. Carvalho, J. J. Rodrigues, and M. L. Proença, "Holt-winters statistical forecasting and ACO metaheuristic for traffic characterization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2013, pp. 2524–2528.

[60] J. Lu, F. Valois, M. Dohler, and M.-Y. Wu, "Optimized data aggregation in WSNs using adaptive ARMA," in *Proc. IEEE 4th Int. Conf. Sensor Technol. Appl.*, 2010, pp. 115–120.

[61] R. Karim. (Dec. 2018). *Animated RNN, LSTM and GRU*. [Online]. Available: http://towardsdatascience.com/.

[62] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," Feb. 2014. [Online]. Available: arXiv:abs/1402.1128.

[63] P. Maji and R. Mullins, "On the reduction of computational complexity of deep convolutional neural networks," *Entropy*, vol. 20, no. 4, p. 305, 2018.

[64] W. Li, "Mutual information functions versus correlation functions," *J. Stat. Phys.*, vol. 60, nos. 5–6, pp. 823–837, 1990.

[65] R. Smith, "A mutual information approach to calculating nonlinearity," *Statistics*, vol. 4, no. 1, pp. 291–303, 2015.

SU P. SONE received the B.E. and M.E. degrees in telecommunications engineering from the Asian Institute of Technology, Bangkok, Thailand, in 2016 and 2018. She is currently pursuing the Ph.D. degree and working as a Full-Time Researcher with the Centre for Wireless Communications, University of Oulu, Finland. She worked as a Research Assistant with BU-CROCCS Lab, Bangkok University, Thailand, from 2015 to 2016. Her research interests include traffic modeling in physical layer of wireless communications. She was an awardee of the Hiromichi Seya Prize and the Wireless Personal Multimedia Communications Prize in 2018.

JANNE J. LEHTOMÄKI (Member, IEEE) received the Doctorate degree from the University of Oulu, Finland, in 2005, where he is currently an Adjunct Professor with the Centre for Wireless Communications. He spent the fall 2013 semester with the Georgia Tech, Atlanta, GA, USA, as a Visiting Scholar. He is currently focusing on spectrum measurements and terahertz band wireless communications. He coauthored paper received the Best Paper Award in IEEE WCNC 2012. He has served as a Guest Associate Editor for the *IEICE Transactions on Communications* Special Section from February 2014 and July 2017, and as a Managing Guest Editor for *NANO Communication Networks* Special Issue in June 2016. He was a General Co-Chair of IEEE WCNC 2017 International Workshop on Smart Spectrum, a TPC Co-Chair for IEEE WCNC 2015 and 2016 International Workshop on Smart Spectrum, a Publicity/Publications Co-Chair for ACM NANOCOM 2015, 2016, and 2017. He is an Editorial Board Member of *Physical Communication*.

ZAHEER KHAN (Member, IEEE) received the M.Sc. degree in electrical engineering from the University of Borås, Sweden, in 2007, and the Dr.Sc. degree in electrical engineering from the University of Oulu, Finland, in 2011, where he is currently an Adjunct Professor. He has also worked as a Tenure Track Lecturer with the University of Liverpool, U.K., from 2016 to 2017, and as a Research Fellow/Principal Investigator with the University of Oulu from 2011 to 2016. His research interests include implementation of advanced signal processing and wireless communications algorithms on Xilinx FPGAs and Zynq System-on-Chip boards, application of game theory to model distributed wireless networks, prototyping access protocols for wireless networks, IoT location tracking systems, cognitive and cooperative communications, and wireless signal design. He was a recipient of the Marie Curie Fellowship from 2007 to 2008.