Predicting Congestion Level in Wireless Networks Using an Integrated Approach of Supervised and Unsupervised Learning

Alisha Thapaliya
Computer Science and Engineering
University of Nevada, Reno
Reno, Nevada 89557
Email: alishathapalia@nevada.unr.edu

James Schnebly
Computer Science and Engineering
University of Nevada, Reno
Reno, Nevada 89557
Email: jschnebly@nevada.unr.edu

Shamik Sengupta
Computer Science and Engineering
University of Nevada, Reno
Reno, Nevada 89557
Email:ssengupta@unr.edu

***Abstract*—The usage data from user devices can be analyzed to answer a number of possible questions in regards to congestion, access point (AP) load balancing, user mobility trends and efficient channel allocation. In this paper, we attempt to identify Wi-Fi usage trends in a dynamic environment and use it to further predict the congestion in various locations. To accomplish this, we use various supervised learning algorithms to find the existing patterns in spectrum usage inside University of Nevada, Reno. Using these patterns, we predict the values for certain key attributes that directly correlate to the congestion status of any location. Finally, we apply unsupervised learning algorithms to these predicted data instances to cluster them into different groups. Each group will determine the level of congestion for any building at any time of any day. This way, we will be able to ascertain whether or not any place at any time in the future might require additional resources, which can be utilized from places with low congestion rate, to be able to deliver wireless services efficiently.**

I. Introduction

We need continuous access to wireless networks with greater capacity, performance, and throughput because there is a growing number of clients in wireless networks with more demand for a high quality internet connection [1]. This widely spreading need for wireless services calls for the deployment of additional resources when demand reaches its peak. This is highly probable in scenarios where it is difficult to identify the Wi-Fi usage pattern, mostly in dynamic public places where the spectrum usage is both time and space dependent. With the disproportionate and time varying usage of wireless services, studies show that some of these Wi-Fi resources are under-utilized in some locations while they are in high demand at other locations [2]. Therefore, instead of adding new resources every time the congestion level increases, it would be more efficient to utilize resources from areas where the congestion level is relatively low. This leads to an optimized resource allocation problem which can be highly useful in a modern day setting where it is impossible to continuously deploy wireless APs as the number of wireless users vary.

According to the Cisco report, traffic from wireless and mobile devices will exceed traffic from wired devices by 2018. It states that in 2018, Wi-Fi and mobile devices will account for 61 percent of IP traffic [1]. Higher traffic equates to more data in terms of traffic load, number of users, quality of wireless signal, etc., which can be utilized for network management, optimization and fine-tuning [3]. Although there is a myriad of information that we can gather via the wireless activity of clients, it is highly crucial to this study that we select the appropriate data types relevant to our research [5]. For this paper, we have chosen four data attributes: number of clients, throughput, frame retry rate and frame error rate. These attributes help us answer the questions: *1)Is there a pattern to how the users take advantage of wireless networks? 2) When does the client activity increase significantly? 3) How can we relate these 4 attributes to congestion?*

In this paper, we attempt to determine the congestion level in various locations inside the University of Nevada, Reno (UNR) based on a certain day and time. Based on our previous work [4], there is a possibility of trend in spectrum usage at UNR. In this study, we are investigating patterns in usage data in terms of number of clients, throughput, frame error rate and frame retry rate. Furthermore, we are applying machine learning algorithms to predict the future values of these attributes based on the pattern. For instance, is the change in throughput values for one Tuesday similar to the change in throughput values for another Tuesday throughout the entire day? The existence of these trends allows us to apply supervised learning algorithms to predict the values [6] of the four attributes given the date and time. However, we still do not know what these predictions mean and how they can be accredited to congestion. That's when the unsupervised learning comes into play, with the main idea to draw inferences from data sets consisting of input data without labeled responses and find hidden groupings in data [7]. Using this mechanism, we can efficiently classify different data items with the four attributes into different groups, where each group defines a certain level of congestion. We use the ideology that if there are more clients connected to an access point then the probability of a higher level of congestion is greater.

We first pick the day and time in the future to determine its

978-1-5386-5541-2/18/$31.00

congestion status, after which we predict the number of clients, throughput, frame retry rate and frame error rate for those inputs. Finally we feed these predicted values to a clustering mechanism which will identify whether these values refer to a high level, medium level or a low level congestion.

To gain a better understanding of how users are taking advantage of wireless networks, we examined the user activity data over a period of 5 weeks inside UNR. We polled the access points using SNMP commands [12] to extract the required information in terms of clients, throughput, packet drop and retry rates. The data was collected every 5 minutes so as not to overload the server with queries. The data consists of the following attributes: Timestamp, Location, Clients, Throughput, Frame Retry Rate, Frame Error Rate. We preprocessed the data in a suitable format to be stored in and accessed from a database. Afterwards, we averaged the data instances to an hourly basis to get a cleaner representation, and its analysis led us to believe that there exists a trend in how the users use Wi-Fi services across different hours of a day. Following this trend, we implemented Support Vector Regression and Polynomial Regression to predict the values of Clients, Throughput, Frame Retry and Frame error in the future for a certain location, day and time. To acknowledge how congestion can be identified with these attributes, we created a clustering model using EM algorithm from the averaged data instances for the 5 weeks and with the 4 attributes. The model successfully grouped these data instances into 3 clusters, each cluster determining the congestion level- low, medium or high. Finally, integrating the whole idea, the predicted values of the 4 attributes for a certain location, day and time given by our supervised learning algorithms were fed to the clustering model, the result specifying the congestion level corresponding to those inputs.

II. Methodology

A. Data Collection

The results presented in this paper are based on the analysis of spectrum usage data collected across a section of UNR. UNR spans across an area of 290 acres and comprises of more than 80 buildings. There are two central controllers that manage a total of approximately 1275 active access points at any point of time throughout the university. Many of these APs are deployed in outdoor spaces as well.

To facilitate the data collection process, we have used SNMP network protocol that allows different devices on a network to share information with one another in a consistent and reliable manner [9]. We polled the APs with different queries (snmpwalk commands) via the controllers that manage these APs. We poll every 5 minutes to obtain information reasonably frequently, within the limits of the computation and bandwidth available on our two polling workstations [11].

At any point of time, separate queries were made to these APs explicitly asking for each of following information:

- number of clients connected to the AP
- location of the AP
- mac address of that AP
- AP bssid for 2.4 GHz and 5 GHz radio
- channels operating in that AP
- throughput of the AP
- frame retry rate of the AP (The number of retry packets as a percentage of the total packets transmitted and received by this AP)
- frame error rate of the AP (The number of error packets as a percentage of the total packets received on this AP)

Due to time constraints, we have limited our study to a geographical section of UNR consisting of 5 co-located buildings: JCSU, MIKC, ARF, WFC, and NJC. The data expands over a period of 5 weeks from December 1, 2017 to January 5, 2018. There are a total of 233 APs within these 5 buildings and the outside premises from which we have extracted all the information relevant to this research.

B. Dataset Creation and Pre-processing

Once the data is gathered using the process described in section II-A, we create individual CSV files that hold average clients per AP_MAC address, average throughput per AP_MAC, average frame retry per AP_MAC, and average frame error per AP_MAC. We compute and predict averages instead of totals in order to avoid any bias presented by one hour having more data than another. If one hour has more data than the others, the totals could potentially show high congestion when actually there is just a difference in the amount of data points resulting in higher totals. Each individual CSV corresponds to a building on campus as well as the day of the week. Using python2.7 and the MySQLdb library, we can write SQL queries and manipulate the results with python before exporting the data to a CSV. Our database contains two tables, namely DATA_TABLE and LOCATION_TABLE. DATA_TABLE contains the TIMESTAMP, AP_MAC, CLIENT, THROUGHPUT, FRAME_RETRY, and FRAME_ERROR. LOCATION_TABLE contains AP_MAC and the corresponding location code AP_LOCATION.

Since we want to filter our queries based on location, an inner join on AP_MAC is executed. From the inner join aliased as S we return the per record averages of CLIENTS, THROUGHPUT, FRAME_RETRY, and FRAME_ERROR. This is done 24 times via for loop in order to get averages from every hour (00-23). This process is repeated for every date in the database. To compute averages we use the built-in SQL aggregate AVG. Since we want to compute averages per MAC and not per record, we multiply the result returned by AVG(clients) by four since there are four BSSIDs per MAC. This results in an average number of clients per MAC address. For the three other attributes, we take the result returned by AVG(attribute) and multiply it by two since there are duplicate values for each BSSID in a channel. This gives us average throughput, frame retry, and frame error per MAC address every iteration of the for loop (every hour). From there we export them to a CSV before next iteration of the for loop.

This process of exporting data to a CSV is done for every date in the database inside five buildings on the UNR campus. The dates corresponding to the same day of the week (Monday - Friday) are aggregated into one CSV that holds all the data for a particular weekday and location. The CSV is named according to the location and day of the week. MIKCmonday.csv, for example, holds all Monday data at location MIKC.

C. Supervised Learning

To predict the average clients, throughput, frame retry and frame error per MAC address of an AP, we first plot each attribute as a function of time (hours) in order to identify trends before fitting an algorithm to the data. We found that the data points represented a normal distribution for all attributes. This normal distribution can be seen in figure 1a where the average number of clients is plotted on an hourly basis. Based on the trend in the charts we were able to identify peak congestion period of times ranging from about 10 am to 2 pm on average across all days and locations. The charts were created using the python3 library Matplotlib [20].

Supervised learning is a subsection of machine learning where the user feeds the model training data which contains labels. The model learns to classify or predict the labels based on the training data that is associated with it. Once training is complete, the model uses its prior knowledge of the training data to predict labels on new data. In this instance, we feed our model training data of Wi-Fi AP attributes (clients, throughput, frame retry and frame error) per hour, and it predicts the attribute at a given hour based on the training data.

After identifying the trends in the data we decided to fit the data with support vector regression (SVR) because it does not take into account the outliers in the dataset and fits to the bulk of the data. SVR creates a regression line by splitting the difference between the two closest data points. Using the Gaussian kernel, we can create nonlinear regression lines by mapping the data points to a higher dimension.

In addition to the SVR, we also fit a polynomial regression in the 2nd degree which takes into account outliers in the dataset. This was done in order to compare our results to the SVR. Lastly we create a weighted combination of both the SVR and the polynomial regression in hopes of finding an algorithm that allows the outliers some weight in the prediction but not so much as to throw off the entire prediction. The SVR and polynomial regression were created and run using the Scikit-learn python3 library [19].

To calculate the accuracy of prediction for each algorithm, Mean Squared Error (MSE) was calculated for all three algorithms every time a prediction was made. The formula for MSE is $MSE = 1/N \sum_{i=1}^{N} (f_i - y_i)^2$ where N is the number of data points, f_i is the predicted value, and y_i is the actual value [18]. On average, SVR had a lower MSE than the polynomial and weighted regressions.

Once we found SVR to be the best preforming regression algorithm for this data, we moved on to unsupervised learning to cluster the data points and define levels of AP congestion.

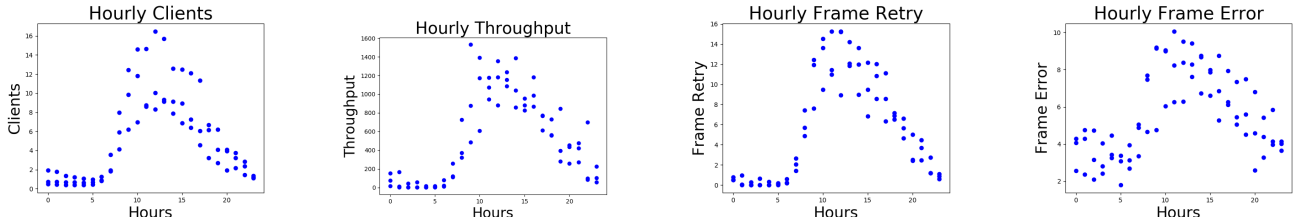
D. Unsupervised Learning

Clustering, an unsupervised learning technique, takes a set of points in n-dimensional space and finds coherent subsets. Each subset consists of points that are clustered together. The advantages of using clustering algorithms is it's ability to categorize data instances automatically and the ability to find groupings that we might not otherwise find [8].

After predicting the values of average clients, throughput, frame retry and frame error per AP MAC address for a specified date and time, we need to relate these values to a certain congestion level. However, there is no definitive way of assigning the predicted values to certain groups. We used EM clustering algorithm for this purpose. EM is a soft-clustering technique where it assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters [16]. Mixture models are a probabilistically-sound way to do soft clustering. We assume our data is sampled from K different sources (probability distributions). The expectation maximization (EM) algorithm allows us to discover the parameters of these distributions, and figure out which point comes from each source at the same time [21]. EM is a method to find the means and variances of a mixture of Gaussian distributions. We specifically chose EM because it allowed us to assign a certain level of probability for any data instance to fall into any cluster (congestion level). This is important because we are not making a hard determination of congestion for the predicted values, instead we are exploring the degree of possibility that those instances can be correlated to congestion . It is also useful when the range of values differs widely between dimensions [10] which is true in our case, as the range of values for throughput is very high as compared to the other attribute values.

Using the EM algorithm, we clustered the hourly averaged data instances per AP MAC address for the entire 5 weeks with just 4 attributes: Clients, Throughput, Frame Retry and Frame Error. There were a total of 863 instances, we chose to divide these data points into three groups, and the algorithm was successfully able to group them into 3 clean clusters. As we analyzed the clusters, we found that each of the clusters can be corroborated to a certain level of congestion. In one cluster, there were data items with very low values for all the attributes; we identified this cluster as low congestion cluster. In another cluster, we had data items with reasonable attribute values which we named as a medium congestion cluster, and finally, for the last cluster, the data instances had very high values for almost all attributes which we designated as a high congestion cluster.

We have used WEKA to perform EM clustering which is a collection of machine learning algorithms for data mining tasks. The software written in Java contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization [13]. The accuracy of EM is measured using log-likelihood. Since all the data points are assigned to their respective clusters on the basis of probability, the objective function is to maximize the probability of likelihood



(a) Clients (b) Throughput (c) Frame retry (d) Frame error

Fig. 1: Attributes plotted as a function of time (hours)

of the data instances belonging to their assigned clusters [15].

Let $D = x_1, x_2, ..., x_n$ be n observed data vectors. Let $Z = z_1, z_2, ..., z_n$ be n values of hidden variables (i.e. the cluster labels). Log-likelihood of observed data given model:

$$L(\theta) = logp(D|\theta) = log \sum_{Z} p(D, Z|\theta) \quad (1)$$

where p is the probability of a certain data vector x_i belonging to a cluster in Z, θ denotes parameters: mean and variance, and both θ and Z are unknown [17].

III. Results

We now present the future prediction by our machine learning models regarding whether a certain location in the future will be congested or not. We demonstrate how the data in this research is being fitted into the algorithms that we chose, what type of outputs are being received, and how they are being processed and analyzed.

A. SVR Prediction Model

In order to get a future prediction of congestion level, the first step was to predict the individual attributes such as average number of clients, throughput, frame retry, and frame error. We took the Support Vector Regression(SVR) from section III-C and created a basic user interface that allowed the user to enter input parameters such as day of week, building, and hour of day. Based on those input parameters, the program would query the SVR model and output a prediction of the aforementioned attributes of Wi-Fi congestion. The output of the program can be seen in table I. For building MIKC at 4 pm, the SVR predicted 14 clients per AP, a throughout of 1742 per AP, a frame retry of 20 per AP, and a frame error of 1 per AP. It is important to note that these values are averages across all AP's in the MIKC building. The MSE for clients tells us the prediction differs from the actual values by about 2.45 clients on average. To find this average difference, we take the square root of the MSE for any particular attribute.

From here, the predicted values are sent to the unsupervised learning EM clustering algorithm to get the resulting predicted congestion level.

B. EM clustering

To perform EM clustering, we have a dataset with 863 instances and 4 attributes as shown in figure 2. The data is in .arff format which is standardized in WEKA. The 4 attributes

TABLE I: SVR Output Prediction (MSE in parenthesis)

Day	Hour	Location	Clients	Throughput	Frame retry	Frame error
Monday	16	MIKC	14 (6.03)	1742 (68078.43)	20 (6.69)	1 (0.04)
Thursday	05	JCSU	0 (0.49)	60 (7009.18)	7 (0.07)	0 (0.43)
Thursday	08	MIKC	5 (4.37)	515 (74291.53)	7 (10.98)	0 (0.75)

are: Clients, Througput, Frame retry and Frame error with the data type as 'numeric' which means real numbers. For each data object, the comma separated numbers denote values corresponding to the attributes in the attribute section. For instance, in line 9 of figure 2, 0 is the client number, 59 is the throughput value, 0 is the frame retry value and 2 is the frame error value. This data is obtained after averaging the 5-minute interval values for all these attributes in an hourly basis per AP MAC address. So, each of these data points refer to the number of clients, throughput, packet retry rate and packet drop rate in an AP for a certain hour of a certain day in a certain location.

```
1   @relation weka_hourly_data
2
3   @attribute Clients numeric
4   @attribute Throughput numeric
5   @attribute Frameretry numeric
6   @attribute Frameerror numeric
7
8   @data
9   0, 59, 0, 2
10  0, 4, 0, 2
11  1, 215, 3, 2
12  5, 440, 6, 3
13  9, 1507, 15, 3
14  9, 1432, 14, 3
15  9, 1244, 13, 4
```

Fig. 2: Data used for EM clustering

We then upload the .arff data file into WEKA and apply the EM clustering algorithm specifying the number of clusters (k) as 3. The default number of clusters generated by this algorithms is 4, but we found out that the clusters are more clearly separated and are of more value when we assign the value of k as 3. This algorithm groups data instances into clusters based on the maximum likelihood estimate of the parameters: mean and variance of each attribute, and assigns the instances into their respective clusters where they have the highest probability of belonging to.

Figure 3 shows the clustering output as performed by the

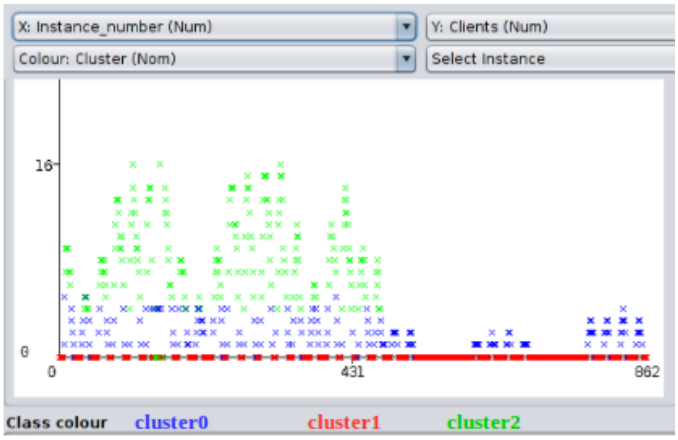


Fig. 3: The 3 clusters generated by the EM algorithm

EM algorithm. The three clusters are identified by colors blue, red and green with auto-generated names cluster0, cluster1 and cluster2 respectively. In this plot, the x-axis is the instance number and y-axis is the number of clients attribute. Due to space constraints, we have only presented one plot with the client attribute in the y-axis. However, the other plots with the rest of the attributes also displayed similar results.

TABLE II: Analysis of data in cluster2, cluster0 and cluster1

Cluster	Instance number	Clients	Throughput	Frame retry	Frame error
cluster2	84	13	1660	21	3
cluster2	85	13	1789	19	3
cluster2	86	13	1675	19	3
cluster0	493	2	258	2	2
cluster0	494	2	207	2	3
cluster0	495	2	258	2	3
cluster1	364	0	0	0	1
cluster1	366	0	4	0	1
cluster1	368	0	44	0	2

To understand how the clusters are formed, we select the different colored points in the plot and compare them. The results are shown in table II. Here, the three data points in cluster2 have very high values for all the attributes. Compared to that, the data points in cluster0 have less attribute values but are still significant. Lastly, the attribute values for all the data points are very less and almost negligible. We analyzed other points in all the clusters and obtained similar results. Based on this analysis, we labeled the data points in cluster2, cluster0 and cluster1 as highly congested, moderately congested and less congested respectively, with the cluster labels for cluster2 as high, cluster0 as medium and cluster1 as low in terms of congestion.

Figure 4 shows the accuracy of our EM clustering model using log-likelihood, the value being -5.56. Since log-likelihood is the logarithmic value of probability, and probability is always between 0 and 1, the log value is negative. There were 4 iterations done by the algorithm to conclude to this result.

```
Number of clusters: 3
Number of iterations performed: 4


                Cluster
Attribute            0        1        2
                  (0.25)     (0.5)    (0.25)
=============================================
Clients
  mean            2.0303        0    9.0304
  std. dev.       1.2759   0.0004    3.8736

Throughput
  mean          303.1015   8.3373 1349.9625
  std. dev.      197.599  15.0098  699.9825

Frameretry
  mean            2.5651   0.0009   15.0428
  std. dev.       1.8183   0.0302   13.1869

Frameerror
  mean            1.8243   1.0565    2.5733
  std. dev.       0.7212   0.5419    0.9126


Time taken to build model (full training data) : 0.04 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      211 ( 24%)
1      431 ( 50%)
2      221 ( 26%)


Log likelihood: -5.56656
```

Fig. 4: Evaluation of the EM clustering model

It also shows the mean and standard deviation values of each of the attributes for each cluster, which provides the range of attribute values within each cluster and gives us an intuitive idea of the level of congestion in these clusters based on that range.

C. Final output

Using the clustering model generated in section III-B, we predict the congestion level for the inputs used in section III-A based on the values of the 4 attributes predicted as shown in table I.

```
1   @relation weka_hourly_test_data
2
3   @attribute Clients numeric
4   @attribute Throughput numeric
5   @attribute Frameretry numeric
6   @attribute Frameerror numeric
7
8   @data
9   14, 1742, 201, 1
10  5, 515, 7, 0
11  0, 60, 0, 2
```

Fig. 5: Predicted data used in the EM clustering model

Figure 5 shows the data that we used as a test data for our clustering model which was generated using the prediction model. We want to determine which congestion level group each one of these instances belong to. As evident from figure 6, the first data instance belongs to cluster2 (high congestion) which is corresponding to MIKC, Monday at 4 p.m. The second data instance belongs to cluster0 (medium congestion) which is corresponding to MIKC, Thursday at 8 a.m. The third data instance belongs to cluster1 (low congestion) which is corresponding to JCSU, Thursday at 5 a.m.

In this way, we can predict the congestion level for any location, day and time in the future.

```
@relation weka_hourly_test_data_clustered

@attribute Instance_number numeric
@attribute Clients numeric
@attribute Throughput numeric
@attribute Frameretry numeric
@attribute Frameerror numeric
@attribute Cluster {cluster0,cluster1,cluster2}

@data
0,14,1742,201,1,cluster2
1,5,515,7,0,cluster0
2,0,60,0,2,cluster1
```

Fig. 6: Output of the clustering model showing the assigned cluster for each data instance

IV. Remarks

As briefly discussed in section I, we believe this integrated Wi-Fi congestion prediction model can be used to control an autonomous resource allocation program. Ideally, the supervised prediction model, which will be fed real time data in order to create a more accurate prediction, will predict hours or even days ahead of time and send those predictions to the clustering algorithm. When the clustering algorithm predicts the future level of congestion, a resource allocater can see that at any time t, building y will have low congestion while building z will have high congestion. The resource allocater can then take resources from building y (for the duration of time that it has low congestion) and assign it to building z in order to better the quality of signal in building z. This autonomous Wi-Fi resource allocater can then act as a load balancer for all buildings that has access to Wi-Fi data.

V. Conclusion

The first step in dealing with Wi-Fi network issues is identifying congestion, when and where it occurs. In this paper, we have accurately predicted the congestion level in certain locations for specified dates and times inside UNR, a public university equipped with thousands of access points and a lot more daily user base. We have chosen two specific buildings for our findings: MIKC, the library and JCSU, the student union, both with a lot of user activity. We have traced the wireless data from the users in these two buildings for 5 weeks and analyzed the trend in the spectrum usage for each day of the week. After realizing that there is in fact a pattern in the network usage for the same days of different weeks, we were able to successfully predict the values of 4 attributes that correlates to congestion: clients, throughput, frameretry and frameerror using SVR based on their historical values. These predictions are made for the chosen location, day of the week and time of the day. Now, all that was left was to affirm whether those inputs corroborated to congestion or not, based on the attribute values that were predicted. For this, we used an EM clustering model to segregate the original dataset, the training data into different clusters with each cluster defining a certain level of congestion: low, medium or high. The predicted attribute values of data instances for different location, day and time are then used as a test data for this model that identifies whether the data instance corresponds to a less, moderate or high congestion. We have integrated two machine learning approaches: supervised and unsupervised learning to determine the congestion level in a wireless network. We believe that this study can be scaled to bigger networks that exhibit similar trends. This study serves as a tool to system administrators who constantly monitor wireless networks. It helps them better optimize the resources and infrastructures and allows them to proactively avoid situations that might lead to a massive network load-balancing issue.

Acknowledgments

This research is supported by NSF Award #1723814.

References

[1] Kamiska-Chuchmaa, A. (2016). Performance analysis of access points of university wireless network. Rynek Energii, 1(122), 122-124.
[2] Song, C., and Zhang, Q. (2010, March). Intelligent dynamic spectrum access assisted by channel usage prediction. In INFOCOM IEEE Conference on Computer Communications Workshops, 2010 (pp. 1-6). IEEE.
[3] Redondi, A. E., Cesana, M., Weibel, D. M., and Fitzgerald, E. (2016, September). Understanding the WiFi usage of university students. In Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International (pp. 44-49). IEEE.
[4] Thapaliya, A., Sengupta, S., and Springer, J.(2018, February). Wi-Fi Spectrum Usage Analytics in University of Nevada, Reno. In International Instrumentation Measurement Technology Conference (I2MTC).
[5] Kim, M., Kotz, D., Kim, S. (2006, April). Extracting a mobility model from real user traces. In INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings (pp. 1-13). IEEE.
[6] Song, L., Kotz, D., Jain, R., He, X. (2004, March). Evaluating location predictors with extensive Wi-Fi mobility data. In INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies (Vol. 2, pp. 1414-1424). IEEE.
[7] Unsupervised Learning. (n.d.). Retrieved from https://www.mathworks.com/discovery/unsupervised-learning.html
[8] Tang, D., Baker, M. (2002). Analysis of a metropolitan-area wireless network. Wireless Networks, 8(2/3), 107-120.
[9] Vandepoele, D. (2016, July). Network Basics: What Is SNMP and How Does It Work? Retrieved from https://www.auvik.com/media/blog/network-basics-what-is-snmp/
[10] Dempster, A. P., Laird, N. M., Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological), 1-38.
[11] Kotz, D., Essien, K. (2005). Analysis of a campus-wide wireless network. Wireless Networks, 11(1-2), 115-133.
[12] Case, J. D., Fedor, M., Schoffstall, M. L., Davin, J. (1990). Simple network management protocol (SNMP) (No. RFC 1157).
[13] Weka 3: Data Mining Software in Java. (n.d.). Retrieved from https://www.cs.waikato.ac.nz/ml/weka/
[14] Sharma, N., Bajpai, A., Litoriya, M. R. (2012). Comparison the various clustering algorithms of weka tools. facilities, 4(7).
[15] Yang, M. S., Lai, C. Y., Lin, C. Y. (2012). A robust EM clustering algorithm for Gaussian mixture models. Pattern Recognition, 45(11), 3950-3961.
[16] http://weka.sourceforge.net/doc.dev/weka/clusterers/EM.html
[17] Retrieved from https://www2.cs.duke.edu/courses/fall07/cps271/EM.pdf
[18] Stephanie. (2013, November). Mean Squared Error: Definition and Example. Retrieved from http://www.statisticshowto.com/mean-squared-error/
[19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.
[20] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in science engineering, 9(3), 90-95.
[21] Lavrenko, V. (2014, January). EM algorithm: how it works. Retrieved from https://www.youtube.com/watch?v=REypj2sy$_5U$