Machine Learning for Channel-Based User Localization in Ad Hoc Systems

Karla Cosio, Mohit Malu, Ahmed Ewaisha

SenSIP REU, School of ECEE, Arizona State University

***Abstract* — Machine learning algorithms have been introduced for user scheduling to improve the performance of 5G systems with the increasing number of users. Currently, machine learning algorithms have been deployed in centralized systems to schedule users to obtain a wireless communication system capable of serving massive numbers of users. In this work, we rely on machine learning algorithms to predict channels to reduce the channel estimation overhead. Our ultimate goal is to improve the scheduling system. Our method primarily focuses on channel-based user localization.**

Index terms: machine learning, user localization, 5G, ad hoc systems, channel estimation

I. INTRODUCTION

Machine learning has been a focus for many researchers due to its ability to classify, predict, and model algorithmic and model deficit problems [6]. With the increasing number of users in centralized systems, machine learning algorithms have been used in scheduling where update-importance-based client scheduling is prioritized [2]. Liu et al., [1] focuses on improving the model convergence of the machine learning algorithm that performs user scheduling. Machine learning has been implemented for load balancing between users [3]. There was little attention in the literature to scheduling users that are directly communicating with each other.

The branch of machine learning that will be used for this research is supervised learning. Supervised learning determines a statistical relationship to find patterns between a known input and output [7]. Machine learning has not been used to schedule users in ad hoc networks. The logistic regression (LR) algorithm and support vector machine (SVM) algorithm are used primarily. Training a machine learning algorithm to schedule users to improve the throughput and latency in ad hoc systems will maximize the performance of applications pertaining to device-to-device (D2D), machine-to-machine (M2M), and vehicle-to-vehicle communications [4]. Moreover, revolutionize healthcare systems with the concept of sending patient's body data to the cloud or healthcare providers [5].

In this paper, we investigate using ML to predict channels to reduce the channel estimation overhead with the ultimate goal of improving the scheduling algorithm. We depend on channel-based user localization to carry out this prediction. To do any data transmission, the channels need to be estimated first. Reducing the amount of time it takes for the channel estimation phase to occur helps with serving more users as the time for data transmission increases especially when using massive multiple-input multiple-output (MIMO) systems. The method we will be performing is based on the relation that location affects the channel. Our contribution in this work will depend on this relation to reduce the channel estimation phase duration. It is critical that the channel is estimated accurately as this will be fed into the scheduling algorithm which schedules the optimal user.

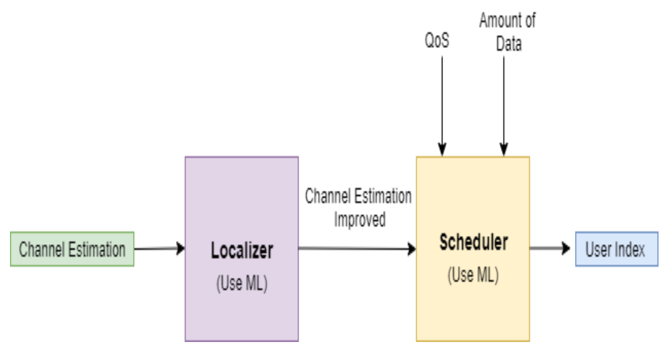

Figure 1: Block diagram of machine learning (ML) algorithm in scheduling users. Our ultimate goal is developing an efficient ML algorithm to schedule users in ad hoc systems.

Once our method of estimating the channels of users is efficient, the scheduling algorithm will decide which user to select based on multiple sources of information including channel estimates, quality of service (QoS), queues, etc. As a result, scheduling performance will be improved so more users will be served with higher throughput and stringent latency.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. Channel Estimation with OFDM System

For this work, we have an OFDM communication system where the observed symbol from subcarrier i is given as follows

$$y_i = x_i H_i + n_i \tag{1}$$

In (1), y_i and x_i refer to the received symbol and transmitted symbol for subcarrier i, respectively. "H_i" refers to the channel for a specific subcarrier, i, in the frequency domain. Determining an estimation of the channel is crucial to schedule users. Equation (1) is written in the time domain as follows

$$y(t) = \sum_n h_n(t) x(t) \tag{2}$$

In (2), $y(t)$ and $x(t)$ refer to the received symbol and transmitted symbol in the time domain. "$h_n(t)$" refers to the channel gain. The summation is accompanied by an index n that represents the path the signal can take as depicted in Figure 2. We will use "location" and "position" interchangeably which represent the environment. The received signal is a superposition of all the paths. Thus, when either the transmitter or receiver change location, the channel is affected in the time and, hence, in the frequency domain. This makes the channel estimation a unique "signature" of the location. This helps reduce the channel estimation duration as will be explained in Section III.

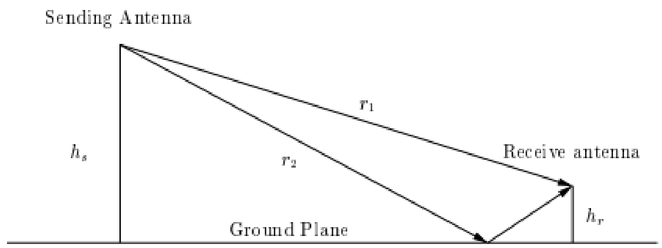

Figure 2: Illustration of multiple paths received at the receiver. [9]

B. USRP Radio

We used the USRP-B200 kits in this work for real-time signal transmission. One USRP radio was the transmitter and the other USRP radio was the receiver.

The GNU Radio is a software development toolkit that uses block diagrams that correspond to radio settings. In our case, a transmitter and receiver block diagram were created and used to control the USRPs and transmit the pilot symbols generated by MATLAB.

MATLAB was used for all baseband signal processing and channel estimation algorithm.

C. Google Colaboratory

Google Colaboratory was used as the Python development environment for the machine learning algorithms that were implemented. It is well suited to machine learning and allows anybody to write and execute python code through the browser [8].

D. Scikit-learn

Scikit-learn is a library for machine learning in Python. The support vector machine (SVM) algorithm and logistic regression (LR) algorithm from the scikit-learn library was used.

III. Solution Approach and methods

This section introduces the relationship between location and channel estimates. We can depend on this relationship to estimate channels at different antennas using ML. In our case, we use a single antenna system to achieve the results and from this can be extended to multiple antennas. Our single antenna system did not work as expected, thus, we relied on simulated-based datasets.

A. Relationship Between Location and Channel Estimate

The transmitter transmits some pilots followed by data to be received by the receiver. The channel is a function of position. By estimating the channel using machine learning, we can localize the users from a single noisy channel. We will manually be changing the position of the transmitter, estimating the channel, changing position, estimating the channel, and so forth.

This is a supervised machine learning problem where the dataset of channel estimates is labeled by "locations." Hence, we can use ML to estimate the channel at other antennas given the channel estimate at one or few antennas. Thus, the two machine learning algorithms used included the logistic regression algorithm and support vector machine algorithm.

B. Channel Estimation by Hardware

A single transmitter and a single receiver were used for our channel estimations. In MATLAB, a pilot signal was generated and saved as a WAV file. This file was then remotely uploaded to GNU Radio which was to be transmitted by the USRPs at a frequency of 2.4GHz. The transmitted signal was then received by the receiver and saved as a WAV file. The WAV file produced by the GNU Radio using the USRPs was uploaded to the channel estimation algorithm in MATLAB where the magnitude of the channel was plotted against the 64 subcarriers.

To generate different dataset points, the location of the transmitter was changed. Each location would result in a unique signature in the received signal due to the different multipath that the signal passes by from the transmitter to the receiver.

C. Alternative Measures

Simulation-based datasets were produced due to the reason that there was not enough variation in the channel among different locations using hardware kits. Figures 3 and 4 depict artificially generated datasets. Simulation-based datasets were needed so that when a dataset is fed into machine learning algorithms, the algorithms can differentiate and locate the different channels.

D. Channel Estimation by Simulation

The dataset depends on two parameters: the SNR and the position. Predicting these two parameters independently is the main goal of this work. It results in a plot where the x-axis is the number of subcarriers versus the magnitude of the estimated channel.

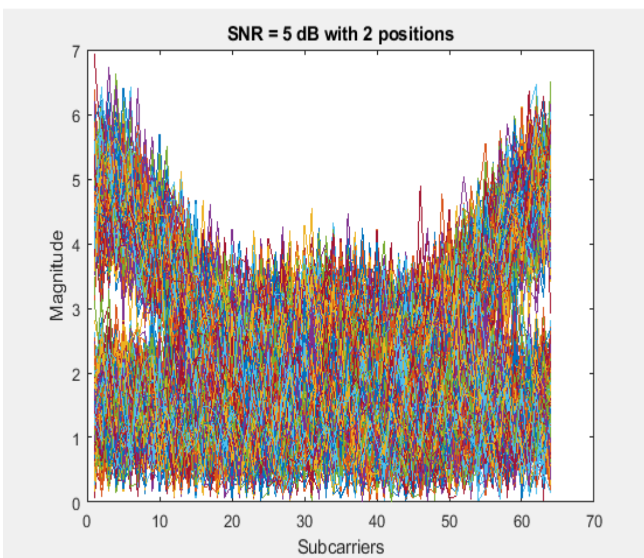

Figure 3: The plot generated with parameters SNR = 5dB and two different positions. The magnitude of the channel vs 64 subcarriers. A total of 1,000 channels estimated.

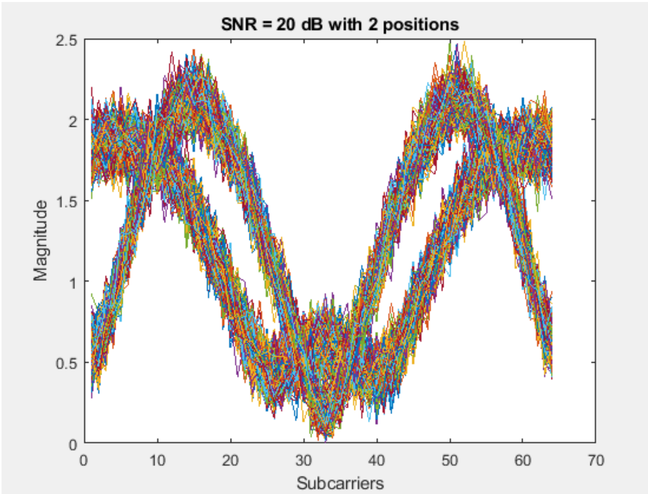

Figure 4: The plot generated with parameters SNR = 20dB and two different positions. The magnitude of the channel vs 64 subcarriers.

IV. Results and Discussion

For our datasets, we want to generate an SNR dataset and positions dataset. A total of 500 dataset points were generated per position. For example, in Figure 4, it displays 1,000 dataset points corresponding to 2 positions. However, the SVM algorithm was applied to a dataset that has 40 different positions. Each dataset was fed into the machine learning algorithms and the classification accuracy was measured. The dataset was divided by 70% training set, 15% validation set, and 15% test set. A confusion matrix, as shown in Figure 5, is produced along with the accuracy of the model predicting the location or SNR value of the channel estimate.

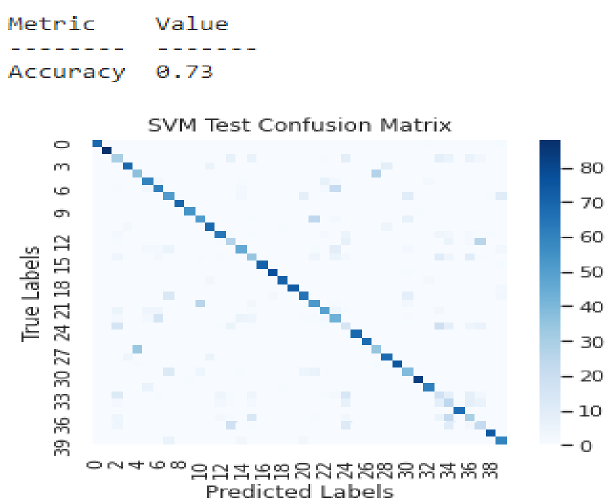

Figure 5: A confusion matrix depicting the support vector machine (SVM) accuracy using 40 different positions with a constant SNR = 5dB. An accuracy of 73% measured using the SVM algorithm with 40 different positions.

Initial results show that when the SNR value is increased a better accuracy is measured in both the SVM algorithm and LR algorithm. As Figure 6 depicts, when the SNR value is 2dB the SVM measures an accuracy of 80% and LR algorithm measures an accuracy of 68%. Whereas, when the SNR value is 10dB both SVM and LR algorithms measure an accuracy of 100%.

Furthermore, when the number of positions is increased the accuracy gradually decreases in both ML algorithms. In Figure 7, at five positions both algorithms measure an accuracy of 100%. At fifty positions, SVM algorithm measures an accuracy of 75% and LR algorithm measures an accuracy of 65%. These results confirm what we expected to see when giving specific data to both machine learning algorithms.

It was also observed that the SVM algorithm outperformed the LR algorithm in both cases tested. As Figures 6 and 7 depict, the SVM dataset continues to have a higher accuracy for all data points whether the SNR value or position change. We expect that this is due to the reason that the SVM algorithm is using the radial basis function (RBF) kernel which projects the data into the higher dimensional space and tries to find the boundary in this high dimensional space. This allows the data that might not be separable in the lower dimensional spaces to be separated at higher dimensions [7]. Meanwhile, the logistic regression model finds the probability distribution of the data points.

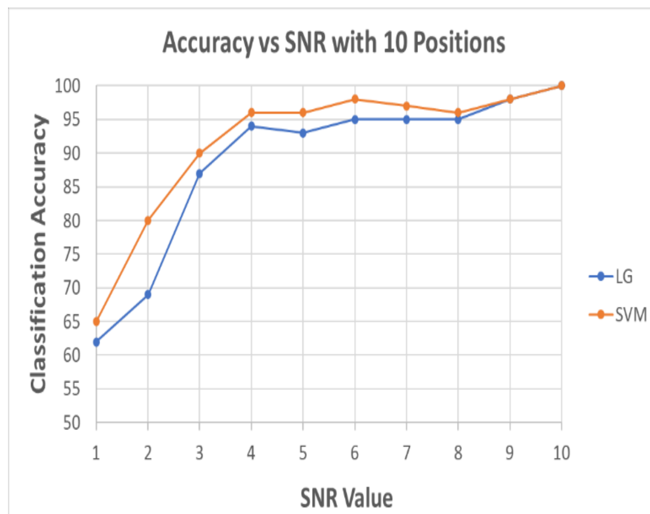

Figure 6: Results from logistic regression (LR) and support vector machine (SVM) algorithms predicting the location of the channel. Classification accuracy vs SNR value (dB) at ten different positions. SVM outperforms LR due to projecting data at a higher dimensional space.

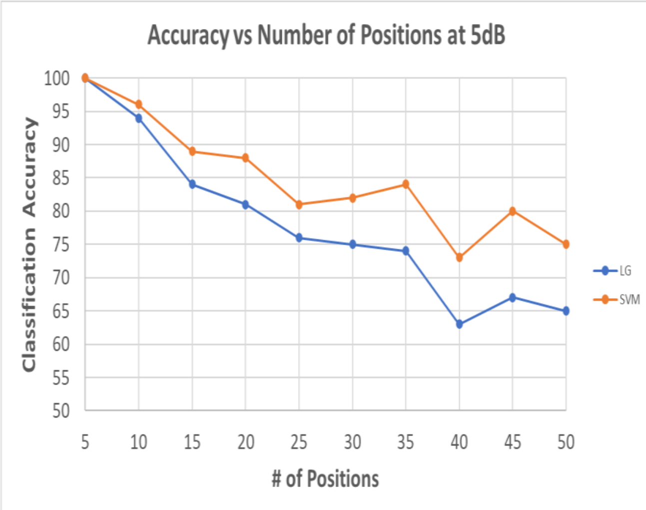

Figure 7: Results from logistic regression (LR) and support vector machine (SVM) algorithms predicting the location of the channel. Classification accuracy vs # of positions at 5dB.

V. FUTURE WORK

We will implement a new method to use hardware kits to represent real world environment and use this data for machine learning. Therefore, we will look at spanning a larger bandwidth for the hardware kits. In addition, other classification algorithms will be investigated to measure the accuracy against the SVM algorithm.

VI. CONCLUSION

In this paper, we demonstrated the ability of machine learning to predict the location of users which could then be used to estimate their channels at other antennas. We implemented a simple hardware system and relied on simulation-based datasets. The results showed the machine learning algorithms are a powerful and useful tool to use when estimating channels because as the noise interfered with the signals, the signals were interfering with each other. Yet, the machine learning algorithms were able to locate and identify which channel was received. The SVM algorithm outperformed the LR algorithm due to the reason SVM uses an RBF kernel while LR finds the probability of the data points.

Thus, implementing machine learning to calculate channel estimation is important because this information can be used for ad hoc networks to schedule users. Ultimately, this leads to scheduling performance being improved so more users will be served with higher throughput and stringent latency.

ACKNOWLEDGEMENTS

This research is sponsored in part by the NSF REU Award 1659871 granted to the Sensor, Signal and Information Processing (SenSIP) Devices and Algorithms REU site.

REFERENCES

[1] Liu, Dongzhu, et al. "Data-importance aware user scheduling for communication-efficient edge machine learning." *IEEE Transactions on Cognitive Communications and Networking* (2020).

[2] Xia, Wenchao, et al. "Federated-Learning-Based Client Scheduling for Low-Latency Wireless Communications." *IEEE Wireless Communications* 28.2 (2021): 32-38.

[3] Sui, Xin, et al. "Virtual machine scheduling strategy based on machine learning algorithms for load balancing." *EURASIP Journal on Wireless Communications and Networking* 2019.1 (2019): 1-16.

[4] A. Ewaisha and C. Tepedelenlioğlu, "Interference-Aware Offloading of Deadline-Constrained Traffic in High Density Cellular Systems," 2019 53rd Asilomar Conference on s, Systems, and Computers, 2019, pp. 1842-1846, doi: 10.1109/IEEECONF44664.2019.9048946.

[5] Fourati, H., Maaloul, R. & Chaari, L. A survey of 5G network systems: challenges and machine learning approaches. *Int. J. Mach. Learn. & Cyber.* **12,** 385–431 (2021). https://doi.org/10.1007/s13042-020-01178-4

[6] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems," IEEE Transactions on Cognitive Communications and Networking, vol. 4, no. 4, pp. 648–664, 2018.

[7] Sergios Theodoridis, Chapter 1 - Introduction, Machine Learning (Second Edition), Academic Press, 2020, Pages 1-17, ISBN 9780128188033, https://doi.org/10.1016/B978-0-12-818803-3.00010-6.

[8] "Google Colaboratory," *Google*. [Online]. Available: research.google.com/colaboratory/faq.html. [Accessed: 15-Jul-2021].

[9] D. Tse and P. Viswanath "Fundamentals of Wireless Communication," *Cambridge University Press*, pp. 29-32, 2004.