Wireless Channel Prediction in Different Locations Using Transfer Learning

Jieying Chen
Purdue University Northwest
Hammond, Indiana, USA
chen3104@pnw.edu

Abdalaziz Sawwan
Temple University
Philadelphia, Pennsylvania, USA
sawwan@temple.edu

Shuhui Yang
Purdue University Northwest
Hammond, Indiana, USA
Shuhuiyang@pnw.edu

Jie Wu
Temple University
Philadelphia, Pennsylvania, USA
jiewu@temple.edu

ABSTRACT

Transfer learning refers to transferring the knowledge of a specific domain to a related domain. In cases where the source and the target learner have similar distribution and parameters, transfer learning can reduce the cost of learning and the construction of the target learner and improve the performance of the target learner. In wireless ad-hoc networks, the users connect to networks based on the service location, and various network channels with different levels of quality-of-service (QoS) are available. The wireless channels represent specific ranges of radio frequencies. When the users move from one location to another, the mobile application may switch channels for good quality of service. This paper predicts the wireless channel based on the user's location. Since channel prediction based on location is feasible in one city, the knowledge of channel prediction in one city can be transferred to another city. Thus, transfer learning is applicable and effective in such applications. The paper uses two cities' wireless mapping datasets to predict network channels and uses transfer learning to predict one city's network channels based on the other city's model. Experiments using different initial learning rates during training and different source and target domain data ratios show that transfer learning is feasible for network prediction among different cities.

CCS CONCEPTS

• **Computing methodologies** → **Instance-based learning**.

KEYWORDS

Cellular networks, deep learning, transfer learning.

ACM Reference Format:
Jieying Chen, Abdalaziz Sawwan, Shuhui Yang, and Jie Wu. 2023. Wireless Channel Prediction in Different Locations Using Transfer Learning. In *The Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '23), October 23–26, 2023, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3565287.3617634

ACM ISBN 978-1-4503-9926-5/23/10.
https://doi.org/10.1145/3565287.3617634

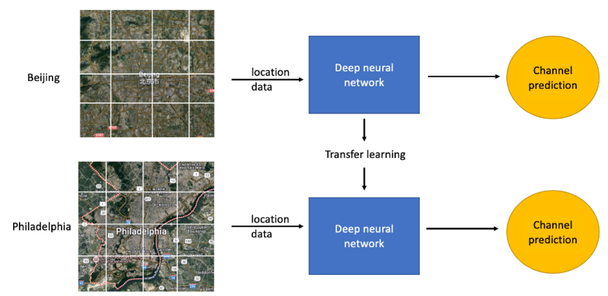

Figure 1: The knowledge gained from training a DNN model from Beijing dataset can be transferred to Philadelphia.

1 INTRODUCTION

Cellular systems are fundamental wireless communication tools in modern communication networks, enabling data to be transferred seamlessly across locations worldwide [1]. Cellular systems provide wireless service by the geological arrangement of individual cellular base stations. Each base station provides a coverage range for a certain radius, and stations locate strategically to avoid signal interference while providing maximum coverage, thus having the name "cellular," where each cell represents the geographic area covered by a base station [1]. When users use mobile devices, they communicate with the base station in their current area via radio waves, allowing data transmission [1].

Each base station provides several radio-frequency (RF) bands that can be divided into individual channels. Each channel represents a distinct range of the electromagnetic spectrum and can perform different tasks, such as TV channels, radio channels, and voice transmission during phone calls [1]. When a user moves from one location to another, their mobile devices gather and analyze the radio-frequency signals from the channels provided by the base station at their current location, then select the channel that provides the strongest signal strength or the quality of service (QoS), and connect to it. This action is called channel switching [1, 2]. Mobile devices switch to a channel that provides the strongest signal strength to improve the call quality, data transfer speeds, network connectivity, upload and download speeds [1]; however, channel switching is a challenging task because (1) It is time-consuming to gather and analyze the channel information, and (2) it is time-consuming to disconnect from the current channel and switch to the new one [2].

Since base stations provide channels based on their current location, we can use location information to predict the channels that might appear in a specific area [2, 3]. We can use machine learning algorithms to carry out such tasks by using location information to train a statistical model or neural network [2, 3]. The intuitive way is to train predictive models individually for each location. However, doing so is time-consuming, and the trained model is data-dependent, which requires a large amount of data for training for each location. A more convenient way is to train a general model that works on a location or several locations, then transfer the knowledge obtained by the model, lastly, train a set of data from a different location to get results. This is called transfer learning [4–7].

Transfer learning is a machine learning technique that uses a pre-trained model's gained knowledge about a task to perform a similar task with different input data [4–7]. Transfer learning improves the model performance of the target domain that is being transferred knowledge to and reduces the amount of time and data needed from the target to perform a similar task as the source domain that transfers knowledge to it [4–7]. A convenient way to perform transfer learning is using deep neural networks (DNN) [2, 3], which are deep learning models composed of multiple neural network layers between the input and output layers [8, 9]. This paper uses DNN as a model architecture for transfer learning.

In this paper, we experiment with the wireless mapping data of two cities and apply transfer learning to them. The two cities are (1) Beijing, which denotes city A in this paper, and (2) Philadelphia, which denotes city B. We wish to examine the model accuracies of channel prediction using transfer learning and the training speed of transfer learning. Figure 1 shows the transfer learning method in this paper and demonstrates how wireless channel prediction utilizes transfer learning.

This paper demonstrates the feasibility of transfer learning in spectral channel prediction based on location in terms of time and data dependency. Our research is summarized as the following:

- We show that transfer learning in channel prediction can be applied among different locations.
- We use different source and target data proportions for training and use different combinations of initial learning rates for DNN during model fine-tuning.
- We evaluate the model accuracy and training time of transfer learning on channel prediction through extensive simulation.

The limitations of this work include, firstly, the sample of cities is limited, with only two cities examined; secondly, the transfer learning model constructed in this paper is preliminary, as the models primarily test the feasibility of transfer learning on network channel prediction via location. There is a trade-off among experimenting with different combinations of data ratio, fine-tuning initial learning rates, and a more sophisticated model that is more time-consuming to run. In practice, the results of transfer learning represent certain level of accuracy, compared to training individual models. To achieve accurate results via transfer learning, comprehensive and extensive simulations of model training across various cities and hyper-parameters are required.

2 RELATED WORK

2.1 Channel prediction

Predicting RF channels can significantly reduce the resources used during channel switching. Biswas and Wu [2] proposed using DNN to predict channels based on GPS locations and combined the channel prediction results and users' mobility patterns to predict the users' future locations. Navabi *et al.* [10] have used neural networks to predict the wireless channel features at base stations that are not directly observable to the base station. Their work gave rise to the potential of predicting unknown channels based on observable channels. In Tumuluru *et al.*'s paper [11], the authors used neural networks and the hidden Markov model (HMM) to predict channel status, i.e., whether a channel is used or unused, which sought to reduce the energy needed by the mobile devices to sense and access unused channels. Azmat *et al.* analyzed the occupancy of the RF spectrum in cognitive radio networks [12] using various machine learning algorithms, and have proposed a new SVM algorithm to classify the channel occupancy information.

2.2 Transfer learning

Transfer learning can be applied to a wide range of tasks. In [4], Zhuang *et al.* performed transfer learning for text-processing and object-detection models. The results show that transfer learning algorithms and approaches allow pre-trained models to be applied to various tasks while saving training time and maintaining certain accuracies. [6] demonstrates the feasibility of transfer learning in human activity recognition tasks. It has characterized the sensor modality for human activity recognition, the source and target environment, the data availability, and the type and amount of data that are transferred. Pan *et al.* [13] have used transfer learning to predict the cross-domain Wi-Fi localization data. They can transfer the knowledge from the source domain, which has a few labeled data, to the target domain containing a large amount of unlabeled data. In the medical field, transfer learning is also applied to transferring one hospital's data to another [14] and can enhance hospital-specific data prediction. [15] demonstrates transfer learning is also feasible for natural language processing (NLP).

3 OVERVIEW

3.1 Wireless channels

The cellular network is essential in telecommunication. It allows radio waves to pass data and voice among wireless devices. The cellular network also enables network access, security, etc. [1]. In a given location, multiple base stations (or cell towers) make up a cellular system, each covering a range of a certain radius. Antennas are placed within the range of the base station, acting as transmitters and receivers of radio waves among mobile devices [1].

When a mobile device is within the range of a base station, the device receives a list of channels. These sections of radiofrequency (RF) bands are called channels occupying specific ranges of frequencies in the electromagnetic spectrum and are expressed in MHz

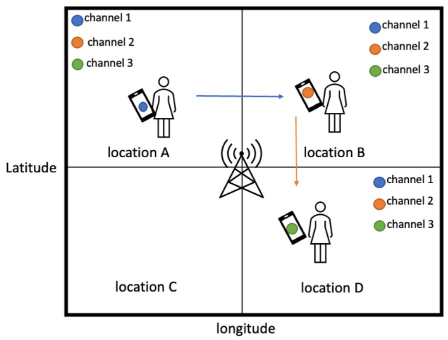

Figure 2: RF channel switch when the mobile devices go from one location to another.

[1, 2]. For example, TV channel 2 has an RF range of 54 – 60 MHz. In the case of Wi-Fi signals, for example, a 2.4 GHz band Wi-Fi ranges from 3000 Hz to 300 GHz within the RF band. There are many channels. Every channel is 20 MHz wide. 5 MHz separates the channels. However, many of them overlap [1].

When a mobile device is placed from one location to another, it receives the RF channels provided by the base station through the antenna. It then analyzes the signal strength of the channels, which also implies network stability, and then chooses the unused channel that provides the most robust signal strength at the current location, disconnects from its existing channel, and connects to the new channel [2, 11]. Figure 2 illustrates the process of channel switching.

3.2 Deep neural network

A deep neural network (DNN) is a neural network class with multiple layers between the input and output layers. DNN learns automatically from the representation of data (i.e., features) through a series of non-linear transformers. DNN is built Conventionally based on Stochastic Gradient Descent (SGD) [8, 9]. When training a DNN model, the goal is to maximize the model's accuracy. Thus, choosing the hyperparameters, such as the initial learning rate, can help minimize the loss function [8]. A loss function measures how well a model performs [8, 9]. If the error is high, the loss function is also high, whereas a decay in the loss function means the model performs well.

In deep learning, stochastic gradient descent (SGD) is a fundamental technique during model training [8], in which it adjusts training parameters iteratively based on the sample's gradient. Still, the required computation complexity is less than that of gradient descent, an optimization algorithm that iterates to find the local maximum of a differentiable function [8]. By adjusting the parameter's gradient at every iteration, SGD or gradient descent goes opposite to the current gradient. Thus, finding the suitable weight for the sample and minimizing the loss function [9].

During the optimization process using gradient descent, the update speed of whether the gradient should change direction is determined by the initial learning rate [8, 9]. A lower initial learning rate allows the optimization function to reach the optimal state after a long time, and a higher initial rate enables the loss function to decay faster. Still, it may lead to fluctuations in model accuracy [8, 9].

3.3 Transfer learning

Transfer learning is a machine learning technique that transfers knowledge from a source domain to a target domain [4–7]. Transfer learning enables a target domain, usually the subject being transferred knowledge to. The target domain learns the knowledge of a specific topic in a shorter time and uses less labeled data than building a model specific to the dataset and domain. This technique is promising when the source and target perform different but related tasks [4–7]. There are two categories of transfer learning: Homogeneous and heterogeneous transfer learning. There are several approaches to transfer learning: Instance-based, parameter-based, feature-based, and relational-based [4–7]. In this paper, we focus mainly on homogeneous transfer learning and instance-based learning.

Homogeneous transfer learning refers to when the source and target domains have similar input features, learning tasks (e.g., network channel prediction), and domain of interest [4–7]. In homogeneous transfer learning, most approaches focus on correcting the marginal distribution differences between the source and target domains [4].

During instance-based transfer learning, we wish to correct the marginal and conditional differences between the source and target domains [4, 5]. There is a sample selection bias or covariate shift when the source and target domains' distribution does not match [16]. Therefore, we need to correct the selection bias and covariate shift. The idea is to assign weights to the loss function of the source domain [4, 16]. The weighting strategy is shown in the following:

$$\mathbb{E}^{T}_{(x,y)}\left[\mathcal{L}\left(x,y;f\right)\right]=\mathbb{E}^{S}_{(x,y)}\left[\frac{\left(P^{T}\left(x,y\right)\right)}{\left(P^{S}\left(x,y\right)\right)}\mathcal{L}\left(x,y;f\right)\right], \quad (1)$$

where $\mathbb{E}_{(x,y)}$ is the expected risk, x is the pattern in the domain, y is the label in the domain, $\mathcal{L}\left(x,y;f\right)$ is the loss function that depends on the parameter f. When the distribution of the source domain ($P^{S}(x)$) is different from that of the target ($P^{T}(s)$), the instances are generalized and are drawn from the target domain ($P^{T}(x)$), denoted in $\frac{P^{T}(x)}{P^{S}(x)}$. The generalized instances are now viewed as the weighting parameter, which is denoted in $\beta\left(x,y\right)$. To estimate the re-weighting coefficient β, the re-weighted regularization risk is first minimized, and $\Omega(f)$ is the regularizer, and n is the number of instances [16].

Therefore, the learning task's general objective function can be written as the following [4, 16]:

$$\min_{f}\frac{1}{n^{s}}\sum_{i=1}^{n^{s}}\beta_{i}\mathcal{L}\left(f\left(x_{i}^{s}\right),y_{i}^{s}\right)+\Omega\left(f\right), \quad (2)$$

where $\beta_i(i=1,2,\ldots,ns)$ is the weighting parameter, and its theoretical value is equal to $P^{T}(x_i)/P^{S}(x_i)$. However, in practice, the values may be hard to determine [16].

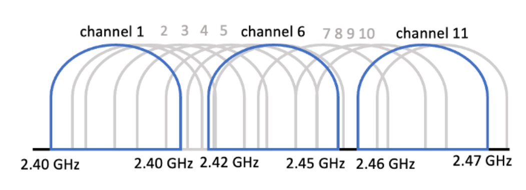

Figure 3: Wireless channels overlap in RF frequency band.

4 METHODOLOGY

This paper uses transfer learning on Beijing (City A) and Philadelphia (City B) wireless mapping datasets. This paper aims to transfer knowledge from city A to city B. We can predict City B's channels using the channel-prediction model originally trained by City A's data.

4.1 Data collection

We collected data from Wigle.net [17], which is a wireless data mapping database. From the website, we downloaded data from two cities: Beijing, as city A, and Philadelphia, as city B. City A has 10,300 observations, and City B has 13,900 observations. Both cities have the same features.

The wireless mapping datasets show the date, type of channel the mobile device is connected to, the type of encryption used for the network, the quality of service of the channel, and other wireless information of the connected user at a specific location. The location information in the dataset includes latitude, longitude, city, country, street number, and more. Network information includes channels, BCN interval, SSID, quality-of-service (QoS), encryption, and more. The channels in the datasets are not represented in radio frequencies in MHz. Instead, they are represented by channels 1, 2, etc. City A has 29 channels, and City B has 38, where some channels are the same, i.e., have the same frequencies.

4.2 Data processing

Because some channel frequencies overlap in the 2.4 GHz RF frequency band [1], many channels in the datasets also overlap. Therefore, some parts of channel 2 may overlap with channel 1, some parts of channel 3 may overlap with channels 1 and 2, and others. Figure 3 shows how wireless channels overlap in the RF frequency band. We can combine the overlapping channels and reduce the possible channels in both cities. Doing so simplifies the channel classification process and reduce the chances of false positive results. City A has 7 channels combined, and City B has 9.

4.3 Feature selection

Since the wireless mapping datasets have a large amount of labeled data, i.e., a specific variable name for the observation, we selected the features and the response variable by hand for supervised learning. This paper uses the wireless channel as the response variable for channel prediction. For the basic model, we selected latitude and longitude as the model features, and this model is used for the simulations.

Besides the basic model, we selected two extra features: encryption and quality-of-service (QoS). We have used the additional features and location features to run some machine learning classification algorithms without transfer learning.

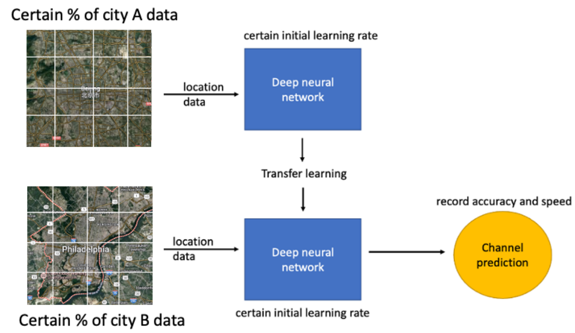

Figure 4: Transfer learning strategy for simulations.

4.4 Building the DNN

For the simulations, we used a 7-layered neural network to perform the model training process. We use DNN because we can fine-tune the training process. The DNN can share the learned weights among the datasets.

The DNN comprises a feature input layer, where the location values are normalized in z-score; a fully connected layer contains 50 nodes, a batch normalization layer, a rectified linear unit layer (reLu) as the activation function. This fully connected layer includes the number of possible network channels, a SoftMax layer, and a classification layer. The training options use the mini-batch size of 30 and use adaptive moment estimation (Adam) as the optimization algorithm.

In the DNN, the input layer takes the latitude and longitude as features. The location values, which are in coordinate systems, are normalized in z-score such that they have a mean of 0 and standard deviation of 1. The fully connected layer that contains 50 nodes are for weight-sharing during the instance-based transfer learning process [4]. During the instance-based learning process, Eq.1 corrects the the selection bias and covariate shift, which simultaneously assigns weights to the loss function of the source domain [16]. In the fully connected layer, the nodes share the assigned weights and multiply them to the input values, then add the bias together [8]. The batch normalization layer makes the DNN training process stable and efficient by correcting the internal covariate shifts, and reduces the chance of over-fitting [9]. The adaptive moment estimation (Adam) optimizer makes the training process memory-efficient and reduces computational power [9].

During the simulation process, we fine-tune the DNN models by adjusting initial learning rates. Learning rates is the parameter updating step size [9]. Since the training process repetitively updates the weights, the learning rates can control the slope of gradient change [8, 9]. Thus, we can observe the performance of the DNN models under different learning rates.

We use the basic model, which uses latitude and longitude as features, to predict the wireless channels. During transfer learning, we first use a certain proportion of City A's data to train the

Figure 5: The data ratios are baseline (100% City A), 100% City A and 20% City B, 80% City A and 20% City B, 60% City A and 40% City B, 40% City A and 60% City B, 20% City A and 80% City B, and upper bound (100% City B). (1) Transfer learning results at different training data ratios, when A's initial learning rate = 0.01. (2) Transfer learning results at different training data ratios, when A's initial learning rate = 0.05. (3)Transfer learning results for initial learning rate fine tuning, when initial learning rate of A = 0.01. (4)Transfer learning results for initial learning rate fine tuning, when initial learning rate of A = 0.025.

prediction model at a certain initial learning rate. Once the last mini-batch of City A's data is trained, we apply a certain proportion of City B's data on the same model at a specific initial learning rate, which may differ from City A's. The transfer learning model is then tested using city B's test data. After each training, we record the test accuracy and the training speed.

5 SIMULATION

In this section, we conduct simulations in three scenarios: (1) Basic model training without transfer learning, (2) Transfer learning from City A to City B, using different proportions of City A and City B's data during training, and fine-tuning the models using different initial learning rates, and (3) training speed and accuracy of transfer learning from training 100% of City A's data, and transfer to different proportions of City B's data, at different initial learning rates.

5.1 Basic model results without transfer learning

In the basic model, latitude and longitude are the features used for channel prediction. Without transfer learning, we want to use the base model results as the baseline and upper bound accuracies for the rest of the transfer learning simulations. To obtain the baseline accuracy, we first train the channel-prediction model with 100% of City A's data and then test the model using 20% of city B's data, without transfer learning. We repeated this process 101 times for different initial learning rates applied to the model, from 0.001 to 1, in steps of 0.01. The baseline value is the average of the results, which is 6.26%.

The upper bound test accuracy is obtained from training and testing solely on city B's data without transfer learning. We split the train-test data in 80/20, and the resulting upper bound accuracy is 28.40%.

5.2 Transfer learning results

In this subsection, we perform transfer learning from City A to City B, using different proportions of City A and City B's data for training over different initial learning rates. For instance, the model trained on 100% of City A's data transfers to 20% of City B's data, the model trained on 40% of City A's data transfers to 60% of City B's data, 80% of City A with 20% city B for training, 60% city A

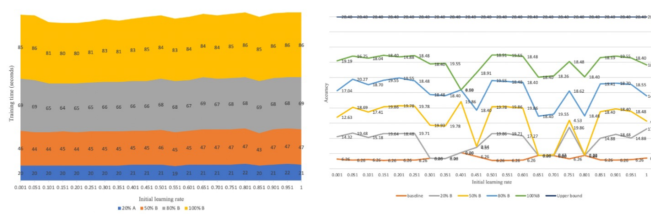

Figure 6: (1)Training speed of transfer A to different proportions of B, over different initial learning rates. (2)Test accuracies that correspond to the training speed..

and 40% city B, and 20% city A and80% city B for training. Figure 4 illustrates the strategy for transferring learning from City A to City B.

We evaluated the test results for city B across various combinations of training data and initial learning rates. Initially, we tested three different initial learning rates for city A: 0.001, 0.01, and 0.05. Then, we trained models for City B using seven different initial learning rates, which were paired with the City A models. We assessed City B's performance for each pairing using five different training data proportions from Cities A and B. We applied five different training data proportions for Cities A and B, resulting in 35 trials. Thus, we ran 105 tests to assess City B's performance under different training scenarios with the specified initial learning rates. The results from Figure 5 show that, in general, as the proportion of City B's training data increase in the model, the higher the model accuracies, regardless of the initial learning rates.

After experimenting with three different initial learning rates, we sought to fine-tune them by adjusting the initial learning rates of the transfer learning models. We then do simulations over more combinations of initial learning rates for Cities A and B. Six initial learning rates for City A are set to be 0.001, 0.01, 0.025, 0.05, 0.075, and 0.1. We also choose seven different initial learning rates for City B to pair with City A. However, the initial learning rates for City B are greater than or equal to City A's.

Figure 5 shows that finding the optimal initial learning rate requires extensive tests and only sometimes guarantees an increase in model accuracy. The results also show that increasing the data of City B used to train the model always results in a greater overall model accuracy compared to using a small City B sample proportion.

5.3 Training speed vs. model accuracy

This section will conduct transfer learning simulations and record their corresponding training speed and test accuracy over different initial learning rates. The initial learning rates range from 0.001 to 1, in steps of 0.05, with 21 initial learning rates. The channel-prediction model is first trained using 100% of City A's data. Then, with the same initial learning rate as city A, different proportions of city B's data are used to train the model. City B's data proportions are 20%, 50%, 80%, and 100%.

Figure 6 shows the training speed of different proportions of city B data used in transfer learning. We see that training 20% of City B's data uses 1/4 of the training time as training 100% of City B's data and training 50% of City B's data uses half of the time. Training 20% of City B's data during transfer learning yields 17.27% accuracy. However, it only takes 1/4 the time to train a model that uses 100% of City B's data, and its accuracy significantly improved compared to the baseline of 6%. The simulation results show that transfer learning can reduce the training time and the amount of data needed to train a model.

The simulation results shows the effectiveness of transfer learning when there is not enough time and data to train individual models. Although the performance of the basic model in simulations is a proportion of the upper bound, however, it can be improved by using a more sophisticated DNN and more advanced models that uses more features.

6 CONCLUSION

In this study, we explored the application of transfer learning for Wi-Fi channel prediction across distinct geographical locations, specifically focusing on the Wi-Fi channel datasets of Beijing (City A) and Philadelphia (City B). Our objective was to train a deep neural network (DNN) model trained on one city (City A) to enhance the prediction accuracy in another city (City B) through instance-based transfer learning.

We began by constructing DNN models for the baseline and upper bound benchmarks of transfer learning, achieving accuracies of 6.26% and 28.4%, respectively. These initial accuracies served as our performance benchmarks for subsequent evaluations. Subsequently, we introduced and implemented an instance-based transfer learning methodology. This approach involved retraining the City-A-trained DNN on subsets of the City B dataset, utilizing two key parameters: the City B dataset's size and the sample's initial learning rate.

Our experimental results demonstrated the effectiveness of the instance-based transfer learning strategy for wireless channel prediction. We conducted extensive simulations by varying the initial learning rates in DNN training for Cities A and B. The accuracy of these transfer learning models consistently outperformed the lower-bound benchmark – a DNN model trained on the City A dataset and tested on City B data. Our transfer learning models exhibited performance improvements, albeit varying degrees, compared to the standalone DNN model trained solely on City B data, indicating the potential to transfer channel prediction knowledge from one city to another.

Furthermore, we compared the speed of transfer learning on different sizes of City B data. The results show that a small proportion of City B data can improve the baseline accuracy and uses less training time compared to training a standalone City B model. This efficiency gain reinforces the practical utility of transfer learning in real-world scenarios where time and resources are limited.

In the context of accuracy, our transfer learning models presented results falling between the upper-bound and baseline benchmarks, validating the value of transfer learning in bridging the gap between distinct geographical domains. Future work can involve applying transfer learning to more cities. Exploring the optimal subset size, initial learning rates, more sophisticated DNN model, and use enhanced training models could yield even better results.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation Grant CNS 2150152 and CNS 2107014.

REFERENCES

[1] Tripathi, N. D., Reed, J. H. (2009). Cellular Communications: A Comprehensive and Practical guide.

[2] Biswas, R.; Wu, J. Minimizing The Number of Channel Switches of Mobile Users in Cognitive Radio Ad-Hoc Networks. J. Sens. Actuator Netw. 2020, 9, 23. https://doi.org/10.3390/jsan9020023.

[3] Tumuluru, V.K.; Wang, P.; Niyato, D. Channel status prediction for cognitive radio networks. Wirel. Commun. Mob. Comput. 2012, 12, 862–874.

[4] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Senior Member, IEEE, Xiong, H., Fellow, IEEE, He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43-76.

[5] Weiss, K., Khoshgoftaar, T.M. Wang, D. A survey of transfer learning. J Big Data 3, 9 (2016). https://doi.org/10.1186/s40537-016-0043-6.

[6] Cook, D., Feuz, K.D. Krishnan, N.C. Transfer learning for activity recognition: a survey. Knowl Inf Syst 36, 537–556 (2013). https://doi.org/10.1007/s10115-013-0665-3.

[7] L. Shao, F. Zhu and X. Li, "Transfer Learning for Visual Categorization: A Survey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 5, pp. 1019-1034, May 2015, doi: 10.1109/TNNLS.2014.2330900.

[8] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. 2018. A Survey on Deep Learning: Algorithms, Techniques, and Applications. ACM Comput. Surv. 51, 5, Article 92 (September 2019), 36 pages. https://doi.org/10.1145/3234150.

[9] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8.

[10] Navabi, S.; Wang, C.; Bursalioglu, O.Y.; Papadopoulos, H. Predicting wireless channel features using neural networks. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.

[11] Tumuluru, V.K.; Wang, P.; Niyato, D. Channel status prediction for cognitive radio networks. Wirel. Commun. Mob. Comput. 2012, 12, 862–874.

[12] F. Azmat, Y. Chen and N. Stocks, "Analysis of Spectrum Occupancy Using Machine Learning Algorithms," in IEEE Transactions on Vehicular Technology, vol. 65, no. 9, pp. 6853-6860, Sept. 2016, doi: 10.1109/TVT.2015.2487047.

[13] S. J. Pan, I. W. Tsang, J. T. Kwok and Q. Yang, "Domain Adaptation via Transfer Component Analysis," in IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199-210, Feb. 2011, doi: 10.1109/TNN.2010.2091281.

[14] Wiens, J., Guttag, J., Horvitz, E. "A study in transfer learning: leveraging data from multiple hospitals to enhance hospital-specific predictions," in Journal of the American Medical Informatics Association, Volume 21, Issue 4, July 2014, Pages 699–706, https://doi.org/10.1136/amiajnl-2013-002162.

[15] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M. amp; Gelly, S.. (2019). Parameter-Efficient Transfer Learning for NLP. Proceedings of the 36th International Conference on Machine Learning, PMLR 97:2790-2799, 2019.

[16] J. Huang, A.J. Smola, A. Gretton, K.M. Borgwardt, and B. Sch¨olkopf, "Correcting sample selection bias by unlabeled data," in Proc. 20th Annual Conference on Neural Information Processing Systems, Vancouver, Dec. 2006, pp. 601–608.

[17] WiGLE: Wireless Network Mapping. Available online: https://api.wigle.net/ (Accessed June 2023).