

Transfer Learning for Wireless Networks: A Comprehensive Survey

Cong T. Nguyen*, Nguyen Van Huynh*, Nam H. Chu*, Yuris Mulya Saputra*, Dinh Thai Hoang*, Diep N. Nguyen*, Quoc-Viet Pham†, Dusit Niyato‡, Eryk Dutkiewicz* and Won-Joo Hwang§

*School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia

†Research Institute of Computer, Information and Communication, Pusan National University, Busan 46241, Korea

‡School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore

§Department of Biomedical Convergence Engineering, Pusan National University, Yangsan 50612, Korea

Abstract—With outstanding features, Machine Learning (ML) has become the backbone of numerous applications in wireless networks. However, the conventional ML approaches face many challenges in practical implementation, such as the lack of labeled data, the constantly changing wireless environments, the long training process, and the limited capacity of wireless devices. These challenges, if not addressed, can impede the effectiveness and applicability of ML in wireless networks. To address these problems, Transfer Learning (TL) has recently emerged to be a promising solution. The core idea of TL is to leverage and synthesize distilled knowledge from similar tasks as well as from valuable experiences accumulated from the past to facilitate the learning of new problems. Doing so, TL techniques can reduce the dependence on labeled data, improve the learning speed, and enhance the ML methods’ robustness to different wireless environments. This article aims to provide a comprehensive survey on applications of TL in wireless networks. Particularly, we first provide an overview of TL including formal definitions, classification, and various types of TL techniques. We then discuss diverse TL approaches proposed to address emerging issues in wireless networks. The issues include spectrum management, localization, signal recognition, security, human activity recognition and caching, which are all important to next-generation networks such as 5G and beyond. Finally, we highlight important challenges, open issues, and future research directions of TL in future wireless networks.

Index Terms—Transfer Learning, wireless networks, 5G/6G, Machine Learning, cognitive radios, security, caching, localization and signal recognition.

I. INTRODUCTION

The last decade has witnessed the rapid growth of Machine Learning (ML) applications in wireless networks thanks to its agility and efficacy, especially in dealing with uncertainty and dynamics in large-scale problems [1], [2]. A few recent reports by Samsung [3] and Ericsson [4] showed that ML has been developing as the backbone of 5G networks, and it will be the main driver of mobile technology and the 6G in the near future. However, some recent studies revealed that conventional ML solutions have shortcomings, especially when they are applied to solve emerging problems in wireless networks, due to special characteristics of wireless communications, such as high mobility, dynamic environments, diverse connections, and interference [5], [6]. In particular, conventional ML techniques are usually trained for a specific scenario. However, the wireless environments are subject to

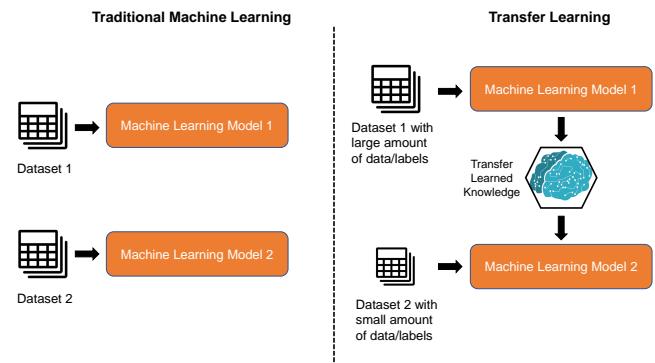


Fig. 1. Traditional ML vs. TL.

significant variations which make the new scenarios different from the previous ones (e.g., mobility of mobile users as well as changes in mobile users’ data demands). Consequently, this can significantly impact the ML techniques’ performances and hinder their applicability. Moreover, the performances of ML techniques mainly rely on the availability of training data, but acquiring a sufficient amount of data might be costly and time-consuming. Even if the training data are sufficient, ML techniques, such as deep learning, usually require a long training time, which makes them impractical for many latency-sensitive applications. Apart from the training time issues, many wireless devices, e.g., IoT devices, are constrained by their limited computing capacity, and thus they are unable to train high-complexity ML models. Moreover, centralized ML systems usually require a huge amount of data to be collected at central servers for processing and training. In addition to communication overhead, sending raw data to the servers may also threaten network users’ privacy because sensitive information, e.g., healthcare, is sent over wireless networks or public networks.

To address these challenges, Transfer Learning (TL) has recently emerged as a highly effective solution. Unlike conventional ML techniques that are trained to solve a specific problem, TL leverages valuable knowledge from similar tasks and previous experiences to significantly enhance the learning performance of conventional ML techniques, as illustrated in Fig. 1. As a result, TL possesses numerous advantages over traditional ML approaches [7], [8] which can be summarized

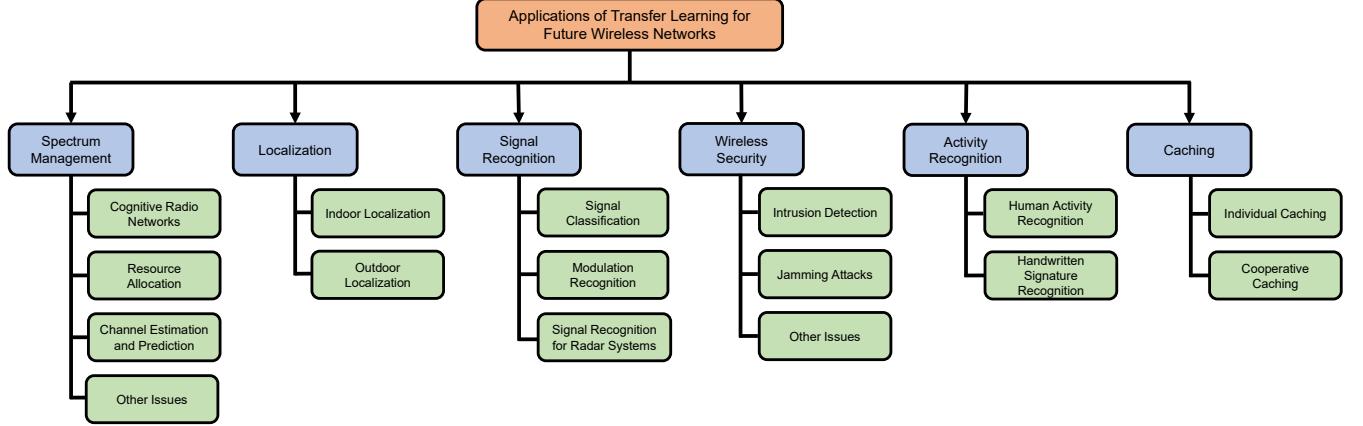


Fig. 2. Taxonomy of the applications of TL for wireless networks.

as follows:

- *Enhance quality and quantity of training data:* One of the most challenging tasks for conventional ML approaches is finding sufficient and high-quality data for the training process. TL can easily overcome this problem by selecting and transferring knowledge from similar domains with a large amount of high-quality data. As a result, TL is considered to be a promising solution for ML-based wireless networks in the near future.
- *Speed-up learning processes:* Instead of learning from scratch like conventional ML approaches, the training process in TL can be significantly sped up thanks to valuable knowledge shared from other similar domains and/or learned in the past. As a result, this can remarkably improve the learning rate, which is especially crucial for the development of ultra-low latency applications for future wireless networks.
- *Reduce computing demands:* Conventional ML approaches usually require a huge amount of computing resources for the training processes. However, with TL, most of the data were trained by other source domains before the trained models are transferred to the target domain, thereby significantly reducing computing demands for the training process at the target domain. This is particularly useful for wireless devices (e.g., smartphones and edge devices) as they usually have hardware constraints.
- *Mitigate communication overhead:* For TL approaches, instead of sending the raw data with large sizes, only knowledge, e.g., weights of trained models, needs to be transferred. As a result, the communication overhead can be substantially reduced for wireless networks.
- *Protect data privacy:* In TL, instead of learning from raw data from other domains, ones only need to learn from their trained models (expressed through weights), and thus data privacy can be protected. This feature of TL is very helpful for privacy-sensitive wireless applications such as healthcare and military communication networks.

With the aforementioned outstanding advantages, TL has been leveraged to address various challenges that traditional ML techniques are facing in numerous wireless network

applications. This article aims to provide an in-depth and comprehensive survey on TL applications in wireless networks. Particularly, we first provide an in-depth tutorial of TL, including formal definitions and some fundamental knowledge of various types of TL techniques. Then, a comprehensive survey on the applications of TL in many areas of wireless networks is presented. For these TL applications, we provide detailed reviews and analyses on how TL is leveraged to overcome the challenges that the underlying ML techniques are facing. Finally, we discuss the current challenges and open issues of implementing TL in wireless networks and introduce some promising research directions for applications of TL in the future.

There are a few surveys on the applications of ML in wireless networks, such as [2], [9]–[12], with different focuses. However, these papers emphasize conventional ML approaches, e.g., supervised learning, unsupervised learning, and reinforcement learning. Alternatively, recent surveys focus on applications of advanced ML techniques in wireless networks such as [13], [14], and [15]. Particularly, [13], [14], and [15] focus on applications of Deep Learning (DL), Federated Learning, and Deep Reinforcement Learning (DRL) in wireless networks, respectively. In addition, there are a couple of survey papers focusing on TL techniques, e.g., [7], and its applications in activity recognition only, e.g., [8]. Nevertheless, from all aforementioned papers, and to the best of our knowledge, there is still no comprehensive survey on the applications of TL in wireless networks. Given the rapid development of wireless communication technologies along with increasing requirements for high Quality-of-Service (QoS), interoperability, robustness, as well as privacy and security from mobile users, TL is in urgent need to address the current limitations of ML-based wireless networks, especially in beyond 5G and 6G networks. As a result, this paper is expected to fill the gap and significantly contribute to the future development of intelligent wireless networks.

The rest of this paper is organized as follows. Section II provides a tutorial on TL including formal definitions, classification, and various types of TL techniques. Then, TL-based applications in various areas of wireless networking (as summarized in Fig. 2) such as spectrum management, localization,

signal recognition, security, human activity recognition, and caching, are presented from Section III to Section VIII. Section IX discusses challenges, open issues, and future research directions of TL. Finally, Section X concludes the paper.

II. TRANSFER LEARNING: AN OVERVIEW

In this section, we first present the fundamental knowledge of TL and its advantages in learning processes. Then, the formal definition of TL is presented. Afterward, important TL strategies are discussed in detail. Finally, we present the current advances in TL.

A. Overview of Transfer Learning

Recently, ML has been successfully applied to various areas and received great attention from both industry and academia. The main goal of ML is to extract and learn data structures to estimate future outcomes. However, existing ML approaches usually assume that the unlabeled data and labeled data share the same data distribution and feature space [16]. In many scenarios, collecting the training data with the same data distribution and feature space as in the testing data is costly and intractable [17]. In this case, the prediction and classification performance is poor as what has been learned from the training data is less useful for the testing data. Another problem of conventional ML techniques is the lack of training data as collecting data in some cases is complex or even impossible [18].

To address these problems, TL is introduced to be an effective solution. Different from traditional ML, the key fundamental of TL is transferring knowledge gained from one source task to improve the learning of a related target task as illustrated in Fig. 1. TL is motivated by the fact that based on knowledge learned before, we can solve new problems more efficient and faster. For example, people who have learned to play violin may learn to play guitar faster than others with no previous experience in playing musical instruments [7], [17].

TL has a long history. The concept of TL may initially be introduced in educational psychology [7]. In the field of ML, TL was reviewed in NIPS-95's workshop on "Learning to Learn". Since then, under different terms like "learning to learn", "life-long learning", and "knowledge transfer", TL has been studied and applied in various aspects of our daily life. It is expected that "TL will be the next driver of ML commercial success after supervised learning" [19].

B. Fundamentals of Transfer Learning

We present the formal definition of TL. As mentioned, in TL, learned knowledge will be transferred from the source domain to the target domain to improve the learning process of the target task. Thus, in the following, we first present the definition of a "domain" and a "task".

Definition 1. "Domain: A domain \mathcal{D} is defined by two parts: (i) a feature space \mathcal{X} and (ii) a marginal probability distribution $\mathcal{P}(X)$ in which $X = \{x_1, \dots, x_n\} \in \mathcal{X}$ where n is the number of feature vectors in X . As such, $\mathcal{D} = \{\mathcal{X}, \mathcal{P}(X)\}$."

Definition 2. "Task: Given domain \mathcal{D} , a task \mathcal{T} is defined by two parts: (i) a label space \mathcal{L} and (ii) a predictive function $f(\cdot)$. The predictive function (or decision function) is learned from the feature vector and label space pairs $\{x_i, l_i\}$, with $x_i \in \mathcal{X}$ and $l_i \in \mathcal{L}$. In other words, a task is defined by $\mathcal{T} = \{\mathcal{L}, f(\cdot)\}$."

In general, the predictive function represents the prediction of the corresponding label $f(x_i)$ given instance x_i [16], [7]. In this case, the predictive function can be defined as $f(x_i) = \{P(l_k|x_i)|l_k \in \mathcal{L}, k = 1, \dots, |\mathcal{L}|\}$. For example, in the signal classification application, \mathcal{L} is the set of all labels which are "0" or "1". The predictive function can predict the probability of bit "0" or bit "1" based on instance x in the feature space. Note that x can contain all the features in the feature space or just critical features (which depend on the considered problem) in the feature space.

It is worth noting that a domain contains various labeled or unlabeled data which are denoted as the domain data. In the context of TL, we denote D_S as the source domain data, i.e., $D_S = \{(x_{S_1}, l_{S_1}), \dots, (x_{S_i}, l_{S_i}), \dots, (x_{S_{n_S}}, l_{S_{n_S}})\}$ with $x_{S_i} \in \mathcal{X}_S$ represents the i -th data instance of D_S and $l_{S_i} \in \mathcal{L}_S$ denotes the corresponding label of x_{S_i} . Similarly, the target domain data can be defined as $D_T = \{(x_{T_1}, l_{T_1}), \dots, (x_{T_i}, l_{T_i}), \dots, (x_{T_{n_T}}, l_{T_{n_T}})\}$, in which $x_{T_i} \in \mathcal{X}_T$ is the i -th data instance of D_T and $l_{T_i} \in \mathcal{L}_T$ is the corresponding label of x_{T_i} . Practically, the number of instances at the target domain is often much lower than that of the source domain, i.e., $0 \leq n_T \ll n_S$. With TL, the learning process of the target domain can be significantly improved by leveraging the knowledge transferred from the source domain. We now formally define TL in Definition 3.

Definition 3. "Transfer Learning: Given a source domain D_S with a corresponding source task \mathcal{T}_S and a target domain D_T with a corresponding target task \mathcal{T}_T , the goal of TL is to learn the target predictive function $f_T(\cdot)$ by leveraging the knowledge gained from D_S and \mathcal{T}_S , where $D_S \neq D_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ ¹."

From Definition 1, the target domain is a pair $\mathcal{D}_T = \{\mathcal{X}_T, \mathcal{P}(X_T)\}$ and the source domain is a pair $\mathcal{D}_S = \{\mathcal{X}_S, \mathcal{P}(X_S)\}$. Thus, $\mathcal{D}_S \neq \mathcal{D}_T$ means that $\mathcal{X}_S \neq \mathcal{X}_T$ and/or $\mathcal{P}(X_S) \neq \mathcal{P}(X_T)$. When $\mathcal{X}_S \neq \mathcal{X}_T$, the learning problem is defined as heterogeneous TL. Differently, when $\mathcal{X}_S = \mathcal{X}_T$, the learning problem is defined as homogeneous TL [17]. $\mathcal{P}(X_S) \neq \mathcal{P}(X_T)$ when the marginal distributions of the target domain and the source domain are different. It is proven that knowledge gained from a given source domain will not achieve the optimal prediction on a target domain if their marginal distributions \mathcal{P} are not the same [16], [17].

Another scenario of TL is $\mathcal{T}_S \neq \mathcal{T}_T$. As mentioned, $\mathcal{T}_S = \{\mathcal{L}_S, f(\cdot)\} = \{\mathcal{L}_S, \mathcal{P}(L_S|X_S)\}$ and $\mathcal{T}_T = \{\mathcal{L}_T, f(\cdot)\} = \{\mathcal{L}_T, \mathcal{P}(L_T|X_T)\}$. Thus, $\mathcal{T}_S \neq \mathcal{T}_T$ represents the cases when $\mathcal{L}_S \neq \mathcal{L}_T$ and/or $\mathcal{P}(L_S|X_S) \neq \mathcal{P}(L_T|X_T)$. When $\mathcal{P}(L_S|X_S) \neq \mathcal{P}(L_T|X_T)$, the conditional probability distri-

¹It is worth noting that when the source domain and target domain are the same (i.e., $D_S = D_T$) and the source task and the target task are the same (i.e., $\mathcal{T}_S = \mathcal{T}_T$), TL becomes traditional ML [16].

butions of the source and the target domains are not the same. The case in which $\mathcal{L}_S \neq \mathcal{L}_T$ represents that the label spaces of the source domain and the target domain are different. Another case that reduces the prediction performance of TL is when $\mathcal{P}(L_S) \neq \mathcal{P}(L_T)$, which happens when the labeled samples of the source and target domains are unbalanced [17].

C. Transfer Learning Techniques

The key fundamental of TL is utilizing knowledge learned from the source domain to improve the learning of the target domain. To guarantee a good transfer performance, the following three main issues need to be taken into account.

- *What to transfer:* Deciding what to be transferred is the most critical step in TL. To address this issue, one needs to decide which part of learned knowledge will be transferred to improve the learning process of the target domain. This stems from the fact that not all learned knowledge from the source domain can be useful for the target domain in many scenarios. For example, some knowledge can be common in both the source and the target domains, whereas some knowledge is specific to the source domain but not the target domain.
- *When to transfer:* Transferring knowledge is not always helpful in speeding up the learning process of the target domain. Thus, one needs to know when the knowledge should not be transferred. For instance, if the target domain does not have anything in common with the source domain, i.e., they are not related, TL may not improve the learning process and even make the learning process less effective (i.e., *negative transfer*).
- *How to transfer:* Once the what and when questions have been addressed, one can proceed to transfer learned knowledge to the target domain. This process requires different techniques and designs to maximize the transferring utilization at the target domain.

Through these three questions, several categorization criteria of TL are proposed. For example, TL can be classified into (i) feature-based, (ii) parameter-based, (iii) relational-based, and (iv) instance-based TL [7]. In particular, feature-based techniques can be used to construct a new feature representation based on the original features. They can be further categorized into (a) symmetric feature-based and (b) asymmetric-based TL. The symmetric method aims to transform the target and source features to a new feature representation, whereas the asymmetric one aims to transform the source features to the form of the target features. Differently, parameter-based techniques aim to transfer hyper-parameters and/or information of the source model. Relational-based techniques transfer knowledge by learning the common relationships between the target domain and the source domain [7]. Finally, instance-based techniques aim to reweight the data samples in the source domain to reduce the differences between the source and the target marginal distributions.

In this survey, we categorize TL based on the problem perspective. In particular, there are three types of TL: (i) inductive TL, (ii) transductive TL, and (iii) unsupervised TL. In the inductive TL, the target domain is the same as the

source domain, but their tasks are different. In this case, the inductive biases of the source domain will be transferred to the target domain and utilized to improve the learning performance of the target task. In the inductive TL, there are two subcategories: (a) self-taught learning and (b) multi-task learning. In self-taught learning, the labeled data are not available at the source domain. Alternatively, in multi-task learning, the labeled data are available at the source domain, and both the target and source domains learn their tasks simultaneously. In transductive TL, the source task and the target task are similar, but their domains are different. Finally, in unsupervised learning, the labeled data are not available at both the source domain and the target domain. These settings of TL are summarized in Table I.

1) *Inductive Transfer Learning:* In general, inductive TL can be defined as follows [16]:

Definition 4. “*Inductive TL: Given a source domain D_S with a corresponding source task T_S and a target domain D_T with a corresponding target task T_T , inductive TL aims to enhance the learning of the target predictive function $f_T(\cdot)$ of target domain D_T based on the knowledge gained in D_S and T_S , in which the source task and the target task are different, i.e., $T_S \neq T_T$.*”

It is worth noting that labeled data should be available at the target domain to guarantee good performance of the target predictive function. This is because the source and the target tasks are not the same. As such, learning only from the labeled data of the source domain may not provide good performance for the target domain. Inductive TL can be further divided into: (i) self-taught learning and (ii) multi-task learning based on the availability of labeled data in the source domain. In particular, when there is no labeled data in the source domain, inductive TL can be categorized as self-taught learning. The key idea of self-taught learning is to obtain a “higher-level feature representation on the inputs” based on the unlabeled data of the source domain [20]. After that, data in the target domain can be analyzed and classified in this representation. In other words, the unlabeled data of the source domain are used to reduce the dimension of the feature space, and the labeled data in the target domain are used for classification in the new feature space. In the case that both the source domain and the target domain have labeled data and they learn their tasks simultaneously, inductive TL can be categorized as multi-task learning [16].

In the inductive TL, knowledge can be transferred to the target domain in the forms of instance, feature representation, parameters, and relational-knowledge. In particular, the instance-transfer approach aims to re-weight a part of labeled data in the source domain in order to reduce the differences in the source and target marginal distributions, and thus improving the learning process of the target domain [21], [22]. Differently, the feature-representation approach constructs a new feature representation and then transform both the source and target features to this new representation to mitigate domain divergence and errors [16]. Note that depending on the availability of labeled data at the source domain, several learning methods can be adopted to construct the new feature representation.

TABLE I
TL CATEGORIZATION

| TL Strategy | Related Research Areas | Source and Target Domains | Source Domain Labels | Target Domain Labels | Source and Target Tasks |
|-----------------|--|---------------------------|----------------------|----------------------|-------------------------|
| Inductive TL | Multi-task Learning | The same | Available | Available | Different but related |
| | Self-taught Learning | The same | Unavailable | Available | Different but related |
| Transductive TL | Domain Adaptation, Sample Selection Bias and Covariate Shift | Different but related | Available | Unavailable | The same |
| Unsupervised TL | | Different but related | Unavailable | Unavailable | Different but related |

For example, if the source domain has many labeled data, supervised learning techniques can be adopted [23], [24]. In contrast, unsupervised learning can be used when the labeled data are not available at the source domain [25]. The parameter transfer approach is used if the models of related tasks have the same parameters or prior distribution of hyper-parameters [16]. Finally, the relational-knowledge transfer approach is applied if the data are not independent and identically distributed (i.i.d), such as networked and social network data [26].

2) *Transductive Transfer Learning*: Transductive TL can be defined as follows [16]:

Definition 5. “*Transductive TL: Given a source domain \mathcal{D}_S with a corresponding source task \mathcal{T}_S and a target domain \mathcal{D}_T with a corresponding target task \mathcal{T}_T , transductive TL aims to enhance the learning of the target predictive function $f_T(\cdot)$ of target domain \mathcal{D}_T based on the knowledge gained in \mathcal{D}_S and \mathcal{T}_S , in which the source domain and the target domain are different, i.e., $\mathcal{D}_S \neq \mathcal{D}_T$ and the source task and the target task are the same, i.e., $\mathcal{T}_S = \mathcal{T}_T$. Furthermore, unlabeled data must be available at the target domain when training.”*

Transductive TL was first introduced in 2007 [27] to address the problem of TL in which labeled data are not available at the target domain. To ensure good learning performance, transductive TL requires all unlabeled data to be available when training. However, the authors in [16] point out that this requirement can be relaxed, and only a part of the unlabeled data are required at the training time. In transductive TL, there are two scenarios corresponding to the settings of feature spaces and marginal distributions as follows:

- The feature spaces of the target domain and the source domain are different, i.e., $\mathcal{X}_S \neq \mathcal{X}_T$.
- The feature spaces of the target domain and the source domain are the same, i.e., $\mathcal{X}_S = \mathcal{X}_T$. However, their marginal distributions are not the same, i.e., $\mathcal{P}(X_S) \neq \mathcal{P}(X_T)$.

These two settings are similar to domain adaptation and sample selection bias [16]. Domain adaptation aims to transfer knowledge learned from one or more source domains to enhance the learning performance of a target domain. Meanwhile, sample selection bias is used to mitigate the differences in data distributions of the source and target domains [28]. Similar to inductive TL, in transductive TL, the learned knowledge can be transferred to the target domain through feature representation-transfer, relational-knowledge-transfer, parameter-transfer, and instance-transfer approaches [16], [29].

3) *Unsupervised Transfer Learning*: Unsupervised TL can be defined as follows [16]:

Definition 6. “*Unsupervised TL: Given a source domain \mathcal{D}_S with a corresponding source task \mathcal{T}_S and a target domain \mathcal{D}_T with a corresponding target task \mathcal{T}_T , unsupervised TL aims to enhance the learning of the target predictive function $f_T(\cdot)$ of target domain \mathcal{D}_T based on the knowledge gained in \mathcal{D}_S and \mathcal{T}_S , in which the source task and the target task are different, i.e., $\mathcal{T}_S \neq \mathcal{T}_T$ and the labeled data \mathcal{L}_S and \mathcal{L}_T are not observable.”*

From Definition 6, it can be observed that unsupervised TL is adopted in the case that labeled data are not available at both the target and the source domains. Moreover, the target task is not the same as the source task. Compared to other TL approaches, unsupervised TL has little research work. However, recently, unsupervised TL has been getting tremendous attention from both industry and academia. The reason is that, in practice, there are many scenarios in which the labeled data are not readily available or impossible to obtain. In [30], the authors propose a self-taught clustering method to improve the learning process of unsupervised TL. The key idea of this method is clustering the source data and the target data to obtain the common features such that the unlabeled data of the source domain can benefit the learning of the target domain. In [31], the authors propose a self-supervised remedy approach to improve the learning performance of unsupervised TL by learning a discriminative space of the unlabeled data at the target domain.

D. Deep Transfer Learning

DL is a sub-domain of ML. Inspired by the human brain structure, DL uses a multi-layered architecture called Deep Neural Network (DNN) consisting of an input layer, an output layer, and multiple hidden layers between them, as illustrated in Fig. 3. DNN also intimates the human brain behaviors. Specifically, human learns how to perform a task, e.g., riding a bicycle, by practicing. The knowledge of this task is “coded” in the brain, and when the task is mastered, the brain will never forget this knowledge. Similarly, after being created, a DNN is trained to perform a specific task such as classification, clustering, or regression. During the learning process, the DNN updates its knowledge, represented by its parameters, according to the training data. After the learning phase, the DNN uses its knowledge to execute the trained task. Indeed, a trained DL model, including its architecture and parameters, can be considered as knowledge obtained from training data. Therefore, DL can be considered to be a powerful tool for TL to transfer knowledge between different domains, namely Deep Transfer Learning (DTL), as illustrated in Fig. 3. There

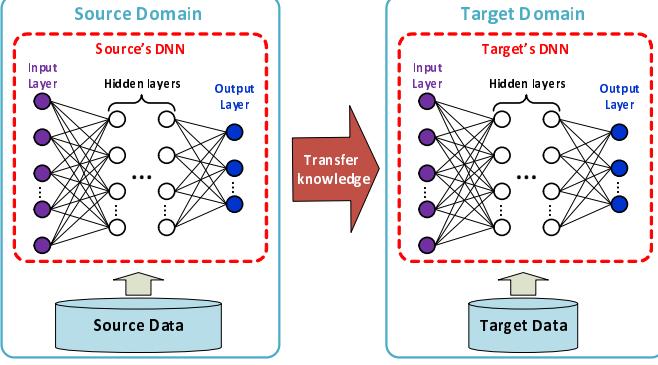


Fig. 3. An illustration of DTL.

are currently three typical strategies used in DTL, including Off-the-shelf Pre-trained Models, Pre-trained Models as Feature Extractors, and Fine-Tuning Pre-trained Models. In the following, we review different DTL strategies and discuss their advantages/disadvantages in different applications.

1) Deep Transfer Learning Strategies:

a) *Off-the-shelf pre-trained models*: Training DL models for complex tasks are data-hungry and time-consuming. For example, it may require a server equipped with 16 TPUv3 chips about three days to train the Bidirectional Encoder Representations from Transformers (BERT) model, a famous model developed by Google for Natural Language Processing (NLP). Note that a single TPUv3 chip is approximately 200 times faster than a Xeon Platinum 8163, one of the most powerful chips from Intel in 2017 when they are used to train ResNet50 [32]. Thus, such a long training process is currently one of the main obstacles hindering DL advancement. In this context, instead of training a DNN from scratch, a pre-trained model, i.e., a model trained from a neighbor domain, can be directly leveraged for the target task. For example, suppose that one wants to perform an image recognition task based on the ImageNet dataset, the pre-trained models such as VGG, GoogLeNet, or ResNet50 can be used directly with reasonable accuracy.

The first fundamental requirement for directly using pre-trained models is their availabilities (i.e., whether these models exist or not). Fortunately, many pre-trained state-of-the-art models are shared freely in the community. These models span from various domains, e.g., computer vision and NLP. Most of the famous pre-trained models can be easily obtained from the Model Zoo section of the Caffe library [33]. Alternatively, Keras, a popular DL library for Python, can provide another handy way to access these models [34].

Although this approach can be straightforwardly implemented, the main disadvantage of this strategy is that the source and target domains, e.g., image datasets, have to be the same, and the source and target tasks, e.g., classifications, need to be similar. In practice, there is hardly a situation that pre-trained models can be directly used to perform target tasks because the target domain and the target task may be similar but not identical to the source domain and the source task, respectively. In addition, if the target and source tasks and domains are quite different. For example, if the source task is

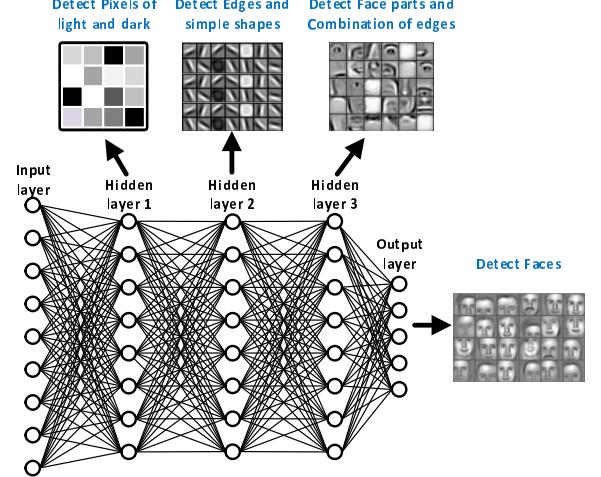


Fig. 4. Hierarchical feature representations [35].

to classify different types of cars while the target task is to recognize whether an image displays a car or not, a pre-trained model may not be able to achieve a good result. Therefore, the off-the-shelf pre-trained model is less popular than other strategies.

b) *Pre-trained models as feature extractors*: For traditional ML algorithms, such as Support Vector Machine (SVM), Decision Tree, and Logistic Regression, raw data cannot be used as input directly. It needs to be preprocessed to extract features. For example, to determine whether an image depicts a human face or not, we must manually identify the human face's unique features, such as shape, nose, and ear. Thanks to DL, the feature extraction step is entirely unnecessary since DNN can automatically learn these features. Leveraging this DL ability, a pre-trained model can be used as a feature extractor in the target domain data. In particular, target data will be passed to the pre-trained model to obtain a new feature representation before being used. As a result, this new representation embedded with the knowledge from the source domain can improve the learning process.

DL can also be considered to be a Learning Hierarchical Representation [35] because it has the ability to learn hierarchical feature representations from a dataset. Specifically, an individual layer of DNN learns different features, from general, i.e., low-level, to more specific, i.e., high-level, as going deeper in DNN. It was proven that features learned by DL are more transferable, which means that they are easier to be reused across similar domains [36]–[38]. These studies also reveal that the higher-level features largely depend on a domain and a specific task, while lower-level features are more general. For example, as illustrated in Fig. 4, when performing a face recognition, the layers closer to the input layer of DNN learn low-level features, e.g., pixels of light and dark. The middle layers learn more complex abstract features that combine these low-level features, e.g., edges and simple shapes. Finally, the layers closer to the output learn the high-level features related to specific tasks, e.g., nose, eye, and chin.

The understanding of the DNN's ability allows us to choose appropriate levels of features from an existing pre-trained

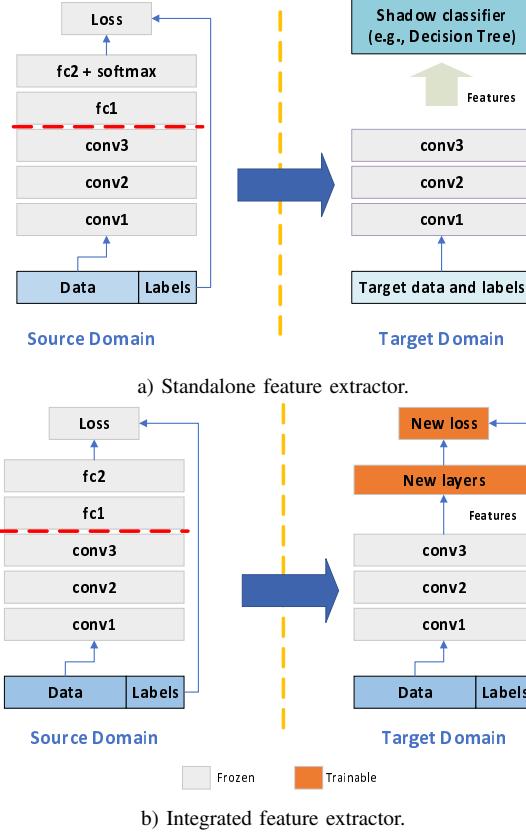


Fig. 5. Pre-trained models as feature extractors.

model. In particular, when the source and target tasks are quite different, it would be appropriate to use features obtained from the first few layers. Alternatively, if they are similar using the output of the deeper layers is more suitable. A pre-trained model could be used as a standalone or an integrated feature extractor. Fig. 5 shows examples of using a pre-trained model as a feature extractor. In the pre-train model, the first three layers are convolution (conv), following by two fully connected (fc) layers. Fig. 5(a) is an example of a pre-trained model as a standalone feature extractor. The target data are fed into the first four layers of a pre-trained model to extract features, and then the output is used to train a shallow classifier, e.g., SVM. In the integration approach, illustrated in Fig. 5(b), layers from a pre-trained model are integrated into a new DNN, but their parameters are not updated during the training process to preserve knowledge from the source model.

c) *Fine-tuning pre-trained models:* Generally, the way of leveraging pre-trained models in this approach is similar to the second strategy, i.e., Pre-trained Models as Feature Extractors. However, instead of freezing all parameters of pre-trained models, certain parts or the whole pre-trained source model can continue to be trained with target data to improve the effectiveness of transfer knowledge further. The fine-tuning of pre-trained models can be carried out in two different ways as follows:

- *Weight Initialization:* The target model parameters are first initialized by the pre-trained model's parameters.

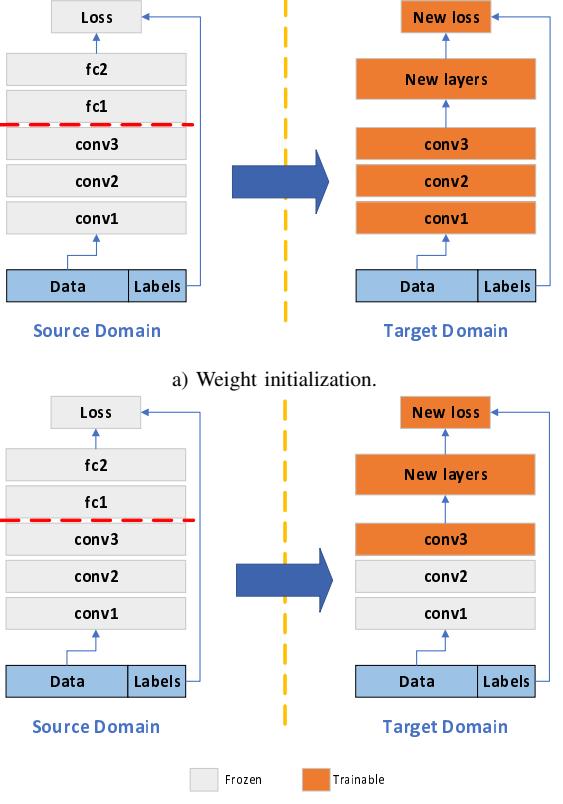


Fig. 6. Fine-tuning pre-trained models.

Then, the new model is trained with target domain data. Weight initialization is often applied when the target domain has plenty of labeled data.

- *Selective Fine-tuning:* Recall that the lower layers in a DNN can learn general features (domain-independent), whereas the higher layers can learn more specific features (domain-dependent). Based on this knowledge, we can determine how many layers of a pre-trained model should be tuned. In particular, if the target data are small and the DNN has a massive number of parameters, more layers should be frozen to prevent an overfitting problem. On the other hand, if target data are large and the number of parameters is small, more layers should be trained with new data.

Figure 6 illustrates the two types of fine-tuning pre-trained models. The target model is created by replacing two top layers of the pre-trained model with new layers. For *Weight Initialization*, as shown in Fig. 6(a), both the pre-trained layers and new layers are trained with target data until the validation loss starts to increase. For *Selective Fine-tuning* demonstrated in Fig. 6(b), the first two layers are frozen during the training process in the target domain.

- 2) *How to select the best DTL strategy:* The comparisons of DTL strategies are provided in Table II. Choosing a suitable DTL strategy depends on several factors, among which the two most important elements are the target data size and the similarity between the source and target data. Some common rules are summarized as follows:

- *Target labeled data are small and similar to source data:* In this case, the fine-tuning should not be selected because the small data may lead to overfitting problems. Instead, since the source and target data are similar, the good choice is to use a pre-trained model as a feature extractor.
- *Target data are small and very different from source data:* Since the data are small, we should choose the feature exactor approach. Moreover, it is better to transfer only the more general features from the first few layers because the source and target data are not similar.
- *Target labeled data are large:* In this case, overfitting is not a big concern, and thus *Weight Initialization* is an ideal choice.

E. Transfer Learning in Deep Reinforcement Learning

In this section, we first provide fundamental background for DRL and then discuss how TL can be applied to improve the performance of DRL algorithms.

1) Deep Reinforcement Learning:

a) *Markov decision process (MDP):* MDP is widely used to formulate dynamic and stochastic decision-making problems. Typically, an MDP is determined by four elements, including a state space \mathcal{S} , an action space \mathcal{A} , transition probabilities \mathcal{P} , i.e., the probability that a state s_t at time t moves to state s_{t+1} at time $t+1$ after action a_t is executed at time t , and a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ that returns an immediate reward r_t after performing action a_t at state s_t . A mapping from the state space to the action space is called a policy, denoted by π . The objective of MDP is to find an optimal policy π^* that maximizes an expected discounted total reward, i.e., $\pi^* = \max_{\pi} \mathbb{E}_{\pi} [\sum_{t=0}^T \gamma^t r_t(s_t, \pi(s_t))]$ where $a_t = \pi(s_t)$, $\mathbb{E}[\cdot]$ is the expectation function, and $\gamma \in [0, 1]$ is the discount factor representing the importance of future rewards. In particular, the larger the discount factor is, the more important future rewards are. In practice, due to outstanding abilities to deal with uncertainties in intelligent systems, MDP has been widely adopted to address various problems in dynamic wireless environments, such as spectrum management, cognitive radios, wireless security, and power control [39].

b) *Reinforcement learning (RL):* RL is a unique class of ML. An RL agent operates in a dynamic environment formulated by an MDP framework, and its goal is to learn an optimal policy to maximize the expected discounted total reward. During the learning process, the agent interacts with the environment and observes results to gradually find its optimal policy. Specifically, at each time step t , the agent observes current state s_t , performs action a_t according to its current policy, then receives an intermediate reward r_t and moves to new state s_{t+1} . After that, the agent adjusts its policy based on the feedback of the environment, i.e., r_t and s_{t+1} . This procedure is repeated until the agent's policy converges to the optimal one.

In practice, Q-learning is one of the most widely used RL algorithms. This algorithm requires evaluating a state-action value function $\mathcal{Q}^{\pi}(s, a)$, also called Q-function, that

specifies how good of performing an action a at a state s under policy π . The value of each pair of state and action is called Q-value. The Q-function under the optimal policy π^* is called the optimal state-value function $\mathcal{Q}^*(s, a)$. Suppose that values of $\mathcal{Q}^*(s, a)$ for all state-action pairs (s, a) are known, the RL agent can obtain the optimal policy at state s by simply taking an action that maximizes $\mathcal{Q}^*(s, a)$ [40]. The Q-learning employs a table, namely Q-table, to store and update Q-values. Specifically, each cell in the Q-table stores an estimation of the Q-value for each state-action pair. Based on interactions between the agent and the environment at time t , i.e., action a_t , intermediate reward r_t , and next state s_{t+1} , the Q-function is updated using the temporal difference (TD), which is the difference between target Q-value, i.e., $Y_t = r_t(s_t, a_t) + \gamma \max_{a_{t+1}} \mathcal{Q}_t(s_{t+1}, a_{t+1})$, and the current estimated Q-value, i.e., $\mathcal{Q}_t(s_t, a_{s_t})$, as follows:

$$\mathcal{Q}_t(s_t, a_t) \leftarrow \mathcal{Q}_t(s_t, a_t) + \zeta_t [Y_t - \mathcal{Q}_t(s_t, a_{s_t})], \quad (1)$$

where ζ_t is the learning rate that represents the impact of new interaction, i.e., TD. If the Q-function is updated by (1) and the learning rate ζ_t satisfies conditions in (2), it is proven that the policy learned by the Q-learning will converge to the optimal policy [41].

$$\zeta_t \in [0, 1], \quad \sum_{t=1}^{\infty} \zeta_t = \infty, \quad \text{and} \quad \sum_{t=1}^{\infty} (\zeta_t)^2 < \infty. \quad (2)$$

It is worth noting that although the convergence of Q-learning is proven, this algorithm can be inefficient in the case of high dimensional state and action spaces since it uses a table for estimating $\mathcal{Q}^*(s, a)$. Thus, DRL has been introduced recently as a highly-effective solution to address the current limitations of RL algorithms.

c) *Deep reinforcement learning:* In a high dimension environment, which has enormous numbers of states and actions, the traditional RL methods, e.g., Q-learning, may not handle it effectively. For example, considering the problem of designing an RL-based agent to play various video games without knowing the rules in advance, states can be represented by images consisting of millions of pixels, making it impractical to construct a Q-table. Furthermore, it is inefficient to learn all state-action values in all states separately as that of the Q-learning algorithm. These challenges can be addressed by leveraging DNN architecture for RL algorithms, i.e., DRL. The existing DRL methods can be grouped into value- and policy-based categories.

In the value-based category, an agent first needs to learn a value function, e.g., the Q-function, then finds the optimal policy based on this function. Most of the value-based DRL methods rely on the Q-learning algorithm, namely Deep Q-learning. Deep Q-learning employs a DNN instead of a Q-table to learn the Q-function $\mathcal{Q}(s, a)$. This DNN takes a state as an input and outputs Q-values for all actions so that the number of activations in the output layer equals the size of the action space. However, RL is unstable and can diverge if the Q-function is estimated by a non-linear function approximator such as a DNN because data obtained from the MDP are

TABLE II
DTL STRATEGIES

| DTL Strategy | Source and Target Domains | Source and Target Tasks | Advantages | Disadvantages |
|--|---------------------------|-------------------------|--|--|
| Off-the-shelf Pre-trained Models | Identical | Similar | - Simple for implementation. - No need to train with target data. | It is hard to obtain source model since the target and source domain must be the same. |
| Pre-trained Models as Feature Extractors | Similar | Different but related | - Flexible in model selection. - No need to train with target data. | Challenge to select a level of feature. |
| Fine Tuning Pre-trained Models | Different but related | Different | Able to utilize models from different domains. | Need adequate target's labelled data to fine-tune the source model. |

high correlation [42]. Thus, we can address this problem by applying two techniques:

- *Experience Replay*: At time t , instead of only using current transition, i.e., (s_t, a_t, r_t, s_{t+1}) , a mini-batch of previous transitions is sampled uniformly and randomly from a memory buffer \mathbf{B} to train the DNN. Accordingly, the correlation in data can be loosen.
- *Quasi-static target Q-network*: This technique employs two DNNs, the Q-network $Q(s, a; \phi)$ for estimating Q-value and the target Q-network $\hat{Q}(s, a; \phi^-)$ for estimating target Q-value. We denote ϕ and ϕ^- as parameters of the source and target Q-network, respectively. The Q-network is trained at every time step, while the target Q-network's parameters ϕ^- will be copied from ϕ at every predefined number of time steps. Therefore, the learning process is more stable.

The Q-network is trained to minimize the gap between the estimated Q-value, i.e., $Q(s_t, a_t; \phi_t)$, and target Q-value, i.e., $Y_t = r_t + \gamma \max_{a_{t+1}} \hat{Q}(s_{t+1}, a_{t+1}; \phi^-)$, so that its loss function is calculated as follows:

$$L_t(\phi_t) = \mathbb{E}_{(s, a, r, s') \sim U(\mathbf{B})} \left[\left(Y_t - Q(s, a; \phi_t) \right)^2 \right]. \quad (3)$$

The loss function in (3) can be minimized by Gradient Descent algorithm (GD), in which the Q-network's parameters are iteratively updated as follows:

$$\phi_{t+1} = \phi_t - \eta_t \nabla L_t(\phi_t), \quad (4)$$

where $\nabla(\cdot)$ is the gradient of a function, $\eta_t > 0$ is a step size at step t . To address the sampling noises, η_t can be decreased gradually to ensure the convergence. Suppose that the training process is complete, the optimal policy is given according to $\text{argmax}_a Q(s, a)$.

Although value-based DRL methods can work very well in high dimension state spaces, it may take a lot of time to calculate Q-values for all possible actions, making it impractical if the given action space is very large or continuous. Fortunately, the policy-based DRL methods can address this problem by directly learning the optimal policy without estimating any value function. In particular, the agent starts with an inaccurate policy represented by a DNN, denoted by $\pi(a_t | s_t; \theta_t)$ where θ_t denotes parameters of the DNN at time t , and then updates its policy based on interactions with the environment to achieve the best performance. Specifically, a performance according to a policy π can be defined as follows:

$$\mathcal{J}(\theta) = \sum_{s \in \mathcal{S}} d^{\pi(\theta)}(s) V^{\pi(\theta)}(s), \quad (5)$$

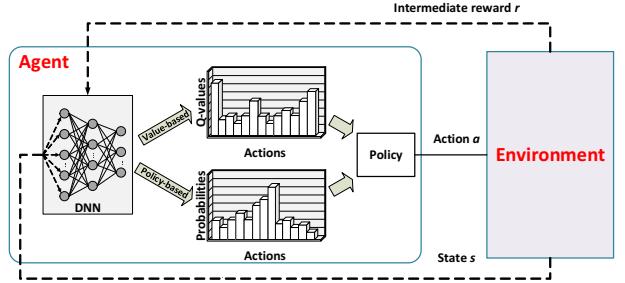


Fig. 7. DRL process.

where $d^{\pi(\theta)}$ is the steady-state distribution under policy $\pi(\theta)$, and $V^{\pi(\theta)}(s)$ is a state value function under policy $\pi(\theta)$ that specifies how good to be of a state [40]. The DNN is trained to maximize the performance function, so that Gradient Ascent (GA) can be used to iteratively update the DNN's parameters by

$$\theta_{t+1} = \theta_t + \eta_t \nabla \mathcal{J}(\theta_t). \quad (6)$$

The policy-based DRL diagram is similar to that of value-based methods, as illustrated in Fig. 7. However, the DNN in the policy-based methods performs a classification task instead of a regression task in value-based methods. In particular, the DNN in policy-based DRL methods takes a state as an input and outputs the probability distribution of actions. Then, actions are selected based on this distribution.

2) *Transfer Learning Strategies in Deep Reinforcement Learning*: DRL is one of the most data-hungry and time-consuming ML types. In particular, at the beginning of the learning process, an agent may spend a lot of time to explore the environment before acquiring an optimal policy. Therefore, the TL techniques can be used to speed up the learning process. However, the DRL's environment is formulated by the MDP framework making it complicated to apply TL. Specifically, the differences between source and target in DRL can be from any MDP elements, i.e., the state space, the action space, the reward function, the transition dynamics, or the task objective. Recall that in Section II.A, TL definition is based on the domain and task. In DRL, data are unlabelled and obtained from interactions between the agent and the environment, and the task is embedded in the MDP. Therefore, the MDP can be referred to both the domain and the task. Thus, TL in DRL can be defined as follows [43]:

Definition 7. “*Transfer Learning in DRL*: Given a source MDP \mathcal{M}_S and a target MDP \mathcal{M}_T , the goal of TL in DRL (TL-

DRL) is learning the optimal policy π_T^* for the target MDP by leveraging the source's knowledge \mathcal{D}_S , i.e., the data, the policy, and the environment dynamics, along with the target information \mathcal{D}_T :

$$\pi^* = \underset{\pi_T}{\operatorname{argmax}} \mathbb{E}_{s \sim S_T, a \sim \pi_T} [Q^{\pi_T}(s, a)], \quad (7)$$

where $\pi_T : S_T \rightarrow A_T$ is a policy of \mathcal{M}_T and estimated by a DNN trained on both \mathcal{D}_S and \mathcal{D}_T .

Based on the form of knowledge to be transferred, the existing strategies for TL-DRL can be categorized into the following groups:

a) *Reward shaping*: In *Reward Shaping* (RS), the prior knowledge from the source MDP is embedded in an additional reward function to adjust the target reward function [44], [45]. In particular, RS uses prior knowledge to design an additional reward function F for target MDP that generates extra rewards along with the environment rewards. The goal of the additional rewards is to guide the agent to select a better action. For example, an action that leads to a good state will have a higher additional reward to direct the agent to the desired policy. As a result, the agent will receive rewards generated from a new reward function $R_{RS} = R + F$ and learn a new optimal policy for a new target MDP in which the reward function is modified. Therefore, RS technique is generally applicable to different DRL algorithms because it only requires minimum changes to the DRL framework.

b) *Learning from demonstrations*: In this approach, the source knowledge forms a demonstration defined by a transition tuple (s, a, s', r) . A demonstration describes the interactions between the agent and the environment, i.e., the agent moves to state s' and receives reward r by performing action a at state s . Demonstrations are obtained through interactions between the agent and environment under a policy that could be optimal, near-optimal, or even suboptimal. Learning from demonstrations is a method aiming to help the agent efficiently explore the environment by utilizing the demonstrations from the source MDP. It can be done in two fashions, online and offline. In the offline approach, target DRL components, e.g., value function, policy, or environment dynamics model, are initialized by using supervised DL with the demonstrations in \mathcal{M}_S before the learning phase in \mathcal{M}_T . In contrast, the online methods use the source demonstrations along during the learning phase to direct the agent's actions.

c) *Policy transfer*: In this approach, the optimal policy from the source MDP will be transferred to the target MDP. Two types of policy transfer are *Policy Distillation* and *Policy Reuse*.

- *Policy Distillation*: The distillation term is introduced by [46] to refer a method for compressing the knowledge ensemble from different source models to a single target model in DL. This method then is extended to DRL with the name *Policy Distillation*, in which policies from one or more source tasks are distilled to a policy for the agent in the target MDP. Typically, a target policy is acquired by minimizing the discrepancy of action distributions between the source policy and the target policy. Because the target model is often shallower than sources models

and capable of executing multiple source tasks, *Policy Distillation* is also a compelling way for compressing model and multi-task RL.

- *Policy Reuse*: Instead of distillation from multiple policies, the second approach aims to reuse sources' policies for the target task directly. *Policy Reuse* guides the target agent's learning process according to how the source policies perform on the target MDP. Specifically, the policies, including source policies and current target policy, are directly reused based on a probability distribution P^{rb} , where the possibility of an individual policy depends on its performance, denoted by G_i in the target MDP as follows:

$$Prb(\pi_i) = \frac{\exp(tG_i)}{\sum_{j=0}^N \exp(tG_j)}, \quad (8)$$

where N is the number of policies, $0 \leq i \leq N$, and t is a parameter that increases over time. Note that G_0 is the current target policy. Typically, Q-function can be used to evaluate the performance of a policy [47], [48].

d) *Inter-task mapping*: To facilitate TL, this approach uses functions to map between the source and target MDPs. Existing studies assume that there is a one-to-one mapping function between the source and target. The subjective of mapping can be a subset or an entire state space [49]–[51] or the representation of the transition dynamic [52], i.e., (s, a, s') . Typically, in inter-task mapping over states, the source state space is split into agent-specific and task-specific representations. Then, the agent-specific space is used to learn the mapping function, and the mapped states are leveraged to reshape the reward function in the target MDP. To preserve source knowledge as much as possible, a DNN based on autoencoder architecture is utilized to learn the mapping function [49]. In inter-task mapping over transition representation, both source and target transition representation are first mapped to a higher dimension. Then, a Sparse Pseudo-Inputs Gaussian Processes is leveraged to learn the mapping function [52]. However, the second approach often relies on assumptions that the source and target are similar in terms of the transition dynamic and the state representation.

e) *Reusing representations*: Instead of learning a mapping function between source and target MDPs, state and action representations can be directly reused in the target or indirectly through a task-invariant feature space. The first way in this strategy is conducted progressively by using a progressive neural network architecture [53] consisting of a sequence of DNNs such that each DNN is trained for a particular task. Specifically, the first DNN learns the first task, then the second DNN is added to the sequence for learning the second task, and so on. While training a new task, the previous DNNs in the sequence are used as feature extractors to produce new representations. As a result, the size of the progressive network grows with the new tasks making it not scalable. The second way can address this disadvantage by using a modular neural network architecture with a fixed size. The authors in [54] propose the PathNet, in which neurons are divided into pathways. Each pathway is analog to a DNN in the progress architecture. In particular, a new pathway is

TABLE III
TL STRATEGIES IN DRL

| TL-DRL Strategy | MDP Difference | Knowledge to transfer |
|------------------------------|--|--|
| Reward Shaping | $\mathcal{M}_S = \mathcal{M}_T$ | Reward function |
| Learning from Demonstrations | $\mathcal{M}_S = \mathcal{M}_T$ | Demonstrations, i.e., (s, a, s', r) |
| Policy Transfer | $S_S \neq S_T$ and $A_S \neq A_T$; $\mathcal{R}_S \neq \mathcal{R}_T$ | Source agent's policy |
| Inter-task Mapping | $S_S \neq S_T$; $S_S \neq S_T$ and $A_S \neq A_T$; $\mathcal{R}_S \neq \mathcal{R}_T$ and $A_S \neq A_T$; $S_S \neq S_T$ and $\mathcal{R}_S \neq \mathcal{R}_T$; $S_S \times A_S \neq S_T \times A_T$ | Source MDP |
| Reusing representations | $S_S \neq S_T$; $S_S \neq S_T$ and $A_S \neq A_T$; $\mathcal{R}_S \neq \mathcal{R}_T$ and $A_S \neq A_T$; $S_S \neq S_T$ and $\mathcal{R}_S \neq \mathcal{R}_T$; $S_S \times A_S \neq S_T \times A_T$ | Representations of states or transition dynamics |

created by randomly selecting neurons to learn the new task. The new pathway's parameters are only updated during the training if they are not in the previous pathways. The modular architecture is also employed to split the policy network into a task- and environment-specific module [55]. The idea behind this division is the task-specific module can be applied to different environments where the task is performed, while the environment-specific module can be applied to different tasks in the same environment.

3) *How to select the best TL-DRL strategy:* In comparison with DRL, TL-DRL is indeed more complex since it involves MDPs. To select an appropriate TL-DRL scheme, it is critical to determine the differences between \mathcal{M}_S and \mathcal{M}_T , which can be in the state space, the action space, the reward function, and the environment dynamics, because this information will define the type of knowledge that can be transferred. Another important aspect needed to be considered is whether a specific TL-DRL method restricts a type of RL algorithm. For example, the authors in [56] propose the Deep Q-learning from the Demonstration framework that can be applied only for Deep Q-Learning. More comparisons for the aforementioned TL-DRL strategies can be found in Table III.

III. APPLICATIONS OF TRANSFER LEARNING FOR SPECTRUM MANAGEMENT

As the demands for frequency spectrum are continuously growing, effective spectrum management becomes a critical issue for wireless networks. Traditional ML techniques have been widely applied in spectrum management for various tasks, such as resource allocation, channel estimation, and spectrum sensing. Nevertheless, there are two main obstacles that hinder the effectiveness of the traditional ML techniques. First, the radio environment is subject to significant variations. This negatively impacts the performance of the ML models which are trained only for a specific scenario. Second, many spectrum management tasks are time-sensitive, and thus there is much less time available to train the ML models. To address these problems, TL has emerged to be an effective solution which can utilize the knowledge gained from similar scenarios in order to achieve highly-effective learning processes.

A. Cognitive Radio Networks

Cognitive Radio Network (CRN) is an effective solution to the ever-growing demands of spectrum-based communications. The backbone of CRN is the cognitive radio techniques that can make optimal spectrum sharing decisions under dynamics and uncertainties of radio environments. To enable such

intelligent and adaptive decision makings, conventional ML techniques have been leveraged for various tasks such as spectrum sensing, handoff prediction, and interference management. However, these ML techniques require a lot of training time and data to be effective for a specific task, which is disadvantageous in the ever-changing radio environments. To improve the effectiveness and robustness of these ML techniques, various TL approaches have been proposed to utilize knowledge from different but related scenarios, i.e., nearby devices or regions.

1) *Spectrum Sensing:* Spectrum sensing refers to the techniques to allow the Secondary Users (SUs) to detect activities of Primary Users (PUs) on the primary channels, and thereby effectively utilizing the channels under the presence of the PUs [57]. Although the traditional ML approaches have been considered effective solutions for spectrum sensing, the sensing performance is significantly reduced when the domains change, i.e., another type of signal or environment. Thus, transductive TL approaches are proposed in [58] and [59] to utilize the knowledge gained from other signal types and environments. Particularly, as illustrated in Fig. 8, the authors in [58] propose a *Fine-tuning Pre-trained Models* TL strategy for spectrum sensing in CRNs. Particularly, a baseline Convolutional Neural Network (CNN) model trained with various types of signals data is transferred to a target task (one type of signal), and then the model is fine-tuned using the target data. Simulation results show that TL can improve the performance, i.e., detection probability and false-alarm rate, of the CNN by up to 25%. However, in the cases of limited target data, the proposed approach performs worse than other non-ML methods, e.g., energy detectors. Similarly, a TL approach is proposed in [59], which also employs the *Fine-tuning pre-trained models* TL strategy. Specifically, the approach transfers a CNN trained with different types of signal data and noise data to the target task and fine-tune the model with target data. This TL approach is very similar to the one in [58], except that [58] does not consider noise data. Experimental results show that the proposed TL process can improve the detection probability of the CNN by up to 30% under low signal-to-noise ratio (SNR) levels.

Different from the approaches in [58] and [59] that utilize knowledge from different types of signals, an inductive TL process is proposed in [60] that leverages knowledge from different environments. Specifically, a four-layer CNN is developed to extract features, and a linear SVM is responsible for classifying signals based on the extracted features. To improve the robustness of the proposed approach, a CNN trained with

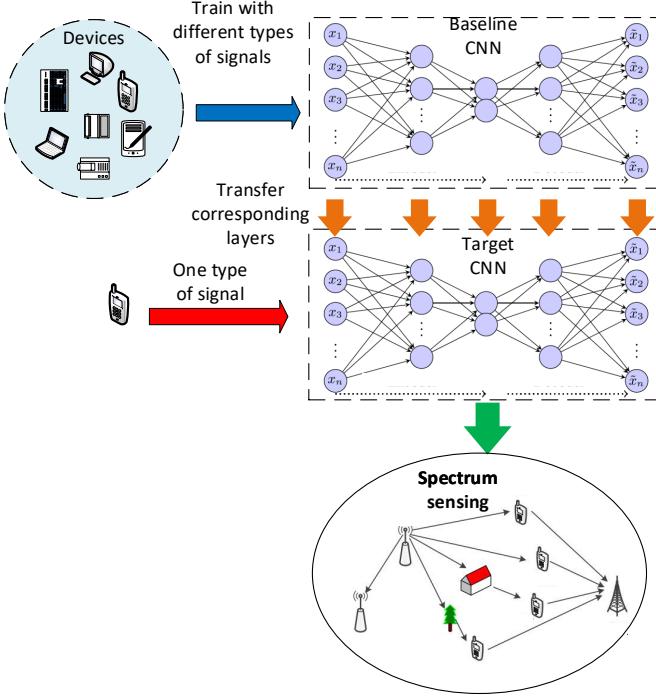


Fig. 8. The TL-based spectrum sensing approach [58].

data from a specific environment is transferred to another environment to train with the target data, while the first two layers of the CNN are frozen. The reason for freezing only the first two convolutional layers is because these layers learn the more general features which are common across different domains. Experimental results show that the TL process can significantly improve the training time and sensing time of the CNN by up to 94% and 20%, respectively.

2) *Channel Selection*: In overlay CRN, the SUs can choose to switch to another channel to avoid interference with the PUs at the current channel [61]. To reduce the delay of such switching process, the SUs can predict when a channel will be occupied by the PUs and then select idle channels for switching in advance. The first step, spectrum prediction, is an important but challenging task. To predict future spectrum state, historical data can be leveraged by conventional ML techniques. However, for a certain spectrum band, historical data might be scarce, which significantly reduces the prediction accuracy. Therefore, a TL approach is proposed in [62], which utilizes historical data from other spectrum bands. Particularly, a two-layer long short-term memory (LSTM) model is developed to predict available channels in a sliding time window manner. Then, the authors propose a transductive TL process to transfer the knowledge (features) learned from different frequency bands to overcome the lack of spectrum data for learning. To further improve the effectiveness of the TL process, the authors utilize the transfer component analysis to choose suitable features for transferring. Experimental results show that the proposed TL approach can improve the accuracy of the LSTM model by up to 2.5 times, especially when the target task data are limited. However, when the domain data are sufficient, training the LSTM without TL

could be better. Moreover, the TL is only effective when the knowledge is transferred between certain pairs of frequency bands, and thus similarity analysis of the frequency bands is needed.

For the second step, i.e., channel selection, RL is a very effective learning technique for the SUs to find the best channel under the dynamic and uncertainty of primary channels. However, RL techniques usually require a long learning time to find the optimal strategy. Therefore, several TL approaches have been recently proposed to speed up the learning process. Specifically, Q-learning approaches are developed in [63]–[65] to find the optimal handoff strategies for SUs. To speed up the learning process of the newly joined SUs, both [63] and [64] employ the *offline Learning from Demonstration* TL strategy. Particularly, the authors propose to transfer the complete Q-table of the nearest SU to serve as an initial point for the new SU. Simulation results show that the TL approaches in [63] and [64] can improve the convergence rate of the new SU's learning process by up to 30% and 14%, respectively. Unlike [63] and [64], [65] proposes to transfer the knowledge of an expert SU selected based on its similarities with the new SU in terms of channel statistics, node statistics, and application statistics. Particularly, the level of similarity is determined using the Bregman divergence of the channel statistics, node statistics, and application statistics among the SUs. Then, the SU with the highest similarity is chosen as the expert SU for the TL process. This TL process aggregates the expert SU's policy π_S and the new SU's policy π_T with a transfer rate θ , i.e., $\pi = (1 - \theta)\pi_T + \theta\pi_S$. At the beginning, π_S performs better since the time to learn the new policy is still short. As the learning time increases, the π_T gradually becomes better. Thus, the authors propose to diminish the transfer rate as the learning proceeds. Simulations are carried out to evaluate the effects of different transfer rates on the Q-learning model. The results show that a high transfer rate ($\theta = 0.8$) achieves better results at the beginning of the learning process than that of the lower transfer rates ($\theta = 0.2$ and $\theta = 0.5$). However, their performances become equal as the Q-learning model has more time to train. Another TL-based deep Q-network (DQN) is developed in [66] for channel selection in CRNs. Particularly, a DQN is proposed to select channels with high QoS for file transmission. The authors propose to transfer the Q-values of the adjacent agents to the newly joined agent. A source agent selection policy is proposed, which selects the agents causing large interference. Then, at the TL process, the Q-values are reversed (i.e., positive rewards become negative) to encourage the new agent to avoid the channels that the source agents have chosen. Simulation results show that TL can significantly improve the throughput of the RL approach, especially in high traffic scenarios.

3) *Interference Management*: In CRN underlay, the SUs can share the same channel as the PUs, as long as the signals from the SUs do not cause unacceptable interference for the PUs. To control the interference, the SUs can control their transmission power [61]. However, in such constantly changing environments, finding an optimal power control policy is not an easy task for conventional ML techniques because of the unsteady number of users sharing the same

resource (different domains) and the small time-window to react (short time to learn). To overcome these limitations, a TL approach is proposed in [67]. Particularly, a distributed Q-learning approach is proposed for interference management in femtocells (SUs) that share the available radio resources with macrocells (PUs), aiming to find the optimal power levels that femtocells can assign to each resource block. Since the domains of the Q-learning models of the resource blocks are similar, the authors propose an unsupervised TL approach which transfers the full Q-table of the previous resource block to the target task resource block. Simulation results show that the TL process can reduce the probability that the macrocell capacity falls below a certain threshold by up to 20 times.

Different from [67] that utilizes the knowledge from previous experiences, the TL approaches developed in [68] and [69] transfer the knowledge from nearby SUs. Specifically, in [68], a DQN, which uses a neural network to estimate the Q-function, is developed for the SUs to determine their optimal power allocation. In CRNs, SUs that are close to each other may have similar environment parameters, and thus their Q-values are also similar. Thus, to speed up the learning process of the DQN, when a new SU joins the network, the Q-function and parameters of the nearest SU can be transferred to the new SU. Similarly, a TL-based DQN is proposed for optimizing spectrum utilization in [69], which transfers the Q-values from the nearest SU to the new SU. Simulation results show that TL can reduce the learning time by approximately 25% and 80% in [68] and [69], respectively.

4) *Radio Map Construction*: Accurate radio map construction can play a crucial role in making optimal resource allocation decisions [70]. Practically, the radio maps are often constructed using measurements from a limited number of measuring devices. To construct the complete radio map from the limited measurements, ML techniques are effective solutions. However, conventional ML techniques are facing serious challenges such as the time-consuming training process and the scarcity of data. To address these problems, transductive TL approaches are proposed in [71], [72]. Particularly, [71] proposes a feed-forward NN to reconstruct the radio map corresponding to an antenna tilt configuration, i.e., antenna tilt degree. To address the lack of data in new tilt configurations, the authors propose a TL strategy to transfer knowledge from existing tilt configurations of the same antenna. Specifically, the authors propose to transfer a number of layers of the feed-forward NN and then retrain the model for the new target task. Experiments are conducted with different numbers of transferred layers to find the optimal TL strategy. The results show that transferring only the first two layers of the feed-forward NN yields the best result, and generally the proposed TL can achieve higher prediction accuracy (up to 12%) compared to that of other approaches such as K-nearest Neighbors (KNN) and random forest. Different from [71], a TL approach is presented in [72], which transfers the knowledge gained from the other regions. In particular, a generative adversarial network (GAN) is developed to estimate the power spectrum maps for underlay CRNs. To address the lack of data, the authors propose to transfer the radio environment features gained from the other regions to the target region.

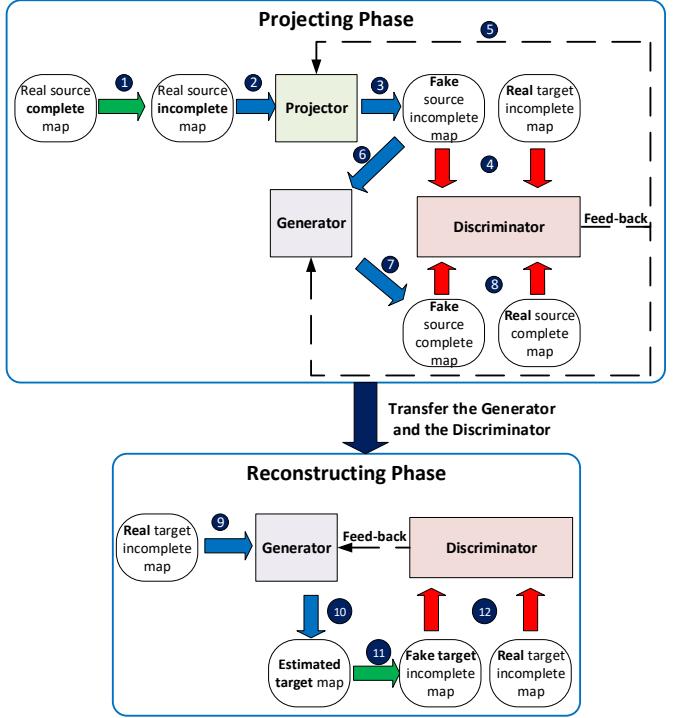


Fig. 9. The GAN-based TL approach [73].

Particularly, as illustrated in Fig. 9, the TL process consists of two phases. In the projecting phase, the complete maps from the source are first fed into a projector, i.e., a DNN, to create (adversarial) incomplete maps. Then, the discriminator will try to distinguish the created adversarial maps and the real maps from the target task. Once the training process is completed, the projector can create incomplete maps very similar to those of the real target task. Then, these fake incomplete maps are fed into a generator to create fake complete maps. The discriminator then is fed with the fake and real complete maps to train the generator. At the end of the projecting phase, the generator can create complete maps that are very similar to the real complete maps. At the beginning of the reconstructing phase, the real target incomplete map is fed into the generator to create the estimated full map. Then, these estimated full maps are transformed back into incomplete maps and fed into the discriminator to train the generator. Once the discriminator cannot distinguish between the fake and real incomplete maps, the training is completed. The generator is finally capable of creating estimated maps based on real incomplete maps. Simulation results show that the proposed TL-based approach provides a more accurate power spectrum map reconstruction performance than those of the traditional methods without TL. However, the effect of the TL process is not comprehensively evaluated.

B. Resource Allocation

In wireless networks, effective resource allocation is an important but also challenging task. Making optimal decisions regarding spectrum, resource block, and power allocation, load balancing, on-off switching, etc., can significantly improve the

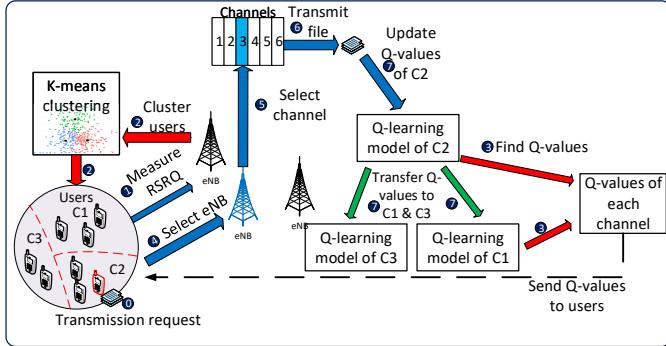


Fig. 10. The TL-based eNB and channel selection approach [73].

network's operation efficiency and user experience. For such optimization problems, ML has emerged to be an effective solution in recent years due to its ability to quickly approximate the optimal solution without requiring expert knowledge of the problem [74]. However, traditional ML models often do not perform well when data is lacking and require a long time to train. To address these problems, TL has been leveraged for many applications as discussed in the following.

1) *Channel Assignment:* Although RL approaches have been widely used due to their outstanding features in handling dynamic issues in resource allocation, there are still some limitations that can be addressed by TL. In [75], a Q-learning algorithm is developed to find the best channel for file transmissions in flexible opportunistic networks. In such flexible opportunistic networks, the network topology may change significantly over time, e.g., due to the aerial eNBs leaving the network. This requires a long time for the traditional Q-learning to learn the optimal channel selection for the new topology. Therefore, to speed up the learning process, a TL strategy is proposed to transfer the Q-values from previous experiences to the current model. Simulation results show that the TL process can improve both the convergence speed and the performance of the Q-learning model. Extended from [75], an unsupervised TL approach is proposed in [73] to optimize the eNB and channel selection for file transmissions. As illustrated in Fig. 10, when there is a file transmission request, the users are clustered based on their Reference Signal Received Quality (RSRQ) measurements. Then, Q-learning is adopted by the eNBs to find the Q-values of each channel for each cluster. The obtained Q-values are then sent to the users for eNB selection. When an eNB is selected, it transmits the file using the channel with the highest Q-value. The result of the transmission, i.e., success or failure, is used to update the Q-values of that cluster. Finally, the Q-values are transferred to other clusters. Simulation results show that the proposed TL approach can reduce the delay and transmission failure rate up to 1.5 times compared to the approach without TL.

2) *Power Allocation and Energy Efficiency:* TL-based approaches have also been developed for effectively learning the optimal control policy, e.g., transmit power allocation and on-off switching of wireless stations and devices, to improve the energy efficiency of wireless networks. Particularly, in [76], a TL-based RL approach is developed for managing access points in Wi-Fi networks, aiming to improve the energy

efficiency by switching on/off the access points. Particularly, an actor-critic RL model is proposed to learn the optimal AP switching scheme based on Wi-Fi traffic data. To improve the learning speed of the RL model, the authors propose to transfer the knowledge gained from previous experience, i.e., previous policy, to the current model. Similar to [65], the previous policy and the new policy are aggregated with a transfer rate, and the transfer rate is diminished as the learning proceeds. Experimental results show that the proposed TL process can improve the energy efficiency of the approach without TL by up to 22%. In [77], a TL-based RL approach is proposed for managing Base Stations (BSs) in radio access networks. Particularly, an actor-critic RL model is proposed to learn the optimal BS switching scheme. To further improve the efficiency of the RL model, the authors propose to transfer the knowledge gained from previous experience and neighboring regions. Similar to [76], the authors propose to aggregate the previous and new policies with a predefined transfer rate, and this transfer rate can be diminished over time. Simulation results show that the proposed TL process can improve the energy efficiency of the actor-critic model by up to 60%, and a high transfer rate (0.5) achieves better results. In [78], a DRL approach is proposed for mode selection and resource management in Fog Radio Access Networks (F-RANs), with the main aim to reduce the network's power consumption. In the considered system, there is a network controller that will determine the user equipments' communication modes and processors' on-off states to minimize the system's long-term power consumption. To this end, the controller utilizes a DQN to learn and find the optimal control policy for the UEs. To speed up this learning process, the authors propose to use TL to transfer the knowledge of the previously trained DQN to the new DQN when the environment changes. Particularly, the authors employ *Weight Initialization*, i.e., transfer the whole weights of the source model to serve as an initial point for the target model. Simulation results show that applying TL can help to speed up the learning process by 37.5% when the environment changes moderately. However, in this work, TL is ineffective when the environment has major changes. In [79], a DL approach is proposed to find the optimal resource allocation policy to minimize the total power consumption of a BS in 5G networks. To this end, the authors develop a cascaded structure of NNs, where the tasks of optimizing bandwidth and transmit power allocation are distributed to the first and second NNs, respectively. Since the wireless environment changes frequently, a DTL approach is proposed. With this TL approach, the NNs are trained in offline mode, and then the first several layers of the two NNs are transferred to the NNs of the target task. Then, the NNs are trained online with the target data. Simulation results show that the TL approach can help the training process to converge much faster, i.e., by 2000-3000 epochs.

3) *Resource Block Allocation:* TL-based approaches can also be used to address resource block allocation problems in wireless networks. In [80], an actor-critic model is developed for assigning transmission blocks to vehicles in Vehicle-to-vehicle (V2V) communications. To this end, the authors propose to employ two DNNs as the actor and critic to train

RL agents, each of which is responsible for assigning resource blocks to vehicles in its designated area. To speed up the learning process of the DNNs in new environments, the DNNs are trained in simplified environments and then transferred to more complex environments to fine-tune. Specifically, the authors employ *Weight Initialization* and fine-tuning all the layers of the DNNs. Simulation results show that the proposed approach outperforms the other methods in terms of packet reception ratio. However, in this work, the impact of the TL process on the DNNs is not evaluated. Similarly, an echo-state-network-based approach is developed in [81] for resource block allocation for wireless virtual reality networks. The authors first formulate the resource allocation problem as a non-cooperative game among the considered small BSs. Then, a RL model based on echo state network is proposed to find the Nash equilibrium of the game. In this game, once the environment changes, e.g., the number of small BSs increases, the RL model has to learn new utility values again, which is time-consuming. Thus, the authors propose a TL approach to transfer previous knowledge, i.e., previous outputs of the RL model, to reduce the time it takes to train the model on new utility values. Particularly, an echo state network is developed to find the relationship between the changes in the environments and the resulting changes in the utility functions. In other words, the effects of the environments on the utility function can be estimated, and thus the RL can learn the new utility values faster. Simulation results show that the TL approach can help to speed up the learning process by 18.2%.

4) Resource Utilization Prediction: In addition to optimizing resource allocation, TL approaches can also be employed to predict resource utilization in wireless networks. In particular, the authors in [82] propose an LSTM to sequentially predict a number of resource utilization parameters. A TL process is employed, which transfers all the layers of a pre-trained model to the target model. Then, the target model is trained with the target data to fine-tune the several last layers, while the first layers are fixed. This TL strategy is expected to be beneficial because the first layers capture more general features, and training only the last layers is more computationally efficient. Simulation results show that the TL process can reduce the training time of the target model approximately by half.

C. Channel Estimation and Prediction

The physical properties of the wireless communication channels are always changing, which causes attenuation, distortion, delays, and phase shift of the signals. Therefore, it is essential to provide accurate and up-to-date predictions of the channel to overcome these limitations and facilitate signal processing. For such prediction tasks, conventional ML techniques have been widely applied. However, these ML techniques usually require a lot of data, have a low learning speed, and are not robust to environmental changes. To address these issues, various TL approaches have been developed. In [83], a DL approach is proposed for 5G/B5G cellular traffic prediction. Particularly, a spatial-temporal cross-domain neural network model is developed for traffic load prediction

of various services such as SMS, call, and the Internet. The proposed model consists of multiple smaller models to simultaneously learn the features of different datasets including spatio-temporal (traffic level at different places over time), timestamp (weekdays or weekends), and cross-domain (BSs and their coordinates) datasets. The extracted features from these datasets are finally fed into a densely connected CNN for traffic level prediction. Since the traffic level of different services has a high correlation, e.g., all services' traffic levels are lower at midnight, a transductive TL process is proposed to transfer a model of one service to another service. Another transductive TL approach is proposed to predict traffic levels of different cellular services in [84]. Similar to [83], this approach also extracts features from different datasets and fuse them to feed into a densely connected CNN. The proposed TL process in [84] is also similar to that in [83], which transfers the model of one service to another. However, the main difference between the two approaches is that [84] does not use the timestamp dataset. Experimental results show that TL can help to improve the performance, in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), of the proposed model in [83] and [84] by up to 17.1% and 12.9%, respectively.

Apart from service traffic level, ML techniques can also be leveraged to predict other parameters such as Channel State Information (CSI), channel quality, and Radio Frequency (RF) distributions. In such tasks, the knowledge from similar scenarios, e.g., frequency bands, can be leveraged to reduce the learning time and improve the prediction accuracy. For example, a transductive TL approach is proposed in [85] to adaptively learn and predict dynamic RF environments. Particularly, a spatio-temporal Gaussian process model is developed to predict the distribution of RF variations. Since the RF environments are dynamic, the authors adopt TL to leverage the knowledge gained from different environments to reduce the training time for a new environment. Moreover, the authors investigate the trade-off between training time and prediction accuracy, the authors propose and compare three TL approaches with varying degrees of transferring, i.e., transferring all hyperparameters, transferring lengthscales and noise hyperparameters, transferring only the lengthscales parameters. Experimental results show that the more hyperparameters are transferred, the more training time can be reduced because the new model needs to learn fewer parameters. However, more transferred parameters may lead to a worse prediction accuracy because of the differences between the environments. Thus, more investigations on fine-tuning methods are needed.

The authors in [86] focus on evaluating different TL approaches for traffic prediction in wireless networks. Specifically, the authors develop and compare a CNN and an LSTM to predict channel quality indicator (CQI) in two scenarios, i.e., different frequency bands of the same 4G cell and different cells using the same frequency band. To address the scarcity of labeled data in the target domains, e.g., a particular frequency band or cell, the authors propose a transductive TL strategy to utilize knowledge from similar domains. Particularly, the authors propose to transfer the first layers of the source models, e.g., a cell or frequency band with sufficient data,

to the target domain for fine-tuning. For both the CNN and the LSTM, all the layers except the last one are transferred to the source model, and the target models are then fine-tuned with target data. Simulation results show that the TL process can improve the prediction accuracy, and the CNN performs better than the LSTM if there is no target data, i.e., the CNN's RMSE value is 6% lower than that of the LSTM. However, when there is more data available, the LSTM's RMSE value is 3% lower than that of the CNN.

D. Other Issues in Spectrum Management

In [87], a TL approach is proposed to predict path loss and delay spread for Unmanned Aerial Vehicle (UAV) communication systems. Specifically, the authors propose two algorithms, i.e., random forest and KNN, for path loss and delay spread prediction based on features such as the UAV's coordinates and the presence of buildings between two UAVs. To address the lack of data issue, especially in a new scenario or at a new frequency, the authors develop two TL strategies, namely frequency-based TL and scene-based TL. The first strategy transfers the models trained at different frequencies to the target frequency, whereas the scene-based TL transfers the models trained at different scenarios but at the same frequency. In both strategies, all the weights of the pre-trained models are transferred to the target models, and then the target models are fine-tuned with target task data, i.e., *Weight Initialization*. Simulation results show that both the frequency-based TL and scene-based TL achieve lower RMSE compared to other approaches.

A TL approach is also used to address the lack of data for configuration troubleshooting problems in femtocell networks [88]. Particularly, an SVM is developed to classify the misconfiguration cases. Since the data in a femtocell network is often scarce due to the small number of users, the authors propose to transfer the output of the other femtocells' models to the target task. Specifically, the authors first determine the similarities among the femtocells by calculating the Kullback-Leibler divergence among their data. Then, each femtocell's model is used to classify the target task. The outputs of these models are then aggregated to be the the output of the target task. Particularly, the final output is the weighted sum of all the models' outputs, and the weights are the similarities among the femtocells. Simulation results show that the proposed TL approach can improve the SVM's prediction accuracy by up to 20%.

Summary: In this section, we discuss diverse applications of TL approaches in order to address existing limitations and challenges of conventional ML techniques for spectrum management problems such as the lack of labeled data, the heterogeneity of devices and environmental conditions, and the slow learning process. As summarized in Table IV, it can be observed that knowledge from previous experiences and similar network conditions can be leveraged to address the lack of data. Furthermore, to improve the ML techniques' robustness to different environments, a baseline model trained with generalized datasets can be utilized. Alternatively, to improve the learning speed, especially in distributed learning

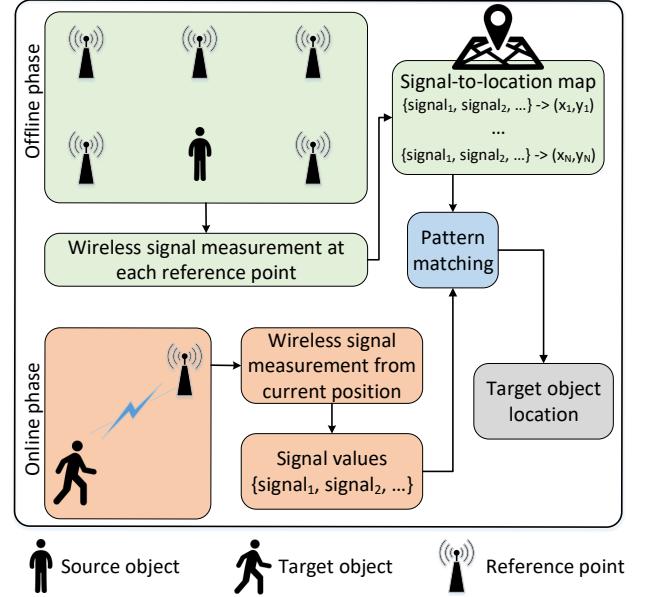


Fig. 11. The general indoor localization framework.

systems, expert knowledge from similar learning agents can be used to initialize the new agent's model. Although the effectiveness of the TL approaches has been demonstrated for many spectrum management problems, there are still several challenges. For example, TL is only effective when the source and the target tasks share many common features. Thus, finding the right source for transfer necessitates proper analysis of the feature spaces of both tasks. However, most of the aforementioned works do not analyze the tasks' domains. Moreover, once the source is selected, there is still a need to determine which part of the source model to be transferred. For example, several approaches transfer only the first layers of the ML models because these layers capture more general features, whereas other TL strategies transfer the whole source model.

IV. APPLICATIONS OF TRANSFER LEARNING FOR LOCALIZATION

Localization plays a very significant role for many applications in order to estimate the location/position of objects, e.g., mobile users, in future wireless networks including smart homes and smart cities. Typically, to implement the localization, wireless signals, e.g., received signal strength (RSS), CSI, and sensor signals, from wireless devices can be used to determine the position/location of the target objects in considered areas. Nonetheless, due to the time-variant signal characteristics of wireless environments, it is costly and impractical to collect and calibrate up-to-date signal data frequently. To address these issues, TL approaches can be applied with minimum signal data re-collection and re-calibration. In this section, we discuss applications of TL to address limitations of localization problems for both indoor and outdoor environments.

TABLE IV
SUMMARY OF TL APPROACHES FOR SPECTRUM MANAGEMENT

| No. | Problem | Knowledge to transfer | ML method | TL type |
|------|---|--|--------------------|--------------|
| [75] | Resource allocation | Q-parameters from previous experiences | Q-learning | Unsupervised |
| [73] | | Q-values from other user clusters | Q-learning | Unsupervised |
| [76] | | Knowledge from previous policies | Actor-critic model | Unsupervised |
| [77] | | Knowledge from previous policies and neighbor regions | Actor-critic model | Unsupervised |
| [78] | | Q-values from other environments | DQN | Unsupervised |
| [79] | | Transfer the NNs pre-trained with offline data to use online | Cascaded NNs | Inductive |
| [80] | | Transfer a pre-trained model | Actor-critic model | Unsupervised |
| [81] | | Knowledge from previous experience | Echo state network | Unsupervised |
| [89] | | Transfer a model pre-trained with offline data to use online | ANN | Inductive |
| [82] | | Transfer a pre-trained model | LSTM | Transductive |
| [58] | Cognitive Radio Network: Spectrum sensing | Knowledge from different types of signals | CNN | Transductive |
| [59] | | Knowledge from different types of signals and noise data | CNN | Transductive |
| [60] | | Knowledge from other environments | CNN, SVM | Transductive |
| [62] | Cognitive Radio Network: Channel selection | Knowledge from other frequency bands | LSTM | Transductive |
| [63] | | Q-table from similar SUs | Q-learning | Unsupervised |
| [64] | | Q-function from nearest SUs | Q-learning | Unsupervised |
| [65] | | Knowledge from a similar SU | Actor-critic model | Unsupervised |
| [66] | | Knowledge from adjacent agents | DQN | Unsupervised |
| [67] | Cognitive Radio Network: Interference management | Q-table from previous resource blocks | Q-learning | Unsupervised |
| [68] | | Q-values of nearby SUs | DQN | Unsupervised |
| [69] | | Q-values of the nearest SU | DQN | Unsupervised |
| [71] | Cognitive Radio Network: Radio map construction | Knowledge from other tilt configuration | Feed-forward NN | Transductive |
| [72] | | Knowledge from other regions | GAN | Transductive |
| [83] | Channel Estimation and Prediction | Transfer a pre-trained model | LSTM, NN, CNN | Transductive |
| [84] | | Transfer a model trained with a different dataset | LSTM, CNN | Transductive |
| [85] | | Transfer parameters of a pre-trained model | LSTM, CNN | Transductive |
| [86] | | Knowledge from different cells and different frequencies | CNN, LSTM | Transductive |
| [87] | Predict path loss and delay spread for UAV communication system | Knowledge from different frequencies and scenarios | Random forest, KNN | Transductive |
| [88] | Configuration troubleshooting in femtocells | Knowledge from different femtocells | SVM | Transductive |

A. Indoor Localization

In indoor environments, the location of an object, e.g., a mobile device, can be estimated by using wireless signals from diverse reference points, e.g., Wi-Fi hotspots through considering offline and online phases. As illustrated in Fig. 11, the reference points can first measure the set of wireless signals received from a source object. Then, a signal-to-location map, i.e., a correlation map to localize an object based on the captured signals, can be deployed offline. In the online phase, wireless signals from a target object in a specific position/location can be measured using a reference point. Upon obtaining the set of wireless signals from the target object, they can be stored for the signal pattern matching, i.e., mapping the most similar wireless signal patterns in the online phase with the ones in the offline signal-to-location map, to estimate the current position/location of the target object. However, the existence of dynamic signal behavior in indoor environments may trigger outdated signal-to-location map. For

that, TL approaches can be used efficiently to address that problem for indoor localization without the need for frequent signal data updates. In the following, we discuss how wireless signals can be utilized to perform the TL process efficiently.

1) *RSS Map-Based Localization*: In a localization problem, the RSS map, i.e., a map which contains the correlation between RSSs and positions/locations of the object, can be generated based on the measurement of RSSs estimated by the reference points (when detecting objects within their considered areas). The RSS map can be easily produced without requiring additional hardware. Additionally, the use of RSS can provide more accurate location than that of the Global Positioning System (GPS) due to the robustness to the obstacles in a small-scale wireless coverage. As a result, RSS map has been widely used as a transferred knowledge in TL methods.

In [90], the authors introduce an adaptive localization prediction leveraging inductive TL approach. In particular, a mapping function between RSS and physical location data

in different periods is first derived based on a Manifold co-Regularization learning method [91]. Then, using the TL method, the mapping function is adjusted for various discrete time dynamically by extracting the meaningful knowledge used to estimate location in another time with minimum recalibration. Extended from the work in [90], the authors in [92] develop the Wi-Fi-based indoor localization problem (WILP) as an inductive TL problem considering the RSS map transfer in different scenarios, i.e., across time periods, locations, and devices. This aims to build an accurate localization model for different periods, areas, and devices with few labeled data separately. Specifically, if the user information, e.g., timestamp of the captured RSS, is available offline and online, a hidden Markov learning model is used to predict discrete location grids. Otherwise, the Manifold co-Regularization learning method can be applied to classify the location grids. With the real experiments, the proposed dynamic TL method can outperform the static non-TL method by 10% in terms of the cumulative probability.

Despite considering the indoor localization problem in different time period, location, and device scenarios, the TL approach in [92] is implemented in each scenario independently. To accommodate more complex conditions adaptively, the authors in [93] study an inductive TL method considering both time and devices simultaneously due to variation of signal distributions. Particularly, a dimensionality reduction method, i.e., manifold alignment, is used to build a mapping function between different RSS and physical location spaces in a low-dimensional space. Then, the built RSS map is utilized to transfer knowledge from an old model to a new model. Through experiments, the joint consideration of time and devices for the TL approach can gradually improve the localization accuracy compared with those of non-TL methods. To investigate the impacts of environment changes on the fingerprinting values, the joint use of fuzzy clustering and inductive TL methods for indoor localization through updating the RSS map is studied in [94]. Particularly, the use of fuzzy clustering can reduce the effect of environment changes to the considered areas. With the obtained result from the fuzzy clustering, the RSS map can be updated using the manifold alignment-based TL approach to improve the accuracy of RSS map for the localization. The performance evaluation shows that the mean location error can be improved up to 71% compared with that of the work in [93].

Although the aforementioned works can reduce the training cost and complexity (through applying RSS map adaptation over various time and devices), they still need to extensively deploy a complete map through fingerprinting to directly localize mobile devices with high accuracy. Additionally, the map accuracy may be reduced when the map adaptation is repeated for several iterations, leading to higher localization errors. To address this problem, the authors in [95] design an inductive TL-based indoor localization scheme without creating a complete RSS map. Particularly, a source dataset with the same spatial correlation of the RSSs is first generated. Then, using the manifold alignment, the RSS map is transferred to a few fingerprints and several RSSs observations with unknown locations, aiming at executing the direct localization

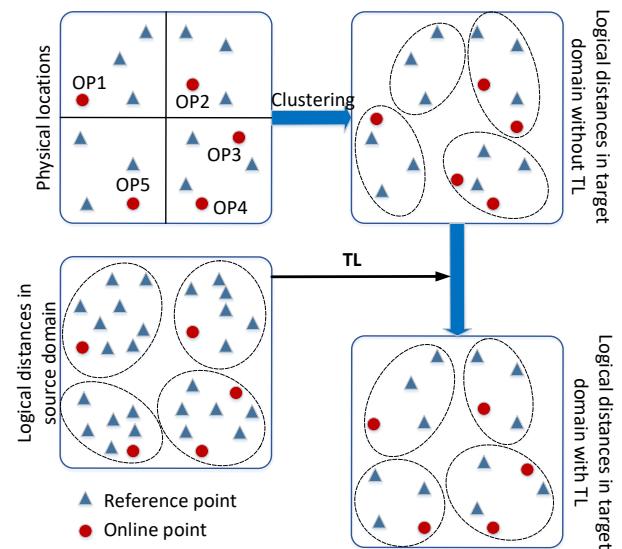


Fig. 12. The TL-based framework for the localization scheme with clustering [96].

of the observed information. As such, the localization accuracy of the moving users can be enhanced through exploring the correlation of the observations. The experimental results then demonstrate that the proposed solution can reduce the mean localization error up to 20% compared with that of the simulated radio map scenario.

In practice, the indoor localization method using the fingerprint may incur a massive time overhead in the offline training step. For that, the authors in [96] propose an inductive TL-based framework using KNN and weighted-KNN (WKNN) for indoor localization problem which can reduce the offline training overhead, thereby improving the system scalability. Specifically, the localization based on the fingerprint is first defined as a clustering problem where the online points' real locations are classified into specific areas according to the real-time RSSs and offline fingerprint dataset. In this case, the source domain can learn the distance metrics through optimizing the statistical dependence between the features of RSS and ground-truth labels. Afterwards, the logical distributions/distances between the online and reference points in the target domain can be calculated and reshaped according to the RSS map from the source domain as illustrated in Fig. 12. As such, the points which are possessed by the same cluster will be closer to each other while the ones which belong to the different clusters will be farther from the others. Using an actual testbed environment, the proposed framework can achieve 85% localization accuracy level with less offline training time overhead up to 30% compared with that of the non-TL method.

Another adaptive localization scheme for dynamic indoor environment leveraging a transfer kernel learning (TKL) with support vector regression (SVR) approach, i.e., TKL-WinSMS, is discussed in [97]. In this case, the commercial-off-the-shelf (COTS) Wi-Fi routers can be used as the online reference points whose locations are extracted from the RSS observations. Leveraging online data together with labeled source data

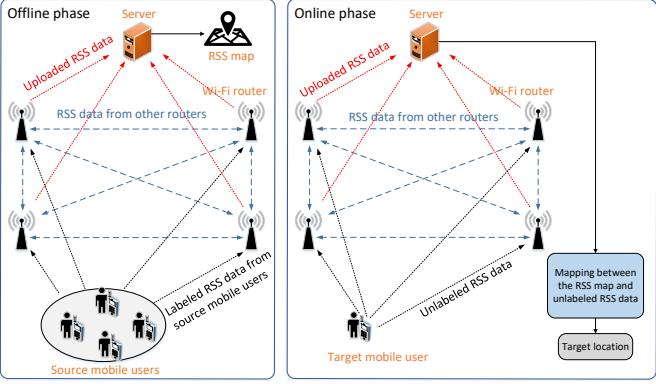


Fig. 13. The TKL-based framework for the robust localization model using Wi-Fi routers [97].

from offline calibrated RSS map, the TKL can be applied to build a robust localization model considering unlabeled RSS data from the target mobile user as shown in Fig. 13. This model can produce a domain-invariant kernel matrix, i.e., the learned matrix built through extrapolating target eigensystem on source data, which can minimize the estimation error to the ground truth source data and has the most similar feature-space distribution between source and target domains. Given the kernel matrix as the input for the SVR training, the trained model from the online process can be used to adaptively improve the offline RSS map. The experiments reveal that the proposed TKL-based method can boost the localization accuracy up to 26% compared with that of SVR without TKL method. Another adaptive indoor localization considering instance-based inductive TL approach is discussed in [98]. When the environment changes, some old instances of RSS data may be obsolete, thereby degrading the localization accuracy. To address this issue, some parts of the old RSS training data which have the same distribution with the current RSS testing data can be selected through fine-tuning the weights of instances. Compared with the methods when the training as well as testing data have different distributions and when insufficient data are utilized, the proposed method can improve the localization accuracy up to 18% and 10%, respectively.

From all the above works, it can be observed that they only utilize linear/shallow representation features, and thus may lead to the poor performance when RSS variations due to the environment changes and heterogeneous hardware, e.g., device A is used in the offline phase while device B is utilized in the online phase, occur more frequently. To solve this problem, a DTL can be applied to extract more robust hidden representation features from the RSS data by learning transferable features deeply. In [99], the authors propose a deep mean correlation alignment (DMCA) network to reduce the domain variance through aligning the hidden representation features between source and target domain in the specific DNN layers. Specifically, the mean embeddings and the second-order statistics of the hidden representation features are aligned between two domains. With the DMCA, the effect of dynamic environment and various hardware can be alleviated without fingerprint and hardware calibrations. Compared with the non-

DL-based learning, the DMCA can achieve a lower average error distance up to 2.76 meters in the experimental evaluation using smartphones on various days.

For a certain real indoor localization scenario, the use of DTL may trigger a heavy burden in building the RSS fingerprint map due to the requirement of a great amount of samples. In practice, it is intractable to obtain sufficient samples for the real localization due to sensor fault or dynamic environment, and thus the DTL may not be applicable in this scenario. To this end, the authors in [100] introduce a hybrid domain-based transductive TL framework, i.e., HDTL, which can transfer important knowledge from multiple source domains to the single target domain. Particularly, the framework first distinguishes the source domains into homogeneous and heterogeneous features with respect to the target domain. Then, a mapping between the homogeneous and heterogeneous features in the source domains can be developed using homogeneous knowledge transfer between the source and target domains. To produce a common homogeneous feature space in both source and target domains, the mapping is applied to fill up some missing heterogeneous data in the target domain. Using the homogeneous feature space, the location of target can be determined accurately under the existence of data loss in the target domain. Compared with the conventional TL with stand-alone source domain, the use of TL with multiple source domains can provide higher localization accuracy, i.e., lower average error distance.

2) *CSI Map-Based Localization:* Instead of using the RSS map for the indoor localization problem, the utilization of CSI map, i.e., a map which contains the correlation between CSIs and positions/locations of the object, can bring more stable signal amplitude, smaller multipath effect, and better representation of the location-dependent feature under the frequency diversity. For example, the authors in [101] introduce an inductive TL-based localization scheme based on CSI fingerprinting, i.e., CRISLoc, using smartphones. Particularly, the CRISLoc first incorporates a passive mode of the smartphones to capture the over-the-air packets for obtaining CSI samples. Then, using the TL method, the high-dimensional CSI fingerprint samples can be re-modeled based on the obsolete fingerprint database and some up-to-date measurements. To track down the location of the smartphone, the KNN algorithm is also implemented. The experiments reveal that the proposed scheme can obtain small mean error within 0.29 meters with the variance up to 8.6 cm when the locations of two APs are changed, thereby ensuring the robustness of the CRISLoc towards dynamic environment.

Another work considering the DTL for device-free passive wireless indoor localization (DFPWL) framework is investigated in [102] under time-varying CSI behaviors. In particular, the CSI is first extracted from a single link without deploying/wearing any devices or using any pre-processed CSI samples. Then, using the DTL method, the framework can learn new hidden representations from the CSI samples as fingerprints, aiming at minimizing the distribution discrepancy between the training and testing fingerprint samples. To predict the location of the target, the KNN algorithm is used to compare the training and testing fingerprint samples. Based

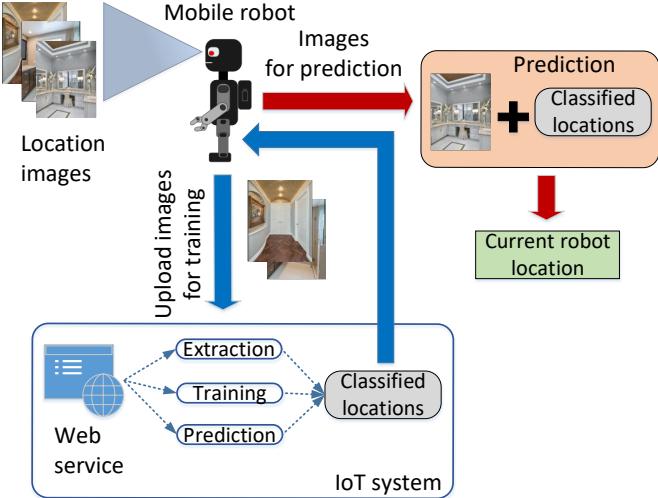


Fig. 14. The TL framework for mobile robot localization [105].

on the experiments, the proposed framework can provide stable localization accuracy continuously, i.e., with the mean localization error up to 1.5 meters, without re-collecting the CSI fingerprint samples.

3) *Sensor-Based Localization*: In addition to the RSS and CSI, sensors embedded in smartphones or IoT devices, e.g., accelerator and camera sensors, also can be applied to localize target objects. For example, the inductive TL-based localization problem considering the knowledge transfer from monitoring outdoor movements to the indoor environment, i.e., iLoom, is introduced in [103]. Specifically, accelerator sensors on the user's smartphone are first employed to obtain the indoor movement traces of the user leveraging a dead reckoning approach [104], i.e., current position calculation of some moving objects based on the previous positions, speed estimation, and heading direction for particular periods. Nonetheless, due to high noise in the data sensing, the use of dead reckoning may suffer from the inaccurate position results. Thus, relatively accurate data obtained from outdoor movements using GPS are added. Then, the inductive TL process from the outdoor dataset to the indoor dataset is executed. Particularly, the TL process is motivated by the fact that for a specific person, his/her movement behaviors, e.g., walking style, do not change significantly whether he/she is in an indoor or outdoor area. As such, the movement knowledge using large training weights from the outdoor environment can be transferred to the indoor environment properly. To further improve the dead reckoning approach with lower error calibration, an acceleration range box (which is built by the TL approach) and some optimization methods are implemented. This acceleration range box works as an acceleration filter which can determine an incorrect acceleration from a set of accelerations in various directions. Via both indoor and outdoor experiments, the proposed method can obtain higher localization accuracy up to 60% with low-complexity training compared with that of the conventional method without acceleration range box.

The application of camera sensors to detect the location of an object has also been investigated recently. In [105], the

authors introduce an inductive DTL using CNN for indoor IoT-based mobile robot localization. As illustrated in Fig. 14, the mobile robot can first capture images of its current locations using embedded GoPro and omnidirectional cameras. With the development of IoT system, the following processes are then conducted. First, the captured images are uploaded to the IoT system using Wi-Fi, and the useful features from them are extracted and trained using CNN. In this case, the last fully connected layer of the pre-trained CNN model can be removed and modified into a one-dimensional vector layer. The modified model can then be transferred to perform the TL process through a classification task using popular ML methods, e.g., Bayes, Multi-Layer Perceptron (MLP), SVM, random forest, and KNN. Based on this classification, the location of the mobile robot within the topological map coverage (based on the new real-time images of the mobile robot) can be predicted. Through experiments, the proposed method can achieve the localization accuracy up to 100% especially when the images of an apartment are captured using the GoPro camera.

B. Outdoor Localization

From the previous section, it is observed that the indoor localization problems are only limited to small-scale wireless environments through using Wi-Fi or wireless local area networks (WLANs). To further increase the localization coverage, the localization can also be adopted in outdoor environments. Specifically, the location of various mobile nodes within a certain large-scale network coverage can be first trained in the offline phase through capturing the RSS. Upon building the RSS map in a similar way as that of the indoor localization, the locations of new mobile nodes can be estimated in the online phase. However, there exists an inherent challenge of using outdoor localization in which the mobile nodes' density and SNR change dramatically, leading to higher localization estimation errors. To cope with this problem, TL approaches can be used through transferring and fine-tuning the RSS map between source and target domain (similar to the indoor localization). In this case, instead of using small WLANs, the outdoor localization employs larger networks, e.g., low power wide area networks (LoRaWANs) to localize an object/a node within the same WAN area.

In [106], the authors propose a cell-based inductive TL (CTTL) method for a large-scale localization in a wireless sensor network. Specifically, as illustrated in Fig. 15, the CTTL system incorporates beacon nodes with known location vector $\mathbf{b} = [b_{1,1}, \dots, b_{M,N}]$. When a pending node requests the information about its position, it can make a communication with the beacon nodes in the same cell, e.g., Cell-1 with the distance vector $\mathbf{a} = [a_{1,1}, \dots, a_{1,N}]$, aiming at reducing the communication cost. Using the RSS information from the pending nodes, the beacon nodes can estimate the pending nodes' locations. To address dynamic changes of pending nodes' density and SNR, a robust localization mechanism using a TL approach along with SVR model is applied to achieve accurate positioning. As shown in Fig. 15, a supercell containing multi cells is first deployed to train many location

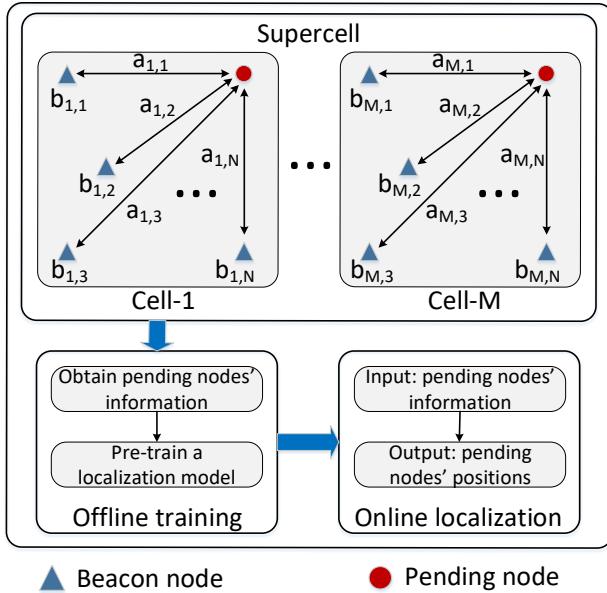


Fig. 15. The CTTL framework for large-scale outdoor localization [106].

samples within a big cell coverage in the offline phase. Then, in the online phase, the extracted location knowledge can be used by each smaller cell through modifying the trained model's weights to determine the pending nodes' positions within each cell. Using the CTTL, the positioning accuracy can reach 95% with high robustness and low communication cost.

To boost the accuracy of the outdoor localization technique, the authors in [107] investigate an inductive DTL approach using low power wireless area networks (LoRaWANs). Specifically, a grid segmentation method is first developed to obtain a sufficient number of virtual labeled locations based on the correlation between the labeled and unlabeled locations. In this case, some features including the RSS information, SNR, and current timestamps are used to pre-train a model. Leveraging the knowledge of labeled-unlabeled location correlation from the pre-trained model, the target model can be fine-tuned through adding new labeled locations, aiming at providing higher location prediction accuracy. Compared with SVM, the proposed DTL can improve the outdoor location accuracy up to 75% after 5000 epochs in a large area. Motivated from the above work, the authors in [108] propose an inductive TL-based localization accuracy augmentation using RSS fingerprint samples in the co-located low power WAN technologies, i.e., LoRaWANs and wireless M-Bus. Particularly, by using an inter-technology method, RSS map calibration can be transferred from one technology to another one by using a pre-trained DNN model with two hidden layers. In this case, the weights of the first hidden layer remain unchanged while the ones of the second hidden layer are fine-tuned according to the RSS samples in the target domain. Through the real experiments, it is demonstrated that the proposed method can improve the localization accuracy up to 16% compared with those of traditional ML methods without knowledge transfer.

Summary: In this section, we discuss the use of TL for various localization applications. It can be observed that for the

indoor localization problem, using the conventional inductive TL methods, e.g., SVM-based TL, KNN-based TL, and SVR-based TL, can improve the localization accuracy through transferring the knowledge map, i.e., RSS map, CSI map, sensing data knowledge, from one location/environment to another one. Additionally, the use of inductive DTL methods, e.g., CNN-based TL and DMCA-based TL, can further boost the accuracy with the minimum update of the map and more robust performance against the dynamic environment. Meanwhile, for the outdoor localization problem, the inductive DTL approach can be applied to enhance the positioning accuracy of mobile devices by fine-tuning the RSS map in large-scale wireless areas, e.g., cellular networks and LoRaWANs. The reason is that the semi-supervised TL method can perform well with the existence of a large amount of unlabeled datasets and few amount of labeled datasets. Nonetheless, most of the existing works in localization problems incorporate the inductive TL in which similar environments are used. To enhance the use of transductive and unsupervised TL approaches, the utilization of different environments, e.g., knowledge transfer from indoor environments to outdoor environments and vice versa can be further investigated. Additionally, the adoption of signal data from the embedded sensors to RSS/CSI map and vice versa may further prove the efficiency of TL approaches. The summary of all the TL approaches for localization applications is presented in Table V.

V. APPLICATIONS OF TRANSFER LEARNING FOR SIGNAL CLASSIFICATION AND MODULATION RECOGNITION

Radio signal classification and modulation recognition play a critical role in many wireless communication systems, especially in non-cooperative communications [110], [111]. In particular, signal classification aims to detect information bits sent from the transmitter by analyzing received signals at the receiver. Differently, the goal of modulation recognition is to recognize the modulation technique of the transmitted signals from the transmitter and then derive the original signals by using the corresponding demodulation mechanism [110]. From the beginning of the wireless communication era, modulation recognition and signal classification have been carefully studied in many signal processing applications. However, traditional approaches require sophisticated mathematical models for specific signal types and properties which limit their applications. With the rapid development in neural network architectures, DL-based methods have been widely proposed to address the existing problems of traditional signal and modulation classification [112]. However, DL-based detectors have several drawbacks. Specifically, DL solutions usually assume that data distribution is i.i.d distributed, which is infeasible in many cases. For example, the receiver may capture transmitted signals at different sampling rates due to the system properties and hardware configuration [112]. As a result, the data distribution changes the result in a low detection performance. In addition, DL applications often requires long training time over large datasets to guarantee good detection performance. However, collecting and processing data are costly and sometimes impossible [113]. Recently, TL

TABLE V
SUMMARY OF TL APPROACH FOR LOCALIZATION

| No. | Problem | Knowledge to transfer | ML method | TL type |
|-------------|-----------------------------------|---|---|--------------|
| [90], [92] | RSS map-based indoor localization | RSS map | TL with manifold co-regularization | Inductive |
| [93] | | RSS map | TL with manifold alignment | Inductive |
| [94] | | RSS map | TL with fuzzy clustering and manifold alignment | Inductive |
| [95] | | Incomplete RSS map | TL with manifold alignment | Inductive |
| [96] | | RSS map | KNN and WKNN-based TL | Inductive |
| [97], [109] | | RSS map | TKL with SVR | Transductive |
| [98] | | Signal strength distribution | Instance-based TL | Inductive |
| [99] | | Knowledge from RSS fingerprint with location labels | DMCA-based TL | Inductive |
| [100] | | Knowledge from homogeneous and heterogeneous features of multiple source domains | Hybrid domain-based TL | Transductive |
| [101] | CSI map-based indoor localization | Outdated CSI fingerprints | TL with KNN | Inductive |
| [102] | | CSI map | DTL | Inductive |
| [103] | Sensor-based indoor localization | Knowledge learned from outdoor motion data | SVM-based TL | Transductive |
| [105] | | Knowledge learned from location images | CNN-based TL | Inductive |
| [106] | Outdoor localization | Extracted location knowledge from a supercell training | SVR-based TL | Inductive |
| [107] | | Knowledge learned from virtual labeled-unlabeled data based on RSSI, SNR, and timestamp | DTL | Inductive |
| [108] | | RSS map calibration from a LoRaWAN technology | DNN-based TL | Inductive |

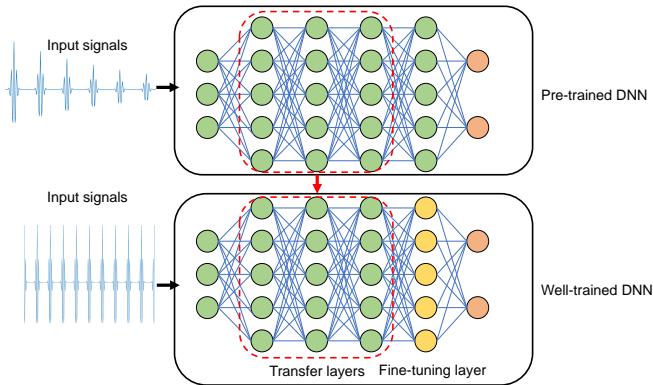


Fig. 16. TL for signal classification [114].

has been emerging as a promising solution to overcome these challenges. Therefore, in this section, we will discuss potential applications of TL for signal recognition in wireless networks.

A. Signal Classification

The effectiveness of TL in signal classification has been demonstrated in several works in the literature. In [115], the authors propose a signal classification method considering the real propagation effects (i.e., over-the-air measurement). In particular, the authors point out that training a model from the beginning requires a very long time over large datasets to achieve a good detection performance. As such, the authors propose to use TL as a convenient alternative. Specifically, the model is first trained with common datasets, e.g., synthetic data. The trained model is then transferred to classify over-the-air data with real propagation effects through fine-tune techniques. The simulation results demonstrate that by using TL, when training on the real data, the DL model can achieve the accuracy close to that of training on synthetic data. In [111], the authors study that the signal classification accuracy of a DNN is greatly relied on signal characteristics,

surrounding environment, and signal conversion in time and frequency domains. Nevertheless, wireless channels may have different properties in different settings and environments. Moreover, unexpected hardware errors may randomly occur, especially in low-cost devices, and thus a single analytical channel/hardware model cannot achieve good classification in all scenarios. For that, the authors in [111] and [116] propose to use TL to transfer knowledge trained in a source task to improve the classification accuracy of new tasks. In particular, a part of the source neural network is transferred to the target neural network to leverage the knowledge gained in the source task. Then, the remaining network layers will be fine-tuned to adapt with the target environment and setting.

Although these works can achieve good classification performance, they may not be effective in dealing with weak signals in sensor networks, especially low-cost and low-power wireless communication systems. For that, the authors in [114] introduce a universal detection framework for backscatter communication networks based on DTL with offline training and online detection. In particular, a pre-trained DNN is first established to obtain the similar features between channel models by training offline over labeled data. After that, a part of the pre-trained model is frozen while the other part is fine-tuned to adapt the current channel coefficients by using TL as illustrated in Fig. 16. Finally, the well-trained model is used for signal classification. Simulation results then demonstrate that the proposed DTL-based detection framework can achieve the bit-error-rate close to that of the optimal detection with perfect channel state information.

B. Modulation Recognition

TL-based applications in modulation recognition have also attracted significant attention from researchers. In [117], the authors propose to use a CNN with TL for automatic modulation recognition. The authors demonstrate that the DNN can achieve a good detection performance when the SNR

is high. However, when the wireless channel quality is unstable, the accuracy may drop below 40%. To address this issue, the weights trained in the high SNR scenarios are transferred for training the model when the channel quality is not good, i.e., low SNR. Through experiments, the authors then demonstrate that TL can effectively address the unstable training problem of conventional DL approaches. Similarly, the authors in [112] also adopt DTL to enhance the modulation classification performance of the target model. In particular, the authors demonstrate that with a simpler DNN architecture (compared to that of the source model), the target model can achieve a good detection performance by using TL. To do that, during the TL phase, the author freeze the source model and perform an adversarial training between the target model and a discriminator with the inputs are the output of the softmax layers in the source and target models. Based on experiment results, the authors then demonstrate that the proposed TL solution can achieve a higher detection performance compared to other existing methods.

In [110], the authors propose a semi-supervised classification framework for zero-forcing aided multiple-input and multiple-output systems with the help of TL. In particular, the authors point out that conventional DL approaches require large datasets with labeled samples for effective recognition performance. Nevertheless, in practice, labeled data is not always available. For that, a TL model is proposed to enhance the recognition performance in real-life settings. Specifically, the proposed TL model has two components: (i) a traditional CNN to train a few labeled samples and (ii) a convolutional auto-encoder utilize massive unlabeled data. To improve the training accuracy and avoid overfitting, the encoder layer can share knowledge (e.g., weights) with the convolutional layers. Through experiments, the authors demonstrate that the proposed TL solution can outperform conventional approaches and achieve classification accuracy close to that of the case in which the channel quality is good.

The authors in [118] study a problem of recognizing radio frequency interference type and its characteristics, i.e., modulation class. They first compute scalograms, i.e., a visual representation of a waveform, for collected signal data, then feed them into a CNN-based classifier. To take advantage of pre-trained DL models, they use several famous models for image classification, including AlexNet, GoogleNet, VGG, and Resnet, as *Feature Extractors* (section II.D). Specifically, classification layers of these pre-trained models, which are fully connected layers after the last convolution layer, are replaced by new classification layers. Simulation results then show that the accuracy of the system greatly relies on data type and feature extraction method.

C. Signal Recognition for Radar Systems

Most of the applications of TL for radio signal recognition are in the radar detection networks. In [119], the authors address the radar emitter signal recognition under a noisy environment recognition. The proposed system consists of four phases, including time-frequency analysis, feature extraction, feature reconstruction, and classification. To overcome the

dynamic environment noise, the *Feature Extractor* strategy in DTL is applied in the feature reconstruction phase. Specifically, a shared-hidden-layer autoencoder network is trained first, then the encoder layer of this network is used to reconstruct the features. In this way, the reconstructed features are robust again the dynamic noise of the environment. Then, these features are fed into a relevance vector machine (RVM)-based classifier. From the simulations, it is demonstrated that the proposed approach can achieve more robustness and accuracy than those of other methods using SVM [120], K-means [121], and fuzzy theory [122].

In [123], the authors study an application of DRL for recognizing radio waveforms. The signal data is first transformed into time-frequency images by using Choi-Williams distribution (CWD). Then features of these images are extracted by a feature fusion algorithm. This algorithm aims to combine two type of features, including texture feature, which contains information about the spatial distribution of colors or intensities, and the depth characteristics, which is extracted by a CNN model. To cope with the lack of training data, they use *Fine-tuning* strategy (section II.D) to transfer the knowledge of pre-trained models, including AlexNet and VGGNet, to their CNN model. The results show that this method achieves good accuracy in a noisy environment, 98.89% and 99.72% when using AlexNet and VGGNet, respectively. Similarly, the authors in [124] also focus on waveforms recognition for cognitive passive radar systems. However, in this work, they introduce a two-channel CNN with bidirectional LSTM (B-LSTM) architecture, named TCNN-BL. In this architect, input data is first fed parallelly into two different CNN-based channels (T-CNN) to obtain multi-scale reception field. Then, the output of T-CNN goes through a B-LSTM layer along with two classification layers. Since the sample rate of a cognitive passive radar highly depends on both operation mode, i.e., detection, parameter estimation, or imaging, and configuration, i.e., bistatic/multistatic and individual/cooperation, it is challenging to obtain adequate training data. To overcome this situation, they first train the model on a specific sample rate and then apply the *Weight Initialization* technique to transfer the knowledge from this pre-trained model to the target model operating at a different sample rate. From simulation results, the proposed approach outperforms both conventional methods [125], [126], in which features are extracted manually, and DNN-based methods [127]–[129].

Summary: In this section, we have discussed the applications of TL for various signal classification and modulation recognition applications. In particular, the signal detection performance largely relies on the signal features, time frequency domain conversion, and wireless environment. However, in different settings and scenarios, wireless channels possess different characteristics. Moreover, signals may be transmitted at different sampling rates due to the system's properties and unexpected hardware errors and configurations. As such, conventional DL approaches may achieve a low detection accuracy as the assumption of independent and identical data distribution is violated. In addition, collecting sufficient labeled data for effective training is costly and even impossible in many scenarios. Finally, training the DL model

from the beginning usually takes a long time and requires high-computing hardware. To tackle these problems, TL can be applied. The most common approach is transferring the trained source model to the target domain. One can reuse the whole trained model or a part of the trained model as the initiation of the target model. Moreover, the trained model can be frozen and then combined with new layers for training in the target model. It is worth noting that the existing works mostly assume that labeled data is available at both the target and the source domains. Nevertheless, in practice, labeled data may not be available at the target domains or even at the source domain. To address this problem, transductive TL and unsupervised TL can be adopted. The summary of the TL approaches for signal classification is highlighted in Table VI.

VI. APPLICATIONS OF TRANSFER LEARNING FOR WIRELESS SECURITY

In the last few years, the rapid development of wireless technologies such as 5G and IoT on the one hand has brought great benefits for human beings, but on the other hand has led to many serious concerns regarding emerging security issues for users as well as service providers [130]. Although conventional ML techniques have been widely employed for wireless security, they have been facing some major challenges. Specifically, the great diversity of devices and environmental conditions have hindered ML's applicability because conventional ML techniques require a large amount of data and training resources to be effective for only a specific problem. To address these limitations, TL has emerged to be an effective solution which can utilize knowledge not only from past experiences but also from other domains, i.e., different types of devices and environments. Therefore, in this section, we discuss applications of TL in security issues in wireless networks, such as intrusion detection, jamming attacks and eavesdropping.

A. Intrusion Detection

Wireless intrusion detection techniques can be classified into signature-based and anomaly-based techniques [131]. Signature-based intrusion detection relies on the known characteristics (signatures) of the attacks to detect intrusion. Although these methods can effectively detect intrusions and generate few false alarms, they require manual updates of the signatures database, and they cannot detect (zero-day) novel attacks [131], [132]. On the other hand, anomaly-based intrusion detection methods first try to model normal network behaviors and then detect intrusion by identifying abnormal behaviors. This type of method can detect zero-day attacks, and the data collected from these zero-day attack cases can be useful to build the signature database for new attacks. Nevertheless, this type of intrusion detection techniques is prone to creating many false alarms because new legitimate behaviors might also raise the alarm [131].

With the ability to classify behaviors automatically, ML techniques have been considered one of the most effective techniques for anomaly-based intrusion detection [131], [132]. However, ML techniques can only perform well if the training

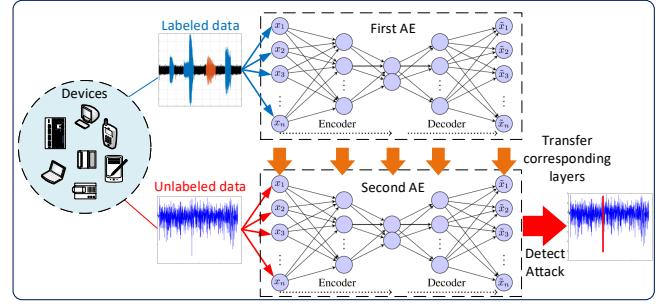


Fig. 17. The TL-based intrusion detection approach [132].

data's and the actual data's distribution are similar, which is highly unlikely in the cases of zero-day attacks [132]. To mitigate this problem, the ML models must be trained with a huge amount of labeled data. However, this is often intractable to achieve in practice and consequently hinders the applicability of ML in intrusion detection [132]. To address this problem, a DTL approach is proposed in [132] for IoT intrusion detection, which can utilize both labeled and unlabeled data to detect attacks such as DDoS, UDP flooding, TCP flooding, and sending spam data. Specifically, the proposed model consists of two AutoEncoders (AEs) as illustrated in Fig. 17. The first AE conducts supervised learning of the source data, whereas the second AE is trained with the unlabeled target data. The knowledge gained from the source domain will be transferred to the target domain (transductive TL) by minimizing the Maximum Mean Discrepancy (MMD) distances between each layer of the first AE and its corresponding layer of the second AE. Another DTL approach is proposed in [133], which also analyzes network traffic to detect anomalies. Particularly, a CNN model is developed to detect abnormal segments of the time series, which is similar to detecting desired objects in images. Moreover, the authors propose a TL strategy that transfers the first 12 layers of a pre-trained CNN to the target model and fine-tune the target model with target data. This TL strategy is expected to be more efficient than transferring the whole model because the first layers usually capture the common features between the tasks, whereas the remaining layers capture the more task-specific features. Experiments results show that the TL strategy can improve the intersection over union (IOU) score of the CNN by up to 21%.

Typically, anomalies in traffic data are the indicators of the attacks that create an abnormal amount of traffic such as DDoS. However, there is another type of attack, namely emulation attack, that cannot be detected using traffic data. For example, in Primary User Emulation (PUE) attack, an attacker pretends to be a PU to prevent the other SUs from competing for a spectrum band, thereby maximizing the attacker's spectrum usage. For such kind of attack, ML techniques can be leveraged to classify the signals coming from the real device and the attacker's device. In [134] and [135], the authors develop approaches to identify emulation attacks based on device fingerprint (i.e., unique features of each device) and environmental effects (i.e., the effects of the physical environment on the device's fingerprint). Specifically, the proposed approaches estimate and model the environmental effects on each

TABLE VI
SUMMARY OF TRANSFER LEARNING APPROACHES FOR SIGNAL CLASSIFICATION

| Paper number | Problem | Knowledge to transfer | ML method | TL type |
|--------------|------------------------|-------------------------------------|----------------|--------------|
| [115] | Signal classification | A part of the pre-trained model | CNN | Inductive |
| [111] | | A part of the pre-trained model | CNN | Inductive |
| [116] | | The pre-trained model | CNN | Inductive |
| [114] | | A part of the pre-trained model | CNN | Inductive |
| [118] | Modulation Recognition | A part of the pre-trained model | CNN | Transductive |
| [117] | | Weights of the pre-trained model | CNN | Inductive |
| [112] | | A part of the pre-trained model | CNN | Inductive |
| [110] | | A part of the pre-trained model | CNN | Inductive |
| [119] | | A part of the pre-trained model | CNN | Inductive |
| [123] | | A part of the pre-trained model | CNN | Transductive |
| [124] | | Parameters of the pre-trained model | CNN and B-LSTM | Inductive |

device's fingerprint and detect abnormal differences between the expected and the measured environmental effects. Since the network devices often have similar environmental effects when they are physically close together, both [134] and [135] propose transductive TL approaches to transfer the knowledge of environmental effects among nearby devices. Particularly, the authors in [134] propose to identify the changes in the environments of nearby devices and use this knowledge to filter out the effects of the environments on the devices' fingerprints. Simulation results show that the implemented TL scheme helps to filter out the effects of environmental changes, which leads to a lower false-positive rate compared to that of the approach without TL. Alternatively, the author in [135] propose a TL strategy that transfers the knowledge learned from a device (e.g., smartphone) to a nearby device of a different type (e.g., sensor). The reason is that if the two devices are physically close to each other, they might share similar environmental effects. Experimental results show that the applied TL approach can improve the detection performance by 70%, and higher transfer weights (i.e., the percent of the source model's weight transferred to the target model) can lead to better results.

Similarly, the authors in [136], [137] also propose approaches to identify IoT devices based on fingerprints. However, these approaches do not consider environmental effects. Instead, they utilize the knowledge gained from past experiences. Particularly, a Bayesian model is developed to classify device fingerprints to detect attacks in [136]. The authors first build an abstract knowledge database which is composed of fingerprints of all devices in each time frame. At a new time frame, new fingerprints are received. Then, an inductive TL strategy is applied to combine the new fingerprints with the old fingerprints stored in the abstract knowledge database to cluster and assign the new fingerprints to the devices. In [137], the authors develop an inductive TL technique that combines a large amount of source data, i.e., data collected in the past, and the smaller target dataset, i.e., current data, to train an SVM model for the classification of IoT devices. The main difference between the two approaches is that the changes over a long time (i.e., one year) in the physical features are taken into account in [137], whereas the time frames in [136] are much shorter. Simulation results show that the TL approaches proposed in [136] and [137] can improve the recognition rate by 3.5% and 9%, respectively. From these results, we can

observe that using the knowledge from a longer time frame might be more beneficial. However, this TL strategy is also much more time-consuming.

Aside from fingerprints, wireless devices can also be recognized using transient signals (which include the amplitude, phase, and frequency) due to the lower hardware requirements than that of the steady-state signals. Nonetheless, the wireless device recognition may suffer from limited number of transient signal data as the wireless devices to be recognized are often non-cooperative. To this end, the inductive TL approach for individual recognition of wireless devices with small amount of samples is discussed in [138]. In particular, a wireless device as the reference AP first captures transient signals from other wireless devices and then classifies signals in the source and target domains. Afterwards, the feature weights of transient signals in the source domain are extracted by using the transient envelope feature extraction with entropy weighting method, and transferred to the target domain. For that, a new KNN-based model can be trained using the transferred knowledge by modifying the feature weights of the source model samples. Through the experiments using 5 wireless devices and an oscilloscope to obtain 200-360 samples and 40 samples of source and target domain, respectively, the proposed TL method can improve the classification accuracy up to 13.2% compared with that of typical KNN without TL method. Additionally, it is possible to recognize similar RF devices based on their emitted signals, even when these devices are in the same batch built by the same manufacturer. The differences in emitted signals come from device noises and manufacturing errors. Therefore, the unique identification of an RF device can be derived from the statistical characteristics of its emitted signals. In [139], the authors introduce a TL-based model to identify an RF device. In particular, they employ the *Weight Initialization* technique to a LSTM model, namely TL-LSTM. They then use eight RF devices, in which four devices generate the source data and the rest of devices generate the target data. The source model is first trained in the source domain. Then, the first layer's parameters of the target model are cloned from that of the trained source model. The results demonstrate that the proposed TL can improve the accuracy in identifying RF devices regardless the noise of the environment.

B. Jamming Attacks

Jamming attack is a common type of security issues in wireless networks, mainly because attackers can easily launch jamming attacks without requiring complete information regarding the target system. To mitigate jamming attacks, conventional methods such as changing the spectrum and transmission power have been widely applied [140]. However, with the development of ML techniques, attackers can launch adaptive jamming attacks to counter these fixed anti-jamming methods. To address these adaptive attacks, reinforcement learning algorithms have been considered to be a very potential solution to dynamically learn and find the optimal anti-jamming control policies. Nevertheless, reinforcement learning algorithms usually require a long learning time to be effective. To address this issue, TL can be used to speed up the learning process of traditional reinforcement learning. In [141], a TL-based approach is proposed for anti-jamming power control of wireless body area networks (WBANs). Specifically, in the considered WBAN, there is a WBAN coordinator which determines the power level and time slot of the transmission of sensing data while taking into account jamming attacks. Since this anti-jamming power control process can be formulated as a Markov decision process, an advanced Q-learning approach is developed to find the optimal power control policy. This Q-learning approach utilizes unsupervised TL to determine the initial Q-learning parameters based on previous experiences, thereby avoiding unnecessary exploration at the beginning and speeding up the learning process. Simulations results show that the proposed approach always outperforms the Q-learning approach without TL in terms of bit error rate (BER), and SINR, and the long-term utility of the jammer can be significantly reduced.

Another approach that combines Q-learning and TL is proposed in [142] for anti-jamming multi-regional communication. Specifically, the approach first applies Q-learning to learn the jamming rules of the attackers and then determine anti-jamming policies accordingly. However, this method is only effective for a particular wireless environment (region). When the wireless environment changes (in a new region), the whole Q-learning process has to start again. Thus, the authors propose to implement unsupervised TL to transfer the previous knowledge, i.e., Q table, gained from other regions to initialize the learning process of a new region. To evaluate anti-jamming performance, simulations are carried out to compare the proposed approach with an independent learning algorithm (without TL). The results show that the proposed approach can converge to the optimal policy faster and achieve better BER compared to those of the independent learning algorithm.

In [143], a DRL approach is developed for UAV aided anti-jamming cellular communications. Specifically, UAVs are deployed to dynamically relay messages of mobile users in the case where the BS is severely jammed. To choose the optimal relay policy, the UAVs employ a CNN that takes into account current BER, channel gains, and jamming power estimation to determine the Q-values for different relay policies. Knowledge from previous experiences is used to initialize the weights and learning parameters of the new CNN, i.e., inductive TL.

Simulation results show that the proposed TL strategy can not only reduce the learning speed of the Q-learning but also improve the performance of the Q-learning approach in terms of BER and energy consumption.

C. Other Security Issues

Apart from intrusion detection and anti-jamming, TL approaches have also been developed for other security issues such as eavesdropping and spam voice calls. In [144], a framework for energy-efficient data communications in energy-harvesting CRNs is proposed, which takes into account hidden eavesdroppers. Specifically, a model-free RL, i.e., the actor-critic model, is developed to find the optimal data encryption policy for transmission. TL is also implemented in this RL process, where the action is determined not only by the reward of the last action but also from the rewards of previous actions. This unsupervised TL of the previous experience is expected to increase the convergence speed of the RL process. Simulations demonstrate that the proposed approach can achieve better channel utilization and transmit more data packets compared to other conventional approaches. In [145], an unsupervised learning approach is developed to filter spam voice calls by assigning trust scores among users. In particular, the approach extracts data from call logs to determine trust scores using an MLP model. Since call data are user-specific, a model that is accurate for one user might not be accurate for another user. Thus, a transductive TL approach is implemented, which first creates a baseline model and then trains the model with a large amount of data. Then, this model is transferred to each specific user, and it will be further trained using the user-specific call data to achieve an adaptive and personalized call filter for each user. Experiment results show that the proposed approach can achieve a prediction accuracy up to 99.88% on a validation dataset.

Summary: As summarized in Table VII, we have discussed various TL-based approaches to address security issues such as intrusion detection, jamming attacks mitigation, anti-eavesdropping data encryption, and spam voice calls filter in wireless networks in this section. For intrusion detection, TL approaches are developed to address the lack of labeled data (by transferring a pre-trained model), the heterogeneity of devices and environment conditions (by transferring knowledge of different device types and environments), and the low learning speed (by transferring knowledge from the past experiences). For jamming attacks mitigation and anti-eavesdropping data encryption, RL is the most commonly applied technique. Since the conventional RL techniques require a long learning time to be effective, knowledge from past experiences is transferred to speed up the learning process. For spam voice calls filter, the call data are scarce and user-specific. An effective solution to simultaneously address those two challenges is to first train a baseline model with data from many users. Then, the baseline model can be fine-tuned with user-specific data to build a personalized model for each user. Although the proposed TL approaches are effective solutions to their respective problems, there are still some challenges. For example, the baseline models used for transfer require a

long time to train because of the large and generalized datasets used for training. Moreover, choosing the right TL parameters, e.g., transfer rates, is crucial to the effectiveness of TL, but it is not analyzed in most of the discussed approaches.

VII. APPLICATIONS OF TRANSFER LEARNING FOR ACTIVITY RECOGNITION

Wireless networks, e.g., Wi-Fi or wireless sensor network, have a capability to provide low-cost wireless sensing for various activity recognition applications by capturing CSI or sensing data. Generally, the activity recognition using wireless signals can only work efficiently for a specific environment and may not be efficient to use in other environments. To this end, the TL approach can be applied as a useful learning method to provide common wireless-based activity recognition system including human activity, handwritten signature, and mobile device recognition without requiring large computing resources.

A. Human Activity Recognition

Human activity recognition (HAR) is commonly utilized to detect various human activities using wireless signals and smart sensors in smartphones or mobile devices, aiming at improving the quality of human life including safety and health services. Specifically, the human movement data in forms of CSI signals or sensing data can be first collected from the wireless devices. Then, using the HAR system powered by ML methods, the human activities can be classified by transforming the CSI signals or sensing data into certain labels for various purposes, e.g., older people tracking, medical diagnosis, crime monitoring, and driving behaviors. Nonetheless, a human activity classification model usually only works well in one specific environment. When a new environment is employed, a new training process may be labor-intensive and time-consuming as the sufficient data and trained model require to be collected and re-built from the beginning, respectively. In this way, TL approaches can be used as very useful methods to recognize the human activities accurately while addressing the above challenges efficiently.

1) CSI Signal-Based Human Activity Recognition: The CSI, which describes how the wireless signals propagate between transmitters and receivers in the communication link, can be used to recognize human activity. The CSI as the channel matrix can be estimated using transmitted and received signals of wireless devices after passing several signal processings, e.g., cyclic prefix removal, demapping, and OFDM demodulation. Compared to the RSS, the CSI can observe the change of fine-grained environment (including finding the causes of the environment changes and recognizing how the environment changes) more effectively, and thus it is applicable to be used for human activity recognition. However, due to the lack of sufficient CSI signal data and high cost to obtain them when the environment changes, the use of TL approach can help to improve the efficiency of using CSI to recognize the human activity.

The adoption of TL approach leveraging CSI signals has been investigated recently to recognize hand and arm movements. In [146], the authors present an inductive TL-based

sign-language recognition system, i.e., WiSign, using time-series of CSI with a sparse labeled dataset. In particular, the system can recognize and extract the meaning of particular hand or arm movements, e.g., “Yes/No”, “Hello/Goodbye”, “Thanks”, and “Sorry”, according to the study of various CSI waveforms. In this case, an inductive TL method, i.e., by transferring knowledge of the CSI waveforms from other users to train a new classifier for a new user, can be applied to achieve a higher sign language accuracy. Using the TL method, the system only needs few labeled training samples and can prevent the use of a generative model. Through real experiments using two Wi-Fi devices, it is demonstrated that the proposed system can achieve a higher mean prediction accuracy than that of classical SVM. Instead of using the CSI waveforms directly for the TL process, the authors in [147] propose a DTL-based classification framework for gesture recognition through CSI waveforms-to-image conversion. Specifically, the framework first converts the amplitude of CSI into an image matrix. Using an existing CNN pre-trained model on a large number of images, the TL method can be applied to reduce the training time and extract the pre-trained model’s important features (by implementing fine-tuning process for the classification tasks). From the real experiments using 8 gesture positions, the gesture recognition can achieve up to 100% accuracy.

Instead of using the full component of CSI signals in the above works, the frequency component of CSI signals can be utilized to detect the velocity of human body movement for human fall detection. Specifically, the authors in [148] discuss an inductive TL-based human activity recognition for fall detection system, i.e., TL-Fall. Typically, a fall detection system in a certain condition does not work well for other conditions due to unique information where the falling occurs, i.e., when a specific person (e.g., elderly or disable person) loses his/her balance and hits the ground. To cope with this problem, the TL-Fall accounts for an SVM-based TL to transfer falling velocity changes knowledge from the trained labeled data in the old condition to the new condition with few labeled data (as illustrated in Fig. 18). In this case, the system first collects the amplitude of CSI from multiple OFDM subcarriers of Wi-Fi signals with IEEE 802.11n standard. Then, the frequency component of the CSI can be extracted to determine the velocity of human body changes when falling happens. Based on the experimental results in an indoor environment, the TL-Fall outperforms other baseline ML-based fall detection methods in terms of fall detection sensitivity, i.e., the percentage of correctly detecting falling activities.

Motivated by the CSI extraction in [148] and CSI-image transformation method in [147], the authors in [149] propose an inductive TL-based framework to detect multiple human activities, i.e., TL-HAR. Specifically, using the variance of MIMO subcarriers, the activity-based CSI is first collected and transformed into CSI images. Then, the specific features from CSI images of human activities are trained using deep-CNN and inferred-CNN. This pre-trained model can be modified by fine-tuning several layers which are closer to the output of this model. The modified model can then be transferred to perform the TL process for other human activity dataset

TABLE VII
SUMMARY OF TL APPROACHES FOR SECURITY IN WIRELESS NETWORKS

| No. | Problem | Knowledge to transfer | ML method | TL type |
|-------|-----------------------------------|--|---------------------------------|--------------|
| [132] | Intrusion detection | Layers of a pre-trained model | Auto Encoders | Transductive |
| [133] | | Layers of a pre-trained model | CNN | Transductive |
| [135] | | Environmental effects of nearby devices | TL | Transductive |
| [134] | | Environmental effects | Infinite Gaussian Mixture Model | Transductive |
| [137] | | Device's physical features from the past | SVM | Inductive |
| [136] | | Fingerprint from the past | Bayesian model | Inductive |
| [138] | | Feature weights of transient signals | KNN | Inductive |
| [139] | | Knowledge from similar devices | LSTM | Transductive |
| [141] | | Knowledge from previous experiences | Q-learning | Unsupervised |
| [142] | Anti-jamming | Knowledge from nearby regions | Q-learning | Unsupervised |
| [143] | | Knowledge from previous experiences | CNN and Q-learning | Unsupervised |
| [144] | Anti-eavesdropper data encryption | Knowledge from previous experiences | Actor-critic model | Unsupervised |
| [145] | Filter spam voice calls | Knowledge from other users | MLP | Transductive |

training with shorter training time. Via the simulation, it is shown that the TL-HAR can enhance the classification accuracy up to 3% and 5.6% for single and multiple MIMO links, respectively, compared with those of other conventional ML methods. Instead of transforming the CSI data to images, the authors in [150] investigate a location-independent inductive TL with CNN-based human activity recognition model using the estimation of channel frequency response containing the amplitude attenuation. Particularly, the CSI data from certain locations are first collected and denoised to remove the noise from raw CSI data. Then, a pre-trained CNN model is built to obtain the feature extractor based on the characteristic of the CSI amplitudes. From this feature extractor, a perceptive-based TL approach is used to predict/classify new i.i.d samples online in different locations. Applying such the TL method, the new activities can be recognized without training the new samples or with only using few training samples. Through the experiments, the proposed framework can reach the activity classification accuracy more than 90% and outperform those of other baseline ML methods, e.g., Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), and KNN, for six different activities.

To investigate the correlation between the spatial and temporal information from CSI, the authors in [151] introduce an inductive DTL framework for human body movement which jointly considers the spatial features trained from CNN and temporal features learned from B-LSTM. In particular, the spatial features first can be extracted from the CSI streams using the CNN. Then, the fully-connected layer of CNN is used as the B-LSTM input to extract the temporal features and produce a pre-trained model. Some specific layers from the spatio-temporal pre-trained model can be fine-tuned (while other layers remain fixed) to implement the inductive TL process, aiming at training a new environment with a smaller number of samples. Through experiments using wireless router to classify four different human movements, the proposed framework can achieve overall accuracy 96.96% with low computational cost.

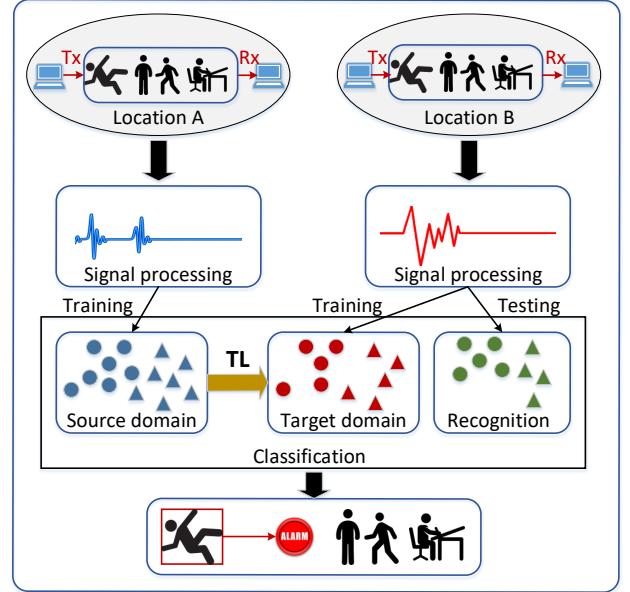


Fig. 18. The TL framework for human fall detection [148].

2) *Sensor-Based Human Activity Recognition*: In addition to the reflected signals, i.e., CSI, wearable sensor devices can be used to detect human activities. In [152], the authors introduce a light DTL framework to detect abnormal human activities, e.g., fall detection, at edge servers. Specifically, the users' smart devices first send raw data (captured via triaxial accelerometer sensors) to their nearest edge servers. Then, a CNN-based prediction model is generated to recognize the abnormal activities at the edge servers. Using the pre-trained CNN model, the high-level embedding features can be extracted to perform the inductive TL process for other abnormal activity models using conventional ML methods, i.e., LR, KNN, DT, NV, RF, and SVM. From the experiments, the framework can provide higher prediction accuracy up to 28% compared with those of the conventional ML methods without using TL. The development of TL approach without using the

edge servers is also discussed in [153] where collaborative wearable devices can be used to automatically detect real-time physical human movements. In particular, there are three main steps to apply an inductive TL method. First, two wearable sensors (referred to as *source view*) are attached to train a base model for the movement recognition. Then, a new wearable sensor (referred to as *target view*) is embedded to train a new model at the target view. Upon completing the new model training, the source and target views (referred to as *multi-view learning*) can be used together to construct a new classification model for activity recognition with a higher accuracy. Via the experiments, it is shown that the activity recognition accuracy can be improved up to 9.3% at the level 83.7% compared with those of conventional approaches where the model from the source view is re-used directly to the target view without any refinement.

Instead of sending raw data to the edge data center, the authors in [154] propose a distributed ML approach where the collected raw measurement data, i.e., 3-axial linear acceleration and 3-axial angular velocity, from embedded accelerator and gyroscope sensors can be trained locally at each user's smartphone. In particular, the collected data from the smartphone's sensors are first pre-processed and sampled locally to obtain a vector of multiple features in time and frequency domain. Then, by adopting a hypothesis transfer learning (HTL), some smartphones with sufficient number of datasets can train local partial models (referred to as *source models*) locally. From the trained partial models, a unique complete model can be obtained by accumulating the source models from the training smartphones. This complete model can then be transferred via wireless network to learn a more accurate model at another smartphone (referred to as *target model*) with insufficient number of datasets, thereby improving the human activity classification accuracy at the smartphone. Considering three static activities, (i.e., standing, sitting, and lying), and dynamic activities (i.e., walking, walking downstairs, and walking upstairs, and static-to-dynamic transitions), the proposed TL method can achieve the comparable classification accuracy performance with that of centralized learning while reducing the wireless network overhead significantly.

Due to the benefits of using distributed ML for sensitive data privacy and security, the authors in [155] introduce a federated TL-based framework, i.e., FedHealth, leveraging CNN for wearable activity recognition in the healthcare application system. This framework can deal with personalization issues where the daily health activity pattern and physical features for each individual user can be trained differently. Particularly, each user with a wearable device can train a lightweight model locally without sharing any actual data with other users. The local trained models from all the users are then collected and aggregated by the cloud server with a homomorphic encryption to update the global cloud model. To incorporate a unique new model for each user, the useful common knowledge from the cloud model can be utilized by the previous local trained models of the users as shown in Fig. 19. In this case, the low-level layers of CNN, i.e., convolution and max-pooling layers, which contain the common activity recognition are frozen. Meanwhile, the fully connected layers which consist

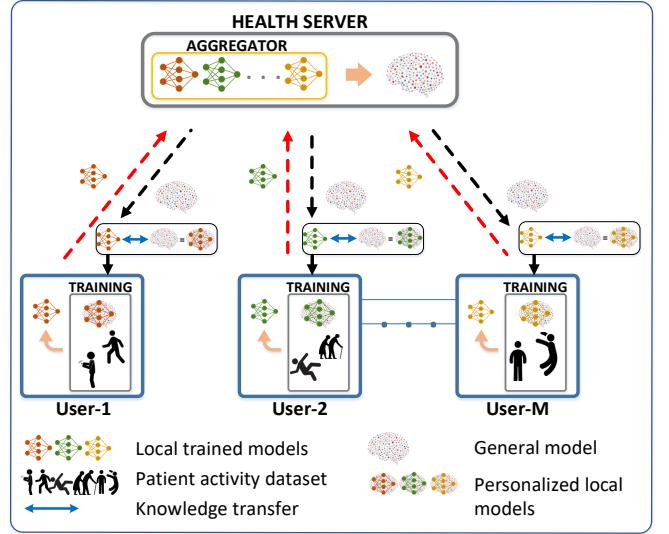


Fig. 19. The federated TL framework for healthcare activity system [155].

of more specific features for each user can be updated using an additional correlation alignment layer between the cloud model and the user's local trained model. Using human activity UCI Smartphone dataset with 6 activities from 30 users [156] and self-producing Parkinson's disease dataset, the proposed TL approach can improve the classification accuracy up to 5.3% and 21.6%, respectively, compared with those of other conventional ML methods, i.e., traditional DL, KNN, SVM, and RF.

3) Probe Response Signal-Based Human Activity Recognition: Generally, capturing CSI waveforms may require user access to the real Wi-Fi hotspots, and thus some authentication procedures need to be performed. Furthermore, the use of real Wi-Fi network needs continuous data traffic even when the Wi-Fi hotspots are idle through transmitting beacon signals. In practice, some emergency services, e.g., search-rescue applications, may have urgent cooperation with the wireless network without using the beacon signals. For that, the authors in [157] present a TL-based passive activity recognition using only Wi-Fi probe response signals generated from a local Wi-Fi device, e.g., Raspberry Pi, without connecting to the actual Wi-Fi network. Specifically, the probe request-response mechanism of Wi-Fi can enable the information exchanges between the local Wi-Fi AP and its users without any encryption or modification to the AP. Using the probe response signals, Doppler measurements to implement human activity recognition can be employed. Nonetheless, due to the low duty cycle of the probe response signal transmission, the classification accuracy may be degraded. For that, an inductive TL approach can be used to improve the accuracy through fine-tuning the convolution layer of the Doppler CNN model. By utilizing the TL method with a probe response signal dataset containing six human activities, the classification accuracy can be improved up to 7% with significant training time reduction.

B. Handwritten Signature Recognition

In addition to the human activity recognition, the inductive TL method can be utilized to verify a specific person's identity

based on his/her handwritten signature as discussed in [158]. Specifically, the Wi-Fi gesture signals of the handwritten signature for a specific position can be first collected through the reflected Wi-Fi CSI. Then, a gradient descent CNN-based algorithm is performed to pre-train a model (as the feature extractor). In this case, the feature extractor contains convolution layers, activation functions, max-pooling layers, and fully connected layers which correspond to the classifier of the pre-trained model. As such, only the common knowledge with the best classification accuracy will be used for the TL process. Upon obtaining the feature extractor, the common knowledge from this particular position can be transferred to detect signature signals from other positions using the kernel and range space projection learning [159]. Since the Wi-Fi signature signals for diverse positions have the same data domain, only the model classifier is required to be re-trained. Based on the experimental study using thousands of Wi-Fi signature signals from 50 user IDs, the proposed method can achieve 96.75% accuracy with faster training CPU/GPU computation time compared with those of typical gradient descent and convolutional network algorithms.

Summary: In this section, we have investigated TL approaches for the applications of wireless-based human activity recognition in wireless networks based on the full/partial component of CSI signals, sensing data, and probe response signals. In particular, due to the diverse human activities in various conditions, e.g., locations and environments, the TL method can be efficiently applied to provide generalized knowledge transfer which can improve the recognition accuracy with less training data and shorter learning time. In this case, the inductive DTL approaches, e.g., CNN and LSTM, have been widely used to improve the activity recognition accuracy by extracting the non-linear features deeply. Additionally, the adoption of distributed ML methods, e.g., HTL and FTL, can further improve the accuracy of activity recognition model for each individual user due to the mutual sharing of local trained models with minimum private data disclosure. Based on the above works, the inductive TL is considered as the most popular TL approach to recognize the human activity where the activities in the target domain are a subset of those in the source domain. The extension capability of practical human activity recognition to new activities unseen in the source domain using transductive and unsupervised TL approaches will be potential to explore for future research works. The summary of all the TL approaches for human activity recognition applications is presented in Table VIII.

VIII. APPLICATIONS OF TRANSFER LEARNING FOR WIRELESS CACHING

Content caching in wireless edge networks has been proposed as one of the most effective paradigms to deal with a massive data traffic demand through distributing popular contents at edge servers (which are deployed in a close proximity to requesting users). To predict the popularity characteristic of contents at the edge servers with high accuracy, proactive caching leveraging ML-based methods can be applied [160].

However, in practice, it is costly and intractable to collect sufficient number of training dataset to create efficient prediction models for content popularity. Moreover, the content popularity matrix containing the correlation between users and contents frequently suffers from data sparsity problem. For that, TL approaches can be used to address these issues. In this section, we discuss the use of TL to cope with the drawbacks of caching applications in wireless networks for two scenarios, i.e., individual caching and cooperative caching.

A. Individual Caching

In this individual caching scenario, the knowledge produced by the TL method can be transferred separately to each individual caching storage of the small cells, e.g., small BSs (SBSs). Thus, there is no cached-content sharing among the participating small cells. The content popularity in the network can be predicted at both small and macro BSs as follows.

1) *Content Popularity Prediction at Small BSs:* In [161], the authors propose an inductive TL-based content popularity prediction policy as illustrated in Fig. 20(a). Specifically, the TL approach can first extract knowledge from content-access history of Device-to-Device (D2D) users within their social community (source domain). Then, using the extracted knowledge, the learning process for the large-scale content popularity matrix in SBS networks (target domain) can be improved. Compared to the random caching and collaborative filtering methods, the proposed TL method can achieve much closer backhaul offloading gain to the ground truth of known popularity matrix for all scenarios with various storage ratios. Extended from the work in [161], the authors in [162] develop an inductive TL-based caching strategy executed at SBSs under the storage constraints, workload, data sparsity, and unknown spatio-temporal traffic demands. In particular, the hidden latent contextual information including the users' content-access history and social ties is obtained from the interactions of D2D users (source domain). To efficiently cache well-received contents at the SBSs, the obtained information is then utilized to predict the content popularity matrix with unknown users' ratings in the SBS networks (target domain). Based on the simulation results, it is shown that the proposed TL method can improve the satisfaction ratio (i.e., the total requests that are served immediately within a time window over the total requests from all users) and backhaul offloading gain up to 22% compared with those of the random caching and collaborative filtering methods when various storage sizes, traffic intensity, and backhaul capacity are taken into account.

2) *Content Popularity Prediction at Macro BSs:* The use of inductive TL to predict the content popularity in heterogeneous networks considering the minimum training time is proposed in [163], [164]. Instead of estimating the content popularity at each SBS, the SBS only incorporates a random caching strategy. Meanwhile, the content popularity prediction is conducted at a macro BS in a centralized manner. The predicted content popularity is then used to optimize the caching strategy through minimizing the training time due to the requested content unavailability. Nonetheless, the minimum training time can be very high to achieve a required prediction accuracy

TABLE VIII
SUMMARY OF TL APPROACH FOR ACTIVITY RECOGNITION

| No. | Problem | Knowledge to transfer | ML method | TL type |
|-------|--|---|--------------------------------|-----------|
| [146] | CSI signal-based human activity recognition | CSI waveform map | SVM-based TL | Inductive |
| [147] | | CSI map | CNN-based TL | Inductive |
| [148] | | Knowledge from falling velocity changes | SVM-based TL | Inductive |
| [149] | | Knowledge from CSI images | Deep and inferred CNN-based TL | Inductive |
| [150] | | Knowledge from the characteristic of CSI amplitudes | CNN-based TL | Inductive |
| [151] | | Knowledge extracted from spatial and temporal features | CNN and B-LSTM-based TL | Inductive |
| [157] | Probe response signal-based human activity recognition | Knowledge from probe response signals | CNN-based TL | Inductive |
| [152] | Sensor-based human activity recognition | Knowledge from embedding features of sensing data | CNN-based TL | Inductive |
| [153] | | Knowledge from movement detection of wearable sensors | Synchronous multi-view TL | Inductive |
| [154] | | Knowledge from features in time and frequency domain | Hypothesis TL | Inductive |
| [155] | | Common knowledge from cloud model to obtain personalized model | CNN-based TL | Inductive |
| [158] | Identity recognition | Knowledge from reflected or interfered Wi-Fi CSI signals of in-air handwritten signatures | CNN-based TL | Inductive |

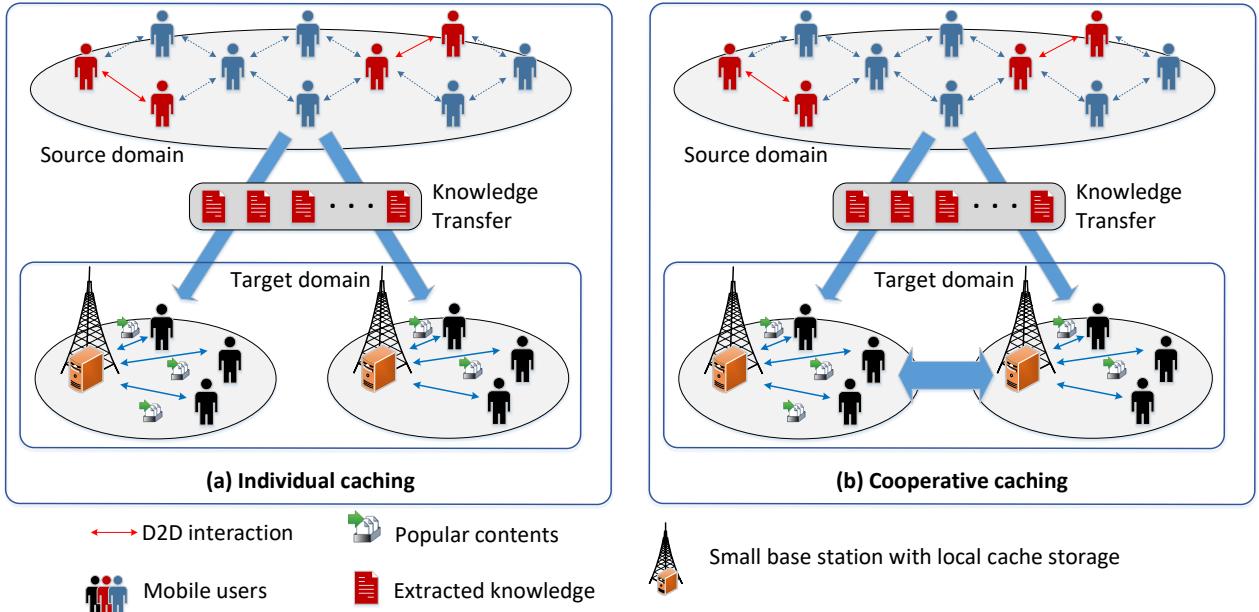


Fig. 20. The knowledge transfer of D2D interactions in the individual and cooperative caching scenarios.

when the training process is performed from the scratch. As such, the TL approach can be applied to further reduce the training time by transferring important knowledge from the user-content interaction (source domain), e.g., popular cartoon movies for kids and popular action movies for adults, to the users' request pattern (target domain). When the number of source domain samples increases, the proposed TL method can reduce the training time linearly compared with the learning without TL method where the training time remains the same for the whole performance. In [165], the authors investigate an efficient dynamic video rate allocation mechanism considering a combination of DRL, imitation learning, and unsupervised TL approaches. In particular, the allocation problem is first modeled as a Markov decision process problem. Then, a DQN, imitation learning, and the unsupervised TL approaches are

utilized to find sub-optimal allocation policy. In this case, the TL approach can transfer the knowledge of DQN parameters and fine-tune them to map the video rate from source to target domain. Leveraging the TL method, the mechanism can obtain the sub-optimal allocation policy faster. From the performance evaluation, the proposed method can outperform other conventional learning methods, e.g., DNNs and random rate allocation, in terms of the average user's reward, i.e., QoS of video streaming.

B. Cooperative Caching

Although the works in [161], [162] demonstrate the superiority of using TL approaches, they assume that the content popularity is predicted independently at each SBS, leading to the redundant caching problem. Furthermore, the authors

in [163]–[165] consider that the content popularity estimation is performed at the centralized BS, thereby increasing the workload and operational cost at the BS. To address these challenges, the authors in [166] propose an inductive TL-based proactive caching mechanism under the cooperative caching scenario among edge servers as shown in Fig. 20(b). In this case, the following two-step proactive caching optimization framework is considered. First, the framework trains the content popularity model by using the TL approach with small training time at multiple edge servers in a cooperative scenario, i.e., where each edge server with a local caching storage can share its cached contents with other edge servers. Using this trained model, the requested content information at another edge server can be predicted. In other words, the target and source domains are the content-access information to be predicted at an edge server and those are at other edge servers in the cooperative scenario, respectively. Then, a content placement policy can be derived using a greedy algorithm. By using such a framework, the cache hit rate can be improved up to 177% compared with those of the baseline caching methods. To further improve the prediction accuracy, a joint ML utilization using a regularized singular value decomposition-based collaborative filtering (RSVD-based CF) and inductive TL methods is investigated in [167]. Particularly, the content popularity matrix is first predicted by using the RSVD-based CF method. The TL method is then utilized to enhance the prediction accuracy under the data sparsity through transferring hidden features, e.g., a social network or a D2D network, from source to target domain. Based on the predicted content popularity, a decentralized iterative algorithm is executed to obtain the optimal caching policy. Via the simulation results, the proposed approach can reduce the content request delay, and thus improve the users' satisfaction.

Summary: In this section, we have discussed the use of TL to address the limitations of caching applications in wireless networks. Specifically, the content popularity prediction may suffer from low accuracy and long training time when typical ML methods are utilized. In this case, the TL approaches are applicable to solve the aforementioned problems through transferring knowledge, e.g., user-content interaction, social network, and video rate map, without a need of training process from the scratch. Furthermore, the use of TL-DRL and RSVD-based TL (inductive TL) can be very efficient to improve the average user's reward and prediction accuracy, respectively. With the TL approaches, the training time of the content popularity prediction can also be reduced, thereby minimizing the content-access delay for the requesting users. However, from all the above works, they assume that the popularity matrix is constant over time. In practice, the popularity matrix may change frequently due to dynamic preferences of many mobile users. To this end, the use of TL approaches which are robust to the popularity matrix changes needs to be further evaluated to maintain high cache hit ratio and low delay of the requested contents for different time. The summary of the TL approaches for caching applications is presented in Table IX.

IX. CHALLENGES, OPEN ISSUES, AND FUTURE RESEARCH DIRECTIONS

Although the surveyed works have demonstrated the effectiveness of TL in solving various problems in wireless networks, there are still many challenges and open issues. In the following, we discuss the current challenges, open issues, and future research direction of TL for future wireless networks.

A. Challenges and Open Issues

1) *Determine the Source Tasks in Wireless Networks:* The effectiveness of TL depends greatly on the similarity between the source and the target task. Nevertheless, choosing the right source task is usually not trivial, especially in wireless network applications where there are multiple sources available, e.g., from nearby users, access points, femtocells, and cellular base stations. For example, [63], [64], [68], [69] choose sources based on their physical distances to the target. The reason is that in this case, the surrounding environments of the source and the target have high similarities, and thus their Q-values might be similar. Alternatively, TL approaches such as [86] and [87] choose to transfer knowledge of the same task in different frequency bands, whereas some other approaches transfer knowledge from previous experiences. However, we can observe that most of current works do not well justify why the sources are chosen. If the source and target's domain are not highly related, TL approaches can even make the target model performing worse than just training with the target data. Thus, solutions and performance analyses for selecting sources in wireless networks need to be further investigated.

2) *Determine What to Transfer in Wireless Networks:* Once the source is chosen, the next step is to determine which part of the source's model to be transferred. Generally, the first layers of the source model capture the more general features that are similar between the two domains, whereas the last layers are more domain-specific. For example, in wireless applications where there are multiple cognitive agents in the same radio environment, more layers can be transferred. However, in other applications, such as [77], where the knowledge is transferred from neighboring regions, only the first few layers should be transferred. Therefore, determining how many layers to transfer is a very challenging task, because it requires an in-depth analysis of the similarities between the source and target. Despite its importance, there are few works that attempt to address this challenge. In [86] and [71], the authors determine the number of layers to transfer through simulations, i.e., testing different numbers of layers and choose the one with the best results. Therefore, more experiments and simulations should be performed to better determine layers to transfer according to different wireless applications, e.g., cognitive radio, localization, and activity recognition.

3) *Determine the TL Parameters in Wireless Networks:* The transfer rate parameter has a significant impact on the effectiveness of the TL process. This parameter dictates how many percent of the source model's weights are transferred to the target model's weights. Few approaches such as [65] test different transfer rates by simulations and conclude that

TABLE IX
SUMMARY OF TL APPROACH FOR CACHING

| No. | Problem | Knowledge to transfer | ML method | TL type |
|--------------|-------------------------|--------------------------------------|-------------------------------|--------------|
| [161], [162] | Non-cooperative caching | D2D user interactions | Collaborative Filtering | Inductive |
| [163], [164] | | User-content interaction | TL-based popularity estimator | Inductive |
| [165] | | Video rate map | TL-DRL | Unsupervised |
| [166] | Cooperative caching | Knowledge from content popularity | K-means clustering | Inductive |
| [167] | | Hidden features of popularity matrix | Regularized SVD-based CF | Inductive |

a high transfer rate (0.8) leads to better results compared to those of lower transfer rates (0.2 and 0.5). Nevertheless, it is shown in [77] that a transfer rate of 0.5 achieves more desirable results. Thus, we can observe that the setting of TL parameters largely depends on the specific context of the application. Moreover, most of the surveyed approaches transfer the whole weights (i.e., the transfer rate is 1) without any clarification. Tuning such an important parameter is crucial to the effectiveness of the TL process, yet attempts to address this challenge are still limited.

4) Trade-off between Performance and Training Time:

Among the surveyed TL approaches, some approaches freeze the first layers of the target model and only fine-tune the last layers, whereas the other approaches fine-tune all the layers. Generally, the more layers we fine-tune, the better the model can perform because all the weights are adjusted to the target's domain. However, this also necessitates more training time. Thus, determining the number of layers for fine-tuning depends greatly on the wireless network applications. For example, in applications with a longer time window for training, such as channel prediction [86] and radio map construction [71], more layers can be fine-tuned compared to latency-sensitive applications such as embedded IoT [168]. Moreover, in the case where the target data is limited, e.g., new environments, fine-tuning all the layers may even lead to overfitting (as shown in the results of [169] and [170]) since the available labeled data do not represent the domain well. Thus, choosing an appropriate number of layers for fine-tuning remains an open issue.

B. Future Research Directions

1) *Beamforming*: Beamforming refers to the techniques that focus the signal to the receiving devices instead of spreading the signal in all directions. To find the optimal beamforming solution, conventional ML techniques have been widely adopted. However, the obtained beamforming solution is only optimal for a specific network setting, and training ML models for new settings takes a lot of time. To address this issue, TL can be a potential solution. Particularly, a DL model consisting of a conventional CNN and a signal processing module is proposed in [171]. The signal processing module can be placed before the input or after the output of the CNN to convert key features from expert knowledge to the target beamforming matrix. A TL process is then proposed to transfer the knowledge from different network settings to the target task. Specifically, the CNN model of the source task is transferred to the target task, and then new input and output layers will replace the corresponding layers of

the source model. Finally, the new target model is fine-tuned with the target task data. However, there are no simulations or experiments conducted to evaluate the TL process. In [169], a beamforming NN is proposed for beamforming optimization for SINR load balancing, which includes a conventional CNN and a beamforming recovery module at the output of the CNN. Since the SINR load balancing problem has some common features across different wireless environments, a transductive TL process is proposed to utilize the knowledge from different environments. Different from the TL approach in [171], [169] proposes to transfer a pre-trained model to the target task. Specifically, all the layers except the fully connected layer, of the pre-trained network are frozen, and the target data is only used to train the fully connected layer. The reason for training only the last layer is that the lack of data at the target task may lead to overfitting of the target model if all the layers are trained [169]. Moreover, this TL strategy is also beneficial in terms of training time, as only the last layer needs to be trained. Simulation results demonstrate that the proposed TL process helps to improve the convergence rate of the CNN. Although beamforming optimization consists of a wide variety of tasks, there are currently very few works that utilize TL for beamforming. The wireless network environments frequently change, and consequently new data need to be collected and the beamforming ML models need to be trained again. Thus, TL is a very promising solution to reduce the need for new data and training time for beamforming designs.

2) *Embedded IoT Devices*: Embedded IoT devices are significantly restricted in terms of computational resources and latency requirements. Consequently, this hinders the implementation of ML techniques on embedded IoT devices. To address this issue, TL can be an effective solution because the training process can take part in a central node and the target model on an embedded device only needs to fine-tune shortly. In [168], the authors propose to utilize TL for embedded IoT devices. Particularly, a DRL framework is developed for embedded IoT wireless devices, including a one-dimensional CNN located on the FPGA, and another CNN located on the edge/cloud. The CNN on the FPGA receives I/Q samples as input and learns to optimally reconfigure the wireless protocol stack in real-time. Due to the strict time constraints, e.g., 20ms, and the limited computational capacity of the FPGA, it is not possible for the CNN to learn the optimal action. Therefore, a TL approach is developed as illustrated in Fig. 21, which first trains the model on the edge/cloud with different spectrum states until it can achieve a high accuracy. Then, to address the latency issue, the model is tested and its parameters are reduced until it satisfies the latency constraint. After that,

the parameters of the edge/cloud's CNN are transferred to bootstrap the CNN on the FPGA, and they are also periodically transferred to the IoT devices to update. Moreover, the CNN on the FPGA is also trained using I/Q samples, and the obtained information, i.e., states, actions, and values, is transferred to the CNN on the edge/cloud to train the CNN there. Thus, the learning and transfer procedure between the two models can be considered a hybrid of TL and federated learning [14]. This can be a promising future research direction that can combine the strengths of both learning frameworks.

3) General Optimization Problem in Wireless Networks: In [172], the authors formulate the general resource allocation problem as a mixed-integer nonlinear programming (MINLP) model. For solving such MINLP optimization problems, the branch-and-bound algorithm [173] is often employed. However, the solving time of this algorithm grows exponentially as the problem size increases. Therefore, to speed up the branch-and-bound algorithm, the authors propose an imitation learning model to help the branch-and-bound algorithm to prune unnecessary nodes (i.e., pruning policy [173]), thereby improving the branch-and-bound algorithm solving time. However, the pruning policy obtained from the imitation learning model is only effective for the specific MINLP problem which the imitation learning model is trained with. For new problems, the imitation learning model needs to be trained again, which is very time-consuming. Thus, the authors introduce two TL approaches, namely TL via fine-tuning and TL via self-imitation. In the first approach, the previous imitation learning model is transferred to a new target task, and then it is fine-tuned with the target task data, i.e., the new MINLP problem. Although this approach can reduce training time, it still needs labeled data for fine-tuning. In this case, the labels of training data are the optimal solutions, which can only be obtained through the time-consuming process of solving the MINLP. Therefore, the authors propose the second TL approach that does not need labeled data. Particularly, this approach transfers the previous policy from the source task to the target task. Then, the target model will utilize that policy as an initial point and start exploring from there. During the exploration, the objective values of the branch are calculated and set to be the labels, thereby creating labeled data. Simulation results show that the TL via self-imitation approach can help to speed up the training process by 9 to 15 times. It is worth noting that the authors formulate a general optimization problem in this work, and thus the proposed approach can be applied for not only resource allocation problems but also other optimization problems in wireless networks.

4) Network Selection: Various traffic types, e.g., virtual reality or high-definition video, and diverse conditions of available wireless networks at a different time may degrade the quality of service (QoS) for the requesting mobile users. For that, a smart network selection leveraging TL can be utilized to maintain high QoS in wireless networks. In [174], the authors propose a reinforcement TL-based framework using three-level context-aware information to select an indoor network, i.e., LTE, WLAN, or VLC. At the first level, the information of asymmetric downlink-uplink network features and traffic specification is taken into account in the utility function.

Then, a context-aware learning algorithm is developed based on the information of traffic type, location, and time at the second level. At the last level, the periodic update rule of load statistical distributions as the knowledge transfer is employed to incorporate the learning algorithm for the online network selection. In particular, the source domain can first train some observations including suitable networks for all traffic types, preferable networks for users, and time-location dependable network statistical distribution. Using these observations, a TL-based Q-learning algorithm can be executed to train the new observations in the target domain. Via the simulations, the proposed reinforcement TL method can achieve a higher average reward with faster convergence compared with that of conventional reinforcement learning.

5) Improving TL with other ML Techniques: In [170], a DTL approach is proposed for downlink CSI prediction in different propagation environments. In particular, a fully-connected NN is proposed to predict downlink CSI based on uplink CSI data. To address the lack of labeled data in a new propagation environment, a transductive TL approach is proposed, in which the fully-connected NN is trained with different datasets to learn the generalized weights that are suitable for many tasks. Afterward, this pre-trained model is transferred to initialize the model of the target task, and then the target model is fine-tuned with target data. However, this approach may lead to overfitting if the target data is not sufficient. Therefore, the authors propose another method, namely meta-learning. This method trains a model with several datasets and then transfers the model to the target task for fine-tuning. The main difference is that the training of this approach is done separately for each dataset. Then, the obtained weights from each dataset are aggregated, thereby taking into account the loss function of every task (dataset). As a result, the aggregated weights can be used to adapt to new tasks more efficiently than those of TL. Simulation results show that the meta-learning approach can reduce the Normalized Root Mean Square (NMSE) of the prediction by up to 15% and 50% compared to TL and the approach without TL, respectively. Obviously, this meta-learning solution could be very potential to extend to many other problems in wireless networks, such as caching, security and localization.

6) Decentralized TL and Blockchain: Most of the aforementioned TL approaches are centralized, i.e., the transfer of knowledge is governed by a single entity such as a network operator. To fully realize the potential of TL, especially in the era of IoT and 5G/6G networks with a huge number of connected devices, decentralized TL is an aspiring approach. Particularly, the valuable knowledge can be obtained from external sources such as neighboring networks (not controlled by the same operators) or network users (e.g., crowdsourcing). However, for such decentralized approaches, security and privacy become a primary concern, e.g., attackers, eavesdropper, and malicious users. In recent years, blockchain, as a cutting-edge technology for enhancing decentralized network's security and privacy, has been successfully applied for various wireless applications [175], [176]. Thus, utilizing blockchain for decentralized TL is a promising research direction. For example, the authors in [177] propose a blockchain-based framework

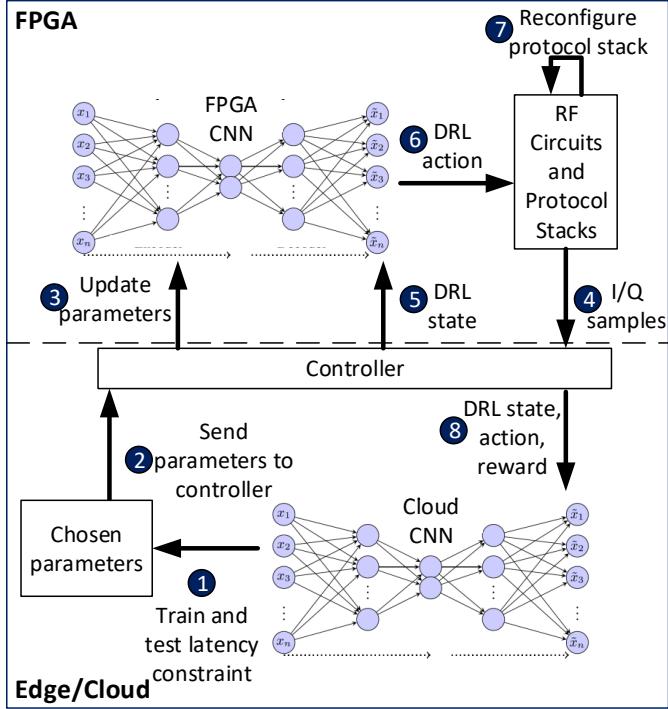


Fig. 21. The TL-based approach for embedded IoT devices [73].

for DTL. Using blockchain and smart contracts [175], [176], the proposed framework allows users to securely share their models, data, and resource. Moreover, the smart contracts also facilitate automatic payment processes, which paves the way for crowdsourcing-enabled TL. Experimental results show that the proposed framework can facilitate ML model sharing over the blockchain network with minimal delay, i.e., less than one minute, while ensuring the users' security and privacy.

X. CONCLUSION

TL is an effective solution to address various challenges that conventional ML techniques are facing in numerous wireless network applications. In this article, we have presented a comprehensive survey on TL applications in wireless networks. We have first provided an in-depth tutorial of TL including formal definitions, classification, and fundamental background of various TL techniques. Then, we have discussed the applications of TL for various issues of wireless networks, including spectrum management, localization, signal recognition, security, human activity recognition, and caching. For these TL applications, we have provided detailed discussions and analyses on how TL can be leveraged to overcome the challenges that the underlying ML techniques are facing. Finally, we have presented the current challenges, open issues, and introduced some potential research directions of TL for future wireless networks.

REFERENCES

- [1] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, June 2019.
- [2] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, Oct. 2012.
- [3] Samsung Research, "The next hyper-connected experience for all," Samsung, Tech. Rep., July 2020.
- [4] Ericsson AI and Automation, "Employing AI techniques to enhance returns on 5G network investments," Ericsson, Tech. Rep., May 2019.
- [5] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 96–103, Apr. 2020.
- [6] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, July 2019.
- [7] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, July 2020.
- [8] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and information systems*, vol. 36, no. 3, pp. 537–556, Sep. 2013.
- [9] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Information Fusion*, vol. 49, pp. 1–25, Sep. 2019.
- [10] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the internet of things: A comprehensive survey," *Ad Hoc Networks*, vol. 93, p. 101913, Oct. 2019.
- [11] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2392–2431, July 2017.
- [12] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, Apr. 2014.
- [13] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, Mar. 2019.
- [14] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, Apr. 2020.
- [15] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, May 2019.
- [16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [17] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, May 2016.
- [18] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*, Sep. 2018, pp. 270–279.
- [19] A. Ng, "Nuts and bolts of building AI applications using deep learning," *NIPS Keynote Talk*, Dec. 2016.
- [20] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, June 2007, pp. 759–766.
- [21] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*, June 2007, pp. 193–200.
- [22] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *AAAI*, vol. 7, July 2007, pp. 540–545.
- [23] S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a meta-level prior for feature relevance from multiple related tasks," in *Proceedings of the 24th international conference on Machine learning*, June 2007, pp. 489–496.
- [24] T. Jebara, "Multi-task feature and kernel selection for SVMs," in *Proceedings of the twenty-first international conference on Machine learning*, July 2004, pp. 55–63.
- [25] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *Proceedings of the 25th international conference on Machine learning*, July 2008, pp. 1120–1127.

- [26] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *AAAI*, vol. 7, July 2007, pp. 608–614.
- [27] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Seventh IEEE International Conference on Data Mining Workshops*, Oct. 2007, pp. 77–82.
- [28] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proceedings of the twenty-first international conference on Machine learning*, July 2004, pp. 114–122.
- [29] E. Sangineto, G. Zen, E. Ricci, and N. Sebe, "We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer," in *Proceedings of the 22nd ACM international conference on Multimedia*, Nov. 2014, pp. 357–366.
- [30] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *Proceedings of the 25th international conference on Machine learning*, July 2008, pp. 200–207.
- [31] J. Huang and S. Gong, "Unsupervised transfer learning with self-supervised remedy," *arXiv preprint arXiv:2006.04737*, June, 2020.
- [32] Y. Wang, Q. Wang, S. Shi, X. He, Z. Tang, K. Zhao, and X. Chu, "Benchmarking the performance and energy efficiency of AI accelerators for AI training," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, 2020, pp. 744–751.
- [33] <https://github.com/BVLC/caffe/wiki/Model-Zoo>, (Accessed on 24/01/2021).
- [34] <https://keras.io/api/applications/>, (Accessed on 24/01/2021).
- [35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [36] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: a deep learning approach," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, June 2011, pp. 513–520.
- [37] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.
- [38] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014, pp. 3320–3328.
- [39] M. Abu Alsheikh, D. T. Hoang, D. Niyato, H. Tan, and S. Lin, "Markov decision processes with applications in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1239–1267, Apr. 2015.
- [40] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
- [41] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [42] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [43] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *arXiv preprint arXiv:2009.07888*, Sep. 2020.
- [44] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *In Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann, 1999, pp. 278–287.
- [45] E. Wiewiora, G. W. Cottrell, and C. Elkan, "Principled methods for advising reinforcement learning agents," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 792–799.
- [46] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, Mar. 2015.
- [47] F. Fernández and M. Veloso, "Probabilistic policy reuse in a reinforcement learning agent," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 720–727.
- [48] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," in *Advances in neural information processing systems*, 2017, pp. 4055–4065.
- [49] A. Gupta, C. Devin, Y. Liu, P. Abbeel, and S. Levine, "Learning invariant feature spaces to transfer skills with reinforcement learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Apr. 2017.
- [50] G. Konidaris and A. Barto, "Autonomous shaping: Knowledge transfer in reinforcement learning," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 489–496.
- [51] H. B. Ammar and M. E. Taylor, "Reinforcement learning transfer via common subspaces," in *International Workshop on Adaptive and Learning Agents*. Springer, 2011, pp. 21–36.
- [52] H. B. Ammar, K. Tuyls, M. E. Taylor, K. Driessens, and G. Weiss, "Reinforcement learning transfer via sparse coding," in *Proceedings of the 11th international conference on autonomous agents and multiagent systems*, vol. 1. International Foundation for Autonomous Agents and Multiagent Systems, June 2012, pp. 383–390.
- [53] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.
- [54] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *arXiv preprint arXiv:1701.08734*, 2017.
- [55] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2169–2176.
- [56] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband *et al.*, "Deep q-learning from demonstrations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018, pp. 3223–3230.
- [57] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE communications surveys & tutorials*, vol. 19, no. 2, pp. 1277–1304, Nov. 2016.
- [58] Q. Peng, A. Gilman, N. Vasconcelos, P. C. Cosman, and L. B. Milstein, "Robust deep sensing through transfer learning in cognitive radio," *IEEE Wireless Communications Letters*, vol. 9, no. 1, pp. 38–41, Sep. 2019.
- [59] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Communications*, vol. 17, no. 2, pp. 138–148, Mar. 2020.
- [60] B. M. Pati, M. Kaneko, and A. Taparugssanagorn, "A deep convolutional neural network based transfer learning method for non-cooperative spectrum sensing," *IEEE Access*, vol. 8, pp. 164 529–164 545, Sep. 2020.
- [61] G. I. Tsipopoulos, O. A. Dobre, M. H. Ahmed, and K. E. Baddour, "Radio resource allocation techniques for efficient spectrum access in cognitive radio networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 824–847, Oct. 2014.
- [62] F. Lin, J. Chen, J. Sun, G. Ding, and L. Yu, "Cross-band spectrum prediction based on deep transfer learning," *China Communications*, vol. 17, no. 2, pp. 66–80, Mar. 2020.
- [63] J. Zhao, W. Shao, F. Li, and Q. Zhou, "A spectrum handoff method based on reinforcement and transfer learning," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, Aug. 2020, pp. 572–575.
- [64] K. Cao and P. Qian, "Spectrum handoff based on dqn predictive decision for hybrid cognitive radio networks," *Sensors*, vol. 20, no. 4, p. 1146, Jan. 2020.
- [65] A. Koushik, F. Hu, and S. Kumar, "Intelligent spectrum management based on transfer actor-critic learning for rateless transmissions in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1204–1215, Aug. 2017.
- [66] Q. Zhao, D. Grace, and T. Clarke, "Transfer learning and cooperation management: balancing the quality of service and information exchange overhead in cognitive radio networks," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 2, pp. 290–301, Feb. 2015.
- [67] A. Galindo-Serrano, L. Giupponi, and G. Auer, "Distributed learning in multiuser ofdm femtocell networks," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, May 2011, pp. 1–6.
- [68] F. Shah-Mohammadi and A. Kwasinski, "Deep reinforcement learning approach to qoe-driven resource allocation for spectrum underlay in cognitive radio networks," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.
- [69] —, "Fast learning cognitive radios in underlay dynamic spectrum access: Integration of transfer learning into deep reinforcement learning," in *2020 Wireless Telecommunications Symposium (WTS)*, Apr. 2020, pp. 1–7.

- [70] C. Phillips, D. Sicker, and D. Grunwald, "A survey of wireless path loss prediction and coverage mapping methods," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 255–270, Mar. 2012.
- [71] C. Parera, Q. Liao, I. Malanchini, C. Tatino, A. E. Redondi, and M. Cesana, "Transfer learning for tilt-dependent radio map prediction," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 829–843, Jan. 2020.
- [72] X. Han, L. Xue, Y. Xu, and Z. Liu, "A two-phase transfer learning-based power spectrum maps reconstruction algorithm for underlay cognitive radio networks," *IEEE Access*, vol. 8, pp. 81 232–81 245, Apr. 2020.
- [73] Q. Zhao, D. Grace, A. Vilhar, and T. Javornik, "Using k-means clustering with transfer and q learning for spectrum, load and energy optimization in opportunistic mobile broadband networks," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2015, pp. 116–120.
- [74] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon," *European Journal of Operational Research*, pp. 405–421, Aug. 2020.
- [75] Q. Zhao, T. Jiang, N. Morozs, D. Grace, and T. Clarke, "Transfer learning: A paradigm for dynamic spectrum and topology management in flexible architectures," in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, Sep. 2013, pp. 1–5.
- [76] S. Sharma, S. J. Darak, A. Srivastava, and H. Zhang, "A transfer learning framework for energy efficient Wi-Fi networks and performance analysis using real data," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Nov. 2016, pp. 1–6.
- [77] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "Tact: A transfer actor-critic learning framework for energy saving in cellular radio access networks," *IEEE transactions on wireless communications*, vol. 13, no. 4, pp. 2000–2011, Feb. 2014.
- [78] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning-based mode selection and resource management for green fog radio access networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1960–1971, Sep. 2018.
- [79] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vučetic, "Deep learning for radio resource allocation with diverse quality-of-service requirements in 5G," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2309–2324, Apr. 2021.
- [80] T. Sahin, R. Khalili, M. Boban, and A. Wolisz, "VRLS: A unified reinforcement learning scheduler for vehicle-to-vehicle communications," in *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*, Sep. 2019, pp. 1–7.
- [81] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state transfer learning for data correlation aware resource allocation in wireless virtual reality," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct. 2017, pp. 1852–1856.
- [82] C. Parera, Q. Liao, I. Malanchini, D. Wellington, A. E. Redondi, and M. Cesana, "Transfer learning for multi-step resource utilization prediction," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, Aug. 2020, pp. 1–6.
- [83] Q. Zeng, Q. Sun, G. Chen, H. Duan, C. Li, and G. Song, "Traffic prediction of wireless cellular networks based on deep transfer learning and cross-domain data," *IEEE Access*, vol. 8, pp. 172 387–172 397, Sep. 2020.
- [84] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389–1401, Mar. 2019.
- [85] N. Wagle and E. W. Frew, "Transfer learning for dynamic rf environments," in *2012 American Control Conference (ACC)*, June 2012, pp. 1406–1411.
- [86] C. Parera, R. A. EC, M. Cesana, Q. Liao, and I. Malanchini, "Transfer learning for channel quality prediction," in *2019 IEEE International Symposium on Measurements & Networking (M&N)*, July 2019, pp. 1–6.
- [87] G. Yang, Y. Zhang, Z. He, J. Wen, Z. Ji, and Y. Li, "Machine-learning-based prediction methods for path loss and delay spread in air-to-ground millimetre-wave channels," *IET Microwaves, Antennas & Propagation*, vol. 13, no. 8, pp. 1113–1121, Mar. 2019.
- [88] W. Wang, J. Zhang, and Q. Zhang, "Transfer learning based diagnosis for configuration troubleshooting in self-organizing femtocell networks," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Dec. 2011, pp. 1–5.
- [89] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 60–69, July 2019.
- [90] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan, "Adaptive localization in a dynamic WiFi environment through multi-view learning," in *Association for the Advancement of Artificial Intelligence*, vol. 2, July 2007, pp. 1108–1113.
- [91] V. Sindhwani, P. Niyogi, and M. Belkin, "A co-regularization approach to semi-supervised learning with multiple views," in *Proceedings of ICML Workshop on Learning with Multiple Views*, vol. 2005, Aug. 2005, pp. 74–79.
- [92] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, "Transfer learning for WiFi-based indoor localization," in *Association for the Advancement of Artificial Intelligence Workshop*, vol. 6, July 2008, pp. 43–48.
- [93] Z. Sun, Y. Chen, J. Qi, and J. Liu, "Adaptive localization through transfer learning in indoor Wi-Fi environment," in *Proceedings of the 7th IEEE International Conference on Machine Learning and Applications*, Dec. 2008, pp. 331–336.
- [94] X. Zhang, Y. Mei, H. Jin, and D. Liang, "TL-FCMA: indoor localization by integrating fuzzy clustering with transfer learning," in *Proceedings of the IEEE International Conference on Network Infrastructure and Digital Content*, Aug. 2018, pp. 372–377.
- [95] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed, "Joint indoor localization and radio map construction with limited deployment load," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 1031–1043, July 2014.
- [96] K. Liu, H. Zhang, J. K.-Y. Ng, Y. Xia, L. Feng, V. C. Lee, and S. H. Son, "Toward low-overhead fingerprint-based indoor localization via transfer learning: design, implementation, and evaluation," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 898–908, Sept. 2017.
- [97] H. Zou, Y. Zhou, H. Jiang, B. Huang, L. Xie, and C. Spanos, "Adaptive localization in dynamic indoor environments by transfer kernel learning," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, Mar. 2017, pp. 1–6.
- [98] A. Hu and L. Zhang, "Adaptive indoor localization with Wi-Fi based on transfer learning," in *Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan*, May 2019, pp. 1–2.
- [99] L. Wang, Y. Shao, and X. Guo, "An adaptive localization approach based on deep adaptation networks," in *Proceedings of the IEEE International Conference on Control, Automation and Information Sciences*, Oct. 2019, pp. 1–5.
- [100] L. H. Yong and M. Zhao, "Indoor positioning based on hybrid domain transfer learning," *IEEE Access*, vol. 8, pp. 130 527–130 539, July 2020.
- [101] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "CRISLoc: reconstructable CSI fingerprinting for indoor smartphone localization," *IEEE Internet of Things Journal*, Sept. 2020.
- [102] X. Rao, Z. Li, and Y. Yang, "Device-free passive wireless localization system with transfer deep learning method," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1–17, Jan. 2020.
- [103] C. Qiu and M. W. Mutka, "Self-improving indoor localization by profiling outdoor movement on smartphones," in *Proceedings of the 18th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*, June 2017, pp. 1–9.
- [104] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket-an experimental study," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, Mar. 2010, pp. 162–170.
- [105] C. M. D. Junior, S. P. da Silva, R. V. da Nobrega, A. C. Barros, A. K. Sangalha, P. P. Reboucas Filho, and V. H. C. de Albuquerque, "A new approach for mobile robot localization based on an online IoT system," *Future Generation Computer Systems*, vol. 100, pp. 859–881, Nov. 2019.
- [106] Z. Tang, J. Zhang, X. Gong, X. Cheng, X. Chen, D. Fang, and W. Wang, "CTLL: a cell-based transfer learning method for localization in large scale wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, pp. 1–18, Aug. 2015.
- [107] Y.-S. Chen, C.-S. Hsu, C.-Y. Huang, and H.-M. Hung, "Outdoor localization for LoRaWans using semi-supervised transfer learning with grid segmentation," in *Proceedings of the IEEE VTS Asia Pacific Wireless Communications Symposium*, Aug. 2019, pp. 1–5.
- [108] A. Pimpinella, A. E. Redondi, M. Nicoli, and M. Cesana, "Machine learning based localization of LoRaWAN devices via inter-technology knowledge transfer," in *Proceedings of the IEEE International Conference on Communications Workshops*, June 2020, pp. 1–6.
- [109] H. Zou, Y. Zhou, H. Jiang, B. Huang, L. Xie, and C. Spanos, "A transfer kernel learning based strategy for adaptive localization in dynamic

- indoor environments: poster,” in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, Oct. 2016, pp. 462–464.
- [110] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, “Transfer learning for semi-supervised automatic modulation classification in ZF-MIMO systems,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, June 2020.
- [111] T. Xu and I. Darwazeh, “Deep learning for over-the-air non-orthogonal signal classification,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, June 2020, pp. 1–5.
- [112] K. Bu, Y. He, X. Jing, and J. Han, “Adversarial transfer learning for deep learning based automatic modulation classification,” *IEEE Signal Processing Letters*, pp. 880–884, May 2020.
- [113] M. A. Selver, T. Toprak, M. Seçmen, and E. Y. Zoral, “Transferring synthetic elementary learning tasks to classification of complex targets,” *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 11, pp. 2267–2271, Nov. 2019.
- [114] C. Liu, Z. Wei, D. W. K. Ng, J. Yuan, and Y.-C. Liang, “Deep transfer learning for signal detection in ambient backscatter communications,” *IEEE Transactions on Wireless Communications*, Nov. 2020.
- [115] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb. 2018.
- [116] S. Chen, S. Zheng, L. Yang, and X. Yang, “Deep learning for large-scale real-world ACARS and ADS-B radio signal classification,” *IEEE Access*, vol. 7, pp. 89 256–89 264, July 2019.
- [117] Y. Xu, D. Li, Z. Wang, Q. Guo, and W. Xiang, “A deep learning method based on convolutional neural network for automatic modulation classification of wireless signals,” *Wireless Networks*, vol. 25, no. 7, pp. 3735–3746, Oct. 2019.
- [118] S. Ujan, N. Navidi, R. Jr Landry *et al.*, “An efficient radio frequency interference (RFI) recognition and characterization using end-to-end transfer learning,” *Applied Sciences*, vol. 10, no. 19, pp. 6885–6901, Oct. 2020.
- [119] Z. Yang, W. Qiu, H. Sun, and A. Nallanathan, “Robust radar emitter recognition based on the three-dimensional distribution feature and transfer learning,” *Sensors*, vol. 16, no. 3, pp. 289–303, 2016.
- [120] G. Zhang and X. Li, “A new recognition system for radar emitter signals,” *Kybernetes: The International Journal of Systems & Cybernetics*, vol. 41, no. 9, pp. 1351–1360, 2012.
- [121] Z. Yang, Z. Wu, Z. Yin, T. Quan, and H. Sun, “Hybrid radar emitter recognition based on rough k-means classifier and relevance vector machine,” *Sensors*, vol. 13, no. 1, pp. 848–864, 2013.
- [122] L. Wang, H.-B. Ji, and Y. Jin, “Fuzzy passive-aggressive classification: A robust and efficient algorithm for online classification problems,” *Information Sciences*, vol. 220, pp. 46–63, Jan. 2013.
- [123] Y. Xiao, W. Liu, and L. Gao, “Radar signal recognition based on transfer learning and feature fusion,” *Mobile Networks and Applications*, pp. 1–9, Nov. 2019.
- [124] Q. Wang, P. Du, J. Yang, G. Wang, J. Lei, and C. Hou, “Transferred deep learning based waveform recognition for cognitive passive radar,” *Signal Processing*, vol. 155, pp. 259–267, Feb. 2019.
- [125] J. Lundén and V. Koivunen, “Automatic radar waveform recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 124–136, May 2007.
- [126] C. Wang, J. Wang, and X. Zhang, “Automatic radar waveform recognition based on time-frequency analysis and convolutional neural network,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 2437–2441.
- [127] T. O’shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [128] N. E. West and T. O’Shea, “Deep architectures for modulation recognition,” in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Mar. 2017, pp. 1–6.
- [129] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Jan. 2018.
- [130] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, “A survey on security aspects for 3GPP 5G networks,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 170–195, Nov. 2019.
- [131] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, Oct. 2015.
- [132] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, “Deep transfer learning for IoT attack detection,” *IEEE Access*, vol. 8, pp. 107 335–107 344, June 2020.
- [133] T. Wen and R. Keyes, “Time series anomaly detection using convolutional neural networks and transfer learning,” *arXiv preprint arXiv:1905.13628*, May 2019.
- [134] Y. Sharaf-Dabbagh and W. Saad, “On the authentication of devices in the internet of things,” in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016, pp. 1–3.
- [135] Y. S. Dabbagh and W. Saad, “Authentication of wireless devices in the internet of things: Learning and environmental effects,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6692–6705, Apr. 2019.
- [136] Y. Sharaf-Dabbagh and W. Saad, “Transfer learning for device fingerprinting with application to cognitive radio networks,” in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Aug. 2015, pp. 2138–2142.
- [137] C. Zhao, Z. Cai, M. Huang, M. Shi, X. Du, and M. Guizani, “The identification of secular variation in IoT based on transfer learning,” in *2018 International Conference on Computing, Networking and Communications (ICNC)*, Mar. 2018, pp. 878–882.
- [138] Z. Yu, R. Zong, and S. Li, “Individual recognition based on transfer learning for wireless network devices,” in *Proceedings of the 3rd International Conference on Advances in Image Processing*, Nov. 2019, pp. 45–49.
- [139] X. Wang, Y. Zhang, H. Zhang, Y. Li, and X. Wei, “Radio frequency signal identification using transfer learning based on LSTM,” *Circuits, Systems, and Signal Processing*, pp. 1–15, 2020.
- [140] S. Vadlamani, B. Eksioglu, H. Medal, and A. Nandi, “Jamming attacks on wireless networks: A taxonomic survey,” *International Journal of Production Economics*, vol. 172, pp. 76–94, Feb. 2016.
- [141] G. Chen, Y. Zhan, Y. Chen, L. Xiao, Y. Wang, and N. An, “Reinforcement learning based power control for in-body sensors in wbans against jamming,” *IEEE Access*, vol. 6, pp. 37 403–37 412, June 2018.
- [142] C. Han and Y. Niu, “Multi-regional anti-jamming communication scheme based on transfer learning and q learning,” *KSII Transactions on Internet & Information Systems*, vol. 13, no. 7, pp. 3333–3350, July 2019.
- [143] X. Lu, L. Xiao, C. Dai, and H. Dai, “Uav-aided cellular communications with deep reinforcement learning against jamming,” *IEEE Wireless Communications*, vol. 27, no. 4, pp. 48–53, Aug. 2020.
- [144] V. Q. Do and I. Koo, “Learning frameworks for cooperative spectrum sensing and energy-efficient data protection in cognitive radio networks,” *Applied Sciences*, vol. 8, no. 5, pp. 722–746, May 2018.
- [145] A. Bhowmik and D. De, “mTrust: Call behavioral trust predictive analytics using unsupervised learning in mobile cloud computing,” *Wireless Personal Communications*, pp. 1–19, Nov. 2020.
- [146] J. Shang and J. Wu, “A robust sign language recognition system with sparsely labeled instances using Wi-Fi signals,” in *Proceedings of the 14th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, Oct. 2017, pp. 99–107.
- [147] Q. Bu, G. Yang, J. Feng, and X. Ming, “Wi-Fi based gesture recognition using deep transfer learning,” in *Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, Oct. 2018, pp. 590–595.
- [148] L. Zhang, Z. Wang, and L. Yang, “Commercial Wi-Fi based fall detection with environment influence mitigation,” in *Proceedings of the 16th IEEE International Conference on Sensing, Communication, and Networking*, June 2019, pp. 1–9.
- [149] S. Arshad, C. Feng, R. Yu, and Y. Liu, “Leveraging transfer learning in multiple human activity recognition using WiFi signal,” in *Proceedings of the 20th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*, June 2019, pp. 1–10.
- [150] X. Ding, T. Jiang, Y. Li, W. Xue, and Y. Zhong, “Device-free location-independent human activity recognition using transfer learning based on CNN,” in *Proceedings of the IEEE International Conference on Communications Workshops*, June 2020, pp. 1–6.
- [151] B. Sheng, F. Xiao, L. Sha, and L. Sun, “Deep spatial-temporal model based cross-scene action recognition using commodity WiFi,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3592–3601, Feb. 2020.
- [152] Y. Zhou, M. Han, J. S. He, L. Liu, X. Xu, and X. Gao, “Abnormal activity detection in edge computing: a transfer learning approach,” in *Proceedings of the IEEE International Conference on Computing, Networking and Communications*, Feb. 2020, pp. 107–111.

- [153] S. A. Rokni and H. Ghasemzadeh, "Autonomous training of activity recognition algorithms in mobile sensors: a transfer learning approach in context-invariant views," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1764–1777, Jan. 2018.
- [154] L. Valerio, A. Passarella, and M. Conti, "Hypothesis transfer learning for efficient data computing in smart cities environments," in *Proceedings of the IEEE International Conference on Smart Computing*, May 2016, pp. 1–8.
- [155] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, Apr. 2020.
- [156] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International Workshop on Ambient Assisted Living*, Dec. 2012, pp. 216–223.
- [157] F. Shi, K. Chetty, and S. Julier, "Passive activity classification using just WiFi probe response signals," in *Proceedings of the IEEE Radar Conference*, Apr. 2019, pp. 1–6.
- [158] J. Jung, J. Kim, and K.-A. Toh, "Transfer learning of Wi-Fi handwritten signature signals for identity verification based on the kernel and the range space projection," in *Proceedings of the IEEE International Conference on Image Processing*, Sept. 2019, pp. 2586–2590.
- [159] K.-A. Toh, "Kernel and range approach to analytic network learning," *International Journal of Networked and Distributed Computing*, vol. 7, no. 1, pp. 20–28, Dec. 2018.
- [160] J. Shuja, K. Bilal, W. Alasmary, H. Sinky, and E. Alanazi, "Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 181, pp. 103 005–103 029, May 2021.
- [161] E. Baştug, M. Bennis, and M. Debbah, "Anticipatory caching in small cell networks: a transfer learning approach," in *Proceedings of the 1st KuVS Workshop on Anticipatory Networks*, Sept. 2014, pp. 1–3.
- [162] E. Baştug, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2015, pp. 161–166.
- [163] B. B. Nagaraja and K. G. Nagananda, "Caching with unknown popularity profiles in small cell networks," in *Proceedings of the IEEE Global Communications Conference*, Dec. 2015, pp. 1–6.
- [164] B. Bharath, K. G. Nagananda, and H. V. Poor, "A learning-based approach to caching in heterogeneous small cell networks," *IEEE Transactions on Communications*, vol. 64, no. 4, pp. 1674–1686, Mar. 2016.
- [165] Z. Zhang, Y. Zheng, C. Li, Y. Huang, and L. Yang, "Cache-enabled adaptive bit rate streaming via deep self-transfer reinforcement learning," in *Proceedings of the 10th International Conference on Wireless Communications and Signal Processing*, Oct. 2018, pp. 1–6.
- [166] T. Hou, G. Feng, S. Qin, and W. Jiang, "Proactive content caching by exploiting transfer learning for mobile edge computing," in *Proceedings of the IEEE Global Communications Conference*, July 2017, pp. 1–6.
- [167] Y. Wang, Y. Chen, H. Dai, Y. Huang, and L. Yang, "A learning-based approach for proactive caching in wireless communication networks," in *Proceedings of the 9th International Conference on Wireless Communications and Signal Processing*, Oct. 2017, pp. 1–6.
- [168] F. Restuccia and T. Melodia, "Deepwiel: Bringing deep reinforcement learning to the internet of self-adaptive things," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, July 2020, pp. 844–853.
- [169] Y. Yuan, G. Zheng, K.-K. Wong, B. Ottersten, and Z.-Q. Luo, "Transfer learning and meta learning based fast downlink beamforming adaptation," *IEEE Transactions on Wireless Communications*, Nov. 2020.
- [170] Y. Yang, F. Gao, Z. Zhong, B. Ai, and A. Alkhateeb, "Deep transfer learning based downlink channel prediction for FDD massive MIMO systems," *IEEE Transactions on Communications*, pp. 1994–1998, Aug. 2020.
- [171] W. Xia, G. Zheng, K.-K. Wong, and H. Zhu, "Model-driven beamforming neural networks," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 68–75, Mar. 2020.
- [172] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Transfer learning for mixed-integer resource allocation problems in wireless networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.
- [173] M. Conforti, G. Cornuéjols, G. Zambelli *et al.*, *Integer programming*, Berlin: Springer, 2014.
- [174] Z. Du, C. Wang, Y. Sun, and G. Wu, "Context-aware indoor VLC/RF heterogeneous network selection: reinforcement learning with knowledge transfer," *IEEE Access*, vol. 6, pp. 33 275–33 284, June 2018.
- [175] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, June 2019.
- [176] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities," *IEEE Access*, vol. 7, pp. 85 727–85 745, June 2019.
- [177] A. ul Haque, M. S. Ghani, and T. Mahmood, "Decentralized transfer learning using blockchain & ipfs for deep learning," in *2020 International Conference on Information Networking (ICOIN)*, Jan. 2020, pp. 170–177.