Received November 23, 2021, accepted December 6, 2021, date of publication December 10, 2021, date of current version December 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134628

Transfer Learning With Time Series Data: A Systematic Mapping Study

MANUEL WEBER[1], MAXIMILIAN AUCH[1], CHRISTOPH DOBLANDER[2], PETER MANDL[1], AND HANS-ARNO JACOBSEN[3], (Fellow, IEEE)

[1]Department of Computer Science and Mathematics, Munich University of Applied Sciences (HM), 80335 Munich, Germany
[2]Chair for Application and Middleware Systems, Technical University of Munich (TUM), 85748 Garching, Germany
[3]Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada

Corresponding author: Manuel Weber (manuel.weber@hm.edu)

This work was supported in part by the Munich University of Applied Sciences, and in part by the German Research Foundation (DFG) through the ''Open Access Publishing'' Program

ABSTRACT Transfer Learning is a well-studied concept in machine learning, that relaxes the assumption that training and testing data need to be drawn from the same distribution. Recent success in applying transfer learning in the area of computer vision has motivated research on transfer learning also in context of time series data. This benefits learning in various time series domains, including a variety of domains based on sensor values. In this paper, we conduct a systematic mapping study of literature on transfer learning with time series data. Following the review guidelines of Kitchenham and Charters, we identify and analyze 223 relevant publications. We describe the pursued approaches and point out trends. Especially during the last two years, there has been a vast increase in the number of publications on the topic. This paper's findings can help researchers as well as practitioners getting into the field and can help identify research gaps.

INDEX TERMS Time series, transfer learning, domain adaptation, deep learning, survey.

I. INTRODUCTION

Time series data has recently emerged as a new application area for deep learning and transfer learning [1]–[3]. While transfer learning (TL) has been extensively studied within the fields of computer vision and natural language processing [4], applications within the research area of time series analysis are still rare. Back in 2006, mining time series data was identified as one of the ten most challenging problems in data mining research [5]. Since then, it has gained high research interest [1], [6]. There are various approaches to time series classification (TSC) and other time series problems, that can, for instance, be based on comparison of the whole time series, on comparison of selected intervals or shapelets, or on dictionaries of pattern counts [6]. Beside these, also model-based approaches are promising. Following the advances towards deep learning in computer vision, there has also been an increase in studies applying deep learning models with time series data [7]. Recent examples of deep time series models are InceptionTime [8] or TimeNet [3]. Two recent literature reviews give an overview of deep learning in the field of TSC: Fawaz *et al.* [1] compare several state-of-the-art deep learning models in diverse time series domains and provide a taxonomy for deep learning approaches in TSC. Ebrahim *et al.* [7] analyze bibliographic metadata of publications found in the electronic database Scopus and identify deep learning as the number one topic in publications on TSC. In addition to this, Wang *et al.* [9] provide an overview of deep learning advances in the specific field of activity recognition based on low-level sensor values. In the course of increasing interest in deep learning, also, studies have come up, that address the concept of TL in the context of time series data [2], [3], [10]. Typical applications of TL are object recognition or detection in images, action recognition in videos, document categorization, or text sentiment analysis [4]. Large pre-trained models such as VGG [11] or AlexNet [12] are well-known for TL with image data. However, regarding time series, such as sensor readings, TL has not been widely investigated in the past. Fawaz *et al.* [2] have shown that TL can effectively improve a TSC model's generalization capability and provide better predictions. Further studies have also addressed TL for other time series prediction problems, such as time series forecasting [13], [14]. As the availability of data is limited in many time series domains, TL can have a great impact in diverse use cases. It has recently been applied for detecting human occupancy in building rooms based on carbon dioxide

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Mei Zhang.

measurements [15]–[17], for the prediction of wind power or speed [18], [19], for human activity recognition based on inertial sensor readings [20], [21], as well as for forecasting financial time series data [22], [23].

To the best of our knowledge, to date, there is no literature review on the work that has been done towards TL with time series data. Several reviews have been published on TL in general [24]–[27] and on deep learning methods in the context of time series data [1], [7], [28], [29]. There are also reviews on TL in specific time series domains, such as human activity recognition [30], [31] or brain-computer interfaces [32]. In this paper, we review literature on TL approaches with focus on time series data, including univariate or multivariate one-dimensional time series. We conduct a comprehensive literature review in the form of a systematic mapping study. According to Kitchenham and Charters [33], a systematic mapping study, or scoping study, is suitable to provide an overview over a broad topic and can help identify more specific research questions. The overall goal of this study is to provide an overview of the first work in this new application field of TL and to identify research trends. The review addresses the following research questions:

Q.1) What are the main application domains where TL with time series data has been investigated?
Q.2) What TL approaches are used for transfer with time series data?
Q.3) What machine learning model types are used within approaches on time series TL?

The approaches covered in this review can help reduce the required amount of data in systems for, e.g., building automation, healthcare monitoring or financial forecasting. Time series TL breaks with the current paradigm of collecting as much data as possible for a specific purpose. It enables the use of machine learning solutions in a wider range of use cases involving limited amounts of sensor data or other time series data.

The remainder of the paper is structured as follows: First, in Section II, we provide an overview of the relevant terms in this review. This includes the definition of TL and time series. Section III describes our review methodology. Section IV reports our findings in regard to the above research questions Q.1-3. After that, Section V points out important future research opportunities, and Section VI lists potential threats to validity of this study.

II. OVERVIEW AND DEFINITIONS

This section provides an overview of the relevant terms and concepts used in this publication.

A. TIME SERIES

Time series data represents observations at different points in time. The aspect of time sets time series data apart from other types of data. It allows carrying information on temporal patterns, such as trends or seasonality. We define a time series as follows.

Definition 1 (Time Series): A time series $T = [x_1, \ldots, x_n]$ denotes an ordered sequence of data points x_i of length n, where each data point is either a real value or a vector of real values, and data points are regularly recorded at a constant time step Δt after the previous point.

Note that in this paper, we do not consider irregular time series without a regular Δt, where data points may appear at any arbitrary time. In this case, data points may refer to certain events, e.g., social media postings. Instead, our definition refers to data that can be continuously recorded, such as sensor time series, or data that is interpolated in order to form a regular time series. A dataset X in context of this work can be a single coherent time series T with subsequences as instances, or a set of time series $\{T_1, \ldots, T_n\}$.

In accordance with the previous definition, we derive the following two definitions for two types of time series data: *univariate* and *multivariate time series*.

Definition 2 (Univariate Time Series): A univariate time series is a time series where $x_i \in \mathbb{R}$.

Definition 3 (Multivariate Time Series): A multivariate time series is a time series where each x_i is a d-dimensional vector of real values $(x_i^1, \ldots, x_i^d)$, $x_i^j \in \mathbb{R}$.

While univariate time series have a single time-dependent variable, e.g., periodic sensor readings from one specific sensor, or the price history of a certain financial asset, multivariate time series combine multiple time-dependent variables. These can represent, for example, different sensor modalities, sensor channels, or values from sensors placed in different locations. For the financial example, it can include values of multiple assets in a market. Definition 3 allows to address sensor data from multiple sensors placed in different locations, where the sensor locations are unknown or documented in some metadata. This information is, however, not contained in the time series data itself. We refer to data that includes temporal as well as spatial aspects as *spatio-temporal data*. With two space dimensions, spatio-temporal data can be realized in form of a grid of values per time step, i.e., $x_i^j \in \mathbb{R} \times \mathbb{R}$. Such data is used for applications in remote sensing or for the description of moving object trajectories. This literature review addresses temporal data only.

B. TIME SERIES PROBLEMS

Time series problems that can be supported by TL include time *series classification*, *regression*, and *clustering*. Problems may be defined on the whole time series or on subsequences, where either each subsequence has a predefined length, or an additional time series segmentation is involved. We define the following problems.

Definition 4 (Time Series Classification): Time series classification (TSC) denotes the problem of assigning a time series or a subsequence within a time series to a class c_i out of a set of classes $C = \{c_1, \ldots, c_n | n \geq 2\}$.

Definition 5 (Time Series Regression): For a time series T, time series regression denotes the problem of predicting a numeric value y or multiple numeric values $y_1, \ldots, y_n$.

Definition 6 (Time Series Clustering): Time series clustering denotes the problem of assigning time series or subsequences of time series to a set of clusters $C = \{c_1, \ldots, c_n | n \geq 1\}$ based on a similarity measure $Sim(a, b)$, where the number of existing clusters n is either pre-defined or to be determined.

Beside these, there are frequently addressed subcategories of TSC and time series regression, named *anomaly detection* and *forecasting*, which we define as follows.

Definition 7 (Time Series Anomaly Detection): Time series anomaly detection denotes the problem of assigning a time series or a subsequence of a time series to one out of two strongly imbalanced classes $\{c_{normal}, c_{anomaly}\}$. c_{normal} represents the majority class of normal state observations, while $c_{anomaly}$ represents the class of rare observations, i.e., anomalies.

Definition 8 (Time Series Forecasting): For a univariate time series $T = [x_1, \ldots, x_n]$, time series forecasting denotes the problem of predicting the next value of the sequence x_{n+1}, or the next m values $x_{n+1}, \ldots, x_{n+m}$. For a multivariate time series, it denotes the problem of predicting the next value(s) in at least one of the d dimensions.

C. TRANSFER LEARNING

Transfer learning (TL) refers to a concept used within the field of machine learning to improve the generalization ability of models [24], [26]. As machine learning models often require large amounts of training data, TL can improve prediction results by also leveraging related data. In many use cases, it is costly to collect specific target data needed to build an individual model, e.g., for a concrete human being, machine, environment setting, or time period, while more general data is readily available. While it is a typical requirement in machine learning, that test data is drawn from the exact same distribution as the training data, TL relaxes this limitation by transferring knowledge from one domain to another similar domain. Besides this, TL can also refer to a knowledge transfer between different prediction tasks. This is closely related to multitask learning, with the difference that tasks are not of equal importance. Instead, learning is optimized towards a specific target task. We formally describe TL according to the following definition given by Pan and Yang [24]:

Definition 9 (Transfer Learning): Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.

Thus, there are two major subcategories of TL, which we call *domain adaptation* for $D_S \neq D_T$, and *task adaptation* for $T_S \neq T_T$. In rare cases, TL addresses a combination of both, $D_S \neq D_T$ and $T_S \neq T_T$. The most typical case is the transfer between differing domains. The term *domain adaptation* (DA) is commonly used in the literature to refer to this form of TL. Literature surveys specifically dedicated to DA can be found in [4], [34]. Domain differences can be in the marginal as well as conditional distribution of the data. In some cases, even feature spaces may differ from each other, which is addressed by heterogeneous TL [35]. We denote input features from the domains D_S, D_T as X_S, X_T, and labels, if available, as Y_S, Y_T. It is further possible that learning is based on multiple source domains, which is widely referred to as multi-source TL [27], [36] or multi-source DA [37].

Problem Settings: There are different problem settings in TL, depending on the availability of labels in target and source dataset. TL is typically applied when there is a limited amount of labeled data in the target domain, which is often not sufficient for independently training an accurate target model. In other cases, there is no label information in the target domain at all. The latter case is often called unsupervised TL. As the terms supervised and unsupervised TL are not consistently used in the literature, we adopt the naming conventions of Cook *et al.* [30]. They propose an additional term *informed* or *uninformed* to refer to a labeled or unlabeled target domain, while supervised or unsupervised refers to the source domain. The most common TL setting is the informed case, which is commonly known as *inductive TL* [24]. In this case, label information is available for the target domain, only the amount of available data is limited. The source domain data can be either labeled or unlabeled. Beside this, there are two different settings, *transductive TL* and *unsupervised TL* [24], where no target domain labels are available. In transductive TL, labeled data is only available for the source domain. For the target domain, artificial labels may be inferred, or the approach does not require labels, as it is, for example, based on the alignment of feature spaces. In unsupervised TL, labels are not available in either of the domains. This changes which data mining tasks can be addressed. While classification and regression are only possible with at least some label information, unsupervised TL builds on clustering or dimensionality reduction. Table 1 provides an overview of the introduced settings.

TABLE 1. TL settings based on [24], [30].

Information	Labeled Source Domain	Labeled Target Domain	Problem Setting	Data Mining Tasks
Informed	✓	✓	Inductive TL	Classification, Regression
	x	✓		
Uninformed	✓	x	Transductive TL	
	x	x	Unsupervised TL	Clustering, Dimensionality Reduction

Solution Approaches: Pan and Yang [24] categorize TL solution approaches into four categories: instance-based, feature-representation-based, parameter-based, and relational-knowledge-based. Instance-based transfer involves the selection or reweighting of samples from the source domain. This is based on the assumption that instances from the source domain are more or less similar to the set of target domain instances, hence, are more or less useful for model training. Feature representation transfer transforms the data into a common feature space so that learning can take place on features representing characteristics of both domains. Parameter transfer reuses parameters from a model pre-trained in the source domain for target model building. It is also known under the more general term model-based transfer. Relational-knowledge transfer is dedicated to relational domains and not applicable to time series. Tan *et al.* [25] provide an alternative classification of solution approaches specifically addressing approaches for deep TL: instance-based, mapping-based, network-based, and adversarial-based. In this classification, instance, mapping, and network transfer correspond to instance, feature, and parameter transfer in [24]. The term network transfer specifies a parameter transfer with parameters of a deep neural network. Adversarial-based refers to a new deep learning-specific approach, which utilizes a model architecture for domain-adversarial learning [38] based on the idea of generative adversarial networks [39]. Time series TL is not restricted to, but commonly involves deep learning (see Section IV). Hence, we use a classification scheme that combines categories of both, [24] and [25], see Figure 1. According to [27], the model-based perspective on TL can be further extended by approaches involving an ensemble of models, and by model control, which refers to changes to the model's objective function.

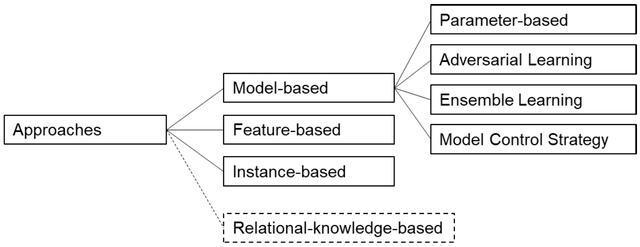

FIGURE 1. **TL solution approaches based on [24], [25], [27]. The dashed line denotes inapplicability to time series.**

III. REVIEW METHODOLOGY

This literature review is designed as an exhaustive summary with selective citation, according to Cooper [40, p. 111]. This means, that all identified relevant publications are used to draw general conclusions, while due to the extensive amount of studies, only a selected subset of the included publications is directly cited. The review is designed and conducted according to the guidelines by Kitchenham and Charters [33]. In the following, we describe the applied process for the literature search, selection and data synthesis. A replication package for this study is provided in [41].

A. SEARCH STRATEGY

1) SEARCHED LITERATURE SOURCES

In this study, we conducted a systematic search over multiple electronic databases. Indexing databases as well as publishers' databases of relevant academic publishers were considered. We used a combination of multiple source databases, as generally most publications cannot be found in all databases. Bramer *et al.* [42] studied 58 published literature reviews and found that 16% of the included publications were obtained only from a single database. We searched the electronic databases listed in Table 2. In addition to the two main bibliographic databases, Scopus and Web of Science, we included multiple databases of scientific publishers of which we found relevant publications in a prior initial literature search. These include, among others, the digital libraries of ACM and IEEE, which are often considered as primary sources in the field of computer science. Depending on the provided search options and the obtained search results, the above-listed databases were searched by either applying a full search over the publication metadata and full-texts or by a restricted search including some metadata. A full search was applied on ACM and Wiley. A full metadata search was applied on Web of Science and ArXiv, as these did not provide a full-text search. For the other databases, we restricted the search, due to the vast amount of search results. Therefore, we applied a metadata search including at least title, keywords, and abstract. As Springer Link does not allow this search setting, for this system, we applied a specific search, where at least one part of the AND-conjunction in our search query (see Section III-A2) has to be contained in the publication title, while the other is only required to appear anywhere in the document.

TABLE 2. **Searched electronic literature databases.**

Database	URL
ACM Digital Library	https://dl.acm.org
ArXiv	https://arxiv.org
IEEE Xplore	https://ieeexplore.ieee.org
MDPI	https://mdpi.com
ScienceDirect	https://sciencedirect.com
Scopus	https://scopus.com
Springer Link	https://link.springer.com
Web of Science	https://apps.webofknowledge.com
Wiley Online Library	https://onlinelibrary.wiley.com

2) SEARCH TERMS

Viewing titles and index terms of an initial set of 17 previously found relevant publications, we identified the following frequently appearing keywords: 'transfer learning', 'domain adaptation' and 'time series'. While all inspected publications contain either 'transfer learning' or 'domain adaptation' in their title or index terms, there is not always a direct indication that the application relates to time series data. Instead, publications may mention a concrete use case, such as occupancy estimation, temperature prediction, or wind power prediction. As it is impossible to consider all possible use cases in our search query, we did not include alternative search terms to indicate the focus on time series. Instead, we added a snowballing procedure (see Section III-C) to obtain further publications, that do not need to directly contain the keyword 'time series'.

For constructing a search query, we considered the identified keywords as well as alternative spellings. 'time-series', a common alternative for 'time series' does not need to be included, as search engines are not sensitive to the hyphen. For 'domain adaptation', we considered the two alternative spellings 'adaptation' and 'adaption'. We combined our keywords with the Boolean operators OR and AND, which are supported by most search engines. The basic search query (SQ) used for the electronic literature search was:

SQ) (''transfer learning'' OR ''domain adaptation''
OR ''domain adaption'') AND ''time series''

This query was modified according to the syntax requirements and available search options of each individual source database.

B. SELECTION CRITERIA

Following the electronic literature search, we conducted a selection process to assess the retrieved publications in regard to their relevance for this literature review. In this subsection, we list our criteria applied in the selection process, to communicate the scope of the review and transparently report what publications are regarded as relevant. As recommended in [33], we separated the criteria into inclusion and exclusion criteria. Inclusion criteria address formal characteristics of publications and the general topic, while exclusion criteria restrict the search results to specific content characteristics. Publications that meet any of the exclusion criteria are excluded from the set of relevant publications. Publications that do not meet all of the inclusion criteria are not included in the first place. We used the following inclusion and exclusion criteria.

Inclusion criteria:

i.1) The publication is written in the English language.

i.2) The full-text is accessible to the researchers.

i.3) The publication is a primary study that presents a proposed method or empirical results.

i.4) The publication describes a TL method for one or more machine learning models.

i.5) If multiple publications refer to the same study, the peer-reviewed publication is included. If there are multiple peer-reviewed publications on the same study, the most recent one is included. If there is no peer-reviewed publication, the most recent non-reviewed is included.

Exclusion criteria:

e.1) The applied TL method is not clearly stated.

e.2) The TL method is not one of the main contributions of the publication.

e.3) The work refers to TL in general but does not focus on univariate or multivariate time series data.

e.4) The work is entirely or partially based on non-time series data, including, for instance, static attributes, text data, or two-dimensional image data.

e.5) The work is entirely or partially based on spatio-temporal data, respectively sequences of any matrix-based data, or sequences of two-dimensional image data (image time series).

e.6) The work simply applies a trained model in a different target domain, without any further adaptation to the target domain.

e.7) The presented TL method is proposed primarily for a different purpose than the increase of prediction performance for a certain learning task (e.g., for improved data security).

e.8) The work describes a purely generative solution that is applied other than for translation between source and target domain data (e.g., for missing data imputation).

e.9) The work applies pre-trained models from the field of image recognition, that were not trained on other time series data.

e.10) The work does not address a one-time transfer between datasets, but a continuous adaptation to new instances (e.g., reinforcement learning).

e.11) The work addresses the case of irregular time series and therefore does not correspond to our definition of time series data (see Definition 1).

C. SELECTION PROCESS & DATA SYNTHESIS

The electronic literature search was carried out on January 8, 2021 and yielded a total of 1329 search results. All retrieved publications were imported into a reference management software and duplicates were removed. The resulting articles were then reviewed in a three-stage process and either included or excluded according to the reported selection criteria. First, the title of each publication was reviewed, in order to exclude publications that clearly do not meet our criteria, such as publications from a different research field or non-primary studies. Second, we reviewed the abstract of each remaining publication. If a publication could not clearly be excluded by the information given in the abstract, or if it appeared to be relevant, the full-text was viewed for a detailed decision. Each decision upon inclusion or exclusion and the reason for each exclusion were documented. As it is argued by Wohlin [44], a pure electronic search poses the difficulty that results strongly depend on the quality of the used search terms. It is therefore advisable to add a manual search to the process of [33]. In our case, this allows to obtain publications that do not explicitly contain the keyword 'time series'. Hence, we added a procedure called snowballing. Within the snowballing procedure, we reviewed all references in the previously included publications, as it can be assumed that these have cited further relevant publications. Note that studies identified by snowballing can be obtained from further literature sources, different from the ones listed in Table 2. In the snowballing procedure, we repeated the previously applied review stages. However, for reasons of practicality, the duplicate check was only applied for publications that were not already excluded by title. Figure 2 presents the complete selection process and the number of staged, included, and excluded publications in each step. From the 1329 publications initially staged for reviewing, 304 duplicates were removed. After a progressive selection by title, abstract, and full-text, 132 publications were identified as relevant. From

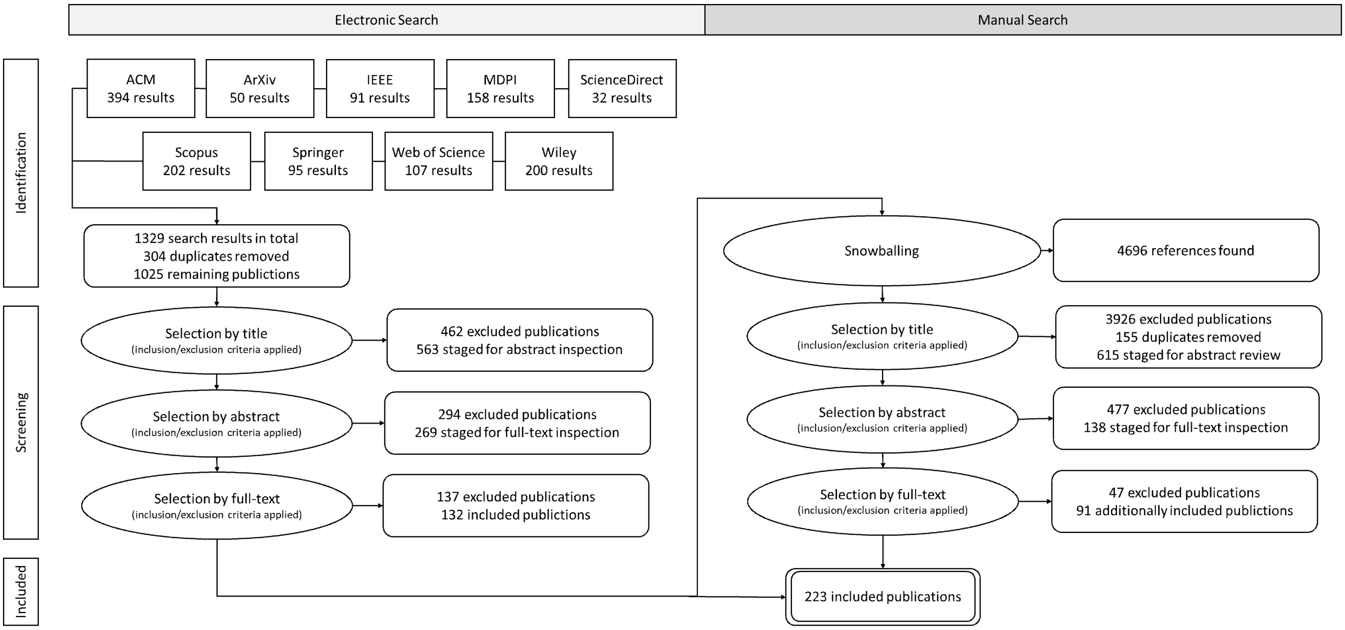
Electronic Search
Manual Search
Identification
Screening
Included
ACM 394 results
ArXiv 50 results
IEEE 91 results
MDPI 158 results
ScienceDirect 32 results
Scopus 202 results
Springer 95 results
Web of Science 107 results
Wiley 200 results
1329 search results in total
304 duplicates removed
1025 remaining publications
Selection by title (inclusion/exclusion criteria applied)
462 excluded publications
563 staged for abstract inspection
Selection by abstract (inclusion/exclusion criteria applied)
294 excluded publications
269 staged for full-text inspection
Selection by full-text (inclusion/exclusion criteria applied)
137 excluded publications
132 included publictions
Snowballing
4696 references found
Selection by title (inclusion/exclusion criteria applied)
3926 excluded publications
155 duplicates removed
615 staged for abstract review
Selection by abstract (inclusion/exclusion criteria applied)
477 excluded publications
138 staged for full-text inspection
Selection by full-text (inclusion/exclusion criteria applied)
47 excluded publications
91 additionally included publictions
223 included publications

FIGURE 2. Selection process shown as PRISMA flow diagram [43].

these, a total number of 4696 references was collected and reviewed in the snowballing phase, resulting in 91 additional relevant publications. This lead to a final set of 223 included publications.

All literature was processed by the first author of the paper. To avoid subjectivity, regular consensus meetings were held with other authors. Also, a validation on a sample of the processed literature was carried out to ensure quality. For this purpose, literature processed by the first author was double-checked by the second author. This validation step was carried out to find fuzzy criteria and to avoid personal bias in the filtering process. As a sample, 10% of the included papers, proportionally from main search and snowballing, were reviewed by full-text analysis. In addition, 10% of the excluded papers from the main search and snowballing were selected for validation. In total, 46 papers were reviewed for correct inclusion or exclusion.

For data synthesis, publication full-texts were analyzed and relevant data was noted in tabular form. The tabular data was used for further categorization. The content analysis approach pursued was mostly inductive. However, regarding transfer approaches, broad categories from previous literature reviews on TL ([24], [25], [27]) were operationalized.

IV. RESULTS

This section presents the main findings of this review. First, Section IV-A provides a basic metadata overview of the identified publications. Then, Section IV-B addresses research question Q.1 (application domains). Section IV-C addresses Q.2 (approaches) and Q.3 (models) from a quantitative point of view. The following subsections provide further explanations and concrete findings regarding the typically applied approaches and models.

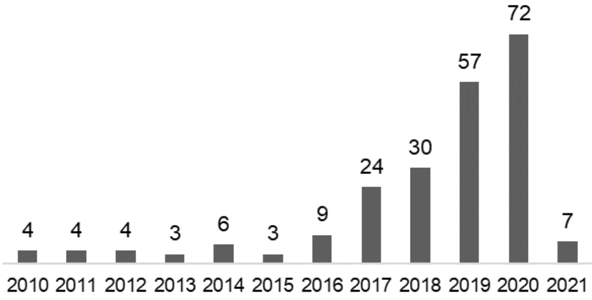

FIGURE 3. Number of included publications by year of publication.

A. ANALYSIS OF PUBLICATION METADATA

Based on the 223 included publications in this literature review, a recent increase in research interest in TL with time series data can be observed. Figure 3 shows the number of publications found per year of publication. It can be seen, that the majority of publications were published within the last four years, and the trend is rising. So far, 2020 was the year with the most relevant publications. Only few publications were identified for 2021, as the literature search was carried out on January 8, 2021. Figure 4 shows the number of publications for different scientific publishers. Also, 14 pre-print publications from arXiv.org were included. Two publications were jointly published by ACM and IEEE. They were counted for both. The publishers with the most relevant publications were IEEE, ACM, Springer, Elsevier, and MDPI. Scientific journals and conferences were almost evenly used for publication. We found 105 publications in journals and 104 publications in conference proceedings. The latter also include 8 workshop papers. The following

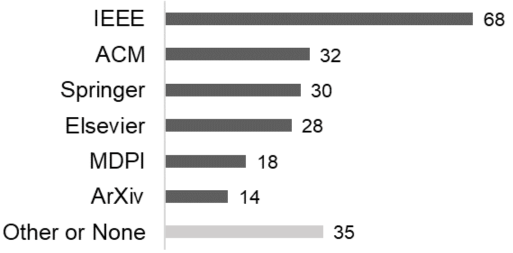

FIGURE 4. Number of included publications by publisher.

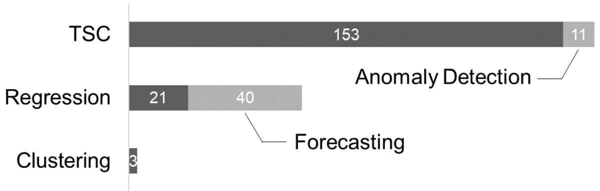

FIGURE 5. Number of included publications by time series problem.

three journals were most frequently used for publication: the MDPI journal *Sensors* (7 publications), the Elsevier journal of the international measurement confederation *Measurement* (7 publications), and *IEEE Access* (5 publications).

B. ANALYSIS OF APPLICATION DOMAINS (Q.1)

This section reports application domains and underlying types of time series data in the included literature. Furthermore, Figure 5 reports the addressed time series problems. The identified research is dominated by time series classification (TSC) problems. As shown in Figure 5, 164 of the found publications address TSC. Among these, only a few specifically address the problem of anomaly detection. The typical time series problem considered is the classification of regular time windows. Only in rare cases, the TSC problem includes a previous time series segmentation phase, as in [45]. Time series regression is addressed in 61 publications. Among these, the fraction of publications on forecasting problems is around two-thirds. Time series clustering is hardly addressed; only in three publications.

Application Domains: As shown in Table 3, there are two main application domains, namely fault diagnosis, respectively fault detection, and human activity recognition. These are followed by research on brain-computer interfaces and electric load forecasting. Applications of *fault diagnosis* and *fault detection* are found especially in context of rotating machinery to analyze bearing faults or gearbox faults. Fault detection is addressed in [46] as an anomaly detection problem to distinguish between normal and abnormal states. The other 34 publications address fault diagnosis, which attempts to classify between fault types or health statuses. Since in the industry data can only be collected under healthy conditions, TL is important in this domain to leverage the very limited amount of fault state data. Fault diagnosis is related to the domain of *remaining useful life* (RUL) estimation [47], [48], which has the goal of predicting the remaining time until machine failure.

In *human activity recognition* (HAR), the second main application domain, the goal is to classify activities of a human being, such as activities of daily living. This is done mostly by multimodal sensor data from an inertial measurement unit carried by the subject, including, for example, accelerometer, gyroscope, or magnetometer. Some literature is based on dedicated body-worn sensors [20], [49], while other publications use sensors of smartphones or wearable devices [21], [50]. A less commonly investigated subdomain is device-free HAR [51], [52], where no device is carried by the subject. It can be based on smart home equipment or Wi-Fi signal interference. TL is especially important in the field of HAR, as personalized models can outperform subject-independent models, but it is impracticable to collect enough training data for each specific subject. Hence, there is a need for subject-adaptation with little or no labeled training data given for the target subject. Moreover, sensor placement, including small changes between different sessions, can have a critical negative impact on classification accuracy. A recent literature review gives an overview of TL for HAR [31]. Several domains are closely related to HAR, including more specific ones, such as fall detection or human identification, as well as the more general human occupancy estimation or detection.

Another frequently addressed domain involves research on *brain-computer interfaces* (BCIs) [53]–[55], which addresses electrical activities of the human brain, typically recorded via electroencephalography (EEG). This includes several subdomains, such as BCI-based emotion recognition [36], [56], [57], attention detection [58], imagined speech decoding [59], [60], or recognition of imagined hand gestures [61]. As in HAR, models for BCIs do not generalize well across different subjects or sessions. Typically, a calibration phase is required for new subjects or new sessions. To avoid this, inter-subject transfer and inter-session transfer are widely addressed in the literature. A review on TL in BCIs can be found in [32]. Further application domains include, for example, electricity load forecasting [62], [63], human occupancy estimation in building rooms [15], [16], wind speed [19] or wind power prediction [18], or acoustic emotion recognition to estimate arousal and valence from music [64] or speech recordings [65]. Also financial applications are frequently addressed, which include crude oil price forecasting [23], stock price forecasting [22] or stock classification [66]. A particularly topical application is epidemic forecasting, including forecasts of confirmed COVID-19 cases [67] or COVID-19 deaths [68]. While most publications address a specific application domain, we also found 26 that refer to TL with time series in general, and 3 that refer to TL with sensor time series. Domain-independent publications typically use evaluation datasets from different domains. A few even report positive effects when transferring knowledge from several arbitrary time series regardless of their domains [3], [69],

TABLE 3. Number of publications by domain of application.

Application Domain	Count
Fault Diagnosis / Detection	35
Human Activity Recognition (HAR)	34
Brain-Computer Interface (BCI)	22
Electricity Load Forecasting	11
Human Gesture Recognition	9
Disease Detection / Prediction	9
Financial Forecasting, Stock Classification	7
Remaining Useful Life Estimation (RUL)	6
Acoustic Emotion Recognition	6
Fall Detection, Fall Risk Classificaton	5
Wind Speed / Wind Power Prediction	4
Phenotype/Mortality Prediction	4
Room Occupancy Estimation / Detection	3
Epidemic Forecasting	3
Human Identification	2
Stress Detection	2
Dissolved Oxygen (Water Quality) Prediction	2
Earth Quake Prediction / Arrival-Time Picking	2
Transport Mode Recognition	1
Pavement Performance Prediction	1
Air Quality Predition	1
Indoor Temperature Prediction	1
Flood Prediction	1
Parking Space Prediction	1
Production Forecasting	1
Container Throughput Forecasting	1
Resource Utilization Prediction	1
Driver Workload Prediction	1
Photovoltaic Power Forecasting	1
Radio Frequency Device Identification	1
Radar Emitter Signal Recognition	1
Channel Quality Prediction	1
KPI Streams Anomaly Detection	1
Dangerous Flight Action Detection	1
Drug Use Detection	1
(Non-)Line of Sight Classification	1
Body Area Network (BAN) Key Generation	1
Construction Vehicle State Prediction	1
Blast Furnance Modeling	1
Wind Turbines Clustering	1
Cloud Monitoring	1
Music Classification	1
Acoustic Scene Classification	1
Muscle Fatigue Detection	1
Lithium-ion Battery Capacity Estimation	1
Sleep Stage Prediction	1
Time Series in General	26
Sensor Time Series	3

TABLE 4. Number of publications by type of time series data.

Type of Time Series Data	Count
Inertial Sensor Data	72
Body-worn Sensors, Smartphone, Wearable Device	38
Vibration Signal from Attached Accelerometer	34
Physiological Signals	39
Electroencephalography (EEG)	21
(Surface) Electromyography (EMG/sEMG)	6
Multiple Physiological Signals	6
Electrocardiogram (ECG)	3
(Functional) Near-Infrared Spectroscopy (NIRS/fNIRS)	2
Skin Conductance	1
Climate Data, Weather Data	16
Electricity Data	15
Audio Signal	14
Financial Data	7
Channel State Information (CSI)	7
Count Values	5
NASA C-MAPSS Datasets of Aircraft Engine Simulations	4
Radio Waves	4
Binary State-Change Sensor Data	3
Vehicle Sensor Data	3
Light Sensor Data	3
Performance Indicator	2
Seismic Activity	2
Photo Reflective Sensor Data	2
Water Quality Data	2
Air Quality Data	1
Ultrasound	1
Server Metrics	1
Wifi Access Data	1
Medical Data	1
Percentage Values	1
Pen Trace Data	1
Blast Furnance Data	1
Coalbed Methane Production Quantity	1
Heating, Ventilation, and Air Conditioning System Settings	1

[70]. An often-used resource is the UCR time series classification archive [71], which provides 128 datasets from various domains.

Data Types: Table 4 gives an overview of the types of time series data used as input data in the set of identified publications. Here, publications are only counted if they address a certain domain, not if they address time series or sensor time series in general. The latter typically use multiple evaluation datasets from diverse domains. As we can see, data types correspond to the addressed application domains. The main domains, HAR and fault diagnosis, are largely based on measurements from inertial sensors, thus the largest part of the research field uses this specific type of sensor data. In case of HAR, this involves several sensor types, especially accelerometers, gyroscopes, or magnetometers, that are carried by the subjects. In fault diagnosis, typically vibration signals are used [72]–[74], from one or multiple accelerometers attached to the examined objects. Physiological signals, especially EEG, are widely used in publications on BCI, or, for example, in case of electromyographic (EMG) data, also for human gesture recognition. Measured electricity data is mainly used for electricity load forecasting. In this research branch, models may additionally consider climate data, such as temperature [63], [75] or solar irradiance [76]. At the same time, climate or weather data is used in further domains such as wind power prediction [18] or room occupancy estimation [16]. In addition to the actual time series, some publications include further information on the time dimension itself. Di *et al.* [75], for instance, include weekend/weekday information, Inoue and Pan [77] use the intraday minutes as an additional feature, Banda *et al.* [78] use several time features including year, month and week in a year, weekday, etc.

C. ANALYSIS OF APPLIED APPROACHES AND MODELS (Q.2 & Q.3)

This section reports the different approaches on TL found in this review and the underlying machine learning models that are applied. As shown in Figure 6, the majority of publications apply neural networks. If a neural network with more

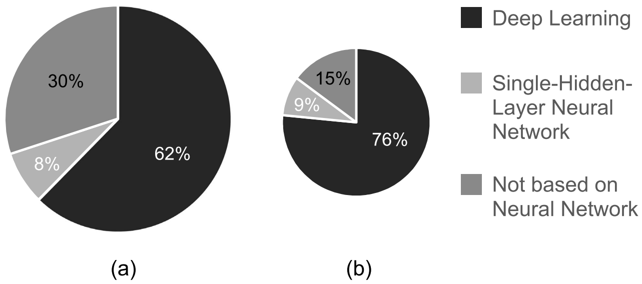

FIGURE 6. Fraction of publications applying deep learning methods, or a neural network, for (a) all included publications and (b) publications on model-based TL. If the method applies to a generic model and the evaluation involves at least one other than a neural network, the publication is counted as not based on neural network.

than one hidden layer is applied, we refer to this method as deep learning. A deep learning model may be used as the prediction model itself or, in some cases, as an auxiliary model for the purpose of TL, while the actual prediction model can still be a non-deep model. One of these possibilities is the case for 62% of the publications included in this review. 30% do not, or not necessarily, apply a neural network. For publications towards model-based transfer, there is an even higher focus on deep learning, with 76%. Only 15% of these do not focus on neural networks. The strong focus on deep learning within the field of time series TL agrees with the findings of Ebrahim *et al.* [7], who identified deep learning as the primary topic in time series classification.

Transfer Approaches: Table 5 reports different approaches on time series TL found in the included literature.

It also states the number of publications in which an approach is pursued, to give an impression of its popularity. The approaches are described in closer detail in the following subsections. As in [24], [27], we differ between model-, feature-, and instance-based transfer approaches. We classify approaches as model-based if the transfer is carried out with the help of model parameters, model objectives, or when an overarching model dedicated to TL is applied for prediction (i.e., ensemble model). The majority of approaches we found for time series TL are model-based. Most of these retrain a model that was previously pre-trained in a source domain. Also, freezing certain layers during retraining is a common practice. Less common are approaches that allow joint training with source and target data in a single training phase. There is also a range of publications on feature-based transfer. These include hand-crafted feature transformation methods, but also neural network-based feature learning. In a relatively high amount of publications, autoencoders are applied to learn a feature representation used for transfer. Only few publications address transfer in the sense of selecting useful source time series instances. Several publications, however, select useful source datasets among multiple alternatives, which can be combined with other approaches. Some publications propose a hybrid method that combines two or three of the identified model-, feature- or instance-based approaches.

TABLE 5. Number of publications by TL approach. Publications are counted multiple times if results are reported for multiple approaches. Approaches are ignored if not included in the evaluation or only considered as a baseline.

Approach	Count
► *Model-based*	
Retraining	82
M.1) Pre-Training & Fine-Tuning	45
M.2) Partial Freezing	30
M.3) Architecture Modification	7
Joint Training	55
M.4) Domain-Adversarial Learning	22
M.5) Dedicated Model Objective	17
M.6) Ensemble-based Transfer	16
► *Feature-based*	
Non-Neural Network-based	33
F.1) Feature Transformation	33
Neural Network Feature Learning	38
F.2) Autoencoder-based Feature Learning	30
F.3) Non-Reconstruction-based Feature Learning	8
► *Instance-based*	
I.1) Instance Selection	8
► *Hybrid*	
M.1+M.2) Temporary Freezing before Full Fine-Tuning	3
M.6+F.1) Ensemble Learning, Feature Transformation	3
M.1+M.6) Ensemble of Fine-Tuned Models	2
M.1+M.6+F.2) Ensemble of Fine-Tuned Autoencoders	1
F.2+M.3) Autoencoder, Adversarial Learning	1
F.1+F.3) Transformation of Encoded Data	1
I.1+F.1) Instance Selection, Feature Transformation	1
I.1+M.1) Instance Selection, Pre-Training & Fine-Tuning	1
► *Source Selection*	17

TABLE 6. Number of publications on model-based approaches by model type.

Model	Count
Convolutional Neural Network (CNN)	51
Recurrent Neural Network (RNN)	27
Long-Short-Term Memory (LSTM)	24
CNN-RNN	16
CNN-LSTM	14
Multilayer Perceptron (MLP)	13
Extreme Learning Machine	2
Support Vector Machine	2
Bayesian Neural Network	1
Wavelet Neural Network	1
Latent Space Model	1
Linear Discriminant Analysis	1
Hidden Markov Model	1
ElasticNet Regression	1
Ensemble Extra Trees	1
Any Base Model	12

Model Types: While feature- and instance-based transfer are widely model-independent, model-based transfer typically accompanies a specific prediction model. Table 6 reports different types of models applied in the included literature on model-based transfer as well as their publication count. Just as in the field of computer vision, the most common model type is the *convolutional neural network* (CNN). In context of time series data, either 2D or 1D convolutions can be applied. The 2D convolution can be used with multivariate time series to learn interdimensional relationships of the data. It is also common to first transform time series data into an image representation (see Section IV-I) before applying a 2-dimensional CNN. The 1D convolution applies the convolution process to 1-dimensional sequential data and is therefore directly suited for time series. 1D convolutions are used for univariate [79]–[81] as well as multivariate [59], [63], [82] time series. In case of multivariate time series, relationships between the dimensions are neglected. An alternative to the CNN is the *recurrent neural network* (RNN). RNNs allow to capture the relation between consecutive input samples and are therefore well suited for time series data. Within the included literature, the *long short-term memory* (LSTM) [83] is clearly more often used than any other type of RNN. An LSTM uses a memory cell that avoids the vanishing gradient problem in RNN training [83]. Several publications apply a combination of both, CNN and RNN, mainly CNN and LSTM [84]–[86]. In a CNN-LSTM model, typically early convolutional layers are used as feature extractor, while later LSTM layers are used to detect temporal relations within obtained feature sequences. Furthermore, several publications apply classic multi-layer perceptron networks, also known as feedforward neural networks. For 12 publications, the proposed method is intended to work with a generic base model, which is especially the case in ensemble-based transfer. As it can be seen from Table 6, model-generic methods account for a large part of non-neural network-based transfer, which amounts to 15% in Figure 6-b. We found only five publications on model-based transfer that include a specific machine learning model other than a neural network. The following subsections go into detail on the previously listed transfer approaches.

D. MODEL-BASED TRANSFER

Most commonly, model-based approaches are used for time series TL (cf. Table 5). The most typical form of model-based TL is a parameter transfer, in which model parameters of a model pre-trained in the source domain are reused for initialization of the target model. In case of a neural network model, this includes trained weights and biases. There are two main approaches based on parameter transfer, which we call *pre-training & fine-tuning* and *partial freezing*. This subsection describes these two and further alternative approaches including architecture modification, adversarial learning, ensemble-based transfer, and the use of an objective function specifically dedicated to knowledge transfer.

M.1) PRE-TRAINING & FINE-TUNING

Source domain *pre-training* and *fine-tuning* the trained model parameters in the target domain is the most frequently used TL approach in the context of time series data (cf. Table 5). In this approach, model parameters of a model pre-trained on source data are fully or partially used to initialize a target model, in order to enhance model convergence during target training and improve prediction accuracy and robustness. In many cases, all model parameters are reused for target training. For example, in [58], an EEG-based CNN model for BCI systems is pre-trained with data from several subjects and directly retrained for a certain target subject. A second common method is transferring all weights to the target model except for the output layer, which is randomly initialized. This is used in [87] in the context of bearing fault diagnosis. When applying a CNN, another method can be transferring weights of convolutional layers and training the subsequent fully connected layers from scratch, see [88].

Adjusted Training Procedure: Apart from the basic approach, where the model is simply retrained with the new target data, fine-tuning often refers to controlled retraining, where the training procedure is altered. This can involve a modification of hyperparameters or the objective function. In neural network training, fine-tuning may be conducted with fewer training epochs or a decreased learning rate. Changed training conditions can be beneficial if only scarce target data is available. It may prevent forgetting of previously learned knowledge, which is a common problem of model retraining, known as catastrophic forgetting. For instance, Wen and Keyes [89] reduce learning rates during fine-tuning of a deep CNN. Moreover, they set specific learning rate multipliers for different layers within the network. The selected multipliers are small values between 0.01 and 1, that become larger towards the model output. This has a similar impact as freezing early layers, which is covered in M.2.

Task Adaptation: TL by fine-tuning is especially applied for adaptation between different feature distributions. In case the target task differs from the source task, i.e., there is a difference between the label spaces, the model architecture needs to be adapted. In [2], an extensive study on fine-tuning-based transfer for TSC tasks, this is done by dynamically setting the number of neurons in the output layer to C, matching the number of C classes in the target dataset. Another example where the output layer is adapted according to a new label space can be found in [74]. In this example, a model is fine-tuned on a fault diagnosis task that contains two more fault classes than the source dataset. We do not regard necessary modifications of the input or output space of a model as a different TL approach. In contrast, we subsume methods that apply more than the required modifications to the source model architecture under the term *architecture modification*, which is described in M.3.

Specific Algorithms: Zhang and Ardakanian [15] introduce an additional re-weighting phase between pre-training and fine-tuning. Appropriate transform matrices are calculated and multiplied with model weights and biases obtained from source pre-training. The adjusted model parameters are then used to initialize the target model. The proposed method addresses the problem of occupancy estimation based on indoor carbon dioxide rates and damper positions. Although the calculation considers differences between building rooms and is not generalizable to other application domains, the idea may be reused with other handcrafted weight transformations.

Apart from neural networks, pre-training & fine-tuning is also applicable to other machine learning models. In [52], a hidden Markov model (HMM) is used, which is often applied for HAR. The source model is learned via maximum likelihood, while the expectation-maximization algorithm (EM) is used to learn the target model based on the estimated source model parameters.

M.2) PARTIAL FREEZING

A prominent special case of fine-tuning, which is also regularly found in the time series literature, is *partial freezing* (or partial fine-tuning). This is a method specifically for neural network-based transfer. Instead of retraining the whole model during a fine-tuning procedure, only selected parts of the model are retrained. A subset of the model's neurons are kept frozen, i.e., their parameters are not changed during fine-tuning. In most cases, this is realized as *layer freezing*, which means that a subset of $n < m$ layers of a network with m layers are frozen. Parameters of frozen layers are taken from the source model. The other, fine-tuned layers are either initialized with source parameters or trained from scratch. In many publications, only the output layer is retrained, while the rest of the network is used as a fixed feature extractor based on the source data [61], [63], [90]. When using a CNN model, a common approach is to freeze the convolutional layers and only retrain fully connected layers at the top of the network [91]–[93]. The idea is to keep the original features, which may be the same for source and target domain, while still adapting higher layers that are more specialized towards the concrete source task. As errors are not backpropagated through the whole network and only a subset of the model parameters are updated, layer freezing is computationally more efficient than fine-tuning the whole network. Hence, freezing may save training time in the target domain. Moreover, freezing layers lowers the risk of catastrophic forgetting, as it limits the impact of training with scarce target data. However, full fine-tuning may result in better predictions if weights in frozen layers are not suitable for the target prediction task due to high differences between source and target. As the performance of the two approaches strongly depends on the data, several publications conduct a comparison of full fine-tuning and layer freezing [63], [69], [82], [86]. Several publications further test different numbers of frozen layers [94] and different combinations of frozen and trainable layers [86]. An alternative to the freezing of complete layers is *sparse learning*. With this technique, certain nodes within layers can be frozen, while others are retrained. Ullah and Kim [95] apply sparse learning for TL in driver behavior identification. They prevent the forgetting of important knowledge by freezing strong nodes and retrain weaker ones in the target domain.

Hybrid Approaches (Freezing & Full Fine-Tuning): He *et al.* [13] propose a method for two source domains A, B: First, a source model is trained with data from source A. The first layer is frozen and the model is retrained with data from source B. Subsequently, the new source model is used for full fine-tuning on target data. Similarly, Wen and Keyes [89] apply a single-source method in which they first freeze early layers, before unfreezing all layers and performing a full fine-tuning in a second step. Strodthoff *et al.* [96] apply a more sophisticated method called *gradual unfreezing* for ECG analysis. Gradual unfreezing has been proposed before by Howard and Ruder [97] in the context of text classification. It allows to fine-tune the entire network, while still providing benefits of layer freezing. Layers are successively unfrozen during the training procedure: At first, only the output layer is set as trainable and the rest of the network is kept frozen. After each training epoch, the last frozen layer is set as trainable and backpropagation is carried out with one more trainable layer. This is repeated until no more layers are frozen. In the last step, the entire network is fine-tuned until convergence.

M.3) ARCHITECTURE MODIFICATION

In some publications, the architecture of the model used during source pre-training is modified for a subsequent fine-tuning phase in the target domain [98]–[100]. We refer to this approach as *architecture modification*, and we delimit it from conventional pre-training & fine-tuning by only considering modifications that go beyond an adaptation of the input or output layer to bridge differences between data spaces. Modifications may, for instance, include the removal or addition of certain layers in a deep learning model architecture.

Top Layers: An intuitive approach involves adding adaptation layers on top of the network, that are only trained on target data. Mun *et al.* [98], for instance, propose an architecture adaptation method used for acoustic scene classification, in which they remove the output layer from the source model and add two additional hidden layers and a new output layer for target adaptation. A more flexible network is proposed in [99]. The authors use a minimum mean squared error (MMSE) criterion to decide on how many layers to transfer and add a new output layer on top of the transferred layers. For p transferred layers from a source model with n layers ($p \leq n$), the redesigned target network then includes $p + 1$ layers. Martinez and De Leon [101] use a model built for multi-class pedestrian activity recognition (ParNet) and modify it for usage towards human fall risk classification. The new network (FallsNet) is obtained by adding a pooling layer and fully connected layer on top of ParNet. The original output layer is removed, and a new output layer for binary classification between high or low fall risk is added.

Inside Layers: Also, additional layers added between certain layers of the source model can facilitate adaptation. Matsui *et al.* [100] use a CNN for HAR to train a subject-independent source model. For adaptation to a specific subject, they add an extra hidden layer after each fully connected layer of the original network architecture and train these on limited target data.

M.4) DOMAIN-ADVERSARIAL LEARNING

Domain-adversarial learning is a recent deep learning-specific approach for TL, introduced by Ganin *et al.* in 2016 [38]. It is inspired by the generative adversarial network (GAN) [39], and borrows the idea of having two adversarial components in a deep neural network that perform a zero-sum

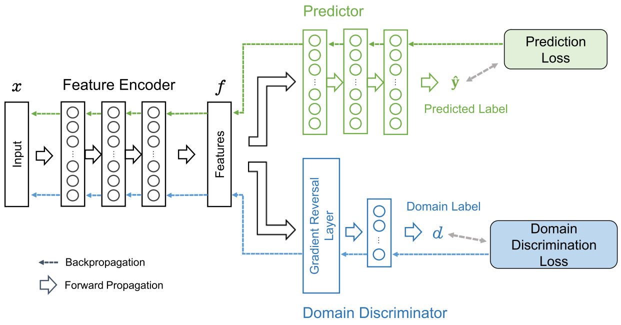

FIGURE 7. DANN architecture according to [38].

game to optimize each other. As illustrated in Figure 7, a deep adversarial neural network (DANN) consists of three components: a feature encoder, a predictor, and a domain discriminator. The feature encoder consists of multiple layers that transform the data into a new feature representation, while the predictor performs the prediction task based on the obtained features. The domain discriminator is a binary classifier that uses the same features to predict the domain from which an input sample is drawn.

Unlike the previously described TL approaches, adversarial-based transfer is not divided into two subsequent phases, for pre-training and adaptation. Models are rather jointly trained on source *and* target data. The predictor is trained via standard supervised backpropagation using the available label information from either of the two domains. In parallel to this, the adversarial objective is to generate domain-invariant features f such that based on f no distinction between target and source domain can be made. This is achieved by calculating an additional domain discrimination loss, and connecting the domain discriminator via a gradient reversal layer (GRL) that negates the gradient during backpropagation. A notable advantage of the adversarial approach over pre-training & fine-tuning is its applicability even if only unlabeled target data is available.

Since its inception, adversarial learning has extensively been studied concerning time series data (cf. Table 5). Also, the approach has been extended several times. Instead of focusing only on domain invariance, Zhao *et al.* [85] further extend the idea by additionally conditioning the discriminator on the label distribution. The goal is to remove conditional dependence on source domains. They propose an adversarial CNN-LSTM model for sleep stage prediction designed to ignore irrelevant subject- or measurement-specific information. Guo *et al.* [81] propose a 16 layer deep adversarial CNN for machine fault diagnosis. For domain adaptation, they add a feature distribution discrepancy loss term measuring the maximum mean discrepancy (MMD) between target and source features. While the domain discrimination loss is maximized, the feature distribution discrepancy is minimized. Adding an MMD term to the objective function is a common attempt also used in other TL approaches (see M.5 and Section IV-E). Li *et al.* [102] extend the idea of the DANN by a bipartite input layer to adapt it to a specific aspect in neural emotion recognition: EEG data from the left and the right hemisphere of a human brain are separately fed into the network via two distinct LSTM layers. This allows considering the two hemispheres' asymmetry to emotional responses. Jiang *et al.* [51] propose an adversarial CNN for HAR. They incorporate an entropy minimization term into the network's predictor module to utilize information from unlabeled data. In addition to this, they propose three additional constraints to prevent overfitting: a confidence control constraint, a smoothing constraint, and a balancing constraint. Purushotham *et al.* [103] propose variational recurrent adversarial deep domain adaptation (VRADA) by adversarially training a variational RNN. The method addresses general time series problems with domain-invariant temporal dependencies in a transductive TL setting without target labels. Wilson *et al.* [37] propose another convolutional adversarial model for time series data in general, named CoDATS, and show that it outperforms VRADA on four out of four evaluation datasets while requiring only a fraction of the training time. CoDATS also allows a multi-source transfer by using a domain discriminator that classifies between n sources.

Apart from purely *discriminative* adversarial networks such as the original DANN, a rather infrequent approach is the application of a GAN to translate between domains, such as in [104]. Here, a GAN is used to generate target from source time series by training a generative model component against a domain discriminator.

M.5) DEDICATED MODEL OBJECTIVE

As in domain-adversarial learning, and in contrast to model retraining, model objective functions specifically dedicated to TL allow using source *and* target data within a single training phase. Hernandez *et al.* [105] investigate modified objective functions for a support vector machine (SVM) in the context of stress recognition from skin conductance. While source data is collected from multiple subjects, they leverage knowledge from unlabeled target subject data for model personalization by (1) inserting suitable class weights for misclassification types, and by (2) integrating an importance weighting based on the similarity between the target subject and source subjects. Other methods typically define a neural network loss function that incorporates a feature distribution measure [79], [84], [106], [107].

Feature Distribution Discrepancy: One way of realizing TL is to force a model to produce similar feature representations for source and target input data. This can be achieved by measuring the distribution discrepancy of features generated when either source or target data samples are fed into the model, and using this measure as a second model objective. As in [106]–[108], an overall loss function $\mathcal{L}$ can be defined by a simple summation of prediction loss $\mathcal{L}_p$ and distribution discrepancy loss $\mathcal{L}_d$, where the influence of $\mathcal{L}_d$ may be

TABLE 7. Variants of MMD loss calculation in multi-layered neural networks.

Variant	Publications
Single-Layer MMD	[14], [84], [112]–[114], ...
Multi-Layer MMD	[79], [106]–[108], ...

weighted by a trade-off parameter $\alpha \in \mathbb{R}^+$:

$$\mathcal{L} = \mathcal{L}_p + \alpha \mathcal{L}_d \tag{1}$$

Besides the application in dedicated objective functions, which we interpret as model-based approaches, distribution discrepancy measures are mainly used in feature-based approaches, addressed in Section IV-E.

MMD: The most commonly used measure of distribution discrepancy in the context of time series TL is the *maximum mean discrepancy* (MMD) [109]. Other than the Kullback-Leibler (KL) divergence, the MMD is a non-parametric discrepancy measure that avoids the calculation of intermediate density. It maps two distributions onto a reproducing kernel Hilbert space $\mathcal{H}$ and calculates the distance in $\mathcal{H}$. For a non-linear mapping function $\phi(\cdot) : X \rightarrow \mathcal{H}$ between the original space X and $\mathcal{H}$, the MMD between two datasets X_S and X_T can be calculated as in [73]:

$$MMD(X_S, X_T) = \left|\left| \frac{1}{|X_S|} \sum_{x_S \in X_S} \phi(x_S) - \frac{1}{|X_T|} \sum_{x_T \in X_T} \phi(x_T) \right|\right|_{\mathcal{H}} \tag{2}$$

Most publications, however, use a squared MMD formulation instead [79], [106], [110].

Adaptation Layers: Within a deep neural network, each network layer represents features on a different level of abstraction. The MMD, or other discrepancy measures, can be calculated on every layer. For a discrepancy loss $\mathcal{L}_d$, such as in (1), it must be specified which layers are included in the loss calculation. Layers whose feature distribution discrepancy between domains is chosen to take influence on the model objective are called *adaptation layers*. The idea of measuring a so-called domain loss in an adaptation layer and to then add this to the model's original loss function was first published under the name *deep domain confusion* in [111]. In this sense, in several publications, one model layer is chosen as adaptation layer. There may be no need to adapt early layers, as they only extract general features. Wang *et al.* [84], for instance, calculate the MMD for features from the last fully connected layer in a deep CNN. Other publications use an earlier fully connected layer [112] or a convolutional layer [14] instead.

Multi-Layer Discrepancy: While several methods in the literature select one single layer as adaptation layer, others consider discrepancies in multiple layers. Table 7 lists publications that use either a single- or multi-layer approach on calculating an MMD-based discrepancy loss.

In case of multi-layer approaches, discrepancies measured at different layers are combined into a single loss function. Zhu *et al.* [107] calculate the sum of MMD discrepancies in the last two fully connected hidden layers of their model. They also use a trade-off parameter to balance each layer's impact. Li *et al.* [106] propose a method that defines a set of adaptation layers L and calculate the sum of discrepancies over all layers in L. This is equal to (3) with $\mu^l = 1$. As in [107], Xiao *et al.* [108] consider importance differences between layers. However, they include all six hidden layers of the applied model and combine discrepancies using a six-dimensional weight vector. Also, Yang *et al.* [79] calculate a weighted sum over four hidden layers of a CNN, including two convolutional and two fully connected layers. Generally, for a set of adaptation layers L, and feature distributions P^l, Q^l for $l \in L$ in source and target domains, as well as a weighting factor $\mu^l \in \mathbb{R}^+$, we find (3) to be a common example of how to combine multi-layer MMD discrepancies.

$$\mathcal{L}_d = \sum_{l \in L} \mu^l MMD(P^l, Q^l) \tag{3}$$

Beside the MMD, as the most typical measure, also other measures may be applied. Khan *et al.* [21], for instance, minimize the sum of layer-wise KL divergences.

Joint Distribution Adaptation: While the MMD only considers the marginal distribution of the data, *joint distribution adaptation* (JDA) [115] is a method that goes further than this, and jointly reduces differences in marginal *and* conditional distributions. For this, it integrates MMD-based marginal distribution discrepancy as well as a reformulation of the MMD to measure conditional distribution discrepancy. At the same time, JDA also preserves principal components, as in transfer component analysis (TCA), which is addressed in Section IV-E, F.1. Although the original JDA can be regarded as a feature-based TL approach, there are works integrating a JDA regularization term into the model objective of a deep neural network, which allows joint training with data from both domains. This is done, for example, for time series TL in fault diagnosis [10], [116].

M.6) ENSEMBLE-BASED TRANSFER

Another TL approach leverages the concept of ensemble learning. Ensemble learning involves the combination of multiple base learners, where each one is trained independently on a subset of the available data. In general, ensemble learning aims to reduce generalization errors. In TL, the aim is restricted to target generalization errors. An ensemble model's final prediction is typically obtained from voting between the individual base learners. These may be equally weighted (bagging) or assigned a weight depending on their individual prediction performance (boosting). Boosting is frequently applied for TL using the prediction performance in the target domain for weight calculation [117]–[119].

Boosting: One of the most prominent ensemble algorithms for TL is *TrAdaBoost* [120], which is based on the original *AdaBoost* [121] algorithm. As in label-based TL, TrAdaBoost assumes that some source data instances are more useful regarding target domain learning than others. It is a boosting algorithm that assigns weights according to target domain performance. The algorithm is frequently used in the context of time series TL. Marcelino *et al.* [118] apply a modified version of the algorithm for pavement performance prediction. They leverage source data from roads in the USA as well as data from the Portuguese road network, which is defined as the target domain. Xu and Meng [117] apply a TrAdaBoost-based regression algorithm for short-term electricity load forecasting. Khan and Roy [122] use TrAdaBoost as part of a hybrid method for HAR. They apply the algorithm to classify instances that are likely to belong to a known activity class. In addition, to consider previously unseen activities, k-means clustering is applied. Shen *et al.* [123] utilize an extended version of TrAdaBoost for fault diagnosis. Their method includes a transferability assessment, which involves the similarity between label distribution and feature similarities per label. This assessment is used to further reduce negative transfer. Just as TrAdaBoost, also similar methods are applied, that weight base models based on their target prediction error. Ye and Dai [119] propose an ensemble of extreme learning machines (ELMs). In addition to the weighting, they further replace source ELMs that exceed a defined error threshold by newly trained ones. In a publication on EEG classification with a specific target subject, Tu and Sun [124] go beyond the idea of boosting. Instead of assigning static weights to base models, they apply a dynamic weighting method that assigns specific weights to different test samples. They propose a two-level ensemble method that involves training multiple filter banks that are either robust (subject-independent) or adaptive (subject-specific). Robust and adaptive filter banks are combined into one robust and one adaptive ensemble model, and weights are dynamically assigned. On level two, the two ensembles are combined into a final ensemble.

Model Stacking: A different type of ensemble learning that can be applied for TL is called model stacking. In this variant, the outputs of multiple models are used as input to a combiner model that learns their optimal combination. An example can be found in [125]. In this publication, two neural networks trained on different datasets are combined by an additional combiner network. The ensemble is used for wind intensity prediction of tropical cyclones. While the target domain, a certain geographic region, is covered by one of the two combined networks, the other receives data from a different region. Wang *et al.* [126] propose a multimodal model for hand motion recognition, where the source domain contains EMG data, whereas the target domain contains EMG as well as inertial data. Parameters from an EMG-based model pre-trained in the source domain are transferred to the corresponding target model. A second target model is constructed for the inertial input data. Both models are connected via a new LSTM layer for feature fusion and trained in parallel on target data.

Hybrid Approaches: Ensemble learning may be combined with other approaches such as pre-training & fine-tuning. Benchaira *et al.* [127] train 12 different CNN-RNN networks, one for each of 12 source domain labels, and fine-tune each one in the target domain. The resulting transferred networks are then combined via XGBoost [128] stacking. Similar to this, Shen *et al.* [129] combine ensemble learning with CNN fine-tuning for capacity estimation of lithium-ion batteries. They train n CNN models on n distinct folds of the source dataset and retrain them on target data. The fine-tuned models are fused by adding an overarching fully connected layer and a regression output layer on top. An approach combining ensemble learning with pre-training & fine-tuning as well as with autoencoders (AEs) can be found in [130]. Without supervision, the authors train multiple stacked denoising autoencoders in the source domain. Each of these is then fine-tuned in the target domain. Finally, they apply a modified voting strategy to combine the transferred models.

E. FEATURE-BASED TRANSFER

Feature-based transfer is based on the reduction of discrepancy between the feature spaces in target and source domain. In contrast to model-based transfer, feature-based approaches are independent of the prediction model. They encode data from one domain into a feature representation that is more similar to the other domain or transform data from both into a common latent feature space. We coarsely divide approaches into ordinary feature transformation approaches and feature learning. In feature learning, a neural network encoder is learned with the goal of encoding data into a more useful feature space. Unlike in model-based approaches, here, the feature learning network is an auxiliary model specifically used for the purpose of transfer. It can be combined with any arbitrary prediction model. In the following, we describe methods that do not involve a neural network (F.1), methods that are based on an autoencoder (F.2), and methods based on a neural network other than an autoencoder (F.3).

F.1) FEATURE TRANSFORMATION

Apart from the application of neural networks for feature learning, feature transformation with the intention of time series TL may be based on signal processing techniques for feature extraction. For example, Natarajan *et al.* [131] use histograms of time series as transferred features in a study on lab-to-field transfer in cocaine detection. Other methods try to align feature distributions. This is often based on the MMD [16], [132]. A prominent MMD-based method for feature-based TL, that is used with time series, is transfer component analysis (TCA).

Transfer Component Analysis: TCA is a method to reduce domain differences in the marginal distribution. It seeks a shared feature space by minimizing the MMD between the domains. At the same time, as in principal component analysis (PCA), it tries to preserve variance in the data. TCA was originally proposed by Pan *et al.* [133] for TL in general. In context of time series, TCA is, for instance, applied in [134] for gearbox fault diagnosis. The authors compare TCA performance under the application of four different kernel functions. It is further applied in [57] for subject-to-subject transfer in the context of BCIs. In this work, the authors compare different subspace projection algorithms, namely TCA, kernel PCA (KPCA), and transductive parameter transfer (TPT).

Seasonal Decomposition: Several methods perform a seasonal decomposition of time series [16], [62]. Based on this, TL can, for instance, be conducted by eliminating typical time series patterns such as trends and seasonality, that may be dataset dependent. Ribeiro *et al.* [62] apply trend and seasonality removal for energy forecasting on related buildings. Arief-Ang *et al.* [16] use a seasonal decomposition model for occupancy estimation from carbon dioxide rates and propose an individual transfer method for each summand of the regression function. The trend term's distribution discrepancy is measured and aligned via MMD, while the seasonality term is adjusted according to pattern sequence repetitions.

Manifold Learning: Several publications apply manifold learning techniques [135]–[137]. In context of fault diagnosis transfer, Zhao *et al.* [136] apply manifold embedded distribution alignment (MEDA), a method originally proposed for TL with image data. Saeedi *et al.* [49] propose a manifold-based transfer method for cross-subject HAR. Their method applies manifold learning in the source domain and later conducts a manifold mapping from target to source data. Rodrigues *et al.* [137] propose a method called Riemannian Procrustes analysis (RPA) in the context of EEG data for BCIs. This method uses symmetric positive definite (SPD) matrices to represent the time series statistics. It estimates the geometric means and re-centers the data in both domains, then stretches the target dispersion and rotates the target SPD matrices to match source dispersion and rotation.

Hybrid Approaches: Feature transformation can be combined with other approaches such as ensemble learning. One example is stratified TL [20], in which, at first, multiple source models are trained and combined into an ensemble to generate candidate labels for unlabeled target data. These are then used to find an embedding into a shared feature space based on the intra-class MMD, which is calculated for each class in source data and target candidate data. Li *et al.* [36] combine ensemble learning with a previously known transformation method named style transfer mapping. The work is based on EEG data used for multisource TL in personalized emotion recognition.

F.2) AUTOENCODER-BASED FEATURE LEARNING

A popular approach for transforming time series data or obtained input features into a new feature space is by using an autoencoder (AE). An AE, also known as sequence-to-sequence model, is a neural network where the number of input nodes k equals the number of output nodes. The goal is to compress the input into a latent representation z. This is achieved by combining an encoder model with a decoder model trying to restore the original data. These two are connected by a bottleneck layer with less than k neurons, containing the learned encoding z. During model training, the network typically aims to minimize the reconstruction error between input and output data. AEs can be used to transform source domain and target domain features into the same subspace [19], [138], but also to transform source domain features into the target space, as in [139], or vice versa. Although a model is trained for transfer, this is different from model-based TL, as the model is only used to generate a feature representation of the original data, while any model can then use the new feature values to perform the actual prediction task. Different AE architectures are applied for time series transfer, including simple single-layer AEs [139], as well as stacked AEs forming a deep model architecture [19], [140]. Table 8 lists specific types of AEs and exemplary studies. Generally, we divide AE-based transfer approaches into two basic strategies: *sequential training* and *parallel training*.

TABLE 8. Autoencoder types used for time series TL.

Autoencoder Type	Publications
Sparse Autoencoder	[139], [141]–[143], ...
Denoising Autoencoder (DAE)	[19], [65], [130], [143], [144], ...
Variational Autoencoder (VAE)	[145], [146], ...
Shared Hidden Layer AE (SHLAE)	[19], [145], [147], [148], ...
Convolutional Autoencoder	[146], [149]–[151], ...

Sequential AE Training: In the sequential training strategy, target and source data are used in two distinct training phases. In some publications, two different AEs, one for the source domain and one for the target domain, are trained one after the other. Akbari and Jafari [149] first train a source AE with labeled source data. In a second step, they train a target AE by minimizing the KL divergence between the output of the (fixed) source AE and the now trained target AE. Faridee *et al.* [151] apply a similar approach using the Jensen-Shannon divergence. Deng *et al.* [65] train a denoising autoencoder (DAE) on target data and subsequently an adaptive DAE (A-DAE). The A-DAE minimizes its reconstruction error and at the same time forces model weights to stay close to the weights obtained from the previously trained DAE.

Other publications only use a single AE for either source or target and use the remaining data and the trained AE directly to learn the prediction model. Deng *et al.* [139] train a single-layer AE with target data and use the AE to reconstruct source data instances. The resulting source data representations are then used to train a classifier, which is intended to be used for target task classification.

Parallel AE Training: In the parallel training strategy, a shared AE is trained for source and target domain simultaneously. This can include either the full AE architecture or only parts of it. For instance, Chai *et al.* [140] feed source and target instances into the same AE, which they call subspace alignment AE. In contrast to this, Deng *et al.* [147] propose a shared hidden layer autoencoder (SHLAE), which uses the same hidden layer neurons for feature encoding, but different output layer neurons for reconstruction. They first applied the SHLAE in the field of acoustic emotion recognition. Since then, it has been further studied with other time series prediction problems, such as radar emitter recognition [148].

Objective Functions: Defining reconstruction error minimization as the single training objective, an AE can be trained in an unsupervised manner and does not require labeled data. If labeled data is available, it can, however, also be applied in combination with training an auxiliary prediction model in order to ensure that relevant features are obtained. In this case, a prediction loss function may be integrated into the training objective. This approach can be found in [65], [73], [149]–[151]. In the typical case of TSC, the loss is measured as classification loss, e.g., the cross-entropy loss, of an auxiliary classifier.

Several studies further penalize distribution discrepancy between features generated for target and source data. This is applied similarly if either features from source AE and target AE are to be compared [149] or if a single AE is used for both domains [73], [138]. Widely applied metrics are MMD or KL divergence. Sun *et al.* [143] use a stacked AE and apply MMD minimization for the output of each layer. To prevent network weights from becoming small and approaching zero to satisfy the distribution discrepancy term, Lu *et al.* [138] introduce an additional weight regularization term to strengthen representative features. A complete objective function $\mathcal{L}$ for AE training can be denoted as the weighted sum of multiple terms.

$$\mathcal{L} = \alpha_{ae}\mathcal{L}_{ae} + \alpha_p\mathcal{L}_p + \alpha_d\mathcal{L}_d + \alpha_w\mathcal{L}_w \tag{4}$$

Equation (4) gives an example including reconstruction loss $\mathcal{L}_{ae}$, prediction loss $\mathcal{L}_p$ of an auxiliary prediction model, distribution discrepancy $\mathcal{L}_d$ such as an MMD term, and a weight regularization term $\mathcal{L}_w$. For each term, a coefficient $\alpha_i \in \mathbb{R}^+$ determines the influence towards the other terms.

F.3) NON-RECONSTRUCTION-BASED FEATURE LEARNING

In addition to the autoencoder approach, there are also ways to learn an encoder model that are not based on the reconstruction of input sequences. An encoder may be trained solely, i.e., without a connected decoder, by using an unsupervised learning scheme as in [152]. Another approach can be model truncation.

Source Model Truncation: Given a pre-trained source model, a possibility to create an encoder is to truncate the model at a certain layer and keep all previous layers. The output of the last remaining layer is then considered as feature representation. Training of the required source model can take place in a standard supervised manner. In [153], [154], and [70] source models are trained to perform a classification in the source domain and the output classification layer is removed to obtain the encoder. Zhou *et al.* [154] conventionally train a CNN via supervised backpropagation using the source dataset. In a second step, the output layer is removed, and the rest of the CNN, including several convolutional layers and a fully connected layer, is used as a feature encoder in the target domain. The encoded features are then used to train an arbitrary classifier, such as a logistic regression classifier or random forest. While Zhou et al. use one source model, Meiseles and Rokach [153] train multiple source models and further perform a source model ranking, which allows selecting the one with the best encoding results. Serrà *et al.* [70] apply a multi-head strategy where a dedicated output layer is used for each of multiple source datasets. The output layers are then dropped to obtain a general encoder model. Similarly, Kashiparekh *et al.* [69] train on multiple source datasets as well, but use two dedicated layers per dataset, a fully connected layer and an output layer, which are both truncated.

F. INSTANCE-BASED TRANSFER

Instance-based transfer involves the selection or weighting of source instances according to their usefulness for target training. In our case, the term instance refers to an individual time series contained in a time series dataset. As the instance weighting-based methods found in this literature review are based on an ensemble model, we categorize these under model-based transfer (see Section IV-D, M.6).

Instance Selection: The remaining instance-based methods select a useful subset of the source data X_S, which can be used as auxiliary data X'_S to train an arbitrary target model. For a selection I, X'_S can be defined as:

$$X'_S = \{(x_i)\}_{i \in I} \subseteq X_S \tag{5}$$

Most selection methods consider the similarity between source instances and the time series contained in the target dataset. Yin *et al.* [56] use instance selection for personalized, EEG-based emotion recognition. Data from the target subject is divided into two clusters for high and low emotional states. Then, instances from source subjects are either selected or discarded according to their distance to the two cluster centers. Vercruyssen *et al.* [155] address TL for time series anomaly detection and propose a cluster-based as well as a density-based method to decide which instances to transfer. In the cluster-based method, k-means clustering is carried out on target data, and the resulting clusters are divided into small and large clusters. The decision upon the selection of source instances is then based on the label, the assignment to either a large or small cluster, and the distance to the cluster center. In the density-based method, they use a Gaussian kernel to estimate the density of multiple subsequences of each time series. Using a normalized sum of subsequence densities as a weighting, the selection is made based on a defined threshold. Shang and Wu [156] use a selection method based on feature discretization. After discretizing all feature values, they consider instances as sufficiently similar if they share the same label and the same discretized feature value for each dimension. Apart from comparing instance similarities, another way of selecting useful instances can be, as in [77], to test the prediction accuracy when using different subsets of the source data, and form a union of the top-performing subsets. This is, however, only possible if target labels are available.

Symbolic Aggregation Approximation: Instance selection can also be carried out by selecting representations of data instances. Two of the identified publications use the symbolic aggregation approximation (SAX) representation for time series data translating each time series into a word [157], [158]. Such word representations can be collected into a bag of words and in this way form a subset of the input data. In [157], the authors construct bags of words for different subjects in fall event detection. The transfer is conducted by collecting a bag of common words, where commonness is measured by the relative term frequency. Fañez *et al.* [158] conduct a clustering of words and only select the cluster centroids into the bag of transferred words.

G. SOURCE SELECTION

Similar to instance selection, which is addressed in the previous subsection, *source selection* denotes another procedure to select useful source data. In contrast to instance selection, this is not done on an instance level. Instead, out of a set of n existing source datasets from distinct domains $D_1, \ldots, D_n$ one or multiple datasets are selected as a whole to form the final source data X_S. For a selection I, X_S can be defined as:

$$X_S = \bigcup_{i \in I} X_i \subseteq \{X_1, \ldots, X_n\} \tag{6}$$

As source selection is typically an upfront procedure carried out before the actual TL, we do not consider it as a method of TL, but rather an additional step of data pre-processing, that may be combined with TL. Analogous to (6), in source *task* selection, a minor subfield of source selection, source tasks Y_S are selected as a subset of multiple tasks $Y_1, \ldots, Y_n$. An example can be found in [159].

Motivation: The idea of source selection is to only reuse knowledge from domains with reasonable similarity to the target. In an experiment with 85 time series datasets from different domains, Fawaz *et al.* [2] show that TL can also lower prediction performance. This negative effect is widely referred to as *negative transfer*. For each pair of datasets, they pre-train a CNN classifier on one dataset and fine-tune it on the other, showing that for most pairs of time series datasets TL has no significant impact on the accuracy. For some pairs, however, it shows either a significant increase or decrease in prediction performance compared to training the model from scratch in the target domain. They conclude that similarity between source and target dataset may play an important role, and also show that choosing the source dataset with the lowest dynamic time warping (DTW) distance to the target reduces the risk of negative transfer. As the choice of a similar source dataset seems to be an important prerequisite for TL, some publications address the problem of how to select appropriate sources from multiple alternatives. Approaches are typically based on prediction performance in the target domain or on the similarity between source and target data.

Target Testing: If labeled target data is available to some extend, appropriate sources may be selected via target domain testing after training with different sources or combinations of sources. Li *et al.* [36] use each source individually for training a prediction model. They test each model in the target domain and select the n top-performing models. The selected models are then combined in an ensemble model. Lotte and Guan [54] test performances for combinations of sources. They apply a search algorithm to search over different combinations and, in each iteration, test target performance when training with the currently selected subset.

Data Similarity: If only unlabeled data is available from the target domain, source selection may be based on intra-dataset similarities instead. While Fawaz *et al.* [2] apply the classic Euclidean distance-based DTW to measure source-target similarity, Ye and Dai [14] also use DTW to align time series lengths, but then calculate the Jensen-Shannon (JS) divergence. Xiao *et al.* [160] divide each time series into multiple segments and calculate the segment-wise Pearson correlation. Wang *et al.* [84] argue that a simple distance measure is not sufficient for their problem and propose a combination of a general and specific distance, where the specific distance is based on human annotation and kinetic aspects relating to HAR. Chen *et al.* [161] propose a stratified distance (SD). They calculate the MMD distance between source and target for each class label individually. Pseudo labels are used for the unlabeled target data. Based on this, the SD is defined as the average distance over all classes. In the end, the source dataset with the minimum SD is selected.

Feature Similarity: Meiseles and Rokach [153] propose a method that is applicable even if only pre-trained source models are available and no original source data. They truncate each source model after a certain layer and compare target encodings of the last retained layer in each model using the mean silhouette coefficient (MSC) based on the cosine distance. Source models are then ranked by their MSC score from low to high.

H. OFF-THE-SHELF ENCODERS

As described in Section IV-E, encoder models can be used to encode data into a feature representation that can then be used as input to a target model. Assuming there are general patterns that appear over diverse time series domains, encoders may be applied to carry general knowledge from various heterogeneous datasets and use this knowledge to support an arbitrary time series prediction task. Recently, there have been some attempts to provide such a general-purpose encoder [3], [69], [70]. Already pre-trained in various time series domains and ready to use, we can call these *off-the-shelf encoders*. The idea is, that the off-the-shelf encoder generalizes well and can be universally applied to extract more informative features. A frequently used source for various time series datasets is the UCR time series classification archive [71]. Prominent models pre-trained on datasets from the UCR repository are *TimeNet* [3] and *ConvTimeNet* [69]. TimeNet is a multi-layered RNN-based autoencoder pre-trained on 18 datasets. It showed improved performance compared to a domain-specific model trained from scratch on 19 out of 31 test datasets [3]. ConvTimeNet is a multi-layered CNN, pre-trained on 24 randomly chosen datasets. Compared to domain-specific training, it showed improved performance on 17, and similar performance on 18, out of 41 test datasets [69]. While TimeNet was trained in an unsupervised fashion, ConvTimeNet was trained using dataset labels. To do so, two task-specific layers were added on top of the ConvTimeNet model for each dataset: A fully connected layer and a softmax classification layer that matches the respective classification task. Another CNN-based encoder trained with a dedicated output layer for each dataset is presented by Serrà *et al.* [70]. It shows promising average results on multiple train-test splits with 85 datasets.

I. TIME SERIES TO IMAGE TRANSFORMATION

As many TL approaches originate from the field of computer vision and address two-dimensional image data, it is a straightforward strategy to transform time series data into a 2D representation and apply traditional approaches to the transformed data. For multivariate time series, a simple transformation method is forming a matrix by setting one time axis and arranging the feature values at the second axis. For a time series of length l with d dimensions, this leads to a matrix of size $d \times l$. After normalizing the features' value ranges, the matrix can be interpreted as an image. This method is applied in [60], [91], [95]. Several other works transform time series data into the time-frequency domain and use the spectrogram as a visual time series representation. This is also applicable to univariate time series data. Typically, the time-frequency transformations are based on a wavelet transform [113], [162] or a Fourier transform, such as fast Fourier transform (FFT) [88] or short-time Fourier transform (STFT) [163]. Hasan and Kim [87] propose the use of the discrete orthonormal Stockwell transform (DOST) to improve time-frequency resolution compared to STFT.

Numerous publications do not transform source and target time series but rather treat a time series dataset as the target and general image data as the source. This allows the application of well-known large CNN models pre-trained on ImageNet, such as VGG [11] or AlexNet [12], in the context of time series data. TL from image source data is not within the scope of this review (see Section III-B, criteria e.9). However, this can be a useful alternative if source time series are not available. The following exemplary publications use CNNs pre-trained on images to enhance model training for time series data: [164]–[166].

V. FUTURE RESEARCH OPPORTUNITIES

Time series TL has developed into its own branch in TL research and offers large potential for future research. In the following, we discuss research opportunities from a methodological, contextual and topical view.

Methodological: Publications on time series TL mostly propose a new method of TL or apply an existing method in a specific domain. Despite the high diversity in the field, only a few compare multiple approaches (e.g., [82], [94], [167]). There is a lack of empirical studies that carve out the advantages of approaches with certain types of time series data or for certain prediction problems, which is necessary to retrieve guidelines for approach selection or method design by practitioners. This review encourages future empirical work that provides insights into context- or topic-related differences and advantages of different approaches. In the same time, both research and practice can benefit from more extensive evaluations of different model architectures and parameters. Evaluation results on multiple different datasets can help find reference architectures.

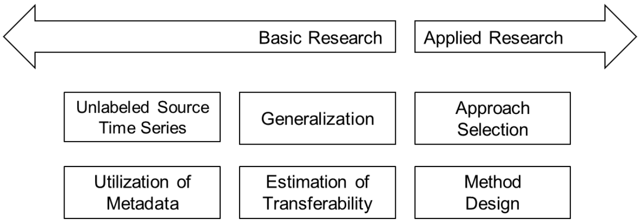

FIGURE 8. Research opportunities from a topical view.

Contextual: As shown in Section IV-B, research on time series TL is still concentrated on only a few application domains. Here, we expect more generalization in the future, as well as an increase in publications in the context of domains that have not yet been widely investigated, such as, for example, occupancy estimation in building rooms. With the application to a wider range of different domains, domain-specific opportunities and impediments will need to be considered. One major impediment for TL, in application domains with personal data, such as in HAR, can be data privacy. Though not within the scope of this review, privacy issues regarding source data is an aspect that needs further investigation. A straightforward idea is to not use source data directly, but rather pre-trained source models, that provide a higher level of data privacy. Ensemble-based TL even allows to integrate models from multiple domains, yet it requires high similarity between sources and target and may not be successful with distant source domains.

Topical: Currently, research on time series TL is still strongly influenced by approaches from other fields. This can be seen from the large amount of publications aiming to make models and methods for computer vision applicable to time series. Further effort is required to find more stand-alone solutions that specifically take into account the nature of time series data. Research opportunities can be found in applied research to bring the current findings into practice as well as towards new solutions. Figure 8 lists some important research directions.

Given the variety of transfer approaches presented in the previous section, selecting the right approach for a problem is not a trivial task. Guidelines need to be found on how to select a transfer approach.

Also, many of the current methods are tailored to a specific prediction problem and need to be customized, for example, in terms of the selection of adaptation layers or frozen layers. Again, guidelines are needed on how to design a concrete transfer method for a given problem, without relying on personal experience or exhaustive testing. On the other hand, more adaptive methods need to be found, which can be generally applied regardless of data and task.

More generalization is directly linked to the ability to transfer knowledge from less similar data. As the transfer from distantly related time series, however, carries the risk of negative transfer, a key challenge is the estimation of transferability. Useful source domains, or useful parts of a source domain, need to be identified for transfer. As shown in this review, only a small part of the literature deals with source selection or instance-based TL, which is essential to allow the use of a wide range of source data. Some publications address general-purpose transfer and have led to achievements such as the first off-the-shelf time series encoders. Research on large pre-trained models, which can be used as a basis whenever dealing with time series, is still young in comparison to the field of computer vision. Equivalents to models such as AlexNet [12] and VGG [11] yet need to be found in the area of time series data. InceptionTime [8] can be named as an early attempt in this direction.

Furthermore, research in the field of time series TL mostly assumes that labels are available in the source domain. Mostly, approaches such as pre-training & fine-tuning are used that require the availability of labels. These approaches may not be applicable in practice, as the cost for data collection and labeling can be too high. Many time series domains involve sensor data, which can be measured cheaply over a long time but often need to be labeled manually. Hence, a greater focus must be placed on transfer from unlabeled sources.

Another opportunity is the combination with non-time series data. In this literature review, we focus on approaches purely based on time series data. These are not applicable for use cases that involve complex objects consisting of time series and other types of data as well. Moreover, it may be beneficial to additionally use some metadata of the time series. Hu *et al.* [168], for example, use information retrieval approaches on web search results of activity names to calculate a similarity between different activities in HAR. These metadata similarities are then used for importance weighting in an instance-based TL approach.

In summary, it can be said that research on time series TL is still in an early stage. While at least some early literature exists for most of the research opportunities pointed out in this section, we could not find any publication that points out the advantages of different TL approaches in different application domains or concerning the nature of different time series. This is a clear research gap, which could help bring time series TL into practice.

VI. THREATS TO VALIDITY

The following threats to validity limit our findings:

Incomplete Selection of Literature: Our study does not include the whole body of literature on the topic of time series TL. Due to the vast amount of existing publications, we only included a sample of the literature. This sample contains 223 publications, which we assume to be sufficiently representative. However, since the research field is rapidly growing, at the time of publication of this study, some new developments may already have come up that we do not cover.

Literature Search Bias: In the electronic search process, we included nine literature sources that we believe to cover the main relevant literature. However, some relevant work may not be included in any of them. In addition to this, we may have missed some relevant literature that does not contain our search terms. Especially, the keyword 'time series' can not be found in all relevant publications, as they may name the concrete type of data instead. The restriction to the applied search query and the selection of literature sources may cause bias in the retrieved search results. We reduced this threat by including an additional snowballing procedure, in which we performed a manual backward search over the references of included publications.

Literature Selection Bias: The inclusion and exclusion criteria applied in this study were elaborated to support the research goal and provide a general overview over the current state of the literature. Still, they pose a limitation to completeness. Literature addressing very specific aspects in relation to the research field, such as privacy issues for example, was not considered. Also, related research fields involving TL with other sequential data, such as image time series or irregular time series, or continuous adaptation to changing data instead of a one-time transfer were not considered. Very active research fields that are not covered in this study are remote sensing or natural language processing.

Researcher Subjectivity: The entire work including the selection process and data synthesis has been conducted by the first author of this paper. To avoid subjectivity, consensus meetings with the other authors were held and a validation was carried out by the second author. Even though we included this additional validation, subjectivity can not be ruled out completely.

VII. CONCLUSION

In this paper, we have conducted a systematic mapping study on the current state of the art in time series TL. In total, we included 223 publications addressing either univariate or multivariate time series. We presented solutions found in the literature and discovered trends regarding three main research questions: (Q.1) what are the main application domains?, (Q.2) what transfer approaches are applied?, and (Q.3) what associated machine learning models are used? We showed that the literature is dominated by deep learning and by few application domains. Highly addressed domains are fault diagnosis, HAR, and BCI. Many transfer methods are adapted to time series but originate from other research branches such as computer vision. The most dominant approach is pre-training a CNN and fine-tuning it in the target domain. For this, time series are, in many cases, first transformed into an image representation. Other prominent approaches are domain-adversarial learning or using autoencoders to transform data into new feature spaces. Some of the reviewed methods are specifically designed for time series data, for instance, by applying an LSTM model. The most frequently applied model, however, is the CNN, originating from computer vision research.

We see large potential for future work in this young research branch and have pointed out some important research directions. Especially, more research is needed towards transfer between distantly related time series and the assessment of transferability, which allows to select useful source time series. Also, applied research is required to bring the current advances in time series TL into practice.

REFERENCES

[1] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, ''Deep learning for time series classification: A review,'' *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Mar. 2019.

[2] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, ''Transfer learning for time series classification,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1367–1376.

[3] P. Malhotra, V. TV, L. Vig, P. Agarwal, and G. Shrof, ''TimeNet: Pretrained deep recurrent neural network for time series classification,'' in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. (ESANN)*, 2017, pp. 607–612.

[4] G. Csurka, ''A comprehensive survey on domain adaptation for visual applications,'' in *Domain Adaptation in Computer Vision Applications*, G. Csurka, Ed. Cham, Switzerland: Springer, 2017, pp. 1–35.

[5] Q. Yang and X. Wu, ''10 Challenging problems in data mining research,'' *Int. J. Inf. Technol. Decis. Making*, vol. 5, no. 4, pp. 597–604, 2006.

[6] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, ''The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances,'' *Data Mining Knowl. Discovery*, vol. 31, no. 3, pp. 606–660, May 2017.

[7] S. A. Ebrahim, J. Poshtan, S. M. Jamali, and N. A. Ebrahim, ''Quantitative and qualitative analysis of time-series classification using deep learning,'' *IEEE Access*, vol. 8, pp. 90202–90215, 2020.

[8] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. W. Eber, G. I. Webb, L. Idoumghar, P.-A. M´uller, and F. Petitjean, ''Inceptiontime: Finding alexnet for time series classification,'' *Data Mining Knowl. Discovery*, vol. 34, pp. 1936–1962, Sep. 2019.

[9] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, ''Deep learning for sensor-based activity recognition: A survey,'' *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[10] T. Han, C. Liu, W. Yang, and D. Jiang, ''Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application,'' *ISA Trans.*, vol. 97, pp. 269–281, Feb. 2020.

[11] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.

[13] Q.-Q. He, P.-I. Pang, and Y.-W. Si, ''Transfer learning for financial time series forecasting,'' in *Proc. Pacific Rim Int. Conf. Artif. Intell. (PRICAI)*, 2019, pp. 24–36.

[14] R. Ye and Q. Dai, ''Implementing transfer learning across different datasets for time series forecasting,'' *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107617.

[15] T. Zhang and O. Ardakanian, ''A domain adaptation technique for fine-grained occupancy estimation in commercial buildings,'' in *Proc. Int. Conf. Internet Things Design Implement.*, Apr. 2019, pp. 148–159.

[16] I. B. Arief-Ang, M. Hamilton, and F. D. Salim, ''A scalable room occupancy prediction with transferable time series decomposition of CO$_2$ sensor data,'' *ACM Trans. Sensor Netw.*, vol. 14, nos. 3–4, pp. 1–28, Dec. 2018.

[17] M. Weber, C. Doblander, and P. Mandl, ''Detecting building occupancy with synthetic environmental data,'' in *Proc. 7th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2020, pp. 324–325.

[18] A. S. Qureshi, A. Khan, A. Zameer, and A. Usman, ''Wind power prediction using deep neural network based meta regression and transfer learning,'' *Appl. Soft Comput.*, vol. 58, pp. 742–755, Sep. 2017.

[19] Q. Hu, R. Zhang, and Y. Zhou, ''Transfer learning for short-term wind speed prediction with deep neural networks,'' *Renew. Energy*, vol. 85, pp. 83–95, Jan. 2016.

[20] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu, ''Stratified transfer learning for cross-domain activity recognition,'' in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.

[21] M. A. A. H. Khan, N. Roy, and A. Misra, ''Scaling human activity recognition via deep learning-based domain adaptation,'' in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–9.

[22] T.-T. Nguyen and S. Yoon, ''A novel approach to short-term stock price movement prediction using transfer learning,'' *Appl. Sci.*, vol. 9, no. 22, p. 4745, Nov. 2019.

[23] J. Xiao, Y. Hu, Y. Xiao, L. Xu, and S. Wang, ''A hybrid transfer learning model for crude oil price forecasting,'' *Statist. Interface*, vol. 10, no. 1, pp. 119–130, 2017.

[24] S. J. Pan and Q. Yang, ''A survey on transfer learning,'' *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[25] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, ''A survey on deep transfer learning,'' in *Proc. 27th Int. Conf. Artif. Neural Netw. (ICANN)*, 2018, pp. 270–279.

[26] K. Weiss, ''A survey of transfer learning,'' *J. Big Data*, vol. 3, no. 1, pp. 1–40, 2016.

[27] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, ''A comprehensive survey on transfer learning,'' 2019, *arXiv:1911.02685*.

[28] M. Längkvist, L. Karlsson, and A. Loutfi, ''A review of unsupervised feature learning and deep learning for time-series modeling,'' *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[29] J. C. B. Gamboa, ''Deep learning for time-series analysis,'' 2017, *arXiv:1701.01887*.

[30] D. Cook, K. D. Feuz, and N. C. Krishnan, ''Transfer learning for activity recognition: A survey,'' *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 537–556, Sep. 2013.

[31] N. Hernandez, J. Lundström, J. Favela, I. McChesney, and B. Arnrich, ''Literature review on transfer learning for human activity recognition using mobile and wearable devices with environmental technology,'' *Social Netw. Comput. Sci.*, vol. 1, no. 2, pp. 1–16, Mar. 2020.

[32] V. Jayaram, M. Alamgir, Y. Altun, B. Schölkopf, and M. Grosse-Wentrup, ''Transfer learning in brain-computer interfaces,'' 2015, *arXiv:1512.00296*.

[33] B. A. Kitchenham and S. M. Charters, *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Princeton, NJ, USA: Citeseer, 2007.

[34] J. J. Jiang, ''A literature survey on domain adaptation of statistical classifiers,'' Univ. Illinois, Champaign, IL, USA, Tech. Rep., 2008. [Online]. Available: http://www.mysmu.edu/faculty/jingjiang/papers/da_survey.pdf

[35] O. Day and T. M. Khoshgoftaar, ''A survey on heterogeneous transfer learning,'' *J. Big Data*, vol. 4, no. 1, pp. 1–42, Dec. 2017.

[36] J. Li, S. Qiu, Y.-Y. Shen, C.-L. Liu, and H. He, ''Multisource transfer learning for cross-subject EEG emotion recognition,'' *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3281–3293, Jul. 2020.

[37] G. Wilson, J. R. Doppa, and D. J. Cook, ''Multi-source deep domain adaptation with weak supervision for time-series sensor data,'' in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1768–1778.

[38] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, ''Domain-adversarial training of neural networks,'' *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.

[39] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[40] H. M. Cooper, ''Organizing knowledge syntheses: A taxonomy of literature reviews,'' *Knowl. Soc.*, vol. 1, no. 1, pp. 104–126, Mar. 1988.

[41] M. Weber, M. Auch, C. Doblander, P. Mandl, and H.-A. Jacobsen, ''Replication package for the paper: Transfer learning with time series data: A systematic mapping study,'' 2021, doi: 10.5281/zenodo.5720364.

[42] W. M. Bramer, M. L. Rethlefsen, J. Kleijnen, and O. H. Franco, ''Optimal database combinations for literature searches in systematic reviews: A prospective exploratory study,'' *Systematic Rev.*, vol. 6, no. 1, p. 245, Dec. 2017.

[43] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, and R. Chou, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *Systematic Rev.*, vol. 10, no. 1, pp. 1–16, Dec. 2021.

[44] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proc. 18th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2014, pp. 1–10.

[45] U. Blanke and B. Schiele, "Remember and transfer what you have learned–recognizing composite activities based on activity spotting," in *Proc. Int. Symp. Wearable Comput. (ISWC)*, Oct. 2010, pp. 1–8.

[46] W. Mao, D. Zhang, S. Tian, and J. Tang, "Robust detection of bearing early fault based on deep transfer learning," *Electronics*, vol. 9, no. 2, p. 323, Feb. 2020.

[47] W. Mao, J. He, and M. Zuo, "Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1594–1608, May 2020.

[48] A. Zhang, H. Wang, S. Li, Y. Cui, Z. Liu, G. Yang, and J. Hu, "Transfer learning with deep recurrent neural networks for remaining useful life estimation," *Appl. Sci.*, vol. 8, no. 12, p. 2416, 2018.

[49] R. Saeedi, K. Sasani, S. Norgaard, and A. H. Gebremedhin, "Personalized human activity recognition using wearables: A manifold learning-based knowledge transfer," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 1193–1196.

[50] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, pp. 2545–2550.

[51] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, "Towards environment independent device free human activity recognition," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 289–304.

[52] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Proc. Int. Conf. Pervasive Comput.*, 2010, pp. 283–300.

[53] W. Samek, F. C. Meinecke, and K.-R. M´uller, "Transferring subspaces between subjects in brain-computer interfacing," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2289–2298, Aug. 2013.

[54] F. Lotte and C. Guan, "Learning from other subjects helps reducing brain-computer interface calibration time," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 614–617.

[55] X. Tang and X. Zhang, "Conditional adversarial domain adaptation neural network for motor imagery EEG decoding," *Entropy*, vol. 22, no. 1, p. 96, Jan. 2020.

[56] Z. Yin, Y. Wang, L. Liu, W. Zhang, and J. Zhang, "Cross-subject EEG feature selection for emotion recognition using transfer recursive feature elimination," *Frontiers Neurorobot.*, vol. 11, p. 19, Apr. 2017.

[57] W.-L. Zheng and B.-L. Lu, "Personalizing eeg-based affective models with transfer learning," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2732–2738.

[58] F. Fahimi, Z. Zhang, W. B. Goh, T.-S. Lee, K. K. Ang, and C. Guan, "Inter-subject transfer learning with an end-to-end deep convolutional neural network for EEG-based BCI," *J. Neural Eng.*, vol. 16, no. 2, Apr. 2019, Art. no. 026007.

[59] C. Cooney, R. Folli, and D. Coyle, "Optimizing layers improves CNN generalization and transfer learning for imagined speech decoding from EEG," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1311–1316.

[60] M.-O. Tamm, Y. Muhammad, and N. Muhammad, "Classification of vowels from imagined speech with convolutional neural networks," *Computers*, vol. 9, no. 2, p. 46, Jun. 2020.

[61] D. Kearney, S. McLoone, and T. E. Ward, "Investigating the application of transfer learning to neural time series classification," in *Proc. 30th Irish Signals Syst. Conf. (ISSC)*, Jun. 2019, pp. 1–5.

[62] M. Ribeiro, K. Grolinger, H. F. ElYamany, W. A. Higashino, and M. A. M. Capretz, "Transfer learning with seasonal and trend adjustment for cross-building energy forecasting," *Energy Buildings*, vol. 165, pp. 352–363, Apr. 2018.

[63] C. Fan, Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, and Y. C. Tseng, "Statistical investigations of transfer learning-based methodology for short-term building energy predictions," *Appl. Energy*, vol. 262, Mar. 2020, Art. no. 114499.

[64] S. Ntalampiras, "A transfer learning framework for predicting the emotional content of generalized sound events," *J. Acoust. Soc. Amer.*, vol. 141, no. 3, p. 1694, 2017.

[65] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, "Autoencoder-based unsupervised domain adaptation for speech emotion recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1068–1072, Sep. 2014.

[66] E. Fons, P. Dawson, X.-J. Zeng, J. Keane, and A. Iosifidis, "Augmenting transferred representations for stock classification," 2020, *arXiv:2011.04545*.

[67] Y. Li, W. Jia, J. Wang, J. Guo, Q. Liu, X. Li, G. Xie, and F. Wang, "ALeRT-COVID: Attentive lockdown-awaRe transfer learning for predicting COVID-19 pandemics in different countries," *J. Healthcare Informat. Res.*, vol. 5, no. 1, pp. 98–113, Mar. 2021.

[68] V. Lampos, M. S. Majumder, E. Yom-Tov, M. Edelstein, S. Moura, Y. Hamada, M. X. Rangaka, R. A. McKendry, and I. J. Cox, "Tracking COVID-19 using online search," 2020, *arXiv:2003.08086*.

[69] K. Kashiparekh, J. Narwariya, P. Malhotra, L. Vig, and G. Shroff, "ConvTimeNet: A pre-trained deep convolutional neural network for time series classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[70] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," in *Proc. Int. Conf. Catalan Assoc. Artif. Intell. (CCIA), Frontiers Artif. Intell. Appl.*, vol. 308, Oct. 2018, pp. 120–129.

[71] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, and Y. Zhu. (Oct. 2018). *The UCR Time Series Classification Archive*. [Online]. Available: https://www.cs.ucr.edu/%7Eamonn/time_series_data_2018/

[72] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, p. 425, Jan. 2017.

[73] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 136–144, Jan. 2017.

[74] R. Zhang, H. Tao, L. Wu, and Y. Guan, "Transfer learning with neural networks for bearing fault diagnosis in changing working conditions," *IEEE Access*, vol. 5, pp. 14347–14357, 2017.

[75] W. Di, B. Wang, D. Precup, and B. Boulet, "Boosting based multiple kernel learning and transfer regression for electricity load forecasting," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML PKDD)*, 2017, pp. 39–51.

[76] S. Zhou, L. Zhou, M. Mao, and X. Xi, "Transfer learning for photovoltaic power forecasting with long short-term memory neural network," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 125–132.

[77] S. Inoue and X. Pan, "Supervised and unsupervised transfer learning for activity recognition from simple in-home sensors," in *Proc. 13th Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, Nov. 2016, pp. 20–27.

[78] P. Banda, M. A. Bhuiyan, K. Zhang, and A. Song, "Transfer learning for leisure centre energy consumption prediction," in *Proc. Int. Conf. Comput. Sci. (ICCS)*, 2019, pp. 112–123.

[79] B. Yang, Y. Lei, F. Jia, and S. Xing, "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings," *Mech. Syst. Signal Process.*, vol. 122, pp. 692–706, May 2019.

[80] M. J. Hasan, M. Sohaib, and J.-M. Kim, "1D CNN-based transfer learning model for bearing fault diagnosis under variable working conditions," in *Proc. Comput. Intell. Inf. Syst. Conf. (CIIS)*, 2018, pp. 13–23.

[81] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7316–7325, Sep. 2019.

[82] T. Alves, A. Laender, A. Veloso, and N. Ziviani, "Dynamic prediction of ICU mortality risk using domain adaptation," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1328–1336.

[83] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[84] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," in *Proc. 3rd Int. Conf. Crowd Sci. Eng. (ICCSE)*, 2018, pp. 1–8.

[85] M. Zhao, S. Yue, D. Katabi, T. S. Jaakkola, and M. T. Bianchi, "Learning sleep stages from radio signals: A conditional adversarial architecture," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 4100–4109.

[86] A. Marczewski, A. Veloso, and N. Ziviani, "Learning transferable features for speech emotion recognition," in *Proc. Thematic Workshops ACM Multimedia Thematic Workshops*, 2017, pp. 529–536.

[87] M. J. Hasan and J.-M. Kim, "Bearing fault diagnosis under variable rotational speeds using Stockwell transform-based vibration imaging and transfer learning," *Appl. Sci.*, vol. 8, no. 12, p. 2357, Nov. 2018.

[88] J. J. Hasan, M. M. M. Islam, and J.-M. Kim, "Acoustic spectral imaging and transfer learning for reliable bearing fault diagnosis under variable speed conditions," *Measurement*, vol. 138, pp. 620–631, May 2019.

[89] T. Wen and R. Keyes, "Time series anomaly detection using convolutional neural networks and transfer learning," 2019, *arXiv:1905.13628*.

[90] D. Buffelli and F. Vandin, "Attention-based deep learning framework for human activity recognition with user adaptation," 2020, *arXiv:2006.03820*.

[91] N. Kimura, I. Yoshinaga, K. Sekijima, I. Azechi, and D. Baba, "Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions," *Water*, vol. 12, no. 1, p. 96, Dec. 2019.

[92] A. Hooshmand and R. Sharma, "Energy predictive models with limited data using transfer learning," in *Proc. 10th ACM Int. Conf. Future Energy Syst.*, Jun. 2019, pp. 12–16.

[93] S. A. Rokni, M. Nourollahi, P. Alinia, I. Mirzadeh, M. Pedram, and H. Ghasemzadeh, "TransNet: Minimally supervised deep transfer learning for dynamic adaptation of wearable systems," *ACM Trans. Design Autom. Electron. Syst.*, vol. 26, no. 1, pp. 1–31, Jan. 2021.

[94] C. Taleb *et al.*, "Detection of Parkinson's disease from handwriting using deep learning: A comparative study," *Evol. Intell.*, 2020, doi: 10.1007/s12065-020-00470-0.

[95] S. Ullah and D.-H. Kim, "Lightweight driver behavior identification model with sparse learning on in-vehicle CAN-BUS sensor data," *Sensors*, vol. 20, no. 18, p. 5030, Sep. 2020.

[96] N. Strodthoff, P. Wagner, T. Schaeffter, and W. Samek, "Deep learning for ECG analysis: Benchmarks and insights from PTB-XL," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 5, pp. 1519–1528, May 2021.

[97] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018, *arXiv:1801.06146*.

[98] S. Mun, S. Shon, W. Kim, D. K. Han, and H. Ko, "Deep neural network based learning and transferring mid-level audio features for acoustic scene classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 796–800.

[99] H. Chen, G. Chen, Q. Lu, and L. Peng, "MMSE-based optimized transfer strategy for transfer prediction of parking data," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 407–412.

[100] S. Matsui, N. Inoue, Y. Akagi, G. Nagino, and K. Shinoda, "User adaptation of convolutional neural network for human activity recognition," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 753–757.

[101] M. Martinez and P. L. D. Leon, "Falls risk classification of older adults using deep neural networks and transfer learning," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 1, pp. 144–150, Jan. 2020.

[102] Y. Li, W. Zheng, Y. Zong, Z. Cui, T. Zhang, and X. Zhou, "A bi-hemisphere domain adversarial neural network model for EEG emotion recognition," *IEEE Trans. Affect. Comput.*, vol. 12, no. 2, pp. 494–504, Apr. 2021.

[103] S. Purushotham, W. Carvalho, T. Nilanon, and Y. Liu, "Variational recurrent adversarial deep domain adaptation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–15.

[104] C. Chen, Y. Miao, C. X. Lu, L. Xie, P. Blunsom, A. Markham, and N. Trigoni, "Motiontransformer: Transferring neural inertial tracking between domains," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8009–8016.

[105] J. Hernandez, R. R. Morris, and R. W. Picard, "Call center stress recognition with person-specific models," in *Proc. Int. Conf. Affect. Comput. Intell. Interact. (ACII)*, 2011, pp. 125–134.

[106] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Multi-layer domain adaptation method for rolling bearing fault diagnosis," *Signal Process.*, vol. 157, pp. 180–197, Apr. 2019.

[107] J. Zhu, N. Chen, and C. Shen, "A new deep transfer learning method for bearing fault diagnosis under different working conditions," *IEEE Sensors J.*, vol. 20, no. 15, pp. 8394–8402, Aug. 2020.

[108] D. Xiao, Y. Huang, L. Zhao, C. Qin, H. Shi, and C. Liu, "Domain adaptive motor fault diagnosis using deep transfer learning," *IEEE Access*, vol. 7, pp. 80937–80949, 2019.

[109] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, Jan. 2012.

[110] Z. Lan, O. Sourina, L. Wang, R. Scherer, and G. R. Müller-Putz, "Domain adaptation techniques for EEG-based emotion recognition: A comparative study on two public datasets," *IEEE Trans. Cogn. Devel. Syst.*, vol. 11, no. 1, pp. 85–94, Mar. 2019.

[111] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.

[112] R. Ding, X. Li, L. Nie, J. Li, X. Si, D. Chu, G. Liu, and D. Zhan, "Empirical study and improvement on deep transfer learning for human activity recognition," *Sensors*, vol. 19, no. 1, p. 57, Dec. 2018.

[113] K. Zhang, G. Cao, K. Zhou, and J. Liu, "Cross-domain fault diagnosis method for rotating machinery based on multi-representation adaptation neural network," in *Proc. 11th Int. Conf. Prognostics Syst. Health Manage. (PHM- Jinan)*, Oct. 2020, pp. 210–214.

[114] S. Yu, Z. Wu, X. Zhu, and M. Pecht, "A domain adaptive convolutional LSTM model for prognostic remaining useful life estimation under variant conditions," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Paris)*, May 2019, pp. 130–137.

[115] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.

[116] W. Qian, S. Li, P. Yi, and K. Zhang, "A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions," *Measurement*, vol. 138, pp. 514–525, May 2019.

[117] X. Xu and Z. Meng, "A hybrid transfer learning model for short-term electric load forecasting," *Electr. Eng.*, vol. 102, no. 3, pp. 1371–1381, Sep. 2020.

[118] P. Marcelino, M. de Lurdes Antunes, E. Fortunato, and M. C. Gomes, "Transfer learning for pavement performance prediction," *Int. J. Pavement Res. Technol.*, vol. 13, no. 2, pp. 154–167, Mar. 2020.

[119] R. Ye and Q. Dai, "A novel transfer learning framework for time series forecasting," *Knowl.-Based Syst.*, vol. 156, pp. 74–99, Sep. 2018.

[120] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 193–200.

[121] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, pp. 119–139, Aug. 1995.

[122] M. A. A. H. Khan and N. Roy, "TransAct: Transfer learning enabled activity recognition," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 545–550.

[123] F. Shen, C. Chen, R. Yan, and R. X. Gao, "Bearing fault diagnosis based on SVD feature extraction and transfer learning classification," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM)*, Oct. 2015, pp. 1–6.

[124] W. Tu and S. Sun, "A subject transfer framework for EEG classification," *Neurocomputing*, vol. 82, pp. 109–116, Apr. 2012.

[125] R. V. Deo, R. Chandra, and A. Sharma, "Stacked transfer learning for tropical cyclone intensity prediction," 2017, *arXiv:1708.06539*.

[126] W. Wang, B. Chen, P. Xia, J. Hu, and Y. Peng, "Sensor fusion for myoelectric control based on deep learning with recurrent convolutional neural networks," *Artif. Organs*, vol. 42, no. 9, pp. E272–E282, Sep. 2018.

[127] K. Benchaira, S. Bitam, A. Mellouk, A. Tahri, and R. Okbi, "AfibPred: A novel atrial fibrillation prediction approach based on short single-lead ECG using deep transfer knowledge," in *Proc. 4th Int. Conf. Big Data Internet Things*, Oct. 2019, pp. 1–6.

[128] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[129] S. Shen, M. Sadoughi, M. Li, Z. Wang, and C. Hu, "Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries," *Appl. Energy*, vol. 260, Feb. 2020, Art. no. 114296.

[130] Z. Di, H. Shao, and J. Xiang, "Ensemble deep transfer learning driven by multisensor signals for the fault diagnosis of bevel-gear cross-operation conditions," *Sci. China Technol. Sci.*, vol. 64, no. 3, pp. 481–492, Mar. 2021.

[131] A. Natarajan, G. Angarita, E. Gaiser, R. Malison, D. Ganesan, and B. M. Marlin, "Domain adaptation methods for improving lab-to-field generalization of cocaine detection using wearable ECG," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2016, pp. 875–885.

[132] X. Qin, Y. Chen, J. Wang, and C. Yu, "Cross-dataset activity recognition via adaptive spatial-temporal transfer learning," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 4, pp. 1–25, Dec. 2019.

[133] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[134] J. Xie, L. Zhang, L. Duan, and J. Wang, "On cross-domain feature fusion in gearbox fault diagnosis under various operating conditions based on transfer component analysis," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2016, pp. 1–6.

[135] O. Yair, M. Ben-Chen, and R. Talmon, "Parallel transport on the cone manifold of SPD matrices for domain adaptation," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1797–1811, Apr. 2019.

[136] K. Zhao *et al.*, "A novel transfer learning fault diagnosis method based on manifold embedded distribution alignment with a little labeled data," *J. Intell. Manuf.*, 2020, doi: 10.1007/s10845-020-01657-z.

[137] P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian Procrustes analysis: Transfer learning for brain–computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 8, pp. 2390–2401, Aug. 2019.

[138] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.

[139] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Proc. Humaine Assoc. Conf. Affect. Comput. Intell. Interact.*, Sep. 2013, pp. 511–516.

[140] X. Chai, Q. Wang, Y. Zhao, X. Liu, O. Bai, and Y. Li, "Unsupervised domain adaptation techniques based on auto-encoder for non-stationary EEG-based emotion recognition," *Comput. Biol. Med.*, vol. 79, pp. 205–214, Dec. 2016.

[141] M. A. A. H. Khan and N. Roy, "UnTran: Recognizing unseen activities with unlabeled data using transfer learning," in *Proc. IEEE/ACM 3rd Int. Conf. Internet Things Design Implement. (IoTDI)*, Apr. 2018, pp. 37–47.

[142] J. Han, S. Miao, Y. Li, W. Yang, and H. Yin, "A wind farm equivalent method based on multi-view transfer clustering and stack sparse auto encoder," *IEEE Access*, vol. 8, pp. 92827–92841, 2020.

[143] M. Sun, H. Wang, P. Liu, S. Huang, and P. Fan, "A sparse stacked denoising autoencoder with optimized transfer learning applied to the fault diagnosis of rolling bearings," *Measurement*, vol. 146, pp. 305–314, Nov. 2019.

[144] E. Coutinho and B. Schuller, "Shared acoustic codes underlie emotional communication in music and speech-evidence from deep transfer learning," *PLoS ONE*, vol. 12, no. 6, 2017, Art. no. e0179289.

[145] X. Duan, N. Chen, and Y. Xie, "Intelligent detection of large-scale KPI streams anomaly based on transfer learning," in *Proc. CCF Conf. Big Data*, 2019, pp. 366–379.

[146] F. Li, K. Shirahama, M. A. Nisar, X. Huang, and M. Grzegorzek, "Deep transfer learning for time series data based on sensor modality classification," *Sensors*, vol. 20, no. 15, p. 4271, Jul. 2020.

[147] J. Deng, R. Xia, Z. Zhang, Y. Liu, and B. Schuller, "Introducing shared-hidden-layer autoencoders for transfer learning and their application in acoustic emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4818–4822.

[148] 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, p. 289, Feb. 2016.

[149] A. Akbari and R. Jafari, "Transferring activity recognition models for new wearable sensors with deep generative domain adaptation," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2019, pp. 85–96.

[150] X. Li, X.-D. Jia, W. Zhang, H. Ma, Z. Luo, and X. Li, "Intelligent cross-machine fault diagnosis approach with deep auto-encoder and domain adaptation," *Neurocomputing*, vol. 383, pp. 235–247, Mar. 2020.

[151] A. Z. M. Faridee, M. A. A. H. Khan, N. Pathak, and N. Roy, "AugToAct: Scaling complex human activity recognition with few labels," in *Proc. 16th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, Nov. 2019, pp. 162–171.

[152] D. Banerjee, K. Islam, G. Mei, L. Xiao, G. Zhang, R. Xu, S. Ji, and J. Li, "A deep transfer learning approach for improved post-traumatic stress disorder diagnosis," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 11–20.

[153] A. Meiseles and L. Rokach, "Source model selection for deep learning in the time series domain," *IEEE Access*, vol. 8, pp. 6190–6200, 2020.

[154] Y. Zhou, M. Han, J. He, L. Liu, X. Xu, and X. Gao, "Abnormal activity detection in edge computing: A transfer learning approach," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2020, pp. 107–111.

[155] V. Vercruyssen, W. Meert, and J. Davis, "Transfer learning for time series anomaly detection," in *Proc. Workshop Tutorial Interact. Adapt. Learn. ECMLPKDD. CEUR Workshop*, vol. 1924, 2017, pp. 27–37.

[156] J. Shang and J. Wu, "A robust sign language recognition system with sparsely labeled instances using Wi-Fi signals," in *Proc. IEEE 14th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2017, pp. 99–107.

[157] J. R. Villar, E. de la Cal, M. Fañez, V. M. González, and J. Sedano, "User-centered fall detection using supervised, on-line learning and transfer learning," *Prog. Artif. Intell.*, vol. 8, no. 4, pp. 453–474, Dec. 2019.

[158] M. Fañez, J. R. Villar, E. Cal, J. Sedano, and V. M. González, "Transfer learning and information retrieval applied to fall detection," *Expert Syst.*, vol. 37, no. 6, Dec. 2020, Art. no. e12522.

[159] Y. Zhang and G. Luo, "Short term power load prediction with knowledge transfer," *Inf. Syst.*, vol. 53, pp. 161–169, Oct. 2015.

[160] J. Xiao, Y. Xiao, J. Fu, and K. K. Lai, "A transfer forecasting model for container throughput guided by discrete PSO," *J. Syst. Sci. Complex.*, vol. 27, no. 1, pp. 181–192, Feb. 2014.

[161] Y. Chen, J. Wang, M. Huang, and H. Yu, "Cross-position activity recognition with stratified transfer learning," *Pervas. Mobile Comput.*, vol. 57, pp. 1–13, Jul. 2019.

[162] S. P. Shashikumar, A. J. Shah, G. D. Clifford, and S. Nemati, "Detection of paroxysmal atrial fibrillation using attention-based bidirectional recurrent neural networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 715–723.

[163] M. J. Hasan, M. M. M. Islam, and J.-M. Kim, "Multi-sensor fusion-based time-frequency imaging and transfer learning for spherical tank crack diagnosis under variable pressure conditions," *Measurement*, vol. 168, Jan. 2021, Art. no. 108478.

[164] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly accurate machine fault diagnosis using deep transfer learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.

[165] J. Zhou, X. Yang, L. Zhang, S. Shao, and G. Bian, "Multisignal VGG19 network with transposed convolution for rotating machinery fault diagnosis based on deep transfer learning," *Shock Vibrat.*, vol. 2020, pp. 1–12, Dec. 2020.

[166] A. Ullah, S. M. Anwar, M. Bilal, and R. M. Mehmood, "Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation," *Remote Sens.*, vol. 12, no. 10, p. 1685, Apr. 2020.

[167] N. Patricia, T. Tommasit, and B. Caputo, "Multi-source adaptive learning for fast control of prosthetics hand," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 2769–2774.

[168] D. H. Hu, V. W. Zheng, and Q. Yang, "Cross-domain activity recognition via transfer learning," *Pervasive Mobile Comput.*, vol. 7, no. 3, pp. 344–358, 2011.

MANUEL WEBER studied information systems at the University of Stuttgart and the University of Hohenheim. He is a Research Assistant with the Department of Computer Science and Mathematics, Munich University of Applied Sciences, since 2019, and is part of the Distributed Cognitive Computing Research Group. Since 2020, he has also been a member of IAMLIS, an institute for machine learning applications at the Munich University of Applied Sciences. His research interests include machine learning, transfer learning, and building occupancy estimation.

MAXIMILIAN AUCH received the Master of Science degree in business information systems from the Munich University of Applied Sciences, Munich. He has been a Research Assistant at the Department of Computer Science and Mathematics, Munich University of Applied Sciences, since 2017, and has also been simultaneously employed as a Software Developer at AUSY Technologies Germany AG, since 2017. His research interests include software engineering, recommendation systems, and knowledge management.

CHRISTOPH DOBLANDER was a Postdoctoral Researcher at TUM, Munich, before going to industry. His research interests include machine learning and using simulation and reinforcement learning as a way to control complex systems.

PETER MANDL has been a Professor, with a focus on ''distributed systems,'' at the Faculty of Computer Science and Mathematics, Munich University of Applied Sciences, Munich, since 2002, a Spokesman of the Competence Center for Information Systems (CCWI), Munich University of Applied Sciences, since 2005, and an Executive Board Member of iSYS Software GmbH, since 1996. Since 2020, he has also been a member of IAMLIS, an institute for machine learning applications at the Munich University of Applied Sciences.

HANS-ARNO JACOBSEN (Fellow, IEEE) received the Ph.D. degree from the Humboldt University of Berlin, Germany. He was involved in postdoctoral research at INRIA, Paris, France, before moving to the University of Toronto, in 2001, where he is currently a Professor of computer engineering and computer science directing the activities of the Middleware Systems Research Group. He conducts research at the intersection of distributed systems and data management, with a particular focus on middleware systems, event processing, and cyber-physical systems.