Manuel Weber

Occupancy Detection with Simulated Environmental Data

Technische Universität München

Technische Universität München

TUM School of Computation, Information and Technology

Occupancy Detection with Simulated Environmental Data

Manuel Weber

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Jens Großklags

Prüfer der Dissertation:

1. Prof. Dr. Hans-Arno Jacobsen
2. Prof. Dr. Peter Mandl
3. Prof. Dr. Felix Dietrich

Die Dissertation wurde am 11.11.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 25.03.2025 angenommen.

Abstract

Reliable detection of the temporal presence of occupants in individual rooms of a building offers numerous opportunities for demand-driven optimization of building operations. This promises significant reductions in emissions compared to the current state of the building sector. Despite extensive research on the automation of presence detection using machine learning, its application in practice remains limited. A major limitation lies in the necessity of collecting room-specific annotated data. This dissertation investigates the application of transfer learning approaches and the use of artificially generated data to reduce the dependence on real-world data. The work focuses on the widely adopted and privacy-preserving approach of deriving presence from measurements of indoor CO_2 concentration.

At the outset of our research, we explore the feasibility of model training assisted by simulated CO_2 values. Subsequently, we conduct a comprehensive analysis of the existing transfer learning literature with respect to time series data. Based on the predominant methods in the literature, we conduct an empirical study on the application of occupancy detection in buildings, evaluating selected methods across different rooms. Finally, we propose a pretrained base model, developed from extensive room simulations, and demonstrate its applicability in real-world scenarios, even in the absence of extensive training data. This dissertation makes a substantial contribution to the enhancement of the applicability of machine learning-based occupancy detection through transfer learning and paves the way for efficient model training based on simulations.

Zusammenfassung

Die zuverlässige Erkennung der zeitlichen Anwesenheit von Nutzern in den einzelnen Räumen eines Gebäudes eröffnet vielfältige Möglichkeiten für eine bedarfsgerechte Optimierung des Gebäudebetriebs. Diese verspricht erhebliche Emissionseinsparungen im Vergleich zum heutigen Stand des Gebäudesektors. Trotz weitreichender Forschung zur Automatisierung der Anwesenheitserkennung mittels maschinellem Lernen, findet diese nur begrenzt Anwendung in der Praxis. Eine wesentliche Limitation besteht in der Notwendigkeit der Erhebung raumindividueller annotierter Daten. Diese Dissertation untersucht den Einsatz von Transfer-Learning-Ansätzen und die Verwendung künstlich generierter Daten, um die Abhängigkeit von Realdaten zu reduzieren. Dabei legt die Arbeit den Fokus auf den weit verbreiteten und privatsphärewahrenden Ansatz, Anwesenheit von Messungen der CO_2-Konzentration im Innenraum abzuleiten.

Zu Beginn unserer Forschung untersuchen wir die Machbarkeit eines Modelltrainings, das durch simulierte CO_2-Werte unterstützt wird. Daraufhin führen wir eine umfangreiche Analyse der existierenden Transfer-Learning-Literatur mit Bezug auf Zeitreihendaten durch. Auf Basis der in der Literatur dominierenden Methoden führen wir für den Anwendungsfall der Gebäudeanwesenheitserkennung eine empirische Studie durch, die ausgewählte Methoden anhand verschiedener Räume evaluiert. Abschließend schlagen wir ein vortrainiertes Basismodell vor, das auf einer Vielzahl durchgeführter Raumsimulationen beruht, und demonstrieren dessen Einsetzbarkeit in realen Anwendungsszenarien, auch unter Abwesenheit umfangreicher Trainingsdaten. Diese Dissertation bietet einen umfassenden Beitrag zur Verbesserung der Anwendbarkeit von Machine-Learning-basierter Anwesenheitserkennung durch Transfer Learning und ebnet den Weg für effizientes Modelltraining auf Basis von Simulationen.

Acknowledgments

This doctoral dissertation and the included research were carried out at the Department of Computer Science and Mathematics at the Munich University of Applied Sciences (HM) under Prof. Dr. Peter Mandl, who supervised my work jointly with Prof. Dr. Hans-Arno Jacobsen from the University of Toronto.

I would like to express my sincere gratitude to both of my professors, whose valuable advice, positive attitude, and encouragement have been indispensable throughout the past years. Peter Mandl has become a role model for me not only in terms of research but also in terms of soft skills and self-management. I would especially like to thank him for the freedom to choose my own research topics and for always being approachable. I am also immensely grateful to my first supervisor, Prof. Dr. Jacobsen, for his timely feedback and helpful meetings, even while being halfway across the globe. His scientific guidance has been crucial in successfully navigating the complexities of my research.

Furthermore, I would like to thank the remaining examination committee: Prof. Dr. Felix Dietrich for agreeing to be my third examiner and Prof. Dr. Jens Großklags for accepting to chair the committee. I am also deeply appreciative of my two mentors, Dr. Christoph Doblander and Prof. Dr. Ruben Mayer, whose important tips and feedback on my paper manuscripts have likely played a significant role in getting papers accepted quickly.

Many thanks to all of my colleagues including Jan Vellmer, Maximilian Auch, Max-Emanuel Keller, and Farzan Banihashemi for their productive collaboration, enriching discussions on various topics, scientific exchange, and also non-scientific conversations that have made this journey memorable. I would also like to give special thanks to Eshref Januzaj, who has been a great office mate and extremely helpful with his advice, particularly at the beginning of my doctoral journey. Apart from the academic world, I am grateful to all my friends and family for their constant encouragement and motivational support. Special thanks to Giselle for her meticulous advice on the English language. Lastly, I want to thank my partner, Lea, for her unwavering emotional support and patience. Your presence has been my anchor during the challenging times of this journey.

Thank you all for being a part of this significant milestone in my life.

Contents

1

Introduction

The building sector holds large potential for reducing energy consumption and greenhouse gas emissions. In building research, a substantial body of literature addresses this challenge by relying on information technology and machine learning, which are increasingly applied in multiple phases of the building life cycle and frequently linked to energy savings [1]. At the same time, there is an ongoing trend towards occupant-centric building design and operation [2]. While today's energy performance predictions are mostly based on building design and leave prospective occupants out of consideration, the focus of research is increasingly shifting toward the occupant.

However, this development requires dynamic data on occupant behavior, which are rarely available for the majority of contemporary buildings. Consequently, occupant-centric research depends on the exploration of methods for occupant data collection, encompassing a variety of sensing technologies facilitated by computational algorithms [3]. Binary presence data form the basis of occupant behavior and serve as an enabler for use cases concerning energy optimization and occupant comfort. Yet, their continuous collection is cost-intensive when involving human effort or dedicated hardware. A more pragmatic approach is to apply occupancy detection from data that can be acquired inexpensively and used for other purposes as well.

Occupancy detection is the task of identifying occupant presence in the form of a binary classification. In this work, we consider time series of indoor CO_2 values as input to the classification problem. CO_2 levels in building spaces correlate with human presence [4] and offer a non-intrusive indication of occupancy [5, 6]. In the literature, CO_2 is a frequently used factor for occupancy detection [7, 8]. Most commonly, data-driven methods, such as neural networks, are used to infer occupancy [7]. However, these require labeled training data, which are scarce in practice.

This work proposes and evaluates transfer learning-based solutions towards occupancy detection, particularly with respect to the learning from simulated data. The goal is to reduce the impact of data scarcity and thereby to advance practical applicability.

1.1 Motivation

According to the International Energy Agency (IEA), the building sector accounts for over 30% of the world's total energy consumption [9]. Despite widespread awareness of the need to reduce greenhouse gas emissions, global annual emissions have continued to increase over the past decade. Within the European Union (EU), buildings are responsible for 40% of energy use, resulting in 36% of carbon emissions [10, 11]. Through the European Green Deal, the EU has set the goal of achieving a net-zero carbon footprint by 2050 [10]. The European Parliament has further decided on the interim goal of reducing emissions by at least 55%, compared to 1990, until 2030 [12]. Given its large share of the total energy consumption, the building sector plays a significant role in achieving these goals. As stated in [11], to meet the current European climate targets for 2030, building emissions must be reduced by at least 60%.

Occupant-centric building control based on real-time occupancy states is a promising approach to increase energy efficiency in buildings. Aligning heating, ventilation, and air conditioning (HVAC) with the presence of occupants in each space of a building helps prevent energy waste while at the same time ensuring occupant comfort. Studies employing occupancy-informed HVAC control strategies have reported significant decreases in energy consumption: Brooks et al. [13] evaluated occupancy-informed control strategies in simulations and reported energy savings of 10 to 48%. In another study, they applied occupant-centric control in a real-world commercial building, reporting a reduction in energy consumption by 37% [14]. These savings are primarily achieved by reducing the air flow rate during room vacancy. This leads to significant energy savings compared to the use of standardized schedules while still maintaining indoor environmental quality. Another approach involves alternating between thermostat setpoints and reduced setback temperatures for times of vacancy. A review of the literature on occupancy-based HVAC control [15] provides an overview of various studies, both simulated and field-tested, that regularly demonstrate substantial energy savings, with the highest reported real-world savings reaching up to 52%. These empirical findings motivate our work by highlighting the potential of occupancy data use for energy optimization.

In addition to building control, the availability of occupancy data may also enhance building performance simulation (BPS). Assessing energy performance and identifying energy-saving opportunities is not only possible during a building's design phase but also throughout its operation. BPS is a commonly applied method to predict building performance with computer-based models. These models typically rely on static occupation schedules that broadly categorize occupancy patterns according to building type. Data collected from occupants, however, allow us to identify more specific patterns that may yield a higher accuracy of energy performance predictions. With occupants being the primary determinants of energy consumption in buildings [16], their behavior should be given greater consideration in building simulations. Occupant behavior is recognized as a major cause of the energy performance gap [17], which denotes the disparity between anticipated and actual building energy performance. To narrow this gap, occupant-related data is required. As occupant presence is the prerequisite for any further occupant behavior, data on presence states are essential in this context.

While occupancy data may be recorded through a variety of sensing technologies, CO_2-based detection offers several advantages. The benefits include cost-effective sensors, non-intrusive data collection, and low dependence on sensor placement. Moreover, CO_2 and other environmental factors can further be used to model occupant behavior, such as window operation behavior [18], which holds additional potential for energy optimization. Unlike other factors such as indoor temperature or humidity, CO_2 is less affected by seasonality, weather changes, or indoor heat sources.

1.2 Problem Statement

Past research has shown the feasibility of detecting occupancy from CO_2 rates. State-of-the-art models reach accuracies above 95% [19, 20, 21, 22], which we consider useful regarding building energy optimization. However, these models are trained in a supervised setting, which makes them depend on labeled training data. Data on CO_2 and other climate factors are generally collected by environmental sensors placed in each room. These can already be part of the building infrastructure that serves the building management system (BMS), or can be deployed as additional Internet of Things (IoT) sensors. After the initial sensor installation, data collection can be carried out without human effort. In contrast, occupancy ground truth collection requires additional effort. Collecting reliable occupancy states involves either manual recording or a sensing method that is significantly more accurate than inferring occupancy from CO_2. In studies, ground truth is collected mainly using cameras to surveil the room and label occupancy times through a manual retrospective review of recordings [4, 5, 21, 22, 23, 24]. This may present practical challenges, as cameras raise privacy concerns and incur additional costs. Moreover, the analysis of camera recordings can be automated to some extent, e.g., by using motion detection software, but may still require manual corrections [19, 21]. In addition to cameras, other studies use manual recordings by the occupants themselves [25, 26, 27]. In [27], for example, buttons are provided for occupants to self-document times of entering or leaving the room.

Such methods are feasible in scientific experiments, but their scalability is limited. Real-world use cases may involve large buildings with hundreds of rooms, where manual recording at room level appears impractical. Consequently, applying methods from state-of-the-art research in practice is hardly feasible. Application in cases with scarce data from the target room collected within only a few days of occupation results in poor accuracy [27]. Without any collected target data, the application is not possible at all. Hence, target data scarcity is a major impediment to the practicality of CO_2-based occupancy detection.

This work aims to enable occupancy detection from CO_2 levels in environments where labeled training data is scarce, specifically focusing on small to medium-sized rooms in commercial buildings. Our essential approach is to address this through transfer learning and data simulation. Accordingly, we identified three relevant research problems to be addressed: The first problem is analyzing existing methods of transfer learning that can effectively be applied on time series data. The second problem is studying selected transfer learning methods in the specific application case of occupancy detection in different rooms. The last problem is finding an approach that involves data simulation to further reduce the need for manual data collection. Addressing these problems will enable the implementation of more data-efficient and potentially more accurate occupancy detection systems. This will support further research and development towards building optimization.

1.2.1 Structuring Fragmented Literature on Time Series Transfer Learning

Transfer learning is a broad concept that can be applied through various specific methods. The general goal is to reduce the need for labeled training data in supervised learning scenarios. It has been investigated across diverse domains and data types, including problems of natural language processing or image classification [28].

As this work addresses the problem of occupancy detection in the form of a time series classification problem, we identified the need to review the related literature that specifically applies transfer learning on time series data. For a description of the addressed problem of occupancy detection and the concept of transfer learning, we refer to Sec. 2.1, respectively, to Sec. 2.3.2. Further details are provided in the publications that this dissertation comprises.

In the context of time series classification, the effectiveness of transfer learning was demonstrated, for instance, by a comprehensive study by Fawaz et al. [29]. In the same time, a substantial body of literature has addressed applications of transfer learning across a diverse range of use cases that involve time series data [30, 31, 32, 33, 34]. Many of the methods proposed in the literature are also applicable to occupancy detection. To gain an overview of these methods and develop an understanding of specific challenges when addressing time series data, we need to comprehensively analyze the existing literature on time series transfer learning. More specifically, we seek to identify the main application domains, transfer learning methods, and the types of machine learning models implemented within these. This allows us to derive suitable transfer learning methods for the addressed problem of occupancy detection.

1.2.2 Evaluating Transfer Learning for Occupancy Detection

Based on general insights from the literature, it is necessary to find methods that specifically address the problem of occupancy detection. Therefore, we need to find a machine learning model that can serve as an accurate classifier in this context when given sufficient training data. In addition to this, we need to find transfer methods to decrease its dependence on labeled training data from the room of interest.

To assess the effectiveness of transfer learning for the problem of occupancy detection, an empirical analysis of promising methods in real-world scenarios is required. Previous studies have primarily applied a single proposed method [35, 36, 37, 38]. However, it is more informative to conduct a comparative analysis of multiple methods across data from various rooms. This provides insight into the effectiveness of transfer learning solutions and helps identify the most promising method.

Additionally, we need to evaluate the same methods also regarding transfer from simulated data. While data simulation reduces the need for real-world training data, model performance may vary significantly compared to results obtained with related real-world data. It is therefore crucial to consider both simulated and real-world datasets.

1.2.3 Overcoming Scarcity of Source Data

Transfer learning from source to target domain can only be effective if sufficient source data are available that are similar to the target data. For straightforward application in practice, public datasets containing ground truth occupancy states and CO_2 levels with adequate granularity can be utilized. However, there are only a few public datasets available [39]. In addition, they typically include observation periods of a few days or weeks, which ignores seasonal changes or changes in personnel during the year [4, 40].

Other datasets that offer longer observation periods do not provide high-resolution data, but instead utilize measurement intervals of 5 or 10 minutes [41, 42, 43]. This limits the application of transfer learning from related real-world scenarios in practice. With its focus on transfer learning between real-world datasets, previous literature overlooks this impediment [35, 36, 37].

Moreover, even under extensive source data availability, source selection remains critical. Dissimilarities between source and target data carry the risk of negative transfer and may potentially harm prediction performance. For instance, if occupants exhibit unique behavior in their interactions with windows or doors, it may lead to patterns in the CO_2 data that do not occur in other datasets and may cause the model to overfit to these specific patterns, reducing its generalizability to other rooms.

To foster practical implementation, we need to find an approach based on simulated source data that allows adaptation to arbitrary real-world target rooms. This includes finding an adequate method for data simulation as well as a reliable transfer method.

1.3 Approach

In this work, we apply transfer learning to achieve accurate occupancy predictions in scenarios with scarce data. We consider a binary classification of occupancy that addresses a specific room inside a building. In this context, we differ between the target domain, representing the room of interest, and the source domain, which either refers to (1) a selected room with higher availability of data in the same or in another building, or (2) to an environmental simulation. We investigate methods of transfer learning and their combination with simulations to reduce the required amount of data from the target domain to a minimum and to eliminate the necessity of real-world source data.

First, we systematically analyze the existing literature on transfer learning with time series data. Based on the most frequent applicable transfer methods identified, we evaluate applications of transfer learning in the context of CO_2-based occupancy detection. This includes an evaluation of the transfer between real-world datasets as well as the transfer from simulated source data. Finally, we describe a simulation-based approach that does not depend on the availability and selection of source data. We support this approach by building a corresponding model and demonstrating its ability to be fine-tuned to real-world scenarios.

1.3.1 Systematic Literature Mapping

To structure the research area on time series transfer learning and analyze existing publications, we conduct a systematic literature mapping. This involves the three consecutive steps of (1) searching for literature, (2) selecting relevant publications, and (3) systematically analyzing the body of selected publications.

The mapping process adheres to the guidelines of Kitchenham and Charters [44] for systematic literature reviews in software engineering. We identify a suitable search term for searching within a collection of relevant electronic literature databases and extend the search by manual snowballing on included publications. We define clear inclusion and exclusion criteria for literature selection, which are applied in a three-step screening of all found publications; first regarding the title, then the abstract, and finally the full-text version of a publication. Publications identified as relevant are then analyzed in a mainly inductive content analysis. Additionally, categories of transfer approaches are adopted from existing literature reviews on transfer learning in general, in case they apply to transfer learning with time series data.

1.3.2 Empirical Analysis of Transfer Learning Methods

To assess the effectiveness of transfer learning in the context of occupancy detection and compare different transfer methods, we collect multiple datasets and implement promising transfer methods. We identify a model type capable of modeling the complexities of CO_2 dynamics. The idea is to use an existing, outperforming model type from the literature and conduct a hyperparameter optimization to determine the specific configuration, including the concrete model architecture. Instead of an exhaustive grid search, we employ a more efficient optimization method leveraging information from prior evaluations.

For evaluation during the optimization process, multiple real-world datasets are used, with the intention of ensuring model robustness and generalizability. Using the optimized model architecture and configuration, we train models for each viable combination of source and target datasets repeatedly for each considered transfer method and different amounts of scarce target data. We evaluate prediction performance regarding different methods and source data, for the range between one and five days of data from the target room being used for transfer. To ensure a wide generalizability of the results, datasets are selected from a variety of scenarios that include multiple different room types. We use five real-world datasets, either collected or reused from other works. We also use a simulated dataset, where the CO_2 rates are generated according to physical building parameters, mass balance, typical human respiration, and randomized room occupancy.

1.3.3 Simulation-trained Base Model

This section describes our approach to solving the problem that is described in Sec. 1.2.3. The underlying concept of our approach is illustrated in Fig. 1.3.1. To remove the necessity to collect or select real-world source data, we use simulated source data alongside real-world target data. Considering the predominant transfer learning method, pretraining and fine-tuning [45], this means that a base model is initially pretrained on simulated data and subsequently fine-tuned on real-world target data.

The base model offers the ability to capture the general principles of indoor CO_2 dynamics, while the transfer step customizes the model to the specific conditions of each room. Incorporating factors such as room type, other metadata of the room, and typical occupancy patterns, we perform two successive simulations:

(1) A simulation to produce synthetic occupancy states for a single occupant or multiple individual occupants. In addition, the simulation encompasses occupant behavior in terms of occupant-specific window operation.

(2) A physical simulation of the indoor environment, with a focus on CO_2 rates, on the basis of data from simulation (1).

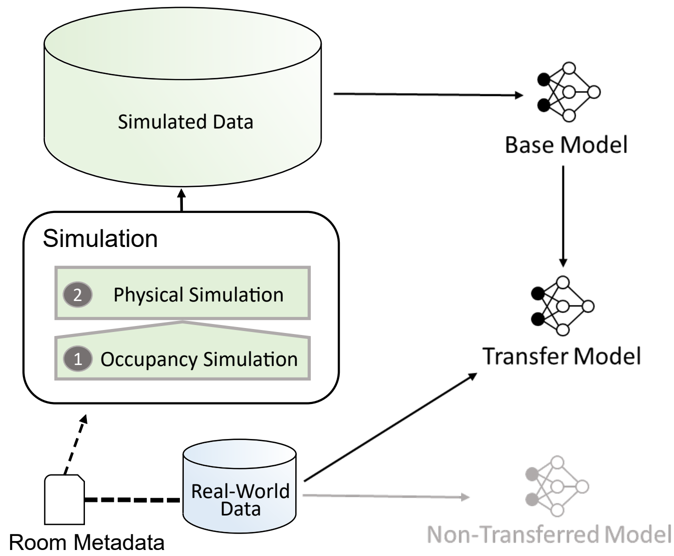

Figure 1.3.1: Approach on occupancy detection with simulated environmental data [46]

It is still necessary to conduct further room-specific training, as varying room conditions, such as room geometry, infiltration, or sensor placement, require adaptation. For the integration of simulations with model training, we distinguish two approaches:

(a) Running a room-specific, upfront simulation and conducting pretraining dedicated to the room of interest: This requires building a digital representation of the room based on its specific characteristics and running a dedicated simulation.

(b) Running a one-time training of a general base model tailored to a group of similar rooms: This requires multiple simulations for randomly generated room characteristics. Besides enhancing performance under sparse training data, this also aims to reduce the computational effort of model training during deployment, as well as the human effort involved in modeling and simulation.

While we focus on approach (b) to limit the extent of human involvement and foster quick applicability in practice, we additionally address approach (a) in [46, 47].

To prepare a general base model, as proposed in approach (b), we utilize the concept of domain randomization. Domain randomization is a technique that facilitates the efficient provision of training data, with the core idea of exposing the model to a diverse range of simulated training scenarios and thereby enhancing its ability to generalize effectively to real-world conditions during application or testing [48]. We design an approach for the simulation of a variety of training data, considering typical office or meeting room scenarios. We apply the approach to generate a large synthetic dataset and use the data to pretrain a base model. Additional real-world data are used to retrain a second model variant. We evaluate the two models with scarce real-world data from both an office and meeting room scenario.

1.4 Contributions

In this work, we analyze the application of transfer learning to occupancy detection in building rooms. The primary goal is to relax the data requirements for supervised training of room-specific deep learning models for occupancy detection. By relying on data simulation, we reduce the need for data collection in real-world rooms. Our main contributions include the following:

i. We describe different transfer learning approaches and problem settings regarding time series problems existing in the literature. Moreover, we provide a comprehensive overview of the existing work on time series transfer learning, covering, among other aspects, models, methods, and application domains. This is useful for our own work and for others to understand current methodologies and develop own approaches. It is further helpful to identify research gaps. In addition, we specifically point out opportunities for future research. [45]
ii. We provide a proof-of-concept demonstrating enhanced prediction performance in scarce-data supervised learning for occupancy detection when supplementary utilizing simulated data. In an experimental evaluation, we show that the envisioned approach achieves comparable performance using only around half of the original target data. By this, we motivate further and more comprehensive work in this direction. [46]
iii. Our work analyzes the impact of transfer learning strategies on occupancy detection performance. It specifically examines how different methods, source data, and varying amounts of target room training data influence model accuracy across

multiple scenarios. This can guide the selection of appropriate transfer learning methods in research and practice. Through a practical example from our evaluation, we specifically emphazise the challenge of potential negative transfer caused by dissimilarities between rooms or occupants, and point out the need for methods that allow for wider generalizability. [49]

iv. We design an approach on occupancy detection based on the simulation of indoor environmental data and deep learning. The effectiveness of the approach is demonstrated in experiments with real-world data. Our approach can be modified and adapted to differing room types or other spatial domains. [50]

v. Our contributions further include a series of artifacts that can be reused in other works. First of all, we provide a collection of preprocessed datasets that contain indoor CO_2 and occupancy data for multiple weeks, including two newly published datasets [49]. We also provide a large simulated dataset for occupancy modeling from environmental data [50]. This helps address the lack of sufficient publicly available datasets, which impedes research on occupancy detection from CO_2. Furthermore, we provide the saved parameters of a ready-to-use simulation-trained model for application through fine-tuning that allows efficient training of a classifier for a specific real-world room [50]. In addition, scripts used for data simulation, data preparation, and model training are made publicly available [49, 50].

Parts of the content and contributions of this work have been published in:

- **Manuel Weber**, Maximilian Auch, Christoph Doblander, Peter Mandl, and Hans-Arno Jacobsen. "Transfer Learning With Time Series Data: A Systematic Mapping Study." In: *IEEE Access* 9 (2021), pp. 165409–165432. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3134628. URL: https://doi.org/10.1109/ACCESS.2021.3134628 **(CORE PUBLICATION 1)**

- **Manuel Weber**, Farzan Banihashemi, Peter Mandl, Hans-Arno Jacobsen, and Ruben Mayer. "Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection." In: *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2023), pp. 1–10. DOI: 10.1145/3600100.3623718. URL: https://doi.org/10.1145/3600100.3623718 **(CORE PUBLICATION 2)**

- **Manuel Weber**, Farzan Banihashemi, Davor Stjelja, Peter Mandl, Ruben Mayer, and Hans-Arno Jacobsen. "Coddora: CO2-Based Occupancy Detection Model Trained via Domain Randomization." In: *International Joint Conference on Neural Networks (IJCNN)* (2024), pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650820. URL: https://doi.org/10.1109/IJCNN60899.2024.10650820 **(CORE PUBLICATION 3)**

- **Manuel Weber**, Christoph Doblander, and Peter Mandl. "Detecting Building Occupancy with Synthetic Environmental Data." In: *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2020), pp. 324–325. DOI: 10.1145/3408308.3431124. URL: https://doi.org/10.1145/3408308.3431124

  An extended version of this publication has further been published under:

  Manuel Weber, Christoph Doblander, and Peter Mandl. *Towards the Detection of Building Occupancy with Synthetic Environmental Data.* (2020). DOI: 10.48550/arXiv.2010.04209. arXiv: 2010.04209

Further publications related to this work include:

- Farzan Banihashemi, **Manuel Weber**, Fatma Deghim, Chujun Zong, and Werner Lang. "Occupancy modeling on non-intrusive indoor environmental data through machine learning." In: *Building and Environment* 254.111382 (2024). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2024.111382

- Farzan Banihashemi, **Manuel Weber**, Bing Dong, Salvatore Carlucci, Roland Reitberger, and Werner Lang. "Window state or action modeling? An explainable AI approach in offices." In: *Energy and Buildings* 298.113546 (2023). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2023.113546

- Farzan Banihashemi, **Manuel Weber**, and Werner Lang. "Deep learning for predictive window operation modeling in open-plan offices." In: *Energy and Buildings* 310.114109 (2024). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2024.114109

- **Manuel Weber**, Philipp Bogdain, Sophia Viktoria Weißenberger, Diana Marjanovic, Katharina Sammet, Jan Vellmer, Farzan Banihashemi, and Peter Mandl. *EnergyPlus Room Simulator.* (2024). DOI: 10.48550/arXiv.2410.19888. arXiv: 2410.19888

1.5 Organization

The remainder of this dissertation is organized as follows. Chapter 2 introduces our methodology and the relevant background of this dissertation. Chapter 3 gives an overview of the publications that this thesis comprises, summarizing approaches, key insights, and the author's contributions. The full texts of these publications are attached to the Appendix. Chapter 4 discusses the work in the context of the existing related literature on building occupancy detection. In this regard, it also highlights the limitations of our work. Chapter 5 concludes the thesis and gives an outlook on future research.

2

Methodology

This chapter provides an overview of the relevant background and design considerations for this thesis and the work that it comprises. Section 2.1 differentiates occupancy detection from other related problem settings and defines the specific prediction problem addressed in this work. Section 2.2 gives an overview of environmental data with a focus on our considerations regarding sensing technology and data simulation. Section 2.3 introduces the high-level technical concepts employed in this work.

2.1 Occupancy Detection

In this thesis, occupancy detection refers to the prediction of current occupancy states within a building context. In general, we distinguish between three dimensions of building occupancy, including the spatial, temporal, and informational dimensions [53, 54].

Regarding the spatial dimension, this work addresses occupancy in a designated building space characterized as an individual room with regular occupation by a small number of occupants. In contrast to the assessment of whole-building occupation, this allows the use of occupation data for optimizing HVAC control in specific spaces. For analysis from a broader perspective, such as building performance estimation, data can be aggregated

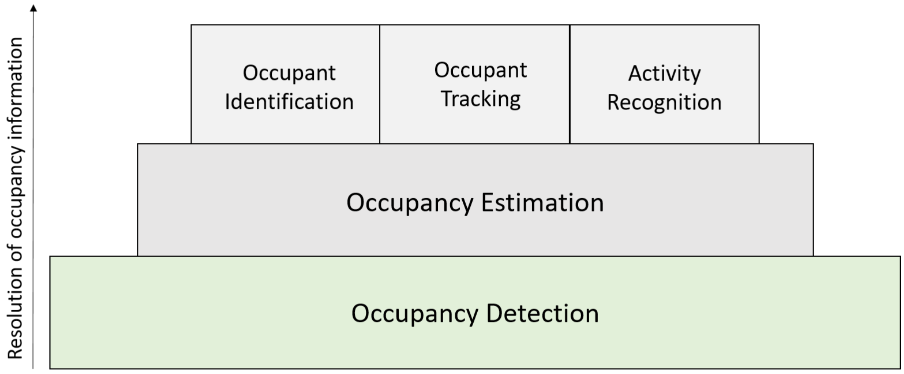

Figure 2.1.1: Problem settings for occupancy information of different resolutions

to larger sections of a building or to the building level. Room-specific occupancy data can also be used as an informative for room planning considerations.

Regarding the temporal dimension, we consider a near-real-time detection of occupancy on a per-minute basis. While related prediction problems involve the prediction of occupancy at a future time-step or occupancy profile learning, current occupancy state detection is the main focus of the literature on occupancy prediction [55] and can be used for both documenting occupancy and dynamic integration with HVAC control.

From the informational perspective, the concept of occupancy may refer to various types of information. Fig. 2.1.1 shows prediction problems commonly found in the literature [54, 56, 57] that relate to occupancy at varying levels of granularity. This work addresses occupancy detection, which predicts occupancy at the lowest possible resolution. The objective is to predict a binary occupancy state that indicates the presence of at least one individual or vacancy. A related problem is occupancy estimation, which aims to predict the degree of occupancy. In this case, occupancy is either categorized into different levels of occupation, such as high, low, or zero, or it is defined as the exact number of present occupants. The latter is also referred to as occupancy counting [57]. While occupancy estimation can especially be interesting in large rooms with a high number of expected occupants and an inhomogeneous climate, this work focuses on room types with few occupants, such as office or meeting rooms. In this context, the impact of variations in occupancy counts on indoor climate requirements is limited, and occupancy detection provides adequate data for the purpose of energy optimization.

Other problem settings address more personal aspects of occupancy, such as an occupant's identity, their trace of movement, or the activity performed. This work is set in the context of general optimization of energy use rather than personalization, thus avoiding privacy invasion. Compared to other types of occupancy data collection, occupancy detection is the least invasive choice.

We define occupancy detection as a binary time series classification problem with the objective of assigning a time series T of CO_2 values with a 1-minute resolution to a class $c \in \{0, 1\}$, where 0 represents vacancy and 1 represents presence at the end of the given time series. Throughout continuous classification, T is a sliding window derived from a continuous stream of measured CO_2 values and shifted by one minute for each successive classification iteration. With regard to this, we define a time series as follows: "A time series $T = [x_1, ..., x_n]$ 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." [47]

In this work, we focus on time steps Δt of one minute and on univariate time series with $x_i \in \mathbb{R}$, which represent consecutive measurements of the indoor CO_2 level. The work can be extended to other time steps or to multivariate time series, where each data point x_i consists of a multidimensional vector of input values, representing, for example, multiple environmental factors.

As the previous decisions are related to the type of data, more details on the input data design are provided in Section 2.2.

2.2 Indoor Environmental Data

In the context of this work, environmental data refers to data on the indoor climate within rooms that are subject to occupancy detection. Indoor climate comprises several environmental factors, including, among others, CO_2 rate, temperature, relative humidity, air pressure, air quality, or illuminance. In the following subsections, we address our considerations regarding data sensing and simulation.

2.2.1 Environmental Sensing

Environmental factors are measured by sensors that are often present in commercial buildings to provide data for the building management system (BMS). In research settings, they are typically measured with dedicated Internet of Things (IoT) devices, as in [4, 51, 58]. Besides environmental sensors, also other technologies may be used for occupancy detection. The literature discusses a variety of sensor types, including optical and thermal cameras, smart meters, acoustic sensors, or passive-infrared (PIR) motion sensors [7]. Each technology has its own specific advantages and drawbacks. With this work, we address the use of environmental sensors, more specifically CO_2 sensors, due to the following advantages:

- **Preservation of privacy**: Unlike, for example, camera-based systems, environmental sensing does not invade the occupants' privacy. It allows for occupancy detection largely without revealing identities, actions, or other personal information.
- **Location invariance**: While PIR sensors, optical, or thermal cameras rely on carefully considered placement and angle, and may have blind spots, environmental sensors can be placed almost arbitrarily within a room, unless it exceeds a critical room size. Climate changes will eventually permeate the entire space. Sensor distance from the occupants only influences the lag until changes are detected.
- **Cost efficiency**: Environmental sensors provide a cost advantage over more expensive hardware such as cameras. Moreover, in many buildings, they are already installed as part of the BMS.
- **Sustainability:** Compared to other technologies, such as cameras, environmental sensors operate with low power consumption. The Sensirion SCD30 sensor used in this work operates at an average current of 19 mA at 3.3-5.5 V [59]. This allows for continuous operation consuming less than 1 kWh per year and does not significantly impact the total power consumption of a building room.
- **Non-cumulative errors**: Sensing methods detecting state changes, such as light barriers at the doors, can lead to errors that are propagated throughout the rest of the day or until the next reset. In contrast, environmental sensing enables occupancy detection that is independent from previous errors.

- **Behavior independence:** Several sensing technologies rely on a specific type of occupant behavior as a proxy for presence. PIR sensors require movement, sound sensors depend on noise, and energy smart meters require the use of electric appliances. In contrast, environmental sensing is based on universal factors such as human respiration or body heat.

Environmental data are frequently used in studies on occupancy prediction, with CO_2 as a dominating factor, which is employed either as a single input or in combination with others [7, 8, 39]. Some examples of works that combine multiple environmental factors, such as CO_2, temperature, air pressure, and humidity, are [4, 19, 38]. A part of the literature, however, applies CO_2 as a single factor [6, 21, 23, 25, 60, 61]. CO_2 was found to be the most informative environmental factor in [62, 63, 64]. Although these findings cannot be generalized to all room scenarios, they underline the relevance of CO_2 data. A literature review by Rueda et al. points out CO_2 as the most commonly used factor in occupancy detection literature [7]. Adding additional factors, besides CO_2, in a multi-sensor approach may improve predictive performance, however, this does not hold in all situations and can also have a negative impact on model performance, as shown in [24]. As we found that accurate occupancy detection is possible with CO_2 as a single factor, and preferring lower model complexity, in this work, we only use CO_2 as input time series. This also fosters practical applicability, as only one sensor is required.

In this work, we focus on CO_2 measured through non-dispersive infrared (NDIR) technology. NDIR sensors provide high accuracy in their CO_2 measurements and have been widely used in related work [4, 19, 23, 60, 61]. CO_2 levels are informative as they are directly influenced by human respiration. However, the relationship between CO_2 levels and occupancy is non-linear and complex due to a variety of influencing factors, such as air exchanges with outdoor air or adjacent indoor spaces, room-specific properties, or the presence of plants. This requires non-linear models capable of capturing this complexity. Yet, time series of CO_2 measured within a building room allow inferring the room's occupancy state [5, 6] as indoor CO_2 concentration closely correlates with human presence [4], with a limited dependence on outdoor weather conditions. Outdoor CO_2 typically ranges from 300 to 500 parts per million (ppm), while the indoor CO_2 level during a period of occupancy is mostly above 500 ppm and can exceed 1000 ppm [65].

During occupancy, human respiration typically causes an increase in indoor CO_2, while during vacancy, CO_2 levels typically decline until they approach the outdoor level. Based on time series of CO_2 values measured over a certain period, this often allows for a straightforward classification in terms of occupancy. Still, as mentioned above, there are various disturbing factors that add additional complexity.

While we apply CO_2 as a single factor, we facilitate further extension of our work that includes additional environmental factors, such as temperature or humidity. In this sense, we apply established simulation software (see Section 2.2.2) that is capable of generating data for additional factors after modifying the simulation parameters. In addition to this, while optimizing models for prediction based on a single input time series, we intentionally focus on model types that are extendable to a multivariate time series setting by small changes in the model architecture (see Section 2.3.1).

2.2.2 Environmental Simulation

There is a variety of building performance simulation (BPS) tools developed to estimate a building's energy performance. Since energy consumption depends on indoor climate conditions, these tools also include the simulation of environmental factors. BPS tools used in building research include, among others, EnergyPlus, IDA ICE, TRNSYS, DOE-2, and DeST [66]. These tools perform discrete-time physical simulations of indoor climate based on a physical building model, provided weather data, and operational schedules. By varying these input factors, they allow the generation of diverse synthetic climate data. In this work, we use EnergyPlus [67], a simulation tool widely adopted in research and engineering. It is a suitable tool for our work for several reasons: EnergyPlus is an open-source tool that is regularly subject to further development. It encompasses a broad range of environmental aspects, including heat transfer, mass transfer, and illuminance [67]. In addition, it offers high extensibility through custom scripts. Although EnergyPlus is a whole-building simulator, it requires building models to be divided into zones, and simulation steps are calculated zone-wise, which allows the simulation of indoor environmental data for a specific room only. The simulation time steps and calculated environmental factors are customizable.

2.3 Deep Transfer Learning

This section briefly introduces and justifies the technical concepts employed in this work. More details on how these are implemented are provided in the publications that this dissertation comprises.

2.3.1 Deep Learning

We approach the problem of occupancy detection from CO_2 levels through deep learning techniques. The term deep learning refers to the use of artificial neural networks with more than one hidden layer. The choice of deep learning is motivated by several factors. Firstly, time series transfer learning literature is dominated by deep learning approaches and provides a wealth of related work [45]. Its successful application in various contexts motivates its exploration also regarding occupancy detection. Secondly, given sufficient capacity and suitable training data, deep learning models offer a universal approximation of any continuous function needed to solve a certain real-world problem. This ensures that our findings can be generalized to other contexts, also apart from occupancy detection and environmental data. Also, modifications to the input data, such as adding another environmental factor besides CO_2, can be made with low effort, as no manual feature engineering is necessary. Moreover, previous work has shown that deep learning outperforms other approaches in detecting occupancy: This was shown not only for environmental variables [19] but also other inputs, such as advanced metering infrastructure (AMI) data [68].

Despite offering highly accurate predictions when substantial training data is provided, deep learning models especially suffer from data scarcity, making our research particularly relevant. The lack of sufficient labeled training data is widely recognized as a major limitation of the application of deep learning, respectively, any data-driven methods, to occupancy detection [2, 7, 39, 69].

2.3.2 Transfer Learning

Transfer learning is a frequently employed concept to address the problem of data scarcity that limits deep learning applications. It is based on the general idea that training and testing data do not need to be drawn entirely from the same distribution. We use the definition according to Pan and Yang [28] that defines transfer learning as follows: "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 $T_S \neq T_T$." Our work addresses the situation where $T_S = T_T$ but $D_S \neq D_T$ in the sense that D_T refers to a real-world target room, while D_S refers to another room or a simulation. The idea is that providing enough general data for D_S will allow accurate predictions even if data for D_T are scarce. Both the generation of suitable source data and the exploration of specific transfer methods are research goals that we address in our publications [49, 50].

Our work is motivated by previous research that has already addressed transfer learning in the context of occupancy detection from environmental data by designing domain-specific transfer methods [35, 36, 37]. While these works rely on domain knowledge and apply calculations dedicated to the specific use case, we identified a research gap in the exploration of more generalizable approaches that remain applicable despite context changes, such as modifications to the input factors. In recent years, this has been addressed by our own work and by others applying similar approaches [38, 70].

Transfer learning can be categorized into different approaches, including informed or uninformed transfer [71]. Informed transfer relies on available labels for some target data, while uninformed transfer does not require any labels. In this work, we address supervised deep learning applications through informed transfer learning. More specifically, we apply informed, model-based transfer, focusing on adapting trained model parameters. According to our literature analysis, this approach is the main focus of the time series transfer learning literature [45] and it is particularly impeded by the lack of labeled training data.

2.3.3 Domain Randomization

Domain randomization is a technique that aims to train machine learning models without or without sufficient real-world training data [48]. The idea is to generate a diverse set of hypothetical samples in sufficient quantity to enable the model to effectively generalize to unseen real-world data. We use this technique in addition to transfer learning to further reduce the need for labeled training data. As indoor CO_2 depends primarily on the room and its occupants, we perform domain randomization by combining a variety of different room parameters and occupant behaviors from various simulations [50]. Domain randomization is not limited to the generation of samples that can naturally appear in the real world. It may be extended to the simulation of the complete physical space. However, we constrain the properties used in our simulations to reflect realistic scenarios, in order to keep data size and training time at manageable levels.

3

Summary of Publications

This thesis is based on four individual peer-reviewed and accepted publications. This chapter provides an overview of these publications, including descriptions of the approach, key achievements, and the author's contributions. The full-text version of each publication is included in the Appendix.

The four publications encompass a systematic literature mapping addressing time series transfer learning (Sec. 3.1), an empirical study of various transfer learning methods to reduce data requirements for occupancy detection (Sec. 3.2), the development of a deep learning base model trained in simulation (Sec. 3.3), and a preceding proof-of-concept for using simulated data to enhance model robustness and efficiency (Sec. 3.4).

3.1 Transfer Learning With Time Series Data: A Systematic Mapping Study

Reference: Manuel Weber, Maximilian Auch, Christoph Doblander, Peter Mandl, and Hans-Arno Jacobsen. "Transfer Learning With Time Series Data: A Systematic Mapping Study." In: *IEEE Access* 9 (2021), pp. 165409–165432. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3134628. URL: https://doi.org/10.1109/ACCESS.2021.3134628
(CORE PUBLICATION 1)

Full-text version enclosed: Appendix A

Summary: Transfer learning (TL) is a well-studied concept in the field of machine learning that relaxes the assumption that training and testing data need to be drawn from the same distribution. Recent success in applying TL in the area of computer vision has motivated TL research also in the 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 the literature on TL with time series data. We run electronic searches across nine scientific literature search databases and conduct a manual snowballing of identified publications. The selection of relevant literature employs a three-step filtering process that includes consecutive title, abstract, and full-text screening. In total, we identify and analyze 223 relevant publications. We provide a systematic literature mapping that addresses aspects such as applied methods and model types, as well as application fields. We describe the approaches pursued and point out trends. The results show a clear dominance of the model-based transfer approach, mostly based on deep learning techniques. Among the identified publications, the majority apply convolutional neural networks (CNNs), followed by long short-term memory networks (LSTMs) or a combination of both. Identified methods mostly involve retraining a model that was pretrained on labeled source data.

Author's contributions: Conceived, developed, and implemented the research approach adhering to established guidelines. Devised optimizations. Conducted literature search, selection, and content analysis. Wrote the paper.

3.2 Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection

Reference: Manuel Weber, Farzan Banihashemi, Peter Mandl, Hans-Arno Jacobsen, and Ruben Mayer. "Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection." In: *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2023), pp. 1–10. DOI: 10.1145/3600100.3623718. URL: https://doi.org/10.1145/3600100.3623718 **(CORE PUBLICATION 2)**

Full-text version enclosed: Appendix B

Summary: Knowing indoor occupancy states is crucial for energy optimization in buildings. While neural networks can effectively be used to detect occupancy based on carbon dioxide measurements, their application is impeded by the need for sufficient labeled training data. In this study, we analyze the prediction performance of three different transfer learning (TL) methods leveraging target room data jointly with data from other rooms. The methods include (1) pretraining and fine-tuning, (2) layer freezing, and (3) domain-adversarial learning. We use data collected from five real-world rooms and one simulated room, including multiple room types.

To the best of our knowledge, at the time of submission to the conference, the paper provided the most extensive evaluation of TL in the field of occupancy prediction from CO_2. This work's contribution further includes the architecture and hyperparameters of a deep CNN-LSTM model for CO_2-based occupancy detection.

Our results indicate that TL effectively reduces the required amount of target room data. Moreover, while previous literature was focused on pretraining with related real-world data, we show that similar performance can be achieved by the more practical approach of leveraging simulated data.

Author's contributions: Conceived, developed, and implemented the approach. Devised optimizations. Conducted analysis and experimental evaluation. Wrote the paper.

3.3 Coddora: CO2-Based Occupancy Detection Model Trained via Domain Randomization

Reference: Manuel Weber, Farzan Banihashemi, Davor Stjelja, Peter Mandl, Ruben Mayer, and Hans-Arno Jacobsen. "Coddora: CO2-Based Occupancy Detection Model Trained via Domain Randomization." In: *International Joint Conference on Neural Networks (IJCNN)* (2024), pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650820. URL: https://doi.org/10.1109/IJCNN60899.2024.10650820 **(CORE PUBLICATION 3)**

Full-text accepted version enclosed: Appendix C

Summary: While there has been a considerable amount of research on using neural networks to automatically detect occupancy from CO_2 sensors, their application in practice is limited due to the scarcity of labeled training data. In this paper, we propose Coddora (CO_2-based occupancy detection model trained via domain randomization), an off-the-shelf deep learning model pretrained on data from randomized room simulations. The model is trained on data resulting from two consecutive simulations, (1) an occupancy simulation generating presence times and window interactions for multiple occupants and (2) a physical simulation of the indoor climate considering a large variety of room models. A second model variation is provided by further retraining on measured real-world data. Our contribution includes the two pretrained model variants as well as the synthetic dataset providing data from simulations with 100,000 room models.
According to our results, Coddora enables efficient adaptation to real-world rooms, requiring only minimal data collection. In our evaluation, fine-tuning with 1-2 target room training days was enough to obtain a solid classifier with an accuracy above 90%. By substantially decreasing the necessary extent of ground truth data collection, compared to multiple work weeks that are required for successful training from scratch, this facilitates a fast adaptation of CO_2-based occupancy detection in practice. In addition to this, in an office room experiment, we show that for fine-tuning on scarce target data, Coddora outperforms a model pretrained on multiple weeks of real-world data from another room.

Author's contributions: Conceived, developed, and implemented the approach. Devised optimizations. Conducted analysis and experimental evaluation. Wrote the paper.

3.4 Detecting Building Occupancy with Synthetic Environmental Data

Reference: Manuel Weber, Christoph Doblander, and Peter Mandl. "Detecting Building Occupancy with Synthetic Environmental Data." In: *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2020), pp. 324–325. DOI: 10.1145/3408308.3431124. URL: https://doi.org/10.1145/3408308.3431124

Full-text version enclosed: Appendix D

Summary: Information about room-level occupancy is crucial to many building-related tasks, such as building automation or energy performance simulation. CO_2 levels and other indoor environmental factors can be used as a proxy to detect occupancy. In this regard, machine learning solutions have been proposed, with solid performance in detecting presence if enough training data is available. The challenge is, to collect sufficient room-specific ground truth data for model training.

In this work, we explore knowledge transfer from synthetic data to reduce the amount of required real-world data. We distinguish between two approaches for the combination of transfer learning with physical simulations, and motivate simulation-based generation of synthetic training data. We then conduct an experiment using data collected over multiple days from a CO_2 sensor in an office room, supplemented by simulated data. For this, we design a simulation method for occupant behavior and corresponding CO_2 levels, and apply it to simulate supplementary data for the experiment. Finally, we provide a proof-of-concept for the transfer from simulated CO_2 data.

Our results demonstrate that the required real-world training data can be reduced by around 50%. At the same time, we achieve improved model robustness. We conclude with an outline of future research implications.

Author's contributions: Conceived, developed, and implemented the approach. Devised optimizations. Conducted analysis and experimental evaluation. Wrote the paper.

4

Discussion

This chapter provides a discussion of our work in the larger context of building occupancy detection in general. We address related work on other approaches and conclude that our work simplifies the adaptation of supervised occupancy detection models while providing state-of-the-art prediction performance. In addition to this, we discuss the limitations of our work and opportunities to overcome them.

It is common for research on occupancy detection to propose accurate detection methods assuming the availability of labeled data (see [5, 19, 22, 63]). The goal is either to demonstrate the feasibility of occupancy detection with a certain technology or method, or to achieve high accuracy in the prediction. In contrast to the existing literature, we specifically address the aspect of data scarcity to enable occupancy detection without extensive effort on data collection. By initially introducing the idea of leveraging simulated data in the first published work of this dissertation [46] and further evaluating concrete approaches [49, 50], we set the path for more data-efficient occupancy detection systems.

Our work shows that the application of transfer learning is highly beneficial in scarce data scenarios, as it significantly increases prediction accuracy compared to training from scratch [49]. This is in line with the findings from related studies [35, 36, 37, 38, 70]. Previous works have proposed domain-specific transfer through predefined calculations based on domain knowledge [35, 36, 37]. These methods are limited in terms of generalizability and become inapplicable in the case of context changes, such as alterations to the input factors. Our work overcomes this limitation by implementing a fully data-driven approach. In that sense, it contributes to the growing body of data-driven research and reflects the broader trend in the field of occupant presence and behavior modeling, where the focus has shifted from analytical and knowledge-based methods to data-driven approaches over the last years [2, 7].

Other works have also addressed the problem by data-driven informed transfer learning between similar building spaces [38, 70]. While our work includes a more comprehensive evaluation [49], their results reinforce our finding that the approach is effective in the addressed context. Contrary results were found by Dridi et al. [72]. They apply uninformed transfer learning in a multi-sensor approach with six sensor types for activity recognition and occupancy estimation, and also evaluate their method towards binary detection. Compared to our work, the study reports rather small performance improvements over training from scratch. With regard to occupancy detection, they even report reduced accuracy for transfer learning with their top-performing method. The smaller effect may be due to the uninformed transfer approach, to lower data intensity in multi-sensor scenarios, or to the applied classification method based on decision trees.

Our work further shows that transfer learning from simulated datasets can perform on par with the transfer from collected real-world datasets [49] and that a wide variety of simulations can be used as a solid basis of data for effective pretraining [50]. To the best of our knowledge, prior to our research, transfer learning from simulated CO_2 data had not been addressed in related work. Since we first introduced this approach in [46, 47], it has been discussed in research both in the context of building occupancy detection [39, 73], as well as in other contexts [74, 75]. Sayed et al. [39] provide a literature review that highlights our approach and describes it in detail.

Prior to our research, data simulation had been applied primarily to evaluate methods for occupancy detection or estimation in simulated environments, such as in [64, 76, 77]. On the other hand, there is also research on the direct application of simulation-trained models. Parzinger et al. created a prediction model based on training data from a building simulation with the BPS tool IDA-ICE and tested it in a real-world scenario [64].

In contrast to our work, this assumes minimal variation between simulated and real-world data, which we do not consider a reliable basis for accurate modeling. They also conducted simulations in a real-world physical environment using a heat dummy to represent an artificial occupant [64]. Similarly, Jin et al. used a CO_2 pump in the studied room to emulate human breath [60]. While the physical simulation of occupants represents an alternative approach, we believe it is less practical than the simulation with BPS tools, as it requires even higher effort than collecting ground truth data in the field.

The prediction performance of our approach is comparable to that of state-of-the-art approaches in the literature [19, 20, 21, 22, 78]. In [50] we report possible accuracies of above 95%, where values around 94% are already reached with only two days of training data. Although accuracy values are not directly comparable to other studies due to different test datasets, we reference several results from the literature to give a general idea of reported performances.

Chen et al. report an accuracy of 95.42% using the same type of deep learning model as in our work but without transfer learning [19, 20]. This, however, is reached by collecting 26 full days of training data for four environmental factors. Our work reduces this effort to only a few days of CO_2 data.

In another multi-sensor approach using CO_2, temperature, humidity, and illuminance, Hitimana et al. report 96.7 to 96.8% accuracy using a multi-layer perceptron (MLP) or LSTM model trained on around 10 days of data [78]. Also Banihashemi et al. [27] report 95-97% accuracy in multi-sensor classification. In this work, CO_2, sound pressure, and illuminance are identified as the top-performing combination of input factors. It is also shown that when using only one input factor, a higher amount of labeled training data is required. This limitation is effectively addressed by our work.

Some studies also report accuracies similar to ours, or higher, with models of lower complexity. In an extensive study by Yang et al. on multiple machine learning models and sensor combinations, the best results were achieved with decision trees, yielding accuracies of 96% to 98.2% [62]. However, this result involves multiple weeks of training data for eleven sensor variables, which we do not consider practical. Using only CO_2, the reported results were substantially worse, with 66.36-89.69%. Reaching a performance above 95% accuracy, as in our work, required at least three sensor types (86.24%-95.21%).

Zhou et al. [22] and Jiang et al. [21] both report accuracies above 97% using only CO_2 data. However, these results are based on multiple work weeks of training data and broader prediction intervals of 10 min and 15 min, compared to per-minute prediction.

A certain branch of our related work relies on computer-vision techniques and cameras, which tend to provide higher accuracy. Zou et al. report an accuracy of 95.3% using image data from surveillance cameras [79], which is comparable to our results. However, Tien et al. reach 97.09-97.32% using camera images with a region-based convolutional neural network (R-CNN) [80]. In another study, they propose a vision-based deep transfer learning approach and report an average accuracy of 98.65% [81]. While cameras offer highly accurate detection, they raise privacy concerns and should only be used in public spaces such as hallways. This limitation is mitigated in our work, as CO_2 allows only minimal conclusions about individuals within a room. Different human CO_2 generation rates may allow for inferences about an occupant's gender or whether the person has recently engaged in physical activity. This, however, becomes increasingly difficult when more than one person is present in the room. Obtaining occupants' identities or information about their appearance or concrete activities is not possible.

An alternative that seeks to combine the benefits of both vision-based and privacy-preserving technologies is the use of thermal sensor arrays. These provide a resolution sufficiently low to prevent occupant identification. Metwaly et al. report an accuracy of 98.90% using low-resolution thermal sensor images [82]. However, beyond a certain resolution, they still allow to determine the occupants' locations and potentially their body heights. Moreover, detection may be misled by other heat sources such as electronic devices, and, like optical cameras, they are limited to the monitoring of a certain angle.

Given that the primary focus of occupancy detection research is on supervised methods [7], we address the lack of labeled data which is impeding supervised learning. Another approach is to fully eliminate the need for labeled training data by using unsupervised learning. Some related work has considered unsupervised solutions for occupancy detection. Ebadat et al. propose an unsupervised occupancy estimation method based on CO_2 and HVAC actuation data [77]. Habib and Zucker use k-means clustering on air quality data recorded by the ventilation system [83]. Primarily, works on unsupervised learning for predicting occupancy use energy consumption data from smart meters. Example works include SHARK [84], PresenceSense [85], or the works of Becker and Kleiminger [86] or Jin et al. [87]. Energy metering has a major limitation, which is that it only allows occupants to be detected when they use any electronic devices. It is not surprising that the reported accuracies are lower than in our work, ranging from 69% to 93% [84, 85, 86, 87]. Yet, when aiming for energy optimization, it may also be considered to accept a lower accuracy in exchange for eliminating the need to collect ground truth data and thereby enabling even faster implementation than our approach.

Our work reduces but does not fully eliminate the need for ground truth data collection, as a small amount of data is still required for adaptation to the target room. As we show in [45], informed model-based transfer learning using some labeled target data is the primary focus of transfer learning literature in the context of time series data. Our work aligns with this well-established methodology. In contrast to this, there are also approaches towards uninformed transfer learning. These mainly involve feature-based transfer, either by feature-transformation methods or by autoencoder-based feature learning [45]. Uninformed transfer and unsupervised learning present opportunities for future work to overcome the current requirement of labeled occupancy datasets.

Due to the limited distance between the locations where CO_2 is generated and where it is measured, occupancy detection systems using single CO_2 sensors are particularly well-suited for small to medium-sized rooms. The scope of our work can be expanded to larger spaces by introducing a concept that uses measurements from multiple CO_2 sensors simultaneously. Larger spaces have been studied in related work targeting, for example, a lecture hall [23], a movie theater [35, 36], an open-plan office space [88], or a subway station [89]. Our approach offers adaptability to differing domains, such as larger spaces, by altering the applied simulation or the deep learning model architecture. The model architecture can be modified to integrate inputs from multiple sensors. BPS tools, such as EnergyPlus, which we apply in this work, are able to simulate indoor climate across multiple modeled building zones. This allows the simulation to be adapted to different types of rooms or spaces. However, for large spaces occupied by crowds of people, the prediction objective may shift from accurate binary detection to the estimation of space utilization. In such cases, other technologies, such as Wi-Fi infrastructure, become increasingly relevant, even if they rely on specific behavior, such as the use of mobile devices, and do not always ensure accurate binary detection. In smaller rooms, such as office rooms, where utilization is limited to a small group or to single occupants, estimates of utilization levels are less useful, as necessary energy consumption is primarily driven by presence in general. Comfortable room climate must be maintained, regardless of whether the space is occupied by a single occupant or by multiple.

In addition to limitations with respect to room size, our work also has several other limitations related to the sensing technology. CO_2 sensors can only detect human-generated CO_2 after it has spread throughout the room. This makes real-time detection impossible. However, for analyzing long-term occupancy patterns that may help to optimize energy-consumption, real-time detection is not necessary. Instead, a rough identification of presence periods is sufficient. In retrospect, the time delay caused by the sensor distance can be estimated and applied to correct the time series of predicted values. This would lead to a further increase in accuracy. For automated HVAC control, the approach inherently involves a brief delay in response time after occupants enter or leave the room.

Another point is that CO_2 sensors are vulnerable to window openings as the indoor climate is disrupted by the influx of fresh air. This may especially be problematic in summer, when high temperatures lead to windows being left open permanently. Yet, in such situations, there is no need for heating, automated ventilation, or air conditioning, which makes accurate occupancy detection with the goal of energy optimization temporarily irrelevant. While we have only addressed naturally ventilated rooms in this work, mechanical ventilation can interfere with CO_2-based occupancy detection by continuously stabilizing the indoor CO_2 level. Despite that, if there are no detectable variations in the breathable zone of the room, CO_2 sensors can also be placed at the ventilation exhaust [90].

Overall, our work addresses the challenge of occupancy detection with minimal labeled data through data simulation and transfer learning, achieving state-of-the-art accuracy while being privacy-preserving and adaptable. Future work may address real-time detection and expand applicability by exploring uninformed transfer learning, unsupervised methods, or alternative sensing technologies.

5

Conclusions

This work provides a comprehensive investigation of transfer learning in the context of CO_2-based occupancy detection in buildings. It covers four publications exploring methods to overcome data scarcity while still providing accurate detection ability through the use of both real-world and simulated source data. Specifically, we advance the discourse on transfer learning-based occupancy detection by proposing the integration of simulations as an alternative to high-effort data collection.

First, we present a proof-of-concept for using simulations to enhance occupancy detection, showing that pretraining on simulated data can effectively improve model training with real-world target data. In an office room scenario, we demonstrate that this approach is able to significantly reduce the required amount of real-world data, while at the same time improving the robustness of the model.

We then systematically map the literature on transfer learning with time series data, identifying trends and dominant approaches. We find that the main focus of the literature is on deep learning and model-based transfer, primarily via pretraining and fine-tuning.

In a subsequent study, we focus on the application of transfer learning specifically in the area of CO_2-based occupancy detection, demonstrating the effectiveness of different methods. The work gives insights into different scenarios and highlights the ability of transfer learning to reduce the amount of labeled data required from the target room to only a few days, instead of weeks.

Finally, we introduce Coddora, our occupancy detection model trained via domain randomization. We show its ability to adapt to real-world environments with minimal additional training, requiring no source data collection and only 1 to 2 days of target data.

Collectively, the papers included in this work contribute to the advancement of building occupancy detection based on CO_2 or other environmental factors. As a result, our work contributes to more informed solutions for building energy optimization and building automation. Combining the insights of our work, we conclude that transfer learning makes effective use of scarce occupancy labels and fosters the practical applicability of CO_2-based occupancy detection by facilitating simplified deployment, particularly in the existing stock of buildings where CO_2 sensors are already in place.

Future research opportunities to extend our work include the exploration of unsupervised learning methods or uninformed transfer learning to potentially further reduce data collection requirements from minimal to zero. Another opportunity is to explore the application of similar approaches in settings involving other sensing technologies or, where appropriate, multiple sensor inputs. However, unlike research focused on occupancy detection in large multi-sensor environments, it is crucial to prioritize practicality in a wide range of contemporary buildings.

Bibliography

[1] Tianzhen Hong, Zhe Wang, Xuan Luo, and Wanni Zhang. "State-of-the-art on research and applications of machine learning in the building life cycle." In: *Energy and Buildings* 212.109831 (2020). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2020.109831.

[2] William O'Brien, Andreas Wagner, Marcel Schweiker, et al. "Introducing IEA EBC annex 79: Key challenges and opportunities in the field of occupant-centric building design and operation." In: *Building and Environment* 178.106738 (2020). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2020.106738.

[3] Bing Dong, Vishnu Prakash, Fan Feng, and Zheng O'Neill. "A review of smart building sensing system for better indoor environment control." In: *Energy and Buildings* 199 (2019), pp. 29–46. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2019.06.025.

[4] Luis M. Candanedo and Véronique Feldheim. "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models." In: *Energy and Buildings* 112 (2016), pp. 28–39. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2015.11.071.

[5] Mustafa K. Masood, Yeng Chai Soh, and Victor W.-C. Chang. "Real-time occupancy estimation using environmental parameters." In: *International Joint Conference on Neural Networks (IJCNN)* (2015), pp. 1–8. DOI: 10.1109/IJCNN.2015.7280781.

[6] Yixuan Wei, Shu Wang, Longzhe Jin, Yifei Xu, and Tianqi Ding. "Indoor occupancy estimation from carbon dioxide concentration using parameter estimation algorithms." In: *Building Services Engineering Research and Technology* 43.4 (2022), pp. 419–438. ISSN: 0143-6244. DOI: 10.1177/01436244211060903.

[7] Luis Rueda, Kodjo Agbossou, Alben Cardenas, Nilson Henao, and Sousso Kelouwani. "A comprehensive review of approaches to building occupancy detection." In: *Building and Environment* 180.106966 (2020). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2020.106966.

[8] Zhenghua Chen, Chaoyang Jiang, and Lihua Xie. "Building occupancy estimation and detection: A review." In: *Energy and Buildings* 169 (2018), pp. 260–270. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2018.03.084.

[9] IEA. *World Energy Outlook 2023 [Online]*. URL: www.iea.org/reports/world-energy-outlook-2023 (visited on 04/30/2024).

[10] European Commission. “The European Green Deal.” In: *COM/2019/640 final* (2019). URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2019%3A640%3AFIN.

[11] European Climate Foundation. *BUILDING EUROPE’S NET-ZERO FUTURE [Online]*. 2022. URL: https://europeanclimate.org/wp-content/uploads/2022/03/ecf-building-emissions-problem-march2022.pdf (visited on 08/19/2024).

[12] European Parliament. “Regulation (EU) 2021/1119 of the European Parliament and of the Council of 30 June 2021 establishing the framework for achieving climate neutrality and amending Regulations (EC) No 401/2009 and (EU) 2018/1999 (‘European Climate Law’).” In: *Official Journal of the European Union* L 243 (2021), pp. 1–17. ISSN: 1977-0677. URL: http://data.europa.eu/eli/reg/2021/1119/oj.

[13] Jonathan. Brooks and Prabir Barooah. “Energy-efficient control of under-actuated HVAC zones in buildings.” In: *American Control Conference* (2014), pp. 424–429. DOI: 10.1109/ACC.2014.6859151.

[14] Jonathan Brooks, Siddharth Goyal, Rahul Subramany, Yashen Lin, Timothy Middelkoop, Laura Arpan, Luca Carloni, and Prabir Barooah. “An experimental investigation of occupancy-based energy-efficient control of commercial building indoor climate.” In: *IEEE Conference on Decision and Control (CDC)* 53 (2014), pp. 5680–5685. DOI: 10.1109/CDC.2014.7040278.

[15] Mohammad Esrafilian-Najafabadi and Fariborz Haghighat. “Occupancy-based HVAC control systems in buildings: A state-of-the-art review.” In: *Building and Environment* 197.107810 (2021). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2021.107810.

[16] Kathryn B. Janda. “Buildings don’t use energy: people do.” In: *Architectural Science Review* 54.1 (2011), pp. 15–22. ISSN: 0003-8628. DOI: 10.3763/asre.2009.0050.

[17] Chris van Dronkelaar, Mark Dowson, E. Burman, Catalina Spataru, and Dejan Mumovic. “Corrigendum: A Review of the Energy Performance Gap and Its Underlying Causes in Non-Domestic Buildings.” In: *Frontiers in Mechanical Engineering* 2 (2016). ISSN: 2297-3079. DOI: 10.3389/fmech.2016.00010.

[18] Farzan Banihashemi, Manuel Weber, Bing Dong, Salvatore Carlucci, Roland Reitberger, and Werner Lang. “Window state or action modeling? An explainable AI approach in offices.” In: *Energy and Buildings* 298.113546 (2023). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2023.113546.

[19] Zhenghua Chen, Rui Zhao, Qingchang Zhu, Mustafa K. Masood, Yeng Chai Soh, and Kezhi Mao. “Building Occupancy Estimation with Environmental Sensors via CDBLSTM.” In: *IEEE Transactions on Industrial Electronics* 64.12 (2017), pp. 9549–9559. ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2711530.

[20] Zhenghua Chen, Chaoyang Jiang, Mustafa K. Masood, Yeng Chai Soh, Min Wu, and Xiaoli Li. “Deep Learning for Building Occupancy Estimation Using Environmental Sensors.” In: *Deep Learning: Algorithms and Applications*. Ed. by Witold Pedrycz and Shyi-Ming Chen. Vol. 865. Studies in Computational Intelligence. Cham: Springer, 2020, pp. 335–357. ISBN: 978-3-030-31760-7. DOI: 10.1007/978-3-030-31760-7_11. URL: https://doi.org/10.1007/978-3-030-31760-7_11.

[21] Chaoyang Jiang, Zhenghua Chen, Rong Su, Mustafa Khalid Masood, and Yeng Chai Soh. “Bayesian filtering for building occupancy estimation from carbon dioxide concentration.” In: *Energy and Buildings* 206.109566 (2020). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2019.109566.

[22] Yaping Zhou, Jiayu Chen, Zhun Jerry Yu, Jun Li, Gongsheng Huang, Fariborz Haghighat, and Guoqiang Zhang. “A novel model based on multi-grained cascade forests with wavelet denoising for indoor occupancy estimation.” In: *Building and Environment* 167.106461 (2020). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2019.106461.

[23] M. S. Zuraimi, A. Pantazaras, K. A. Chaturvedi, J. J. Yang, K. W. Tham, and S. E. Lee. “Predicting occupancy counts using physical and statistical Co2-based modeling methodologies.” In: *Building and Environment* 123 (2017), pp. 517–528. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2017.07.027.

[24] Pawalai Kraipeerapun and Somkid Amornsamankul. “Room Occupancy Detection using Modified Stacking.” In: *Proceedings of the 9th International Conference on Machine Learning and Computing (ICMLC)* (2017), pp. 162–166. DOI: 10.1145/3055635.3056597.

[25] Davide Calì, Peter Matthes, Kristian Huchtemann, Rita Streblow, and Dirk Müller. “CO2 based occupancy detection algorithm: Experimental analysis and validation for office and residential buildings.” In: *Building and Environment* 86 (2015), pp. 39–49. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2014.12.011.

[26] Ramoni Adeogun, Ignacio Rodriguez, Mohammad Razzaghpour, Gilberto Berardinelli, Per Hartmann Christensen, and Preben Elgaard Mogensen. “Indoor Occupancy Detection and Estimation using Machine Learning and Measurements from an IoT LoRa-based Monitoring System.” In: *Global IoT Summit (GIoTS)* (2019), pp. 1–5. DOI: 10.1109/GIOTS.2019.8766374.

[27] Farzan Banihashemi, Manuel Weber, Fatma Deghim, Chujun Zong, and Werner Lang. “Occupancy modeling on non-intrusive indoor environmental data through machine learning.” In: *Building and Environment* 254.111382 (2024). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2024.111382.

[28] Sinno Jialin Pan and Qiang Yang. “A Survey on Transfer Learning.” In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359. ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191.

[29] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. “Transfer learning for time series classification.” In: *IEEE International Conference on Big Data (Big Data)* (2018), pp. 1367–1376. DOI: 10.1109/BigData.2018.8621990.

[30] Debrup Banerjee, Kazi Islam, Keyi Xue, Gang Mei, Lemin Xiao, Guangfan Zhang, Roger Xu, Cai Lei, Shuiwang Ji, and Jiang Li. “A deep transfer learning approach for improved post-traumatic stress disorder diagnosis.” In: *Knowledge and Information Systems* 60 (2019), pp. 1693–1724. ISSN: 0219-3116. DOI: 10.1007/s10115-019-01337-2.

[31] Dengyu Xiao, Yixiang Huang, Lujie Zhao, Chengjin Qin, Haotian Shi, and Chengliang Liu. “Domain Adaptive Motor Fault Diagnosis Using Deep Transfer Learning.” In: *IEEE Access* 7 (2019), pp. 80937–80949. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2921480.

[32] Renjie Ding, Xue Li, Lanshun Nie, Jiazhen Li, Xiandong Si, Dianhui Chu, Guozhong Liu, and Dechen Zhan. “Empirical study and improvement on deep transfer learning for human activity recognition.” In: *Sensors* 19.57 (2018). DOI: 10.3390/s19010057.

[33] Priyanka Gupta, Pankaj Malhotra, Jyoti Narwariya, Lovekesh Vig, and Gautam Shroff. “Transfer learning for clinical time series analysis using deep neural networks.” In: *Journal of Healthcare Informatics Research* 4.2 (2020), pp. 112–137. ISSN: 2509-4971. DOI: 10.1007/s41666-019-00062-3.

[34] Qi-Qiao He, Patrick Cheong-Iao Pang, and Yain-Whar Si. “Transfer Learning for Financial Time Series Forecasting.” In: *Pacific Rim International Conference on Artificial Intelligence (PRICAI)* (2019), pp. 24–36. DOI: 10.1007/978-3-030-29911-8_3.

[35] Irvan B. Arief-Ang, Flora D. Salim, and Margaret Hamilton. “DA-HOC: Semi-Supervised Domain Adaptation for Room Occupancy Prediction using CO2 Sensor Data.” In: *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys)* (2017), pp. 1–10. DOI: 10.1145/3137133.3137146.

[36] Irvan B. Arief-Ang, Margaret Hamilton, and Flora D. Salim. “A Scalable Room Occupancy Prediction with Transferable Time Series Decomposition of CO2 Sensor Data.” In: *ACM Transactions on Sensor Networks (TOSN)* 14.3-4 (2018), pp. 1–28. ISSN: 1550-4859. DOI: 10.1145/3217214.

[37] Tianyu Zhang and Omid Ardakanian. “A domain adaptation technique for fine-grained occupancy estimation in commercial buildings.” In: *Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI)* (2019), pp. 148–159. DOI: 10.1145/3302505.3310077.

[38] Mohamad Khalil, Stephen McGough, Zoya Pourmirza, Mehdi Pazhoohesh, and Sara Walker. “Transfer Learning Approach for Occupancy Prediction in Smart Buildings.” In: *12th International Renewable Engineering Conference (IREC)* (2021), pp. 1–6. DOI: 10.1109/IREC51415.2021.9427869.

[39] Aya Nabil Sayed, Yassine Himeur, and Faycal Bensaali. “Deep and transfer learning for building occupancy detection: A review and comparative analysis.” In: *Engineering Applications of Artificial Intelligence* 115.105254 (2022). ISSN: 0952-1976. DOI: 10.1016/j.engappai.2022.105254.

[40] Krzysztof Arendt, Aslak Johansen, Bo Nørregaard Jørgensen, Mikkel Baun Kjærgaard, Claudio Giovanni Mattera, Fisayo Caleb Sangogboye, Jens Hjort Schwee, and Christian T. Veje. “Room-level occupant counts, airflow and CO 2 data from an office building.” In: *Proceedings of the First Workshop on Data Acquisition To Analysis (DATA)* (2018), pp. 13–14. DOI: 10.1145/3277868.3277875.

[41] Zeynep Duygu Tekler, Eikichi Ono, Yuzhen Peng, Sicheng Zhan, Bertrand Lasternas, and Adrian Chong. “ROBOD, room-level occupancy and building operation dataset.” In: *Building Simulation* 15 (2022), pp. 2127–2137. ISSN: 1996-8744. DOI: 10.1007/s12273-022-0925-9.

[42] Marcel Schweiker, Michael Kleber, and Andreas Wagner. “Long-term monitoring data from a naturally ventilated office building.” In: *Scientific Data* 6.293 (2019). ISSN: 2052-4463. DOI: 10.1038/s41597-019-0283-3.

[43] Simon Pommerencke Melgaard, Hicham Johra, Victor Ørsøe Nyborg, et al. “Detailed operational building data for six office rooms in Denmark: Occupancy, indoor environment, heating, ventilation, lighting and room control monitoring with sub-hourly temporal resolution.” In: *Data in Brief* 54.110326 (2024). ISSN: 2352-3409. DOI: 10.1016/j.dib.2024.110326.

[44] Barbara A. Kitchenham and Stuart M. Charters. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Tech. rep. EBSE-2007-01, Keele University. 2007.

[45] Manuel Weber, Maximilian Auch, Christoph Doblander, Peter Mandl, and Hans-Arno Jacobsen. "Transfer Learning With Time Series Data: A Systematic Mapping Study." In: *IEEE Access* 9 (2021), pp. 165409–165432. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3134628. URL: https://doi.org/10.1109/ACCESS.2021.3134628.

[46] Manuel Weber, Christoph Doblander, and Peter Mandl. "Detecting Building Occupancy with Synthetic Environmental Data." In: *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2020), pp. 324–325. DOI: 10.1145/3408308.3431124. URL: https://doi.org/10.1145/3408308.3431124.

[47] Manuel Weber, Christoph Doblander, and Peter Mandl. *Towards the Detection of Building Occupancy with Synthetic Environmental Data.* (2020). DOI: 10.48550/arXiv.2010.04209. arXiv: 2010.04209.

[48] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2017). DOI: 10.1109/IROS.2017.8202133.

[49] Manuel Weber, Farzan Banihashemi, Peter Mandl, Hans-Arno Jacobsen, and Ruben Mayer. "Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection." In: *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2023), pp. 1–10. DOI: 10.1145/3600100.3623718. URL: https://doi.org/10.1145/3600100.3623718.

[50] Manuel Weber, Farzan Banihashemi, Davor Stjelja, Peter Mandl, Ruben Mayer, and Hans-Arno Jacobsen. "Coddora: CO2-Based Occupancy Detection Model Trained via Domain Randomization." In: *International Joint Conference on Neural Networks (IJCNN)* (2024), pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650820. URL: https://doi.org/10.1109/IJCNN60899.2024.10650820.

[51] Farzan Banihashemi, Manuel Weber, and Werner Lang. "Deep learning for predictive window operation modeling in open-plan offices." In: *Energy and Buildings* 310.114109 (2024). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2024.114109.

[52] Manuel Weber, Philipp Bogdain, Sophia Viktoria Weißenberger, Diana Marjanovic, Katharina Sammet, Jan Vellmer, Farzan Banihashemi, and Peter Mandl. *EnergyPlus Room Simulator.* (2024). DOI: 10.48550/arXiv.2410.19888. arXiv: 2410.19888.

[53] Ryan Melfi, Ben Rosenblum, Bruce Nordman, and Ken Christensen. "Measuring building occupancy using existing network infrastructure." In: *International Green Computing Conference* (2011), pp. 1–8. DOI: 10.1109/IGCC.2011.6008560.

[54] Mikkel Baun Kjærgaard and Fisayo Caleb Sangogboye. "Categorization framework and survey of occupancy sensing systems." In: *Pervasive and Mobile Computing* 38 (2017), pp. 1–13. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2016.09.019.

[55] Xilei Dai, Junjie Liu, and Xin Zhang. "A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings." In: *Energy and Buildings* 223.110159 (2020). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2020.110159.

[56] Preethi K. Mane and K. Narasimha Rao. "Review of Research Progress, Trends and Gap in Occupancy Sensing for Sophisticated Sensory Operation." In: *Cybernetics and Algorithms in Intelligent Systems*. Ed. by Radek Silhavy. Cham: Springer International Publishing, 2019, pp. 212–222. ISBN: 978-3-319-91192-2.

[57] Kailai Sun, Qianchuan Zhao, and Jianhong Zou. "A review of building occupancy measurement systems." In: *Energy and Buildings* 216.109965 (2020). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2020.109965.

[58] Yunwan Jeon, Chanho Cho, Jongwoo Seo, Kyunglag Kwon, Hansaem Park, Seungkeun Oh, and In-Jeong Chung. "IoT-based occupancy detection system in indoor residential environments." In: *Building and Environment* 132 (2018), pp. 181–204. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2018.01.043.

[59] Sensirion AG. *Datasheet Sensirion SCD30 Sensor Module [Online]*. URL: https://sensirion.com/media/documents/4EAF6AF8/61652C3C/Sensirion_CO2_Sensors_SCD30_Datasheet.pdf (visited on 08/19/2024).

[60] Ming Jin, Nikolaos Bekiaris-Liberis, Kevin Weekly, Costas J. Spanos, and Alexandre M. Bayen. "Occupancy Detection via Environmental Sensing." In: *IEEE Transactions on Automation Science and Engineering* 15.2 (2018), pp. 443–455. ISSN: 1545-5955. DOI: 10.1109/TASE.2016.2619720.

[61] Hwataik Han, Kyung-jin Jang, Changho Han, and Junyong Lee. "Occupancy estimation based on CO2 concentration using dynamic neural network model." In: *Proceedings of the 34th AIVC - 3rd TightVent - 2nd Cool Roofs' - 1st venticool Conference* (2013).

[62] Zheng Yang, Nan Li, Burcin Becerik-Gerber, and Michael Orosz. "A systematic approach to occupancy modeling in ambient sensor-rich buildings." In: *SIMULATION* 90.8 (2013), pp. 960–977. ISSN: 0037-5497. DOI: 10.1177/0037549713489918.

[63] Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. "A methodology based on Hidden Markov Models for occupancy detection and a case study in a low energy residential building." In: *Energy and Buildings* 148 (2017), pp. 327–341. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2017.05.031.

[64] Michael Parzinger, Lucia Hanfstaengl, Ferdinand Sigg, Uli Spindler, Ulrich Wellisch, and Markus Wirnsberger. "Comparison of different training data sets from simulation and experimental measurement with artificial users for occupancy detection — Using machine learning methods Random Forest and LASSO." In: *Building and Environment* 223.109313 (2022). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2022.109313.

[65] ANSI/ASHRAE. *ANSI/ASHRAE Standard 62.1-2022: Ventilation and Acceptable Indoor Air Quality*. Atlanta, Ga.: ASHRAE, 2022.

[66] Tianzhen Hong, Yixing Chen, Zsofia Belafi, and Simona D'Oca. "Occupant behavior models: A critical review of implementation and representation approaches in building performance simulation programs." In: *Building Simulation* 11.1 (2018), pp. 1–14. ISSN: 1996-3599. DOI: 10.1007/s12273-017-0396-6.

[67] U.S. Department of Energy. *EnergyPlus [Online].* URL: https://energyplus.net/ (visited on 09/09/2024).

[68] Cong Feng, Ali Mehmani, and Jie Zhang. "Deep learning-based real-time building occupancy detection using AMI data." In: *IEEE Transactions on Smart Grid* 11.5 (2020), pp. 4490–4501. ISSN: 1949-3053. DOI: 10.1109/TSG.2020.2982351.

[69] Pratiksha Chaudhari, Yang Xiao, Mark M. Cheng, and Tieshan Li. "Fundamentals, Algorithms, and Technologies of Occupancy Detection for Smart Buildings Using IoT Sensors." In: *Sensors* 24.7 (2024). DOI: 10.3390/s24072123.

[70] Davor Stjelja, Juha Jokisalo, and Risto Kosonen. "Scalable Room Occupancy Prediction with Deep Transfer Learning Using Indoor Climate Sensor." In: *Energies* 15.6 (2022). DOI: 10.3390/en15062078.

[71] Diane Cook, Kyle D. Feuz, and Narayanan C. Krishnan. "Transfer learning for activity recognition: a survey." In: *Knowledge and Information Systems* 36 (2013), pp. 537–556. ISSN: 0219-3116. DOI: 10.1007/s10115-013-0665-3.

[72] Jawher Dridi, Manar Amayri, and Nizar Bouguila. "Transfer learning for estimating occupancy and recognizing activities in smart buildings." In: *Building and Environment* 217.109057 (2022). ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2022.109057.

[73] Oumayma Dalhoumi, Manar Amayri, and Nizar Bouguila. "A Review of Neural Networks for Buildings Occupancy Measurement." In: *IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)* (2022), pp. 29–35. DOI: 10.1109/IAICT55358.2022.9887508.

[74] Tala Talaei Khoei, Hadjar Ould Slimane, and Naima Kaabouch. "Deep learning: systematic review, models, challenges, and research directions." In: *Neural Computing and Applications* 35 (2023), pp. 23103–23124. ISSN: 1433-3058. DOI: 10.1007/s00521-023-08957-4.

[75] Peng Yan, Ahmed Abdulkadir, Paul-Philipp Luley, Matthias Rosenthal, Gerrit A. Schatte, Benjamin F. Grewe, and Thilo Stadelmann. "A Comprehensive Survey of Deep Transfer Learning for Anomaly Detection in Industrial Time Series: Methods, Applications, and Directions." In: *IEEE Access* 12 (2024), pp. 3768–3789. DOI: 10.1109/ACCESS.2023.3349132.

[76] Simon Breslav, Rhys Goldstein, Ben Doherty, Dan Rumery, and Azam Khan. "Simulating the sensing of building occupancy." In: *Proceedings of the Symposium on Simulation for Architecture & Urban Design (SimAUD)* 15 (2013), pp. 1–8.

[77] A. Ebadat, G. Bottegal, D. Varagnolo, B. Wahlberg, H. Hjalmarsson, and K. H. Johansson. "Blind identification strategies for room occupancy estimation." In: *European Control Conference (ECC)* (2015), pp. 1315–1320. DOI: 10.1109/ECC.2015.7330720.

[78] Eric Hitimana, Gaurav Bajpai, Richard Musabe, Louis Sibomana, and Jayavel Kayalvizhi. "Implementation of IoT Framework with Data Analysis Using Deep Learning Methods for Occupancy Prediction in a Building." In: *Future Internet* 13.3 (2021). ISSN: 1999-5903. DOI: 10.3390/fi13030067.

[79] Jianhong Zou, Qianchuan Zhao, Wen Yang, and Fulin Wang. "Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation." In: *Energy and Buildings* 152 (2017), pp. 385–398. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2017.07.064.

[80] Paige Wenbin Tien, Shuangyu Wei, and John Calautit. "A Computer Vision-Based Occupancy and Equipment Usage Detection Approach for Reducing Building Energy Demand." In: *Energies* 14.1 (2021). DOI: 10.3390/en14010156.

[81] Paige Wenbin Tien, Shuangyu Wei, John Kaiser Calautit, Jo Darkwa, and Christopher Wood. "Vision-based human activity recognition for reducing building energy demand." In: *Building Services Engineering Research and Technology* 42.6 (2021), pp. 691–713. ISSN: 0143-6244. DOI: 10.1177/01436244211026120.

[82] Aly Metwaly, Jorge Peña Queralta, Victor Kathan Sarker, Tuan Nguyen Gia, Omar Nasir, and Tomi Westerlund. "Edge Computing with Embedded AI: Thermal Image Analysis for Occupancy Estimation in Intelligent Buildings." In: *Proceedings of the INTelligent Embedded Systems Architectures and Applications Workshop 2019 (INTESA)* (2020), pp. 1–6. DOI: 10.1145/3372394.3372397.

[83] Usman Habib and Gerhard Zucker. "Automatic occupancy prediction using unsupervised learning in buildings data." In: *IEEE International Symposium on Industrial Electronics (ISIE)* (2017), pp. 1471–1476. DOI: 10.1109/ISIE.2017.8001463.

[84] Guoming Tang, Kui Wu, Jingsheng Lei, and Weidong Xiao. "The meter tells you are at home! Non-intrusive occupancy detection via load curve data." In: *IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2015), pp. 897–902. DOI: 10.1109/SmartGridComm.2015.7436415.

[85] Ming Jin, Ruoxi Jia, Zhaoyi Kang, Ioannis C. Konstantakopoulos, and Costas J. Spanos. "PresenceSense: zero-training algorithm for individual presence detection based on power monitoring." In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys)* (2014), pp. 1–10. DOI: 10.1145/2674061.2674073.

[86] Vincent Becker and Wilhelm Kleiminger. "Exploring zero-training algorithms for occupancy detection based on smart meter measurements." In: *Computer Science - Research and Development* 33 (2017), pp. 25–36. DOI: 10.1007/s00450-017-0344-9.

[87] Ming Jin, Ruoxi Jia, and Costas J. Spanos. "Virtual Occupancy Sensing: Using Smart Meters to Indicate Your Presence." In: *IEEE Transactions on Mobile Computing* 16.11 (2017), pp. 3264–3277. ISSN: 1536-1233. DOI: 10.1109/TMC.2017.2684806.

[88] Poh Lam Khee, Höynck Michael, Dong Bing, Andrews Burton, Chiou Yun-Shang, Zhang Rui, Benitez Diego, and Choi Joonho. "Occupancy detection through an extensive environmental sensor network in an open-plan office building." In: *Proceedings of Building Simulation 2009: 11th Conference of IBPSA* (2009), pp. 1452–1459.

[89] Cheolyong Shin and Hwataik Han. "Occupancy estimation in a subway station using Bayesian simulation based on carbon dioxide and particle concentrations." In: *International Journal of Mechanical Systems Engineering* 1.106 (2015). ISSN: 2455-7412. DOI: 10.15344/2455-7412/2015/106.

[90] Gen Pei, Donghyun Rim, Stefano Schiavon, and Matthew Vannucci. "Effect of sensor position on the performance of CO2-based demand controlled ventilation." In: *Energy and Buildings* 202.109358 (2019). ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2019.109358.

Appendix A

Transfer Learning With Time Series Data: A Systematic Mapping Study

Reference:

Manuel Weber, Maximilian Auch, Christoph Doblander, Peter Mandl, and Hans-Arno Jacobsen. "Transfer Learning With Time Series Data: A Systematic Mapping Study." In: *IEEE Access* 9 (2021), pp. 165409–165432. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3134628. URL: https://doi.org/10.1109/ACCESS.2021.3134628

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 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.

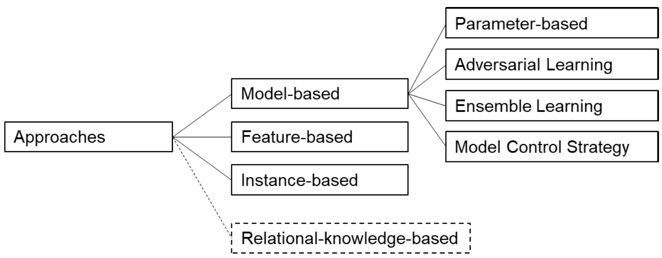

FIGURE 1. TL solution approaches based on [24], [25], [27]. The dashed line denotes inapplicability to time series.

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

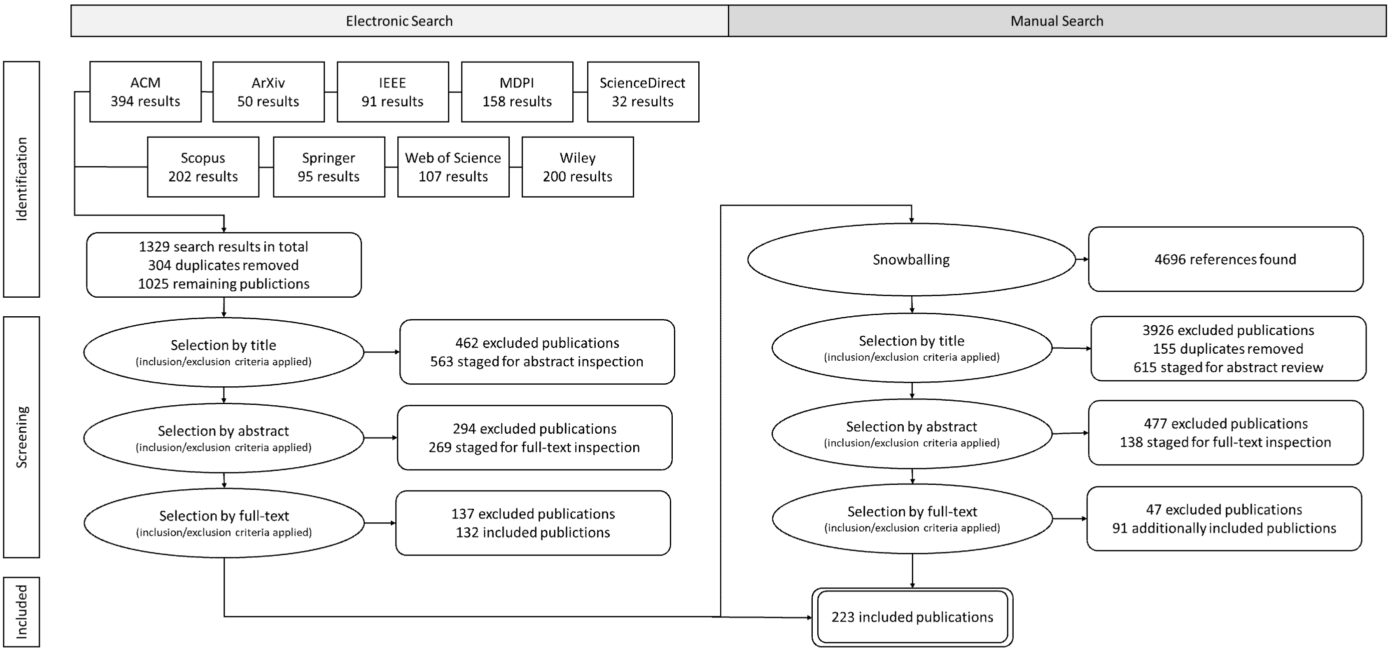

FIGURE 2. **Selection process shown as PRISMA flow diagram [43].**

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.

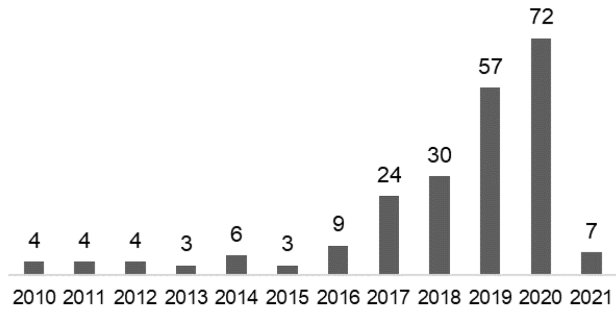

FIGURE 3. **Number of included publications by year of publication.**

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

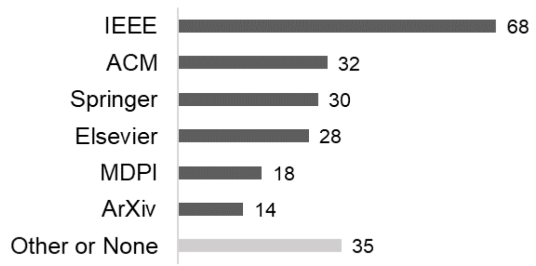

FIGURE 4. Number of included publications by publisher.

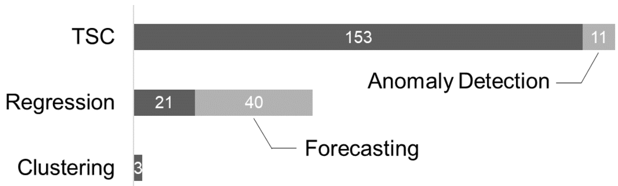

FIGURE 5. Number of included publications by time series problem.

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

[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.

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

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

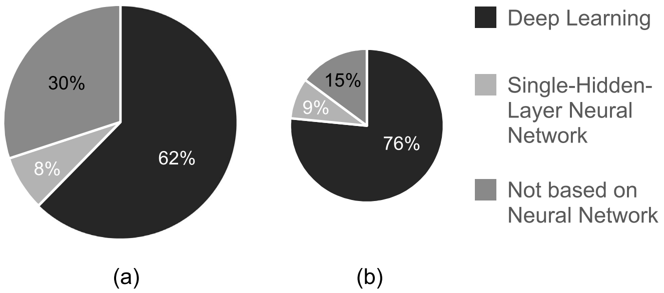

FIGURE 6. Fraction of publications applying deep learning methods, or a neural network, for (a) all included publications and (b) publications on model-based TL. If the method applies to a generic model and the evaluation involves at least one other than a neural network, the publication is counted as 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

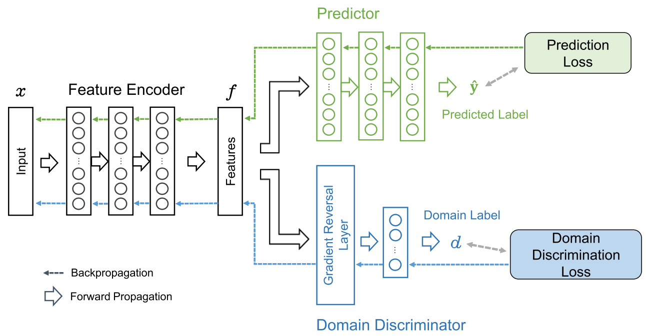

FIGURE 7. **DANN architecture according to [38].**

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 $\phi(\cdot) : X \to \mathcal{H}$ between the original space X and $\mathcal{H}$, the MMD between two datasets X_S and X_T can be calculated as in [73]:

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

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.

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

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:

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

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

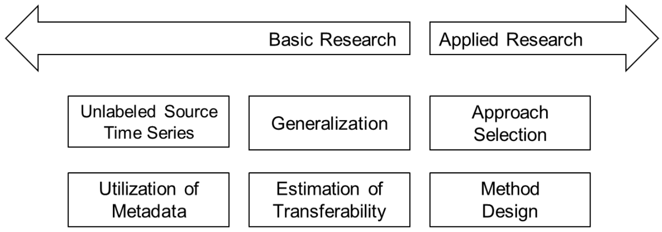

FIGURE 8. Research opportunities from a topical view.

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.

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. Ḿuller, "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/%7Eeamonn/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.

Appendix B

Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection

Reference:

Manuel Weber, Farzan Banihashemi, Peter Mandl, Hans-Arno Jacobsen, and Ruben Mayer. "Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection." In: *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2023), pp. 1–10. DOI: 10.1145/3600100.3623718. URL: https://doi.org/10.1145/3600100.3623718

Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection

Manuel Weber
Munich University of Applied Sciences (HM)
Munich, Germany

Farzan Banihashemi
Technical University of Munich
Munich, Germany

Peter Mandl
Munich University of Applied Sciences (HM)
Munich, Germany

Hans-Arno Jacobsen
University of Toronto
Toronto, Canada

Ruben Mayer
University of Bayreuth
Bayreuth, Germany

ABSTRACT

Knowing indoor occupancy states is crucial for energy optimization in buildings. While neural networks can effectively be used to detect occupancy based on carbon dioxide measurements, their application is impeded by the need for sufficient labeled training data. In this study, we analyze the prediction performance of three different transfer learning (TL) methods leveraging target room data jointly with data from other rooms. The methods include (1) pretraining and fine-tuning, (2) layer freezing, and (3) domain-adversarial learning. Using data from five real-world rooms and one simulated room, including multiple room types, we provide the most extensive evaluation of TL in the field of occupancy prediction from environmental variables to date. This work's contribution further includes the architecture and hyperparameters of a deep CNN-LSTM model for CO2-based occupancy detection. Our results indicate that TL effectively reduces the required amount of target room data. Moreover, while previous literature was focused on pretraining with related real-world data, we show that similar performance can be achieved by the more practical approach of leveraging simulated data.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Human-centered computing → Ubiquitous and mobile computing**.

KEYWORDS

Transfer Learning, Deep Learning, CNN-LSTM, Building Occupancy Detection, Indoor Carbon Dioxide

ACM Reference Format:
Manuel Weber, Farzan Banihashemi, Peter Mandl, Hans-Arno Jacobsen, and Ruben Mayer. 2023. Overcoming Data Scarcity through Transfer Learning in CO2-Based Building Occupancy Detection. In *The 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '23), November 15–16, 2023, Istanbul, Turkey*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3600100.3623718

BuildSys '23, November 15–16, 2023, Istanbul, Turkey

ACM ISBN 979-8-4007-0230-3/23/11.
https://doi.org/10.1145/3600100.3623718

1 INTRODUCTION

Data on building occupancy enables various use cases in terms of building energy performance simulation or building automation [7, 22]. This can have a high impact in the current fight against climate change. As stated by the International Energy Agency (IEA), the buildings sector is responsible for 30% of global energy consumption, with CO_2 emissions annually increasing over the past decade [19]. Moreover, the COVID-19 pandemic has shifted office occupancy away from standardized schedules, which further increases the need for optimization. Integration of occupancy information into heating, ventilation, and air-conditioning (HVAC) systems was experimentally shown to result in energy savings of 37% in [22] and up to 48% in [7]. However, since manual data collection is impractical, especially at the room level, occupancy should be predicted automatically. In [38], the authors outline how occupancy information inferred from environmental sensors can be used for automatic heating set point scheduling. We differentiate between two prediction problems: occupancy detection, which is a binary prediction of presence in general, and occupancy estimation (or counting), the prediction of the number of occupants. In this study, we address the problem of occupancy detection. For the majority of room types, including offices or residential spaces, utilization is limited to a few occupants, and, with respect to energy optimization, exact occupant count information has little advantage over binary states. As the vast majority of today's buildings are not equipped with any dedicated sensing technology, there is a growing branch of research on how to use available technologies or integrate further technology [29, 30]. Technologies explored in the literature include [29], among others, optical or thermal cameras, climate sensors, passive-infrared (PIR) motion sensors, sound sensors, or smart meters to measure energy consumption [11]. Rinta-Homi et al. [27], for example, analyzed occupancy detection based on low-resolution thermal sensors, and in association with this, the trade-off between prediction ability and costs in terms of privacy and deployment. Despite being less accurate than approaches based on camera images, environmental sensing of factors such as carbon dioxide, temperature, or humidity has gained considerable research interest due to its low-cost implementation and low privacy-intrusive nature. Therefore, various machine learning approaches [8, 9, 15, 21, 40], including deep learning [12], have been proposed to infer occupancy from indoor climate. However, scarce data was identified as one of the key challenges for accurate data-driven occupancy prediction [25, 29]. CO_2 rates or other indoor climate factors show different behavior depending on various factors such as room dimensions, infiltration rate, ventilation, or presence of humans or plants. Collecting room-specific labeled training data over large time periods is not applicable in practice. An approach to tackle the impediment of data scarcity is transfer learning (TL). After its huge success in the fields of computer vision and natural language processing, TL has recently made its way into the domain of time series data [33]. In this study, we adopt the definition of TL according to Pan et al. [26]: *"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 $T_S \neq T_T$."* We assume that the detection task remains unchanged ($T_S = T_T$), while D_S and D_T may refer to different rooms, as in [3, 31], or D_S may involve simulated data [34, 35]. Recent literature studies [14, 30] reviewing the application of TL methods for occupancy estimation show that, to date, only a few TL-related studies exist for environmental sensing. Moreover, previous works in this context mainly propose a single transfer method and often evaluate it in one specific scenario. This leaves a research gap regarding empirical comparisons of transfer methods on different datasets, which this paper aims to fill. Our contributions include:

- architecture and hyperparameters for a CO_2-based deep learning occupancy detection model found via an extensive hyperparameter tuning run with data from five rooms,
- the evaluation of three transfer methods, namely pretraining and fine-tuning, layer freezing, and domain-adversarial learning, with six different datasets,
- a comparison between the transfer from similar rooms and a sim-to-real transfer from simulated CO_2 data, and
- the provision of a preprocessed collection of datasets that can be reused in further works on occupancy detection.

2 RELATED WORK

Over the last years, several publications have addressed the estimation or binary detection of occupancy based on indoor climate factors including CO_2. While some works directly analyze physical dependencies and propose to infer occupancy, for instance, based on the mass balance equation [8], the majority of relevant literature is focused on machine learning (ML) solutions [3, 9, 10, 21], including numerous works leveraging neural networks [12, 18, 40]. ML seems more appropriate in this context since explicit modeling is difficult due to the complexity of indoor CO_2 dynamics and other climate factors. Recently, some works have come up that apply TL in the addressed research area in order to improve adaptability to different rooms. Table 1 gives an overview of these. AriefAng et al. [3] initially used a seasonal decomposition model to show that data from an only slightly related domain, in terms of an academic office, can be used to improve prediction performance for a cinema hall. In [2], the authors extended their evaluation by further domains, including classrooms and study zones. Zhang and Ardakanian [39] proposed a transfer between similar rooms in a building using a long short-term memory (LSTM) network. Despite being more generalizable than [2, 3], the approach involves a weighting step relying on domain knowledge. Weber et al. [34, 35] and Stjelja et al. [31] recently applied a deep learning model from [12] to conduct transfer via pretraining and fine-tuning, allowing more generalizability. While [31] focused on similar rooms in a building, [34, 35] proposed the utilization of simulated data to improve predictions for a real-world target office. Another work, by Khalil et al. [23], applied pretraining and fine-tuning with a multi-layer perceptron and a stacked LSTM. With respect to their experimental results on the transfer between offices in a university building, the authors again pointed out the usefulness of TL for occupancy prediction [23]. As well as most of the previous literature, they focus on transfer from similar real-world rooms and do not consider the potential of a

Table 1: Literature on transfer learning for occupancy prediction from CO_2 and other environmental factors

Publication	Problem	Factors	Model	Method	Dataset	Room Type	Dataset Size	Data Rate
Stjelja et al. 2022 [31]	Both	CO_2, Temp.	Deep NN [12]	Fine-Tuning, Layer Freezing	own[1]	2 Meeting Rooms	19 days, 64 days	3 min
Khalil et al. 2021 [23]	Detect.	CO_2, Temp., Humidity, Motion	Deep NN (MLP and stacked LSTM)	Fine-Tuning	own[1]	3 Offices	27 months, 3 months, 3 months	5 min
Weber et al. 2020 [34, 35]	Detect.	CO_2	Deep NN [12]	Fine-Tuning	own[1]	1 Office, 1 Simulated Office	7 days, 500 days	5 sec[2], 1 sec[2]
Zhang & Ardakanian 2019 [39]	Est.	CO_2, Air Flow	LSTM, NN	Fine-Tuning with Reweighting	[1], own[1]	2 Study Zones, 2 Classrooms, 3 Meeting Rooms	15 days, 1 year	1 min, 10 min
AriefAng et al. 2018, 2017 [2, 3]	Both	CO_2	Seasonal Decomposition	Domain-Specific	own[1], [1, 37]	1 Office Room, 1 Cinema Hall, 2 Study Zones, 2 Classrooms	2 months, 45 days, 15 days	5 min, 30 sec, 1 min

Detect.: occupancy detection. Est.: occupancy estimation or counting. Temp.: temperature.
NN: neural network. MLP: multilayer perceptron. LSTM: long short-term memory.

[1] Data unpublished. [2] Downsampled to 1 min for model training.

sim-to-real transfer using purely synthetic data from a dedicated simulation. We assume that this can further reduce the need for real-world data. A recent study by Rotem et al. [28] indicates that even pretraining on randomly generated synthetic time series can lead to improvements in time series classification.

While part of the literature on occupancy prediction combines several climate factors [9, 12, 23], such as CO_2, temperature, humidity, and air pressure, other works include only CO_2 [8, 18, 20, 21, 36, 40], which was, for instance, identified as the most informative factor in [10]. The CO_2 rate is closely related to human occupancy and generally shows patterns that can clearly be attributed to presence or absence of individuals in a room. Using more input factors can potentially help the model in inconclusive situations but does not necessarily lead to improved overall prediction performance. To the best of our knowledge, there is no broad evaluation showing a clear advantage of using multiple climate factors over using CO_2 only. Kraipeerapun et al. [24] even reported a higher accuracy when using CO_2, compared to the combination of CO_2 and temperature. In addition to this, even if the same or a slightly higher accuracy could be achieved, temperature is strongly seasonal. This makes an evaluation more difficult, as models trained during the winter season, for instance, may not perform the same on data from a warmer time of the year. Other factors may be more useful. Banihashemi et al. [5] reported a prediction improvement when combining CO_2 with sound pressure level and illuminance. However, while CO_2 sensors are already available in many modern buildings and can be used to optimize ventilation as well, these factors require dedicated sensors. In practice, additional sensor equipment on a room level produces high costs. Aiming at a lower model complexity and better applicability in practice, we use CO_2 time series as single model input. In this setting, TL is particularly important, as it was shown in [5] that, compared to the combination with sound pressure level or illuminance, training only with CO_2 requires larger amounts of training data until substantial results are reached.

3 METHODOLOGY

This section introduces the datasets used in this study, the applied deep learning model and transfer methods, as well as the evaluation metrics and the hyperparameter tuning process.

3.1 Datasets

In this study, we use CO_2 and occupancy data from multiple real-world datasets of different room types and scenarios, as well as a dataset obtained from simulation.

Table 2: Overview of datasets

ID	Name	Room Type	Days	Occ.[1]	PR[2]
A	Office A	office	80	0-2	0.32
B	Office B	office	20	0-2	0.26
C	Home	living & sleeping	50	0-2	0.75
D	Candanedo	office	8	0-2	0.35
E	Stjelja	meeting room	26	0-12	0.11
F	Simulated	hypothetical office	100	0-1	0.26

[1]Occ.: typical number of occupants, [2]PR: presence rate.

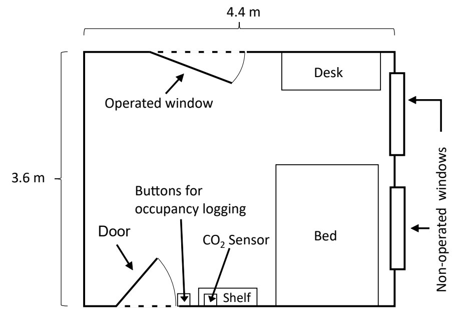

Figure 1: Layout of the residential room for dataset C.

To ensure comparability, each dataset was resampled to a common 1-min resolution, and occupancy values were binarized to represent presence (1) or absence (0). Incomplete or unoccupied days were removed. Five of the prepared datasets are made publicly available on GitHub.[1] All datasets are summarized in Table 2.

A) Office A: These data were collected at 80 working days between September 2021 and June 2022 in an office room of a university building with about 24 m², located in Munich, Germany. The data were collected through a sensory device and previously used for occupancy detection in a study by Banihashemi et al. [5]. The device consists of multiple sensors to capture indoor environmental qualities (IEQ), including ambient light, sound pressure level, air temperature, indoor air quality, relative humidity, and CO_2. The data were published under [4]. In this study, we only use the CO_2 data, measured with a Sensirion SCD30 sensor, and the occupancy ground truth, documented by additional manual recording. The room is a two-person office but was predominantly used by a single person due to Covid-19 restrictions.

B) Office B: Dataset B contains 20 working days from a second office room in another part of the same building used for dataset A. The two rooms have similar dimensions and measurement was conducted equally. The data were as well published under [4].

C) Home (Living & Sleeping): For the Home dataset, we collected data from 50 days of occupation in a residential building in Munich, Germany, within the period between 30 April 2020 and 05 July 2020. Due to the Covid-19 lockdown regulations, the room was occupied most of the time and used for working, living, and sleeping. Therefore, this dataset reflects an untypical occupation behavior and contrasts the office datasets with lower occupation rates. The room is located on the ground floor and measures approximately 16 m^2. CO_2 levels were measured with a Sensirion SCD30 sensor in intervals of 1 sec and later downsampled to a 1-min resolution by averaging. The sensor was placed on top of a bookshelf at a height of approximately 2 m. The room was regularly occupied by two occupants who manually documented their occupancy via button clicking. Figure 1 shows the layout of the room.

[1]https://github.com/CCWI/transfer_learning_for_occupancy_detection/

D) Candanedo et al. 2016 (Office): Dataset D contains data measured by Candanedo et al. [9], which were previously used for occupancy detection. It was collected in a two-person office of approximately 20 m^2 within the period between 2nd and 18th of February 2015. CO_2 rates were measured with a Telaire 6613 sensor multiple times per minute and downsampled to a 1-min resolution by averaging. Ground truth was manually inferred from camera pictures. As some days within the data were either not measured completely or fully unoccupied, we selected a subsample of eight complete days. We prepared two further subsamples, in which we additionally removed (D.1) unexpected sensor behavior during non-office hours, and (D.2) the same as (D.1) plus slowly falling CO_2 values during occupation due to opening of the office door.

E) Stjelja et al. 2022 (Meeting Room): Dataset E contains data measured by Stjelja et al. [31], which were previously used for occupancy detection. It was collected in a meeting room of a hospital building in Finland. The room measures 21 m^2 and is designed for up to 12 occupants. The measurement was conducted in March and April of 2021. Ground truth was documented in the form of a binary occupation state using an infrared people-counting camera. The authors collected further data, which we did not use, as the room was not operated with constant airflow anymore. From the data collected in [31], we selected a total of 26 occupied days under constant airflow. As the data was available at a 3-min resolution, we applied linear interpolation to obtain 1-min records.

F) Simulated (Hypothetical Office): Dataset F contains simulated data for a hypothetical one-person office room with the same dimensions as the room used for dataset A. The data was previously used for occupancy detection by Weber et al. in [34, 35]. First, occupancy was simulated by considering typical office hours for basic status transitions as well as a Markov chain to model random movement. A second Markov chain was used to model the window opening behavior. Afterward, CO_2 rates were determined for each time step based on mass balance calculations. Further details on the simulation can be found in [35]. We selected 100 simulated working days into dataset F.

3.2 Deep Learning Model

The deep learning model used in this study is inspired by the convolutional deep bidirectional long short-term memory (CDBLSTM) proposed by Chen et al. [12]. It was introduced for occupancy estimation based on environmental factors outperforming various previous approaches. Moreover, it has been successfully applied for TL in the context of occupancy prediction [31, 34, 35]. As depicted in Fig. 2, the model combines a one-dimensional convolutional neural network (CNN), for the extraction of local features, and a bidirected long short-term memory (BLSTM), that takes temporal dependencies between these features into account. Subsequently, a feed-forward network of fully connected (FC) layers is used for final classification, with a layer-wise dropout for regularization purposes. The combination of CNN and long short-term memory (LSTM) is a typical approach that is widely applied in TL studies on other time series prediction problems as well [33].

In contrast to [12], we focus on binary occupancy detection, hence, we use sigmoid activation in the output layer instead of softmax. In addition, we treat the number of layers in each of the three model parts as hyperparameters as we assume that the optimal layer numbers depend on the problem setting, including input factors and binary versus multi-class classification. After hyperparameter tuning (see section 3.5), our model consists of two convolutions with 200 and 50 filters, and with kernel sizes of 5 and 3 in the two 1D convolutional layers. Each of the two is followed by a max pooling layer with a pooling size of 2. The convolutional network is connected to three subsequent BLSTM layers with 50 cells per layer. While a stateful BLSTM shows advanced performance if input sequences are not shuffled, we apply a stateless BLSTM and shuffling to avoid overfitting. This is especially required for the domain-adversarial model introduced in Sec. 3 to ensure that the domain classifier receives sequences from both domains alternately. The final FC classifier consists of one dropout layer with a dropout rate of 0.5 and one FC layer with 100 neurons. The model is trained with a batch size of 128 and Adam optimizer. We use input sequences of 30 minutes, generated by a sliding window technique on the original time series. To avoid data leakage, we apply sliding windows separately to distinct days in the dataset and use sequences from the same day exclusively for either training or testing.

3.3 Evaluation Metrics

For model evaluation, we use two metrics. One is the accuracy (*acc*) as it is the most widely used evaluation metric in similar studies. It measures the percentage of correct predictions and is given by Eq. 1, where tp, tn, fp, and fn refer to true positive, true negative, false positive, and false negative predictions.

$$acc = \frac{tp + tn}{tp + fp + tn + fn} \tag{1}$$

Since the accuracy is of limited informative value for imbalanced class distributions, it is not well suited, as most rooms, for instance, office rooms, strongly tend to vacancy, especially during the night. Hence, we use Cohen's kappa [13] as a second, more informative metric. Cohen's kappa coefficient (κ) traditionally measures the agreement between two annotators and can be interpreted as the extent to which a model performs superior over a random classifier. It is calculated according to Eq. 2, where p_e is the expected accuracy by chance, calculated by p and n, predicted positives and negatives, as well as p^* and n^*, positives and negatives in the ground truth.

$$\kappa = \frac{acc - p_e}{1 - p_e}, \qquad p_e = \frac{p \cdot p^* + n \cdot n^*}{(p+n)^2} \tag{2}$$

3.4 Transfer Learning Methods

In this paper, we apply the following transfer methods, representing the three most widely adopted approaches for model-based time series TL [33].

Pretraining and Fine-Tuning: A neural network is pretrained on data from the source domain D_S, and the trained model parameters are reused to initialize a model for consecutive training, called fine-tuning, with data from the target domain D_T.

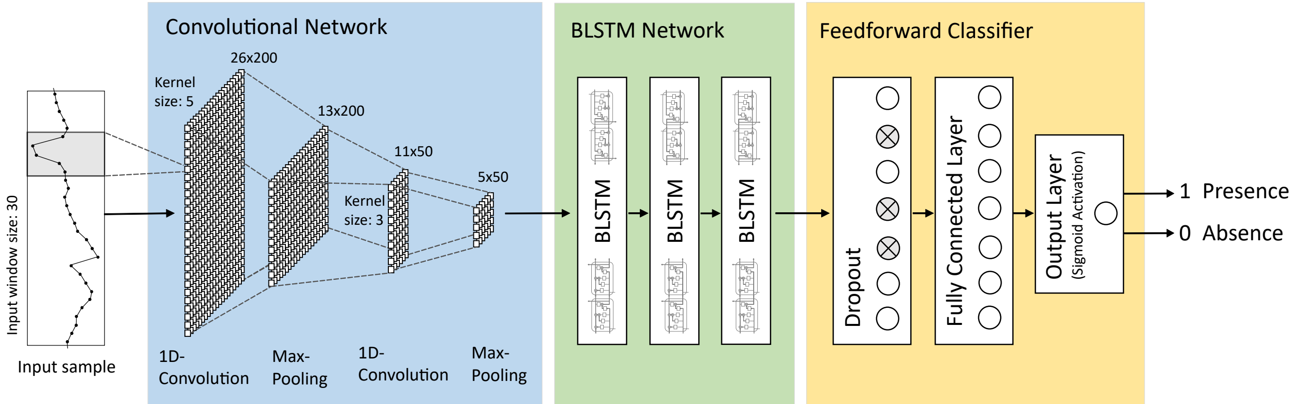

Figure 2: Deep learning model architecture for occupancy detection.

While fine-tuning may include a modification of the training procedure, in this paper, we retrain with training parameters remaining unchanged. We use the last 20% of the source dataset for validation during pretraining. For validation, during fine-tuning, we use one additional target day and apply two variants: In the first, we use only target data, which we call vanilla fine-tuning. In the second, we add an equal amount of source validation data already seen during pretraining.

Layer freezing: A common variation of pretraining and fine-tuning is layer freezing [33]. In this method, fine-tuning on target data only applies to a subset of k out of m model layers l_i. We fine-tune the last k layers. Previous layers $\{l_1, ..., l_n\} \subset \{l_1, l_2, ..., l_m\}$ with $n = m - k$ are frozen, which means their parameters remain unchanged during retraining. We apply two variants, (1) freezing the layers of the CNN part of the model, and (2) freezing both CNN and BLSTM.

Domain-Adversarial Learning: This method was introduced by Ganin et al. [17] and uses two competing classifiers, the task classifier (TC) and an additional domain classifier (DC). Both receive features from a preceding feature generator as depicted in Fig. 3. We calculate two loss functions, $\mathcal{L}_t$ and $\mathcal{L}_d$, for TC and DC. In both cases, we apply the binary cross-entropy. A gradient reversal layer (GRL) [17] negates the loss from the domain classification branch. This leads to the following aggregated loss for the feature generator, where λ is a weighting factor that we set to $\lambda = 1$ in this study:

$$\mathcal{L}_{agg} = \mathcal{L}_t - \lambda \mathcal{L}_d \quad (3)$$

Since the DC tries to discriminate between target and source samples, gradient reversal forces the feature generator to extract domain-invariant feature representations. We apply two variants, placing the DC (1) after the CNN or (2) after the BLSTM.

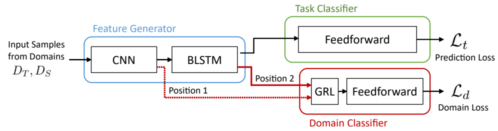

Figure 3: Domain-Adversarial Architecture. GRL: Gradient Reversal Layer.

3.5 Hyperparameter Tuning

To find the model architecture and training parameters described in Sec. 3.2, we conducted a hyperparameter tuning process divided into two phases: In phase 1, the task classifier, i.e., the base CDBLSTM, was tuned. In phase 2, the domain classifier of the domain-adversarial model was tuned, reusing task classifier parameters from phase 1.

Table 3: Hyperparameters, cardinality (#), and search values

Hyperparameter	#	Search Values
A. Training Parameters:		
Batch Size	3	[32, 64, 128]
Window Size	3	[15, 30, 60]
Optimizer	4	[SGD, RMSprop, Adam, Adadelta]
B. Number of Layers:		
Number of Convolutions	2	[1, 2]
Number of BLSTM Layers	4	[1, 2, 3, 4]
Number of Dense Layers	4	[1, 2, 3, 4]
Number of Layers in DC[3]	3	[0, 1, 2]
C. Further Hyperparamters:		
Stateful BLSTM	2	[True, False]
Convolutional Filters[1]	4	[50, 100, 150, 200]
Convolution Kernel Size[1]	2	[3, 5]
Number of BLSTM Cells[1 2]	10	[50, 100, 150, ..., 500]
Number of Dense Neurons[1 2]	5	[100, 200, ..., 500]
Number of Neurons in DC[1 2 3]	50	[10, 20, ..., 500]
Dropout Rates[1 2]	3	[0.1, 0.3, 0.5]
Dropout Rates in DC[1 2 3]	3	[0.1, 0.3, 0.5]

[1] May exist multiple times, depending on the selected number of layers (B.), with possibly different values each.

[2] The listed values apply to the first layer. Consecutive layers are restricted to have the same or half the value of the previous (rounded up to the next listed discrete value), or the minimum.

[3] Refers to the domain classifier (DC) tuned separately in phase 2.

Phase 1: Base model tuning was conducted by applying a Bayesian optimization (BO). BO is an informed, sequential optimization strategy based on the Bayes theorem that selects promising parameter combinations according to previous evaluations. It is well-suited for computationally expensive evaluations and outperforms uninformed strategies such as grid search or random search [32]. We conducted 1000 iterations of BO to approximate the optimal hyperparameter setting within the search space, and used all real-world datasets (A-E) for the tuning procedure. For each dataset, we reserved the first 80% of the data for hyperparameter tuning, the remaining 20% were held out for testing purposes. Among the first 80%, we randomly selected 5 consecutive days of data for training and the following 3 days for validation. We decided on a training data amount of 5 days, as our preliminary examinations showed this to be the minimum amount of data needed for the untuned deep learning model to produce solid results. By keeping the training data amount small, we ensure fast training times and allow a large number of parameter settings to be searched in the tuning procedure. Table 3 lists all parameters and their considered values included in the hyperparameter search. We included (A) parameters related to the training process and input data, i.e., different batch sizes and optimizers, as well as input window sizes of either 15-, 30-, or 60-minute CO_2 sequences, (B) different numbers of layers for all model components, and (C) further hyperparameters such as the number of neurons, filters, and BLSTM cells. In the case of multiple layers per model component, we reduced the complexity of the search space by constraining the number of neurons or cells in consecutive layers to either (1) the minimum allowed value, (2) the same as the preceding layer, or (3) half of the preceding layer's value rounded up to the next allowed value.

Phase 2: In phase 2, we tuned the number of layers and neurons as well as dropout rates in the domain classifier branched from the base model for domain-adversarial learning. We applied a BO with 100 iterations. Within each iteration, we tested each of 20 possible source-to-target combinations within the set of datasets A-E, with 5 repetitions each, and calculated the average Cohen's kappa accordingly. While again using five days of training data from the source dataset, we used one additional day from the target dataset for training and three target days for validation.

3.6 Experimental Setup

We applied a min-max normalization separately to each training, validation, or test split. Model training and interference were conducted on an Nvidia GeForce Tesla V100 SXM2. Keras and Tensorflow were used for implementation. All training procedures were run with early stopping after five epochs without loss improvement. The source code used for the experiments is made publicly available on GitHub.[2]

4 RESULTS

To analyze transfer performance, we conducted multiple experiments applying each of the transfer methods introduced in Sec. 3.4 to each possible combination of source datasets within A-F and distinct target datasets within A-E. Simulated data (F) was not used as target, as this would not represent any real-world use case. Each experiment was repeated 20 times to increase reliability.

Fig. 4 shows the results for an exemplary selection of experiments where two days of target data were used. The results indicate that depending on the specific data, different transfer methods result in remarkable differences in prediction performance. While pretraining and fine-tuning, for instance, performs best for combinations 3 and 5, domain-adversarial learning shows superior results in case 4 (Stjelja to Candanedo) and layer-freezing in case 1 (office B to A). In addition to this, there are also differences in reliability. Especially layer-freezing often shows large standard deviations.

[2] https://github.com/CCWI/transfer_learning_for_occupancy_detection/

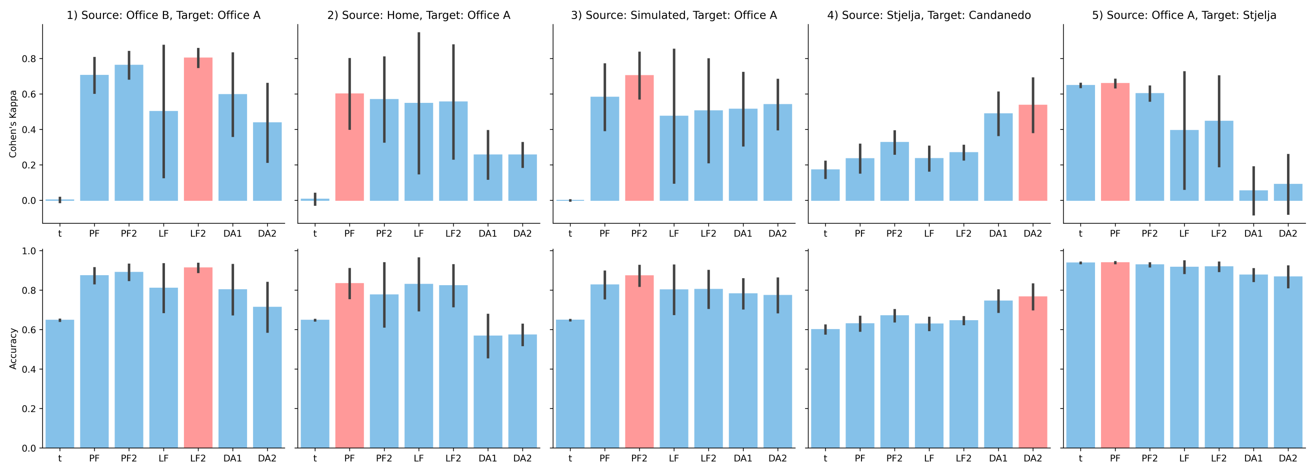

Figure 4: Mean and standard deviation of Cohen's kappa and accuracy by transfer method for five selected experiments with two days of target data. The highest mean values are highlighted in red. t: target-only training (no transfer), PF: pretraining and vanilla fine-tuning, PF2: pretraining and fine-tuning with source and target data, LF: layer freezing (frozen CNN), LF2: layer freezing (frozen CNN-BLSTM), DA1/DA2: domain-adversarial training variant 1 or 2.

In our experiments, the best transfer method almost always performs superior to non-transfer, regardless of the source dataset, except for some exceptions targeting the meeting room dataset (E) by Stjelja et al. Combinations with dataset E suffer the most from negative transfer. This may be due to the data interpolation from 3-min to 1-min resolution, which was only necessary for this dataset, or to the room type being the most dissimilar compared to the other datasets. All other rooms are regularly used by 1-2 occupants, whereas for E, the room is mostly unoccupied and there are up to 12 occupants. Also, transfer to E shows low benefit, as the model can already reach substantial results with scarce training data only. When using two days of data from E, for example, in comparison with training from scratch, pretraining on A only increases Cohen's kappa from 0.648 to 0.659 and accuracy from 0.938 to 0.939. For other target datasets, however, training a model from scratch with such scarce data is not possible, which can be seen from accuracies around 0.6 and kappa values close to zero. This offers large potential for TL.

4.1 Analysis of Transfer Methods

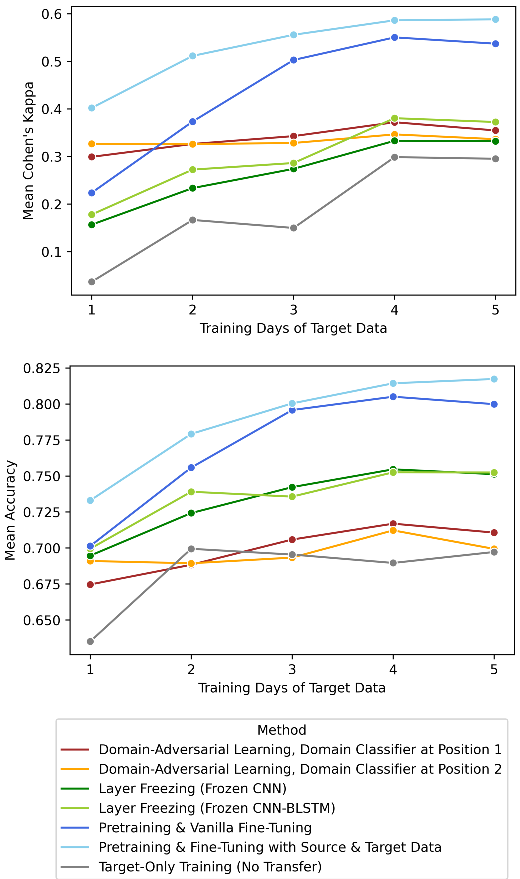

Figure 5: Mean Cohen's kappa and accuracy by target training days and transfer method over all source-to-target combinations between sources A-F and targets A-D.

Fig. 5 shows the mean performances over all source-to-target combinations with target datasets A-D. It can be seen that domain-adversarial learning shows remarkable performance in terms of Cohen's kappa in cases with extremely scarce data of only one target day. In comparison to the other methods, however, it appears to fail in making use of more target data. In our evaluation, for multiple training days, the mean accuracy remains close to the baseline accuracy without transfer learning, and below the accuracy of other transfer methods. Pretraining and fine-tuning stands out as the best method when averaging over the experiments. Especially when fine-tuning is conducted jointly with data from target and source domain, there is a further performance advantage. The approach intends to reduce catastrophic forgetting during fine-tuning with scarce data by repeatedly training with data previously seen during pretraining. Regarding the results shown in Fig. 5, it seems that this is particularly useful in cases with extremely scarce target data. With three target days or more, the performance of vanilla fine-tuning approaches a similar level, although still performing weaker. It shall be noted that target performance also depends on the selected source data. A comparison of source datasets is addressed in subsection 4.2. For specific sources, methods can reach a higher or lower performance than the mean values reported in Fig. 5. To show this, Fig. 6 provides a more detailed overview showing the mean Cohen's kappa values of the methods for each source dataset for one and five target training days respectively. Even with extremely scarce target data of only one day, most of the method-source combinations show significantly higher performance than the baseline without transfer, which is close to zero and cannot be regarded as a useful model. With five days, performance further increases in most cases. The top performance is reached on source dataset C by pretraining and fine-tuning with source and target data. Negative transfer can be observed in some cases, mostly with datasets D or E and with layer freezing or domain-adversarial learning. Hence, these methods seem less advisable when datasets are too diverse. Pretraining and fine-tuning appears more flexible. With five target training days, it shows positive transfer for each of the sources. This can be explained by the necessity of early layers being retrained when there are massive pattern changes, which is omitted in the layer freezing method.

4.2 Analysis of Transfer Sources

Fig. 7 shows the mean Cohen's kappa and accuracy values grouped by the source datasets for 1-5 days of target training data. Dataset E was excluded as target dataset in this evaluation to avoid negative transfer affecting the results. For each source, values are reported for the best transfer method, as we suggest that in each case the best method would be applied in practice. A more detailed overview regarding the combination of sources and methods is shown in Fig. 6. The results show a clear dominance of transfer learning over target-only training for all source datasets. Datasets D and E perform weaker than other sources. This may again be due to major dissimilarities to the other datasets. The two offices A and B perform similarly. For more than one target day, the simulated data show similar results. This indicates that it is possible to successfully apply TL even without extensive real-world data collected from other rooms.

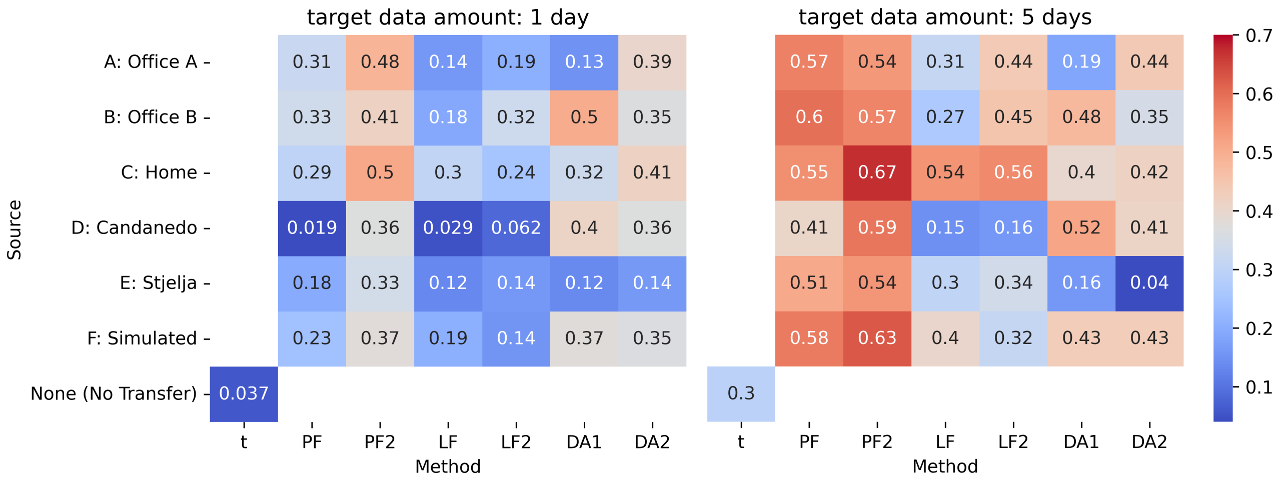

Figure 6: Mean Cohen's kappa over target datasets A-D by source dataset and transfer method for 1 day (left) or 5 days (right) of target data used for training. t: target-only training (no transfer), PF: pretraining and vanilla fine-tuning, PF2: pretraining and fine-tuning with source and target data, LF: layer freezing (frozen CNN), LF2: layer freezing (frozen CNN-BLSTM), DA1/DA2: domain-adversarial training variant 1 or 2.

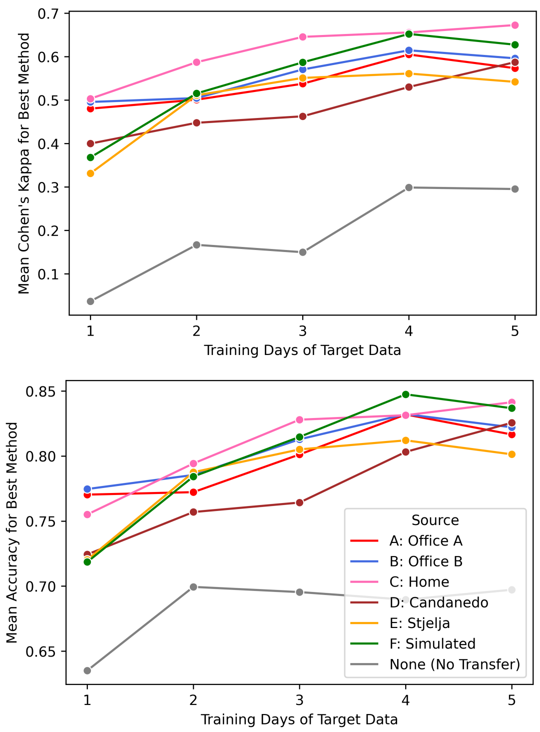

Figure 7: Mean Cohen's kappa and accuracy by target training days and source dataset for the best transfer method in each source-to-target combination between sources A-F and targets A-D.

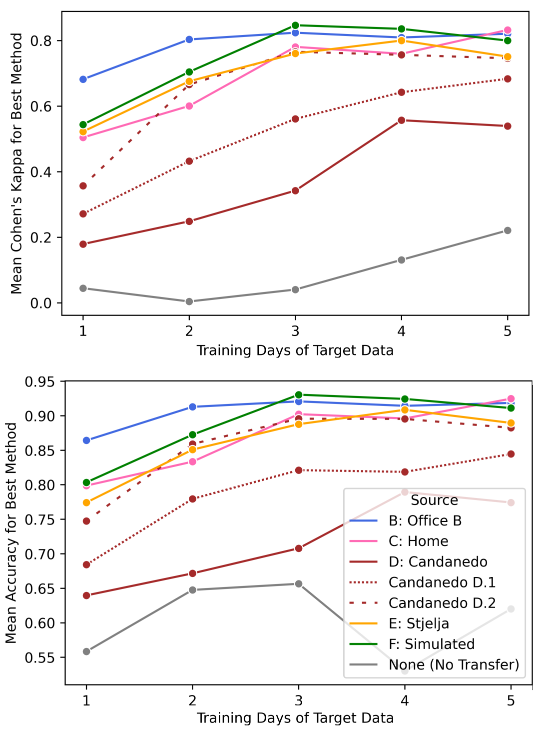

Figure 8: Mean Cohen's kappa by target training days and source dataset for the best transfer method in each experiment with sources B-F and office A as the target. Results for two selected subsets of D are shown in dashed lines.

The home dataset (C) stands out as the best source by average. With its high presence rate of 75% and dataset size of 50 days, it provides the most instances from the positive class. However, regarding a specific target room and with little target data, other sources may be more suitable, as can be seen in Fig. 8.

Fig. 8 shows the kappa values for different sources when office A is the target dataset. As expected, the most similar room, office B, appears to be the most suitable real-world source. However, with more than two target days, the simulated dataset F performs on par with office B.

Dataset D shows noticeable poor performance in transfer to A, although it was measured in a two-person office as well. Closer investigation showed some peculiar CO_2 measurements during non-office hours that, according to the authors, cannot be attributed to any known real-world phenomenon. Removing these from the data increases performance to some extent (see D.1). A further regular scenario that differs from the other datasets involves slowly decaying CO_2 values despite occupancy. The authors explain this by opening of the office door to let fresh air in. In dataset A, however, ventilation took place by window opening. This causes a fast drop of CO_2 that is more distinctive from vacancy. Removing also these unique decay scenarios (see D.2), performance becomes more comparable to other sources. This indicates that, besides the similarity of rooms, also the particularities of their usage are relevant regarding transfer capabilities. Despite these difficulties, the transfer from dataset D to A, just as from all other source datasets, improves the results to an important degree compared to training without transfer learning.

5 DISCUSSION

Our results show that TL broadly benefits occupancy detection based on CO_2 in many different scenarios. While this study is focused on the problem of scarce data and, hence, reports results for up to five working days in the target room, we observed further prediction improvements with larger amounts of target data as well. For instance, when using 40 working days from dataset A for target training, we still observed a slight advantage when using a model pretrained on 40 days of the simulated dataset D, with a Cohen's kappa of 0.8879 compared to 0.8808. This may motivate the use of TL even in cases where data scarcity is not a major problem, although it can be assumed that the performance advantage of TL tends towards zero with increasing target dataset size.

However, since data is often scarce in practice, TL may play a key role in making the approach applicable. A remaining difficulty is the observed performance variance depending on the concrete source dataset being used. Dissimilarities between source and target domain may appear due to differences in

1. properties of the room (size, ventilation type, etc.), or
2. occupant behavior (e.g., door or window opening behavior).

Indoor climate, for example, fundamentally differs between rooms with air-conditioning versus natural ventilation. While the applied methods are transferable, in this study, we focused on naturally ventilated rooms. A dissimilarity caused by occupant behavior found in this study was due to occupants using the office door instead of the window to ventilate the room.

Dissimilarities leading to negative transfer should be prevented in practice, which is why selecting the right source dataset and transfer method is crucial. Approaches on how to prevent negative transfer may be:

1. Selecting appropriate sources, either by actually testing the prediction performance or based on estimation. Regarding TL for time series classification in general, Fawaz et al. [16] suggested calculating a dynamic time warping (DTW) distance as a proxy for dataset similarity.
2. Selecting a transfer method that reduces the risk of negative transfer. In our evaluation, pretraining and fine-tuning showed less tendency to negative transfer. In contrast, freezing layers during retraining carries the risk that parameters representing patterns deviating from the target domain cannot be overwritten.
3. Multi-source training: Leveraging multiple source datasets at once, in order to train a more general model, that is less overfitting to a specific source room or occupant behavior.

Regarding multi-source training, however, it is a challenge to collect sufficient source data, as public datasets are rare. To avoid the restriction of limited real-world data being available, we propose leveraging simulated data instead. As shown in our evaluation, there is no discernable discrepancy between simulated source data and real-world source data. When transferring from the simulated dataset with more than two days of target data, we observed performances similar to the top-performing real-world datasets (cf. Fig. 7 and Fig. 8). Hence, we point out that the application of simulations in occupancy detection modeling needs to be further addressed in future research. Two alternative ideas were introduced in [34], which are (1) to replicate the conditions in the target room as closely as possible or (2) to simulate under a broad variety of conditions to generate data for multi-source training. The first option requires a large amount of manual work, including the tasks of room modeling, simulating, data preprocessing, and two phases of model training, whereas the latter may allow the preparation of a more general base model that can quickly be applied by fine-tuning on any given real-world data. This further reduces human involvement compared to collecting suitable real-world datasets for TL. As an alternative future research direction, we suggest to also compare the approach to unsupervised learning algorithms as, for instance, investigated in the context of energy consumption data in [6].

6 CONCLUSIONS

In this paper, we proposed a model architecture and configuration for the task of building occupancy detection from CO_2, based on an existing deep learning approach. We evaluated the model's capability of transfer learning, comparing three different transfer methods on a variety of datasets. The results suggest that pretraining and fine-tuning is generally a good choice in the addressed domain of application, even though it is not necessarily the best method in all cases. According to our experiments, its performance can be increased by fine-tuning jointly with data from target and source domain. In addition, in experiments with sufficient fine-tuning, pretraining with simulated data performed on par with using real-world data from a similar room. This finding underlines the usefulness of simulations in practice, as it allows us to train models with minimal data collection. In future work, we intend to pretrain an off-the-shelf model based on a variety of simulations.

ACKNOWLEDGMENTS

We would like to thank Davor Stjelja from Aalto University in Finland for kindly providing us with the meeting room data. Further thanks go to IBM and the *OpenPower@TUM* project for providing us with the GPU computing infrastructure.

REFERENCES

[1] Krzysztof Arendt, Aslak Johansen, Bo Nørregaard Jørgensen, Mikkel Baun Kjærgaard, Claudio Giovanni Mattera, Fisayo Caleb Sangogboye, Jens Hjort Schwee, and Christian T. Veje. 2018. Room-level occupant counts, airflow and CO 2 data from an office building. *First Workshop on Data Acquisition To Analysis (DATA)* (2018), 13–14.

[2] Irvan B. Arief-Ang, Margaret Hamilton, and Flora D. Salim. 2018. A Scalable Room Occupancy Prediction with Transferable Time Series Decomposition of CO2 Sensor Data. *ACM Transactions on Sensor Networks* 14, 3-4 (2018), 1–28.

[3] Irvan B. Arief-Ang, Flora D. Salim, and Margaret Hamilton. 2017. DA-HOC: Semi-Supervised Domain Adaptation for Room Occupancy Prediction using CO2 Sensor Data. *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys)* (2017), 1–10.

[4] Farzan Banihashemi and Manuel Weber. 2023. "Non-intrusive indoor environmental data in two double-occupied offices". *Mendeley Data* (2023). https://doi.org/10.17632/v7kw9ycccp.1

[5] Farzan Banihashemi, Manuel Weber, Fatma Deghim, Chujun Zong, and Werner Lang. 2023. Occupancy modeling on non-intrusive indoor environmental data through machine learning. *SSRN* (2023). https://doi.org/10.2139/ssrn.4477794

[6] Vincent Becker and Wilhelm Kleiminger. 2018. Exploring zero-training algorithms for occupancy detection based on smart meter measurements. *Computer Science - Research and Development* 33, 1 (2018), 25–36.

[7] Jonathan Brooks and Prabir Barooah. 2014. Energy-efficient control of underactuated HVAC zones in buildings. *2014 American Control Conference* (2014), 424–429.

[8] Davide Calì, Peter Matthes, Kristian Huchtemann, Rita Streblow, and Dirk Müller. 2015. CO2 based occupancy detection algorithm: Experimental analysis and validation for office and residential buildings. *Building and Environment* 86 (2015), 39–49.

[9] Luis M. Candanedo and Véronique Feldheim. 2016. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings* 112 (2016), 28–39.

[10] Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. 2017. A methodology based on Hidden Markov Models for occupancy detection and a case study in a low energy residential building. *Energy and Buildings* 148 (2017), 327–341.

[11] Dong Chen, Sean Barker, Adarsh Subbaswamy, David Irwin, and Prashant Shenoy. 2013. Non-Intrusive Occupancy Monitoring Using Smart Meters. *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BuildSys)* (2013), 1–8.

[12] Zhenghua Chen, Rui Zhao, Qingchang Zhu, Mustafa K. Masood, Yeng Chai Soh, and Kezhi Mao. 2017. Building Occupancy Estimation with Environmental Sensors via CDBLSTM. *IEEE Transactions on Industrial Electronics* 64, 12 (2017), 9549–9559.

[13] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.

[14] Jawher Dridi, Manar Amayri, and Nizar Bouguila. 2022. Transfer learning for estimating occupancy and recognizing activities in smart buildings. *Building and Environment* 217 (2022), 109057.

[15] Afrooz Ebadat, Giulio Bottegal, Damiano Varagnolo, Bo Wahlberg, and Karl H. Johansson. 2013. Estimation of Building Occupancy Levels through Environmental Signals Deconvolution. *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BuildSys)* (2013), 1–8.

[16] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2018. Transfer learning for time series classification. *IEEE International Conference on Big Data* (2018), 1367–1376.

[17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research* 17 (2016), 1–35.

[18] Hwataik Han, Kyung-jin Jang, Changho Han, and Junyong Lee. 2013. Occupancy estimation based on CO2 concentration using dynamic neural network model. *Proc. 34th Air Infiltration and Ventilation Centre Conference (AIVC)* (2013).

[19] IEA. 2022. World Energy Outlook 2022. https://www.iea.org/reports/world-energy-outlook-2022

[20] Chaoyang Jiang, Zhenghua Chen, Rong Su, Mustafa Khalid Masood, and Yeng Chai Soh. 2020. Bayesian filtering for building occupancy estimation from carbon dioxide concentration. *Energy and Buildings* 206 (2020), 109566.

[21] Ming Jin, Nikolaos Bekiaris-Liberis, Kevin Weekly, Costas J. Spanos, and Alexandre M. Bayen. 2018. Occupancy Detection via Environmental Sensing. *IEEE Transactions on Automation Science and Engineering* 15, 2 (2018), 443–455.

[22] Jonathan Brooks, Siddharth Goyal, Rahul Subramany, Yashen Lin, Timothy Middelkoop, Laura Arpan, Luca Carloni, and Prabir Barooah. 2014. An experimental investigation of occupancy-based energy-efficient control of commercial building indoor climate. *IEEE Conference on Decision and Control* 53 (2014), 5680–5685.

[23] Mohamad Khalil, Stephen McGough, Zoya Pourmirza, Mehdi Pazhoohesh, and Sara Walker. 2021. Transfer Learning Approach for Occupancy Prediction in Smart Buildings. *12th International Renewable Engineering Conference (IREC)* (2021), 1–6.

[24] Pawalai Kraipeerapun and Somkid Amornsamankul. 2017. Room Occupancy Detection using Modified Stacking. *Proceedings of the 9th International Conference on Machine Learning and Computing (ICMLC)* (2017), 162–166.

[25] William O'Brien, Andreas Wagner, Marcel Schweiker, Ardeshir Mahdavi, Julia Day, Mikkel Baun Kjærgaard, Salvatore Carlucci, Bing Dong, Farhang Tahmasebi, Da Yan, Tianzhen Hong, H. Burak Gunay, Zoltan Nagy, Clayton Miller, and Christiane Berger. 2020. Introducing IEA EBC annex 79: Key challenges and opportunities in the field of occupant-centric building design and operation. *Building and Environment* 178 (2020), 106738.

[26] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.

[27] Mikko Rinta-Homi, Naser Hossein Motlagh, Agustin Zuniga, Huber Flores, and Petteri Nurmi. 2021. How Low Can You Go? Performance Trade-Offs in Low-Resolution Thermal Sensors for Occupancy Detection: A Systematic Evaluation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3 (2021).

[28] Yarden Rotem, Nathaniel Shimoni, Lior Rokach, and Bracha Shapira. 2022. Transfer Learning for Time Series Classification Using Synthetic Data Generation. *Cyber Security, Cryptology, and Machine Learning (CSCML): 6th International Symposium* (2022), 232–246.

[29] Luis Rueda, Kodjo Agbossou, Alben Cardenas, Nilson Henao, and Sousso Kelouwani. 2020. A comprehensive review of approaches to building occupancy detection. *Building and Environment* 180 (2020), 106966.

[30] Aya Nabil Sayed, Yassine Himeur, and Faycal Bensaali. 2022. Deep and transfer learning for building occupancy detection: A review and comparative analysis. *Engineering Applications of Artificial Intelligence* 115 (2022), 105254.

[31] Davor Stjelja, Juha Jokisalo, and Risto Kosonen. 2022. Scalable Room Occupancy Prediction with Deep Transfer Learning Using Indoor Climate Sensor. *Energies* 15, 6 (2022).

[32] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. 2021. Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. *Proc. NeurIPS 2020, PMLR* 133 (2021), 3–26.

[33] Manuel Weber, Maximilian Auch, Christoph Doblander, Peter Mandl, and Hans-Arno Jacobsen. 2021. Transfer Learning With Time Series Data: A Systematic Mapping Study. *IEEE Access* 9 (2021), 165409–165432.

[34] Manuel Weber, Christoph Doblander, and Peter Mandl. 2020. Detecting Building Occupancy with Synthetic Environmental Data. *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2020), 324–325.

[35] Manuel Weber, Christoph Doblander, and Peter Mandl. 2020. Towards the Detection of Building Occupancy with Synthetic Environmental Data: arXiv:2010.04209.

[36] Yixuan Wei, Shu Wang, Longzhe Jin, Yifei Xu, and Tianqi Ding. 2022. Indoor occupancy estimation from carbon dioxide concentration using parameter estimation algorithms. *Building Services Engineering Research and Technology* 43, 4 (2022), 419–438.

[37] Jörg Wicker, Nicolas Krauter, Bettina Derstorff, Christof Stönner, Efstratios Bourtsoukidis, Thomas Klüpfel, Jonathan Williams, and Stefan Kramer. 2015. Cinema Data Mining: The Smell of Fear. *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), 1295–1304.

[38] Dominic Wörner, Thomas von Bomhard, and Felix Wortmann. 2013. Occupancy-Based Heating Control for Residential Buildings Using Environmental Sensors. *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BuildSys)* (2013), 1–2.

[39] Tianyu Zhang and Omid Ardakanian. 2019. A domain adaptation technique for fine-grained occupancy estimation in commercial buildings. *Proc. 2019 Internet of Things Design and Implementation (IoTDI)* (2019), 148–159.

[40] M. S. Zuraimi, A. Pantazaras, K. A. Chaturvedi, J. J. Yang, K. W. Tham, and S. E. Lee. 2017. Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. *Building and Environment* 123 (2017), 517–528.

Appendix C

Coddora: CO2-Based Occupancy Detection Model Trained via Domain Randomization

Reference:

Manuel Weber, Farzan Banihashemi, Davor Stjelja, Peter Mandl, Ruben Mayer, and Hans-Arno Jacobsen. "Coddora: CO2-Based Occupancy Detection Model Trained via Domain Randomization." In: *International Joint Conference on Neural Networks (IJCNN)* (2024), pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650820. URL: https://doi.org/10.1109/IJCNN60899.2024.10650820

Accepted version. The publication was modified for the conference.

Accepted version of paper submitted to International Joint Conference on Neural Networks (IJCNN). Published in IJCNN, 2024, pp. 1-8, DOI: 10.1109/IJCNN60899.2024.10650820, URL: https://ieeexplore.ieee.org/document/10650820

Coddora: CO2-Based Occupancy Detection Model Trained via Domain Randomization

Manuel Weber*, Farzan Banihashemi†, Davor Stjelja‡, Peter Mandl§, Ruben Mayer¶, and Hans-Arno Jacobsen‖
*§Department of Computer Science and Mathematics, Munich University of Applied Sciences (HM), Munich, Germany
†Chair of Energy Efficient and Sustainable Design and Building, Technical University of Munich, Germany
‡Department of Mechanical Engineering, Aalto University, Espoo, Finland
¶Faculty of Mathematics, Physics and Computer Science, University of Bayreuth, Germany
‖Department of Electrical and Computer Engineering, University of Toronto, ON, Canada
Email: *manuel.weber@hm.edu, §mandl@hm.edu, ‖jacobsen@eecg.toronto.edu

***Abstract*—Information about human presence in indoor spaces is crucial for building energy optimization. While there has been a considerable amount of research on using neural networks to automatically detect occupancy from CO2 sensors, their application in practice is limited due to the scarcity of labeled training data. In this paper, we propose Coddora, an off-the-shelf deep learning model pretrained on data from randomized room simulations. Coddora enables quick adaptation to real-world rooms, requiring only minimal data collection. Our contribution includes two base models for application via fine-tuning or zero-shot classifying, as well as the synthetic dataset providing data from simulations based on 100,000 room models.**

***Index Terms*—building occupancy detection, deep learning, neural networks, off-the-shelf model, domain randomization**

I. Introduction

According to the International Energy Agency (IEA), the buildings sector accounts for a substantial 30% of global energy consumption, with annual emission increases over the past decade [1]. An approach to energy savings lies in the integration of occupancy data into heating, ventilation, and air-conditioning (HVAC) systems in terms of occupant-centric control. Experimental results in a commercial building showed energy savings of 37% [2]. Besides applications in building operation, the availability of occupancy data also allows for more realistic building performance simulations, which are currently based on static, standardized schedules. As manual data collection on a large scale is impractical, it is indispensable to develop applicable automatic methods for occupancy detection. Various technologies have been investigated including climate sensors, optical or thermal cameras, passive-infrared motion sensors, sound sensors, and smart meters for energy consumption measurement [3]. This paper addresses the branch of research inferring occupancy from indoor CO_2 concentration or other climate factors such as temperature or humidity. The area has gained high research interest due to low-cost implementation and low privacy invasion. Several works have shown that neural networks are able to successfully model indoor CO_2 dynamics to predict occupancy and, when given enough training data, are able to outperform other approaches including physical models, hidden Markov models, or random forests [4]–[9]. However, data scarcity is considered a primary impediment [3]. Indoor climate relies on specific conditions, including room properties as well as occupant behavior, which makes it necessary to collect room-specific labeled training data. Lately, several studies have shown that the required amount of ground truth data can be reduced significantly by applying transfer learning (TL) with data from other rooms [10]–[13], making CO_2-based occupancy detection more practicable. However, the application of TL is still impeded by the limited availability of public datasets and by the risk of negative transfer from specific source data. While labeled real-world data are rare, data from simulations may be used alternatively. In [14], [15] pretraining with simulated CO_2 time series allowed for a reduction of the required amount of real-world data by 50%. In a further empirical study with one simulated and five real-world datasets [10], transfer from simulated CO_2 data achieved performances comparable to real-world source data.

The goal of this work is to provide a reusable, general base model for occupancy detection using domain randomization. Domain randomization is a recently introduced concept with the aim of inexpensively providing training data for neural networks [16]. The idea is to provide the model with a sufficient variety of simulated training data to enable it to generalize effectively to the real world during testing [16]. As shown by Rotem et al. [17], randomly generated time series enable improved predictions in time series classification. We assume that randomization becomes particularly beneficial when generated time series are tailored to the domain of interest. For occupancy detection, domain randomization seems practical, as indoor climate depends on the laws of physics, allowing for straightforward simulations. While related work has shown the effectiveness of deep learning and TL for occupancy detection (cf. Sec. II), there is a research gap regarding the development of a readily applicable model that does not require extensive data collection or data simulation to be trained. We address this gap with the following contributions:

1. a dataset providing indoor climate data from 10 simulated workdays for 100,000 randomized rooms, half office rooms and half meeting rooms,
2. an off-the-shelf deep learning model (Coddora) pretrained on CO_2 sequences from the provided dataset that can be

Accepted version of paper submitted to International Joint Conference on Neural Networks (IJCNN). Published in IJCNN, 2024, pp. 1-8, DOI: 10.1109/IJCNN60899.2024.10650820, URL: https://ieeexplore.ieee.org/document/10650820

fine-tuned with data from a real-world target room, and
3. a zero-shot version of the pretrained model (CoddoraRW) additionally retrained on some real-world data.

The paper is structured as follows: Sec. II introduces the most relevant related work. Sec. III describes the dataset and simulation methodology, Sec. IV the model architecture and training procedures. Sec. V provides an evaluation for two real-world rooms, followed by a discussion of implications on energy efficiency and limitations in Sec. VI.

II. Related Work

Several works have proposed neural networks [4]–[9] and TL approaches [10]–[15] in the context of providing occupancy information from indoor climate. Chen et al. [6] proposed a deep CNN-LSTM-based model outperforming previous state-of-the-art approaches. To reduce its data intensity, the same model type has been applied in a row of TL studies [10], [11], [14], [15]. Stjelja et al. [11], conducted a transfer from a small meeting room to a larger one and were able to reduce required ground truth data from multiple weeks to two days. A similar reduction was shown in [10] for diverse combinations of datasets including different room types. Further works have used other model types such as multilayer perceptron (MLP) [12] or long short-term memory (LSTM) models [12], [13] to transfer between office [12], meeting, or study rooms [13]. For a structured overview of TL studies in this context, we refer to [10]. Apart from using real-world source data, Weber et al. have successfully applied a sim-to-real transfer from simulated CO_2 data [10], [14], [15], which further reduces the need to collect labeled data in real-world scenarios.

While previous works have clearly demonstrated improved practicality through TL, approaches still rely on considerable effort for model training and data collection, or respectively, data simulation. With this work, we further enhance model creation by preparing a large dataset based on a variety of indoor climate simulations and training a reusable model.

III. Data Simulation

This section introduces the synthetic dataset and describes the applied simulation methods. The dataset is made publicly available[1]. Sec. III-A describes the properties of the dataset. The following subsections describe the simulation approach (III-B) as well as the concrete methods used to simulate occupancy (III-C) and indoor climate (III-D).

A. Properties of the dataset

The published dataset contains data for 100,000 randomly generated models of commercial building rooms; half with regard to office rooms, the other half to meeting rooms. For each room, it contains synthetic occupancy counts, window opening fractions, and room climate throughout 10 workdays.

This includes three climate factors: CO_2, relative humidity, and temperature. Fig. 1 shows their distributions.

[1]https://doi.org/10.5281/zenodo.10507614

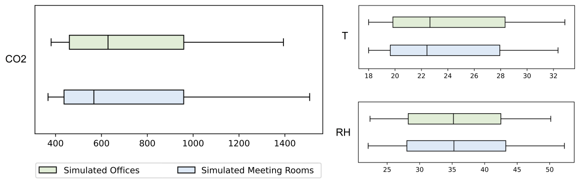

Fig. 1. Box-Whisker plot for CO_2 in ppm, indoor temperature (T) in °C, and relative humidity (RH) in % by room type. Whiskers show the 0.1 and 0.9, boxes the 0.25, 0.5, and 0.75 quantiles.

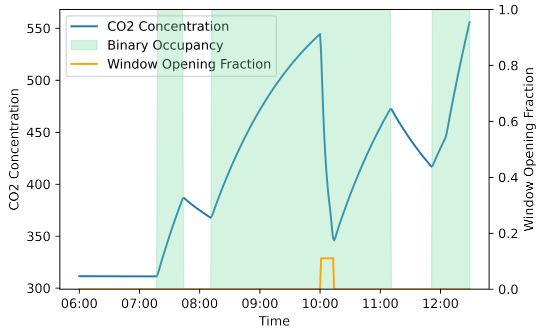

Fig. 2. Example of simulated CO_2 data.

While the provided data for temperature and humidity may be used in other research works, in this paper, we only use CO_2 data, aiming for simple applicability and low model complexity.

The simulated CO_2 rates typically lie within a realistic range of 300 to 1500 parts per million (ppm), with a median at 629 ppm for the office and 567 ppm for the meeting room data. Approximately 10% of values cover the range of 1500 to 5000 ppm, indicating poor air quality. With less than 0.03% of values in the dataset, CO_2 levels hardly exceed 5000 ppm, which is the permissible exposure limit established by the American Occupational Safety and Health Administration (OSHA) [18]. Besides the environmental data, the dataset includes binary occupancy states, precise occupant counts, and window opening fractions in the range from 0 to 1, representing the proportion of the area used for ventilation relative to the total window opening area of a room. Fig. 2 shows an example for CO_2 data in the dataset.

Note that each day in the dataset contains occupancy. We do not consider unoccupied weekends or holidays, as these do not contain useful information regarding occupancy detection patterns. In addition to this, the time series contained in the dataset completely result from simulated occupancy and ventilation. Hence, they do not contain noise as present in real-world measurements, but still allow to implicitly learn the involved laws of physics.

Accepted version of paper submitted to International Joint Conference on Neural Networks (IJCNN). Published in IJCNN, 2024, pp. 1-8, DOI: 10.1109/IJCNN60899.2024.10650820, URL: https://ieeexplore.ieee.org/document/10650820

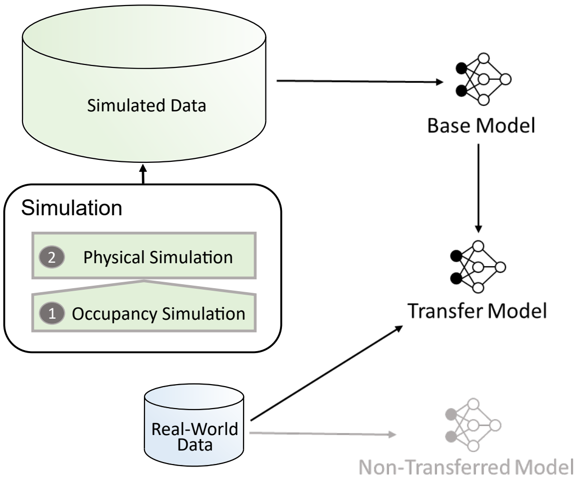

Fig. 3. Simulation approach based on [14]

B. Simulation approach

The presented dataset was generated following the approach from [14], [15], as illustrated in Fig. 3. It involves the automated generation of a large set of synthetic data for base model training, with the idea of using scarce real-world data only for fine-tuning instead of building a model from scratch. As proposed in [14], the process applied to generate the final dataset is divided into two consecutive simulation steps: Step 1 simulates occupancy times and counts, as well as ventilation behavior, without considering room properties. Step 2 simulates the environmental conditions within different room models based on the data from Step 1. Binarized occupancies from Step 1 are then used as labels and combined with the corresponding sequences of environmental values from Step 2 to create the final dataset. The simulation methods of each phase are described in more detail in the following subsections. In both simulation steps, we differ between the two most typical room types in commercial buildings: office rooms and meeting rooms.

C. Occupancy simulation

This simulation aims to generate occupancy patterns for office and meeting rooms. For each room type, we consider multiple maximum values of occupants allowed in the room: For office rooms, we consider 1 to 4 occupants; in meeting rooms, 4 to 12. Large open-space offices with typically more than four regular occupants are neglected, as for those, one can not assume a uniform room climate measurable with a single sensor. For each combination of room type and occupancy limit, we simulate 1000 different workdays. To simulate a workday, we combine two aspects of occupant behavior previously considered by Chen et al. [19]: status transitions and random movement. Status transitions refer to changes in the occupant's basic presence status, such as arrival at work. A simple approach on modeling status transitions in workplaces was used within Reinhart's LIGHTSWITCH-2002 [20] model for lighting control and considers events for arrival, departure, lunch break, and two short breaks. Transition times are based on typical office schedules but randomly selected within a defined 30-minute time interval. In addition, occupants may enter or leave a room several times, for instance, to prepare coffee or meet colleagues. This behavior can be referred to as random movement [19]. Wang et al. suggested a Markov chain approach to model the stochastic behavior of random movement [21]. In this work, we apply modified versions of the occupancy simulation approaches by Reinhart [20] and Wang et al. [21], both carried out for each individual occupant.

TABLE I
TIME SPANS FOR OCCUPANCY SIMULATION IN OFFICES

ID	Event	Starting Time	Duration
E0	arrival	between 6:00 and 13:00	-
E1	lunch	3-6 hours after arrival	30-90 min
E2	departure	6-10 hours after arrival	-
E3	short break 1	between arrival and lunch	5-20 min
E4	short break 2	between lunch and departure	5-20 min

Basic occupancy: For basic office occupancy, we use the same concept as the Lightswitch-2002 method but with a wider range of event times. This allows for more variety in the generated data. The time spans are defined in Table I. Events E0-E2 occur every simulated day, while short breaks have a 50% chance of being skipped. For meeting rooms, the described method is not applicable. In this case, we randomly define meeting schedules, where each meeting takes between 30 min and 2 hours, and each break between meetings takes 5 min to 4 hours. Additionally, meetings are restricted to begin between 8 am and 6 pm. For each meeting, we randomly select the number of meeting attendants between 1 and the maximum room occupancy. The attendants' arrivals and departures are then uniformly distributed within a time interval of ± 10 minutes around the beginning and end of the meeting.

Random movement: For random movement, Wang et al. [21] assume n building zones between which occupants may move and use a transition probability matrix of size $(n+1) \times (n+1)$. While this intends to simulate consistent occupancies across an entire building, occupant movement through the building is irrelevant for occupancy detection in a single zone. Hence, we use matrix P from (1), with a size of 2×2, representing inside and outside.

$$P = \begin{bmatrix} 1 - (\frac{1}{s_0}) & \frac{1}{s_0} \\ \frac{1}{s_1} & 1 - (\frac{1}{s_1}) \end{bmatrix} \tag{1}$$

For each simulated day and each potential occupant, we randomly select mean sojourn times s_0 and s_1, where s_0 refers to transitions from absence back to presence and s_1 from presence to absence. By allowing a broad range of mean sojourn times, we intend to mimic the behavior of different occupants and add more variety to the generated dataset. Specifically, we select minute values $s_0 \in [5, 20]$, $s_1 \in [120, 480]$ in office rooms and $s_0 \in [5, 20]$, $s_1 \in [12 \cdot 60, 48 \cdot 60]$ in meeting rooms. For meeting rooms, we choose a low probability of leaving, as we assume that occupants attending a meeting are less likely to leave during this time. Furthermore, we use another Markov chain with a transition matrix P' to model the occupants' stochastic window opening behavior. P' is equivalent to P but uses mean sojourn times $s_0 \in [4 \cdot 60, 12 \cdot 60]$, $s_1 \in [5, 20]$ in office rooms and $s_0 \in [60, 4 \cdot 60]$, $s_1 \in [5, 20]$ in meeting rooms. Here, s_0 refers to the expected minutes until the next opening action and s_1 until the next closing action. Both are selected individually for each day and occupant. Each present occupant may cause window opening or closing actions according to their individual probabilities.

D. Physical simulation

Based on previously simulated occupancy data, we populate a variety of room models followed by a physical simulation of their indoor climate. The simulation is conducted with the building energy simulation software EnergyPlus (EP) at version 22.2.0. EP is an open source software widely adopted in science and practice. While it is primarily used for building energy performance simulation, its functionality also includes the simulation of indoor climate conditions. By using EP, we rely on a well-established simulation engine and allow the method to be extended by further climate factors in the future. For both office and meeting room simulation, we use an initial room template modeled in EP's proprietary format IDF that is modified for each iteration of the simulation. The template we designed is a straightforward rectangular room with three initial windows. Fig. 4 shows a visualization of this room model. Table II shows the parameters to be modified and their respective value ranges. To ensure a representative sampling of the parameter space, values are chosen by Latin hypercube sampling (LHS). For this, we use the algorithm by Jin et al. [22] as implemented in [23]. First, LHS sampling determines values for the parameters P0-P6, and the corresponding room is constructed. Windows are added by placing as many windows as spatially feasible on the facade. Parameters P7 and P8, which implicitly result from room geometry (P0-P2) and maximum occupancy (P6), are then used to verify a realistic room size for the number of occupants and reasonable lighting conditions with respect to the ratio of window area to floor area. German building standards were used as guidance for setting the boundaries. Although, we assume that unrealistic room properties would not damage a deep learning model, we decided to limit simulations to realistic parameter selections to be able to train a useful model with less data.

After preparing the room models, we run an EP simulation over a period of 10 days for each room. A weather file for Munich, Germany, is used for all simulations and the start date is randomly selected between January 1st and December 22nd, 2023, to cover all seasons of the year. The human CO_2 generation rate is derived from the ANSI/ASHRAE standard 62.1-2007 [24] and set to $3.82 \cdot 10^{-8} m^3/sW$.

The activity level is set to $108\ W/1,8m^2 = 60\ W/m^2$. This equals the recommendation by the ANSI/ASHRAE standard

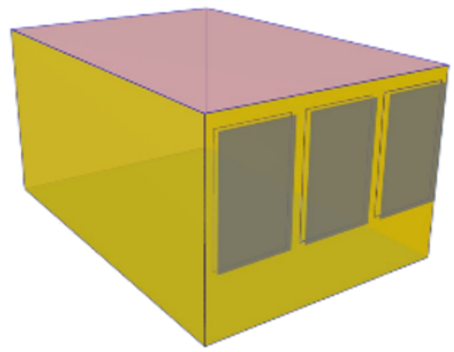

Fig. 4. Visualization of initial room model.

55-2010 [25] for a typical office activity that considers writing occupants. Ventilation is modeled in the way that all windows are opened simultaneously, each with an opening area of $1.025\ m^2$. For each window opening in the occupancy data, we assign a random fraction between 0.01 and 1 used as a multiplier for the opening areas, representing partially opened windows and increasing variety in the resulting room climate. EP applies the CO_2 balance equation given by (2) to calculate the change rate $\frac{dC_z}{dt}$ of the current CO_2 concentration C_z at timestep t for a zone z [26].

$$\rho_{\text{air}} V_z C_{\text{CO2}} \frac{dC_z}{dt} = 10^6 \cdot \sum C_{\text{sched}} + \sum C_{\dot{m}_{zi}} + C_{\dot{m}_{\text{inf}}} + C_{\dot{m}_{\text{sys}}} \quad (2)$$

V_z is the zone volume in m^3, ρ_{air} the air density in kg/m^3, C_{CO2} an air capacity multiplier, $\sum C_{\text{sched}}$ the sum of scheduled internal CO_2 loads in kg/s, and $C_{\dot{m}}$ corresponds to the CO_2 transfer in $ppm \cdot kg/s$ due to adjacent zone air mixing ($C_{\dot{m}_{zi}}$), infiltration and ventilation ($C_{\dot{m}_{\text{inf}}}$), or system supply ($C_{\dot{m}_{\text{sys}}}$).

IV. The Model

This section describes the model architecture and pretraining. All source code is available on GitHub[2].

[2]https://anonymous.4open.science/r/Coddora-88F3

TABLE II
VALUE BOUNDS FOR GENERATED ROOM MODELS

ID	Parameter	Min	Max	Unit
P0	length	2	10	m
P1	width	2	10	m
P2	height	2.4	4	m
P3	infiltration per exterior area	0.000085	0.00085	m³/m²s
P4	outdoor CO_2	300	500	ppm
P5	facade orientation	0	359	degrees
P6	maximum occupancy	1[a] 4[b]	4[a] 12[b]	
P7	space per occupant[c]	6[a] 2[b]	22[a] 6[b]	m² m²
P8	window-to-floor ratio[c]	0.1	0.6	

[a]applied for office rooms [b]applied for meeting rooms
[c]additional constraint on the selection of P0-P2

Accepted version of paper submitted to International Joint Conference on Neural Networks (IJCNN). Published in IJCNN, 2024, pp. 1-8, DOI: 10.1109/IJCNN60899.2024.10650820, URL: https://ieeexplore.ieee.org/document/10650820

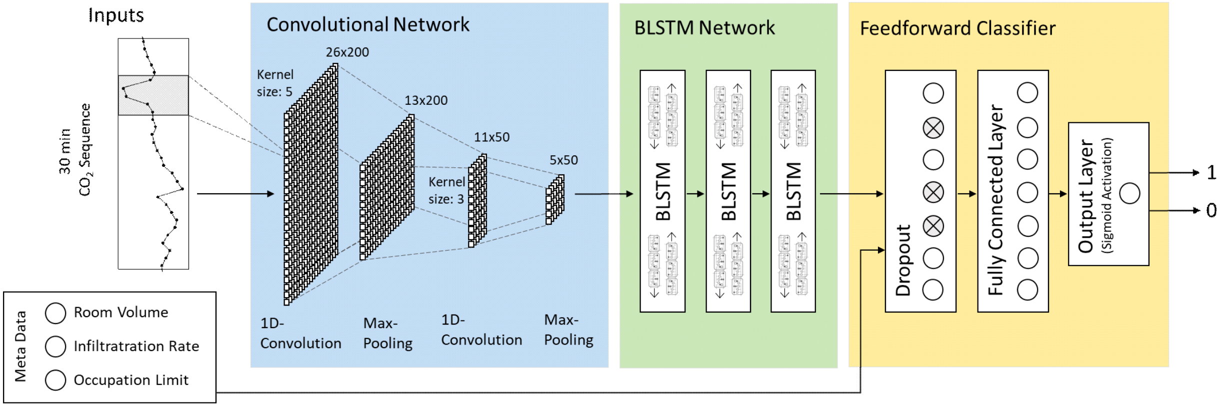

Fig. 5. Model architecture: A modified version of the architecture in [10], including room metadata as a second input.

A. Model Architecture

The proposed model is a convolutional deep bidirectional long short-term memory (CDBLSTM) inspired by the approach of Chen et al. [6] for predicting occupancy from environmental factors. It combines a 1-dimensional convolutional neural network (CNN) with a long short-term memory (LSTM), which is a typical approach also across other time series domains [27]. As input to these model components, we use CO_2 time sequences of 30 values at 1-minute intervals, which are obtained from the complete time series using a sliding window method. The task is to predict whether there is occupancy at the end of the given sequence. While the CNN extracts local patterns from the time sequence, the LSTM puts these into a temporal context. This is followed by a dense feedforward classifier to process the information on a higher abstraction level and results in a sigmoid output layer for final classification. Besides the features extracted from the time sequence, the feedforward classifier is additionally fed with metadata of the room to allow it to learn the relation to room properties explicitly. Here, we use three input features, which we assume most relevant regarding CO_2 dynamics: (1) the room's air volume in m^3, (2) the infiltration rate in m^3/m^2s, and (3) the expected maximum number of occupants. This is a major modification to the original model from [6], which was intended to be trained for a specific room only. The number of layers and other model parameters are derived from a hyperparameter tuning with data from five different rooms that was conducted in [10]. The according model is illustrated in Fig. 5 and consists of two 1D convolutions using 200 and 50 filters with kernel sizes of 5 and 3, each followed by max pooling with size 2, three bidirected LSTM (BLSTM) layers with 50 cells each, one dense layer with 100 neurons, and a dropout of 0.5. Backpropagation is done using binary cross-entropy loss and Adam optimizer.

B. Model Training

The following paragraphs describe the pretraining procedure based on the model architecture from Sec. IV-A. We trained two versions of the model, one using simulated data only (Coddora), and one with additional real-world data (CoddoraRW).

Coddora: To generate the Coddora base model, the model was randomly initialized and pretrained with the entire dataset from Sec. III, with 1000 of the simulated rooms of each room type being used for validation. During data preprocessing, all values were normalized by a min-max normalization. CO_2 values were normalized to the acceptable range between 300 and 5000 ppm. Outliers were clipped to ensure that all values are within the range of 0 to 1. Metadata were normalized according to the reported value bounds from Table II. The training was conducted with a batch size of 512. For implementation, we used the Tensorflow framework at version 2.6. Model training was run on an NVIDIA GeForce Tesla V100 SXM2 GPU. As multiple passes over the training data, up to epoch 6, showed no improvement in our evaluation, we selected the model weights after epoch 1 for the final model.

CoddoraRW: After preparing the purely simulation-based Coddora model, we retrained it with real-world data from a residential room in Munich, Germany, that was published in [10]. The dataset contains 50 days of CO_2 data, of which we used 45 for training and 5 for validation. Based on validation results, we applied early stopping after 5 epochs without model improvement. Due to the lower amount of training data, in contrast to the pertaining on simulation data, the batch size was reduced to 128 for real-world retraining. The lowest validation loss was reached at epoch 14, resulting in the selection of its model weights as the final model. CodorraRW is intended to show improved adaptability to other real-world scenarios. The dataset for retraining was selected according to results from [10], where it showed higher generalizability in TL compared to other real-world datasets. An advantage is that, while long periods of absence do not provide informative data, this dataset offers an untypically high presence rate, as it was measured during the Covid-19 lockdown between April and July 2020. For more details on the dataset, we refer to [10].

Accepted version of paper submitted to International Joint Conference on Neural Networks (IJCNN). Published in IJCNN, 2024, pp. 1-8, DOI: 10.1109/IJCNN60899.2024.10650820, URL: https://ieeexplore.ieee.org/document/10650820

V. Evaluation

For evaluation, we report performances in the zero-shot case and after fine-tuning on target data. In both cases, we use two datasets containing data from an office room and a meeting room, respectively. The office dataset was published in [28] as *Office A* and previously used for occupancy detection in [9], [10]. It contains data from an academic office in Germany. The meeting room data were previously used for occupancy detection in [10], [11]. The dataset equals the one used in [10] and contains data from a 12-person hospital meeting room in Finland. The data were initially measured at a 3-min resolution [11] but upsampled to 1-min intervals via linear interpolation [10]. The following subsections introduce the evaluation metrics and present the results from the zero-shot and fine-tuning evaluations.

A. Evaluation metrics

We evaluate by accuracy and Matthews correlation coefficient (MCC). These are calculated according to (3) and (4), where tp and tn are true positives and negatives, and fp and fn are false positives and negatives.

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + tn + fn} \tag{3}$$

$$\text{MCC} = \frac{tp \cdot tn - fp \cdot fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \tag{4}$$

The accuracy was used due to simple interpretability and popularity in related research. Besides, MCC serves as an additional metric to better assess the imbalance in the data: rooms in commercial buildings tend to have longer periods of vacancy in comparison to occupation. While the accuracy only considers tp and tn predictions, MCC gives a balanced evaluation with regard to all four prediction categories.

B. Zero-shot

We tested the two versions of our model on the last 20 days from both evaluation datasets. The results are reported in Tab. III. While the simulation-trained Coddora model did not show a useful performance, the results for CoddoraRW indicate that it may already make sense to apply this model in certain real-world applications without additional training. For both room types, it reached a solid accuracy of above 90%. However, the MCC shows that it is more suitable for the office room, where it shows a strong correlation between predicted and true labels (0.8123). In the meeting room case, the correlation is moderate and the high accuracy value is possible due to large periods of vacancy that are correctly classified.

TABLE III
Zero-shot results using Coddora models

Model	Office Room		Meeting Room	
	Accuracy	***MCC***	***Accuracy***	***MCC***
Coddora	0.5690	0.3237	0.3383	0.0204
CoddoraRW	0.9149	0.8123	0.9205	0.3735

C. Target fine-tuning

Enhanced and more reliable performance is expected when using labeled target data for fine-tuning. The following evaluation addresses fine-tuning with different amounts of data available for the target room. Fig. 6 shows the performance for multiple experiments with up to 40 target training days in the office and up to 20 in the meeting room scenario. Data from one day correspond to 1411 CO_2 sequences obtained with a 30-min moving window. For each experiment with a number of k days of training data and a total number of l days included in the dataset, the subsequent $l-k$ days were used for testing. The office dataset contains $l = 80$ days; the meeting room dataset $l = 26$ days. Within each train-test split, the last 20% of training data were used for validation. Again, training was conducted with a batch size of 128 and early stopping was applied after 5 epochs without validation loss improvement. Each experiment was repeated 20 times using distinct seed values. For comparison, we trained the model without metadata from scratch, as well as after pretraining with the same real-world dataset used to train CoddoraRW.

Scarce data: As shown in Fig. 6, in our evaluation, it was not possible to train a useful model from scratch with scarce real-world data. With less than six days from the office room dataset, accuracies below 0.75 and MCC values between 0 to 0.2 indicate weak performance close to random predictions. In contrast to this, fine-tuning Coddora led to a model performing reasonably well even with a single workday used for training. Data from two workdays were sufficient to reach a substantial performance that required multiple workweeks of data when training from scratch: In the office scenario, 17 days of data were required to reach a similar MCC; in the meeting room scenario 13 days.

CoddoraRW: When fine-tuning with a single training day, CoddoraRW performed superior to Coddora in both evaluation scenarios. However, the advantage diminished with larger training amounts. With at least two target days, both provided a performance above 90% accuracy and with larger training datasets, both were able to reach performances above 95%.

Non-scarce data: For the meeting room, larger training data of more than 13 days allowed to approach the performance of the fine-tuned models also without pretraining. However, in the office room case, the fine-tuned models constantly outperformed the training from scratch, even with extensive training data. While Fig. 6 shows the results for up to 40 workdays, we extended the evaluation up to 75, which is equivalent to 15 workweeks of labeled training data, and continuously observed superior performance of Coddora.

Performance stability: As data collection on different days does not always provide data with the same informative value, it is possible that using an additional day for training weakens the model's performance. This can, for instance, be observed in Fig. 6 (a) from the performance declines when adding day 5 or day 11. In case of fine-tuning with Coddora or CoddoraRW, however, specific training days did not cause major performance decreases. The same positive effect was also observed

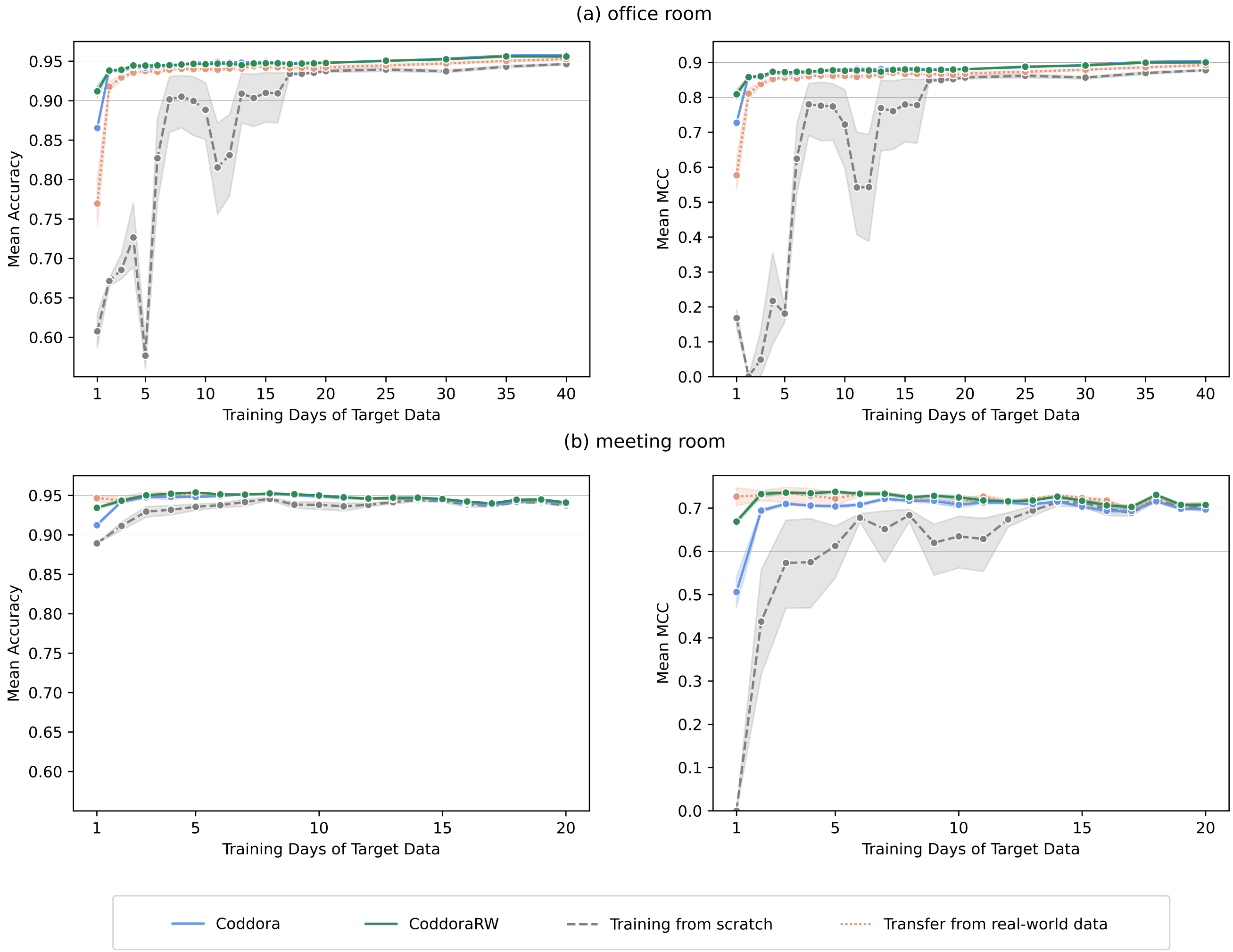

Fig. 6. Mean test accuracy and MCC by base model and target data when fine-tuning on (a) office or (b) meeting room data. Each value is calculated from 20 repetitions. Shaded areas show the 95% confidence intervals.

for TL from a large real-world dataset. Moreover, training from scratch resulted in high variations across repeated experiments using identical input data. In contrast, the fine-tuned models yielded more consistent results.

Comparison to real-world transfer learning: Transfer from other real-world datasets promises similar advantages over training from scratch, in case sufficient suitable data is available. In the meeting room evaluation, this approach performed similar to CoddoraRW, which showed no additional advantage from simulation-based pretraining. However, for the office room, the Coddora models showed clear superiority in scarce data scenarios with less than five training days, while also yielding slightly higher evaluation values with increased quantities of target data.

VI. Discussion

This work focuses on the enhancement of model creation for occupancy detection in an arbitrary room of interest, primarily in a commercial building context. Bringing deep learning to practice in this context can drive energy savings in building engineering, having a direct impact towards climate change. With this regard, the primary focus lies on ensuring practicality of the approach rather than reaching maximum accuracy. According to the results from Sec. V, Coddora provides a stable performance beyond 90% accuracy despite data scarcity. CoddoraRW was able to achieve accuracies above 90% even when applied as a zero-shot classifier. Both base models enable training a strong classifier for a particular room without extensive manual data collection. The observed accuracies in the range of 90-95% indicate that although the model is unable to determine the correct occupancy state for every minute of the day, it is reliable in terms of identifying major periods of presence or absence. For the analysis of long-term presence patterns or occupant-centric HVAC control, temporary occupancy fluctuations are of minor importance. Hence, according to our results, we consider models trained upon Coddora or CoddoraRW feasible for integration into building energy optimization. This can, for instance, be accomplished by using an occupancy-based strategy for HVAC control. Regarding this approach, Kong et al. [29] analyzed the impact of different levels of occupancy detection accuracy in a side-by-side experimental study. In weekly experiments with 76% and 87% of accuracy, they obtained mean energy savings of 19% and 24% while maintaining occupant comfort.

However, our results were demonstrated for a specific office and meeting room, and may not generalize to all possible target rooms. Potential threats to model performance include:

1. *Room type:* The model may not generalize well to rooms that strongly differ from the applied simulation, i.e., types other than offices or meeting rooms. For instance, we do not expect a similar performance in a lecture hall.
2. *Architecture:* The conducted simulation considers rectangular floor plans and specific configurations defined in the room template. Particular architectural aspects such as protrusions may have negative impact on model performance, especially when paired with suboptimal sensor placement.
3. *Room properties outside the boundaries:* As the model was trained on data generated for room parameters within the value bounds from Table II, it is not intended for prediction on differing rooms, such as a room designed for more than 12 regular occupants or with a length longer than 10 m.
4. *Occupant behavior:* As it was previously pointed out in [10], anomalous behavior of occupants, such as ventilating through the office door, may cause negative transfer. In cases where a certain unique behavior is common, a possible solution is to implement this specific behavior into simulation and train a more specialized base model.
5. *Regional differences:* The simulation is based on a weather file from Munich, Germany, and outdoor CO_2 was set to random values between 300 and 500 ppm. Air pressure and CO_2 levels vary for different parts of the world. However, we expect only a marginal impact, since outdoor CO_2 levels are typically lower than indoor levels and patterns do not differ significantly.

Our approach can be extended to further spacial domains, such as other types of rooms currently not supported by Coddora, by a dedicated simulation, modifying room template, weather data, occupancy patterns or room parameters.

VII. Conclusions

In this paper, we demonstrated the effectiveness of transfer learning from simulated data for building occupancy detection. The proposed model can be used as an out-of-the-box pretrained model to detect occupancy from CO_2 and provides high accuracy after being retrained on scarce data from the target room. By substantially decreasing the necessary extent of ground truth data collection, this facilitates a fast adaptation of CO_2-based occupancy detection in practice. In our evaluation, fine-tuning with 1-2 training days from the target room was sufficient to obtain a solid classifier with above 90% accuracy.

References

[1] IEA, "World energy outlook 2023," 2023. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2023

[2] J. Brooks, S. Goyal, R. Subramany, Y. Lin, T. Middelkoop, L. Arpan, L. Carloni, and P. Barooah, "An experimental investigation of occupancy-based energy-efficient control of commercial building indoor climate," *IEEE Conf. Decision and Control (CDC)*, vol. 53, pp. 5680–5685, 2014.

[3] L. Rueda, K. Agbossou, A. Cardenas, N. Henao, and S. Kelouwani, "A comprehensive review of approaches to building occupancy detection," *Building and Environment*, vol. 180, p. 106966, 2020.

[4] M. K. Masood, Y. C. Soh, and V. W.-C. Chang, "Real-time occupancy estimation using environmental parameters," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2015.

[5] T. Liu, Y. Li, Z. Bai, J. De, C. V. Le, Z. Lin, S.-H. Lin, G.-B. Huang, and D. Cui, "Two-stage structured learning approach for stable occupancy detection," *International Joint Conference on Neural Networks (IJCNN)*, pp. 2306–2312, 2016.

[6] Z. Chen, R. Zhao, Q. Zhu, M. K. Masood, Y. C. Soh, and K. Mao, "Building occupancy estimation with environmental sensors via cd-blstm," *IEEE Transactions on Industrial Informatics*, vol. 64, no. 12, pp. 9549–9559, 2017.

[7] M. S. Zuraimi, A. Pantazaras, K. A. Chaturvedi, J. J. Yang, K. W. Tham, and S. E. Lee, "Predicting occupancy counts using physical and statistical co2-based modeling methodologies," *Building and Environment*, vol. 123, pp. 517–528, 2017.

[8] M. Jin, N. Bekiaris-Liberis, K. Weekly, C. J. Spanos, and A. M. Bayen, "Occupancy detection via environmental sensing," *IEEE Trans. Automation Science and Engineering*, vol. 15, no. 2, pp. 443–455, 2018.

[9] F. Banihashemi, M. Weber, F. Deghim, C. Zong, and W. Lang, "Occupancy modeling on non-intrusive indoor environmental data through machine learning," *SSRN*, 2023. [Online]. Available: http://dx.doi.org/10.2139/ssrn.4477794

[10] M. Weber, F. Banihashemi, P. Mandl, H.-A. Jacobsen, and R. Mayer, "Overcoming data scarcity through transfer learning in co2-based building occupancy detection," *Proc. 10th ACM BuildSys*, pp. 1–10, 2023.

[11] D. Stjelja, J. Jokisalo, and R. Kosonen, "Scalable room occupancy prediction with deep transfer learning using indoor climate sensor," *Energies*, vol. 15, no. 6, 2022.

[12] M. Khalil, S. McGough, Z. Pourmirza, M. Pazhoohesh, and S. Walker, "Transfer learning approach for occupancy prediction in smart buildings," *Intern. Renewable Engineering Conference (IREC)*, pp. 1–6, 2021.

[13] T. Zhang and O. Ardakanian, "A domain adaptation technique for fine-grained occupancy estimation in commercial buildings," *Proc. Internet of Things Design and Implementation (IoTDI)*, pp. 148–159, 2019.

[14] M. Weber, C. Doblander, and P. Mandl, "Detecting building occupancy with synthetic environmental data," *ACM BuildSys*, p. 324–325, 2020.

[15] ——, "Towards the detection of building occupancy with synthetic environmental data," *arXiv preprint arXiv:2201.01234*.

[16] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *IEEE/RSJ IROS*, 2017.

[17] Y. Rotem, N. Shimoni, L. Rokach, and B. Shapira, "Transfer learning for time series classification using synthetic data generation," *Cyber Security, Cryptology, and Machine Learning: 6th International Symposium (CSCML)*, pp. 232–246, 2022.

[18] OSHA, "Carbon dioxide: Osha occupational chemical database," 2023. [Online]. Available: https://www.osha.gov/chemicaldata/183

[19] Y. Chen, T. Hong, and X. Luo, "An agent-based stochastic occupancy simulator," *Building Simulation*, vol. 11, no. 1, pp. 37–49, 2018.

[20] C. F. Reinhart, "Lightswitch-2002: a model for manual and automated control of electric lighting and blinds," *Solar Energy*, vol. 77, no. 1, pp. 15–28, 2004.

[21] C. Wang, D. Yan, and Y. Jiang, "A novel approach for building occupancy simulation," *Building Simulation*, vol. 4, no. 2, pp. 149–167, 2011.

[22] R. Jin, W. Chen, and A. Sudjianto, "An efficient algorithm for constructing optimal design of computer experiments," *Journal of Statistical Planning and Inference*, vol. 134, no. 1, pp. 268–287, 2005.

[23] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins, "A python surrogate modeling framework with derivatives," *Advances in Engineering Software*, p. 102662, 2019.

[24] ANSI/ASHRAE, *ANSI/ASHRAE Standard 62.1-2007: Ventilation for Acceptable Indoor Air Quality*. ASHRAE Atlanta, Ga, 2007.

[25] ——, *ANSI/ASHRAE Standard 55-2010: thermal environmental conditions for human occupancy*. ASHRAE Atlanta, Ga, 2010.

[26] U.S. Department of Energy, "Energyplus™ version 22.2.0 documentation: Engineering reference," 2022. [Online]. Available: energyplus.net/assets/nrel_custom/pdfs/pdfs_v22.2.0/EngineeringReference.pdf

[27] M. Weber, M. Auch, C. Doblander, P. Mandl, and H.-A. Jacobsen, "Transfer learning with time series data: A systematic mapping study," *IEEE Access*, vol. 9, pp. 165 409–165 432, 2021.

[28] F. Banihashemi and M. Weber, "Non-intrusive indoor environmental data in two double-occupied offices," *Mendeley Data, V1*, 2023. [Online]. Available: https://dx.doi.org/10.17632/v7kw9ycccp.1

[29] M. Kong, B. Dong, R. Zhang, and Z. O'Neill, "Hvac energy savings, thermal comfort and air quality for occupant-centric control through a side-by-side experimental study," *Applied Energy*, vol. 306, p. 117987, 2022.

Appendix D

Detecting Building Occupancy with Synthetic Environmental Data

Reference:

Manuel Weber, Christoph Doblander, and Peter Mandl. "Detecting Building Occupancy with Synthetic Environmental Data." In: *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys)* (2020), pp. 324–325. DOI: 10.1145/3408308.3431124. URL: https://doi.org/10.1145/3408308.3431124

Poster Abstract: Detecting Building Occupancy with Synthetic Environmental Data

Manuel Weber
manuel.weber@hm.edu
Munich University of Applied Sciences
Munich, Germany

Christoph Doblander
christoph.doblander@in.tum.de
Technical University Munich (TUM)
Munich, Germany

Peter Mandl
peter.mandl@hm.edu
Munich University of Applied Sciences
Munich, Germany

ABSTRACT

Information about room-level occupancy is crucial to many building-related tasks, such as building automation or energy performance simulation. Carbon dioxide levels and other indoor environmental factors can be used as a proxy to detect occupancy. In this regard, machine learning solutions have been proposed, with solid performance in detecting presence, as well as counting the number of present occupants, if enough training data is available. The challenge is, to collect sufficient room-specific ground truth data for model training. With this poster, we address the use of knowledge transfer from synthetic data to reduce the amount of required real world data. We outline two approaches for the combination of transfer learning with physical simulations, and motivate the generation of additional synthetic data. Our results show that the required real world training data can be reduced by 50%.

CCS CONCEPTS

• Human-centered computing → Ubiquitous and mobile computing; **• Computing methodologies → Machine learning**.

KEYWORDS

Synthetic data, CO2, transfer learning, deep neural network

ACM Reference Format:

Manuel Weber, Christoph Doblander, and Peter Mandl. 2020. Poster Abstract: Detecting Building Occupancy with Synthetic Environmental Data. In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3408308.3431124

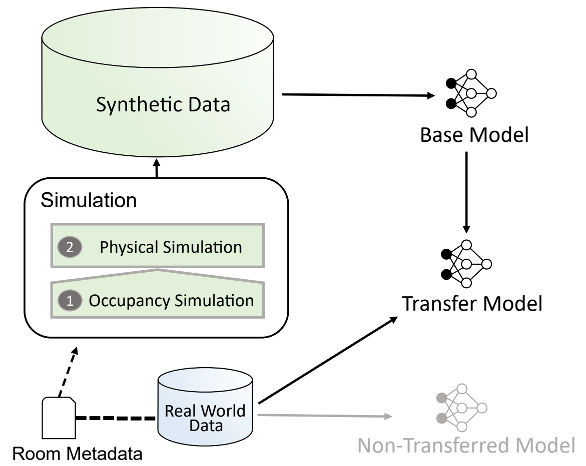

Figure 1: Simulation-aided approach to occupancy detection

1 INTRODUCTION

Human presence is correlated with changes in the indoor climate, in particular the CO_2 level. Although there is a natural lag between human CO_2 generation and sensor perception, it allows to derive occupancy within a reasonable time window, especially for small or medium-sized rooms. Knowing building occupancy on a room-level is essential in many building-related use cases. Application areas include energy performance simulation, building automation or emergency evacuation. Environmental sensing provides several advantages over other sensing technologies. It is more privacy preserving than cameras, and in contrast to light barriers, it does not accumulate errors when occupants enter or leave a room undetected. Machine learning models trained on environmental factors are able to provide solid near-time predictions towards occupancy presence (binary), as well as the count of present occupants [1, 2, 4, 8]. CO_2 was identified as the most important climate factor to detect occupancy [2]. A CO_2-based estimation model can effectively be used in any use case where no down-to-the-minute occupancy data is required, e.g., for annual energy performance simulation.

However, regarding large commercial buildings, the effort of collecting ground truth data in each room impedes applicability in practice. In the same time, simple thresholding solutions are less promising, as measured CO_2 values are influenced by a variety of factors, such as sensor placement, infiltration rates or office plants. Simulated data has been used in other building-related learning tasks, for example in [3], to reduce training times of a reinforcement learning model controlling heating, ventilation, and air conditioning (HVAC) systems. To the best of our knowledge, it has not been studied regarding occupancy detection, which can be used not only in context of HVAC control, but for a variety of applications. Previous studies [1, 7] that apply a transfer of an occupancy model between different rooms show improvements when inferring occupancy from CO_2. Our contribution is a method to lower the amount of required training data for occupancy detection by pretraining with simulated data (see Fig. 1). Further details can be found in [6].

BuildSys '20, November 18–20, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8061-4/20/11.
https://doi.org/10.1145/3408308.3431124

2 SIMULATION-AIDED PREDICTION

Fig. 1 shows our approach on occupancy detection. The idea is to first train a base model on simulated data, and later retrain the model with the given real world data. While the base model can already fit general rules of indoor CO_2 dynamics, the transfer step adjusts the model to room-specific conditions. Taking into account room types and typical occupancy profiles, as well as room metadata such as its dimensions, we conduct two consecutive simulations: (1.) An *occupancy simulation* to generate synthetic ground truth data, including occupancy states for one or multiple occupants, and possibly also occupant behavior (e.g., window opening).
(2.) A *physical simulation*, based on (1.), to generate synthetic CO_2 data, and possibly further environmental data such as temperature or humidity. Room-specific training is still necessary, as conditions differ between rooms, in terms of sensor placement, infiltration rates etc. We distinguish two approaches:

(a) Upfront simulation and pretraining dedicated for the specific room of interest. The simulation process may be automated to complement training procedures.
(b) One-time training of a reusable, general base model. In this case, the base model is dedicated for a group of similar rooms, and simulations are performed for multiple randomly generated room characteristics. Apart from performance improvements with sparse training data, this approach further aims to reduce training times during model application.

In this work, we use approach (a) to conduct a first experiment.

3 RESULTS

We conducted a first experiment in a two-person office room of size 77.5 m³ in a university building in Munich, Germany, to demonstrate the positive effects of transfer learning from synthetic data. We generated 400 days of occupant data for one virtual occupant, and further derived CO_2 values using physical equations described in [5]. Within the simulations several simplifications were made. Infiltration rate, outdoor CO_2 rate and human CO_2 generation rate were held constant, and set to realistic values of 0.0046 m³/s, 360 ppm and 0.24 l/min. Real world data was collected for 7 working days via a Sensirion SCD30 non-dispersive infrared (NDIR) CO_2 sensor and manual recording of occupancy. We applied the deep learning architecture proposed in [4], adjusting it to the case of binary detection from CO_2 only. The model was trained once only on real world data, and once including pretraining. In the latter case, the model was first trained on the simulated data, and model weights were then used for initialisation of the real world training. For evaluation, we applied a cross validation using each day in the dataset, respectively each 2 or 4 consecutive days, for training in one iteration, and the remaining days as test data. Tab. 1 shows the resulting test accuracies and F1 scores for training with 1, 2 and 4 days of training data. It further provides a baseline comparison. Given the scarce amount of data, the model showed clear improvement in the transfer setting: With one day (2 days) of training data, the transfer model exceeded the performance of the non-transferred model trained on 2 days (4 days). Hence, only half of the training data was required. In the same time, model robustness was improved, with standard deviations reduced by up to 50%.

Table 1: Test accuracy and F1 score for the transfer model, non-transferred model and logistic regression (LR) baseline with 1, 2 and 4 days of training data*

		1 Day	2 Days	4 Days
Transfer	Accuracy	0.875 (±0.057)	0.915 (±0.020)	0.933 (±0.015)
	F1 Score	0.764 (±0.171)	0.863 (±0.035)	0.898 (±0.028)
No Transfer	Accuracy	0.715 (±0.086)	0.874 (±0.040)	0.914 (±0.029)
	F1 Score	0.565 (±0.162)	0.791 (±0.065)	0.859 (±0.053)
LR	Accuracy	0.860 (±0.073)	0.879 (±0.058)	0.920 (±0.016)
	F1 Score	0.767 (±0.112)	0.803 (±0.080)	0.866 (±0.038)

*mean values (± standard deviations) are reported for 10 runs in each split

4 CONCLUSION & FUTURE RESEARCH

We propose a simulation-aided approach on occupancy detection in building rooms. Our experiment showed promising results: Required real world data could be reduced by half. The approach has the potential to significantly reduce the time to put a model into production. It also showed remarkable improvement in model robustness. In future work, we intend to find a concrete method concerning transfer as well as simulation of data that is most useful for pretraining. More detail may be added to the simulation, e.g., using historic weather data instead of a constant outdoor CO_2 rate. We are also planning on investigating other model architectures. The method needs to be validated on a variety of rooms, as it may be more or less beneficial for some room types or occupancy profiles. Considering imperfect air mixing, for large rooms, it can also be necessary to include data from multiple sensors. Moreover, it is particularly interesting to explore the limits of a transfer from simulated data, and to compare results to simulation-free alternatives for data generation, such as methods for time series data augmentation.

ACKNOWLEDGMENTS

Special thanks go to the *OpenPower@TUM* project and the industry partner IBM for providing us with the computing infrastructure.

REFERENCES

[1] I. B. Arief-Ang, F. D. Salim, and M. Hamilton. 2017. DA-HOC: Semi-Supervised Domain Adaptation for Room Occupancy Prediction using CO2 Sensor Data. *4th ACM BuildSys* (2017), 1–10.
[2] L. M. Candanedo and V. Feldheim. 2016. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings* 112 (2016), 28–39.
[3] B. Chen, Z. Cai, and M. Bergés. 2019. Gnu-RL: A Precocial Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy. *6th ACM BuildSys* (2019), 316–325.
[4] Z. Chen, R. Zhao, Q. Zhu, M. K. Masood, Y. C. Soh, and K. Mao. 2017. Building Occupancy Estimation with Environmental Sensors via CDBLSTM. *IEEE Transactions on Industrial Electronics* 64, 12 (2017), 9549–9559.
[5] P. Parsons. 2014. Determining Infiltration Rates and Predicting Building Occupancy Using CO2 Concentration Curves. *Journal of Energy* (2014), 1–6.
[6] M. Weber, C. Doblander, and P. Mandl. 2020. Towards the Detection of Building Occupancy with Synthetic Environmental Data. arXiv:2010.04209
[7] T. Zhang and O. Ardakanian. 2019. A domain adaptation technique for fine-grained occupancy estimation in commercial buildings. *Proceedings of the 2019 Internet of Things Design and Implementation (IoTDI)* (2019), 148–159.
[8] M. S. Zuraimi, A. Pantazaras, K. A. Chaturvedi, J. J. Yang, K. W. Tham, and S. E. Lee. 2017. Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. *Building and Environment* 123 (2017), 517–528.