Journal of Intelligent Manufacturing (2022) 33:441–455
https://doi.org/10.1007/s10845-021-01804-0

Towards scalable and reusable predictive models for cyber twins in manufacturing systems

Cinzia Giannetti[1] · Aniekan Essien[2]

Received: 23 December 2020 / Accepted: 16 June 2021 / Published online: 29 July 2021

Abstract
Smart factories are intelligent, fully-connected and flexible systems that can continuously monitor and analyse data streams from interconnected systems to make decisions and dynamically adapt to new circumstances. The implementation of smart factories represents a leap forward compared to traditional automation. It is underpinned by the deployment of cyberphysical systems that, through the application of Artificial Intelligence, integrate predictive capabilities and foster rapid decision-making. Deep Learning (DL) is a key enabler for the development of smart factories. However, the implementation of DL in smart factories is hindered by its reliance on large amounts of data and extreme computational demand. To address this challenge, Transfer Learning (TL) has been proposed to promote the efficient training of models by enabling the reuse of previously trained models. In this paper, by means of a specific example in aluminium can manufacturing, an empirical study is presented, which demonstrates the potential of TL to achieve fast deployment of scalable and reusable predictive models for Cyber Manufacturing Systems. Through extensive experiments, the value of TL is demonstrated to achieve better generalisation and model performance, especially with limited datasets. This research provides a pragmatic approach towards predictive model building for cyber twins, paving the way towards the realisation of smart factories.

Keywords Cyber physical systems · Transfer learning · ConvLSTM · Smart manufacturing · Deep learning

Introduction

Cyber Manufacturing Systems (CMS), also referred to as Cyber Physical Production Systems (CPPS), are considered the building blocks of digitalised production systems. They are defined as advanced mechatronic systems, which use and transform data (and knowledge) from interconnected systems into predictive and prescriptive manufacturing operations to achieve resilient performance (e.g. self-optimisation, self-maintenance and self-learning) (Jeschke et al. 2017). CMS are composed of collaborating and automated computational entities that enable the connectivity of operations and physical processes and, through data exchange, support rapid decision-making (Lee et al. 2018). A layered architecture for the implementation of CMS is proposed in Lee et al. (2015), consisting of five hierarchical levels: (i) smart connection; (ii) data to information conversion; (iii) cyber; (iv) cognition; (v) configuration.

At the conversion level, cyber-twins (or digital twins) of each component/machine are responsible for collecting sensory data and synthesizing future steps to provide self-awareness and self-prediction capabilities (Lee et al. 2015). The data, which is typically in the form of time series, is either directly measured by sensors or obtained from controllers and enterprise manufacturing systems. Information and knowledge from the conversion level is then passed to the subsequent hierarchical level (i.e. cyber) to achieve self-configuration and optimisation of the machine fleet.

Developing robust predictive models for cyber twins is typically considered a challenging task, given the dynamic and often stochastic nature of time series signals. Traditional (or shallow) ML approaches, which have been successfully used to analyse batch data in production processes (Ransing et al. 2013; Giannetti et al. 2014; Ransing et al. 2016;

✉ Cinzia Giannetti
c.giannetti@swansea.ac.uk

Aniekan Essien
A.E.Essien@sussex.ac.uk

[1] Future Manufacturing Research Institute (FMRI), Faculty of Science and Engineering, Bay Campus, Swansea University, Swansea, UK

[2] Department of Management, University of Sussex Business School, Falmer, Brighton, UK

Giannetti and Ransing 2016), fail to capture the dynamic behaviour of machines in manufacturing lines, resulting in predictive models that require re-calibration and re-training on a continuous basis. Deep Learning (DL) has been identified as a promising technology to realise self-prediction and self-awareness features of CMS due to its ability to discover complex patterns in high dimensional data (Lee et al. 2020). DL broadly refers to a particular class of techniques for learning high-level features from data in a hierarchical manner using stacked, layer-wise architectures (Goodfellow et al. 2016). However, from an implementation viewpoint, a major drawback of DL is the reliance on large amounts of data and high-performance computing resources for model training. Additionally, finding the best network architecture and model hyper-parameter optimisation require specific data science expertise, which is typically unavailable in manufacturing organisations. Therefore, the development (and deployment) of DL models for the various machines/components in a manufacturing plant may become impractical, hindering the realisation of smart production systems.

Transfer Learning (TL) is an emerging Machine Learning (ML) paradigm that promotes the efficient training of predictive models by reusing knowledge gained from a previous task(s) on a new task or domain. Lee et al. (2020) argue that TL can play an important role to promote the rapid implementation of cyber models. Through reuse of pre-trained models, TL can support the development of scalable predictive models for CMS, where scalability refers to the ability to train and deploy predictive models without using large computational and time resources. TL is also beneficial when there is a limited supply of the target data. This is either due to the data being rare (such as in extremely rare occurrences), expensive to label (i.e. cost of human labelling), or noisy/unclean. Therefore, TL can also support the development of robust predictive models, where robustness is intended as the ability of a predictive model to keep a satisfactory level of performance on out-of-distribution/noisy samples. For instance, TL can be useful when the prediction task is performed using data from different machines or when data distributions change over time. This scenario is very common in manufacturing where signals from production machines and equipment are dynamic, as they are subject to transient disturbances and/or system degradation over time. The application of TL approaches to the manufacturing domain is an emerging—but relatively new—research area. So far, the vast majority of TL studies in the manufacturing domain address specific challenges in fault diagnosis and prognostics (Xu et al. 2019; Wen et al. 2019; Cao et al. 2020; Zhang et al. 2019; Sun et al. 2019; Zellinger et al. 2020). In general, TL was used to: (i) leverage advances in image recognition (through reuse of pre-trained CNN models); (ii) transfer knowledge between simulated and real environments; (iii) domain adaptation, taking into account changes in data distribution for different fault conditions. However, neither of these studies consider the transfer of knowledge across different machines or factories to improve scalability and the rapid deployment of predictive models for cyber twins of machines, as hypothesised in Lee et al. (2020).

To address this research gap, this paper presents an extensive experimental study to demonstrate the effectiveness of TL approaches for building robust and scalable predictive models for Cyber Manufacturing Systems. More specifically, different TL strategies are applied across machines in a manufacturing plant to classify the speed of a can body-maker machine in a metal packaging manufacturing process. The experimental study shows the effectiveness of applying TL to improve generalisation performance of predictive models when there is limited data, including significant reduction in training time and computational requirements. These findings pave the way for the effective implementation of TL in digital factories, supporting the development of robust and scalable DL predictive models for cyber twins.

The paper is structured as follows. In Sect. 2, previous research related to TL in the manufacturing domain is presented. The research problem statement and methodology are described in Sect. 3. Section 4 presents the results of the empirical analysis, while the paper is concluded in Sect. 5.

Transfer learning background

Different from the conventional ML paradigm, Transfer Learning (TL) breaks the constraint that training and test data should have the same distribution (Pan and Yang 2010). TL is used to improve a model in a target domain by transferring information from another (i.e. source) related domain. In contrast to traditional ML approaches, TL is effective even in instances when the domains, tasks, and distributions used in training and testing are different (Lu et al. 2015). To date, TL methods have been applied to many real-world scenarios, including time series forecasting and classification (Zellinger et al. 2020), natural language processing (Zeng et al. 2019), sentiment analysis and image recognition (Lu et al. 2015; Weiss et al. 2016; Flynn and Giannetti 2021). In particular, TL has gained popularity in solving image recognition problems due to advances in the field of computer vision and the availability of pre-trained models that have been trained on large image dataset such as ImageNet (Deng et al. 2009). The availability of large time series repositories has made TL more attractive, resulting in its increased research interest and application towards sequential or time series problems, such as text classification (Wang and Mahadevan 2011) and time series forecasting (Kashiparekh et al. 2019; Fawaz et al. 2019).

TL in manufacturing

A literature review related to the application of Transfer Learning to manufacturing was carried out and relevant papers are summarized in Table 1. All the studies presented in the table have applied TL, demonstrating its value in improving generalisation and predictive performance in a manufacturing domain. In the table, the column “Real Dataset” is used to refer to studies that have used either real-world datasets or online/synthetic/simulated datasets (for instance, the CMAPPS Engine failure Turbofan dataset).

As can be seen from Table 1, the majority of the studies reviewed are in the area of fault diagnosis (Xu et al. 2019; Wen et al. 2019, 2017, 2019; Xiao et al. 2019; Yang et al. 2019; Li et al. 2020; Zhao et al. 2020) and only four of them utilise real-world datasets. In Xu et al. (2019), TL is used to implement a digital-twin-assisted fault diagnosis system for car body-side production, transferring knowledge between the virtual and real space. In Wen et al. (2019), the authors proposed a negative correlation ensemble transfer learning model (NCTE) for fault diagnosis based on convolutional neural networks (CNN), which uses pre-trained CNN models for feature extraction. Cao et al. (2020) proposed a bearing state recognition method based on transfer learning that adopts Stacked AutoEncoder (SAE) Neural networks to predict different working conditions. The method is demonstrated using a benchmark dataset, which contains drive-end bearing vibration data. Similarly, in Zhang et al. (2019), a deep transfer learning model based on Wasserstein distance guided multi-adversarial networks (WDMAN) is proposed to improve the performance of intelligent fault diagnosis, addressing changes in data distributions related to different operation demands. Li et al. (2020) propose an integrated approach for fault diagnostics with different kinds of components, which combines two deep learning methods—Convolutional Neural Network (CNN) and Multi-layer Perceptron (MLP)—for faults diagnosis using the Case Western Reserve University bearing (CWRU) and 2009 PHM Data Challenge gearbox datasets. Similarly, Zhao et al. (2020) propose a TL framework, which combines bidirectional gated recurrent unit (BiGRU) and Manifold Embedded Distribution Alignment (MEDA) for fault diagnosis. In addition, Sun et al. (2019) propose a deep transfer learning method based on SAEs to predict Remaining Useful Life (RUL) of cutting tools in an offline process and, in the process, transfer knowledge to a new tool for online RUL prediction. In this case, TL is used to address the limited availability of real world data. TL has also been applied to other areas, including production modelling of time series in cyclical manufacturing and production planning. For example, a multi-source transfer learning method was proposed in Zellinger et al. (2020), for modelling time series signals from sensors having different distributions. In another study, which applied TL to the field of production planning, Huang et al. (2019) propose a two-stage transfer learning-based prediction method using both historical production data and real-time order data to improve accuracy and generalization performance when there are insufficient data. The approach is validated with a case study using IoT-enabled machining workshop. In Tercan et al. (2018), TL was used to support project planning in injection moulding, enabling the development of a predictive model via transfer of knowledge between the simulation and real process phase.

As evidenced in the above literature review, to the best of authors’ knowledge, there is the lack of empirical evaluation of TL in real factory environments to improve scalability and rapid deployment of predictive models to enable the realisation of smart production systems as hypothesised by Lee et al. (2020), which forms the motivation of this study. Beyond the manufacturing domain, TL has been applied to many other different fields and the interested reader can refer to Zhuang et al. (2020) for a comprehensive literature review.

Problem statement and methodology

In this paper, TL is applied to a time series forecasting problem to classify the internal speed (measured as the number of strokes per minute) of aluminium can body-maker machines, which can produce up to 300 aluminum cans per minute. Bodymaker machines are large-scale equipment that operate at high-speed and are employed in metal can manufacturing to produce the full length can body from a small (metal) cup that is forced through a series of iron rings. A production line typically includes several bodymakers machines operating at high speed as shown in Fig. 1. Monitoring current and future performance of individual bodymakers is important because the performance of individual bodymakers can affect the efficiency and profitability of the line due to unexpected downtime, increased spoilage and a decrease in can output. In the production planning process, a predictive model that can forecast machine speed can be used to create a cyber twin of individual machines. At the cyber level, aggregated predictive models of all machines can then be used to optimise production schedules by allowing the real-time adjustment of the individual operating speeds for other upstream or downstream machines. Furthermore, at the cognition and configuration levels, aggregated information from cyber twins components can be used to achieve self-configuration and optimisation of production planning.

The development of a forecasting model for machine speed is challenging as speed values typically exhibits a combination of periodic patterns such as normal production schedule and episodic, sporadic patterns caused by abnormal operations or unplanned stoppage. In a recent publication, Essien and Giannetti (2020), a novel deep learning architec-

Table 1 Summary of literature review obtained by performing searches using GoogleScholar and Web of Science with keywords 'Transfer Learning in Manufacturing'

S/no.	Source	Real dataset	Time series	Problem type	Problem area
1.	Sun et al. (2019)	Y	Y	Regression	RUL
2.	Ferguson et al. (2018)	N	N	Classification	Defect detection
3.	Huang et al. (2019)	Y	Y	Regression	Production monitoring
4.	Jiao et al. (2020)	N	N	Classification	Quality control
5.	Mao et al. (2019)	N	N	Regression	RUL
6.	Wen et al. (2019)	N	N	Classification	Fault diagnosis
7.	Cao et al. (2020)	N	Y	Classification	Fault diagnosis
8.	Zhang et al. (2019)	Y	Y	Classification	Fault diagnosis
9.	Wen et al. (2017)	N	Y	Classification	Fault diagnosis
10.	Wen et al. (2019)	N	Y	Classification	Fault diagnosis
11.	Xiao et al. (2019)	Y	N	Classification	Fault diagnosis
12.	Xu et al. (2019)	Y	N	Classification	Fault diagnosis
13.	Yang et al. (2019)	N	N	Classification	Fault diagnosis
14.	Zellinger et al. (2020)	Y	Y	Regression	Cyclical manufacturing
15.	Tercan et al. (2018)	N	N	Regression	Injection moulding
16.	Wang et al. (2021)	Y	Y	Regression	Machining
17.	Zhao et al. (2020)	Y	Y	Regression	Fault diagnosis
18.	Li et al. (2020)	N	Y	Regression	Fault diagnosis

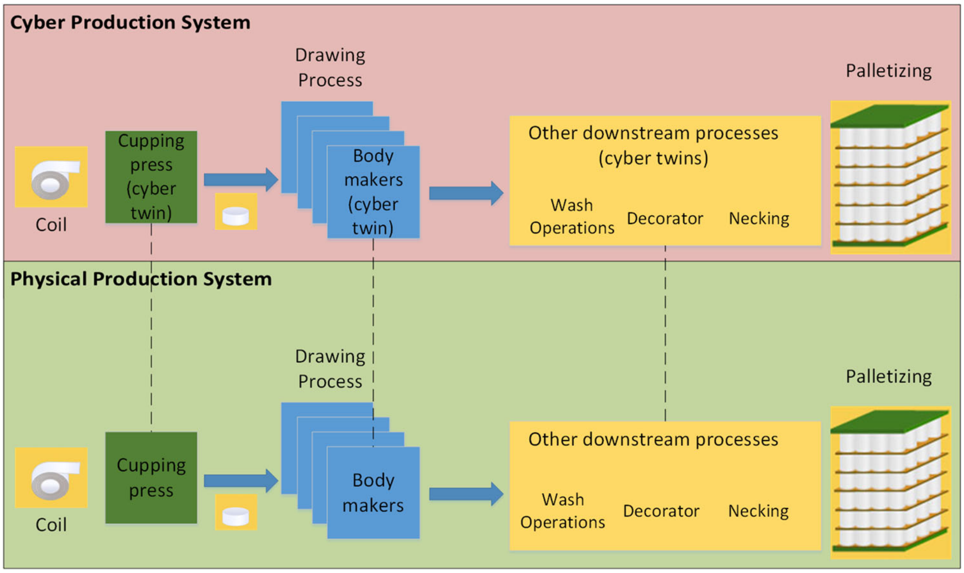


Fig. 1 A schematic representation of the body maker machines and their cyber twins

ture for multi-step machine speed forecasting was proposed. The architecture showed superior performance to benchmark state-of-the-art methods such as ARIMA models. In that study, the model was trained on a body maker machine using a large dataset of historical data. However, in ideal manufacturing settings, such large-scale dataset may not be available for newly acquired, configured, or refurbished machines. Besides, it cannot be guaranteed that a predictive model trained on data obtained from a single machine can be used across the entire fleet of machines, due to changes in data distributions caused by machine degradation or unknown faults. In such scenarios, machine-specific models need to be trained across different production lines and/or factories. The development, training and maintenance of these predictive models may become impractical due to time constraint, computational resources, inherent complexity of DL models (i.e. training and optimisation), as well as scarcity of training data for a particular set of machines. Hence, TL offers the potential to reuse models previously trained on a (source dataset) machine across the entire factory on various target machine signals for rapid development and deployment of cyber twin models across factories or domains.

Table 2 Time series length and class distributions

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9
Low	112,957	275,340	279,979	305,931	270,954	319,020	312,035	303,802	271,000
Medium	68,150	206,924	202,268	193,578	216,474	186,618	193,846	184,767	219,203
High	61,536	43,337	43,353	26,092	38,172	19,962	19,720	37,031	35,397
Total	242,364	525,601	525,600	525,601	525,600	525,600	525,601	525,600	525,600

Description of TL terminologies

This study adopts the notations and definitions that are similar to that found in Pan and Yang (2009). First, let a domain be defined as $\mathcal{D} = \{\mathcal{X}, n\}$, where $\mathcal{X}$ is the feature space and n is the number of observations. Let D_s denote the source domain dataset, such that $D_s = \{(x_{s1}, y_{s1}), (x_{s2}, y_{s2}), \ldots (x_{sn}, y_{sn})\}$ and D_t denote the target domain dataset, such that $D_t = \{(x_{t1}, y_{t1}), (x_{t2}, y_{t2}) \ldots (x_{tk}, y_{tn})\}$. In this specific domain, a task T is used to denote the outcome of a predictive model training task defined as $T = \{y, f(x)\}$, where $x_i \in X$ are observations in the input space, $y_i \in Y$ is the response and $f(.)$, is a predictive function. Let the source task be denoted by T_s and the target task be denoted by T_t. Given a source domain $\mathcal{D}_s$ and a learning task T_s, and a target domain $\mathcal{D}_t$ and a learning task T_t, transfer learning aims to improve the learning task T_t in the target domain by using knowledge in $\mathcal{D}_s$ and T_s, where $\mathcal{D}_s \neq \mathcal{D}_t$ or $T_s \neq T_t$.

Problem formulation

In this work, TL is applied to a multi-step time series forecasting problem. The time series signals consist of categorical data, which corresponds to three distinct speed settings of a can body-maker machine. The goal is to use the previously observed (i.e. lagged) input sequences to classify a fixed-length sequence of the future class. To achieve this, the data is transformed using a sliding window method, as described in Essien and Giannetti (2020), which converts the sequential input data to a supervised learning problem (i.e. inputs and outputs). The number of previous time steps is referred to as the window width/size.

Consider an input time series of machine-measured speed $\mathcal{S}$, which represents a sequence of n machine speed measurements taken in regular intervals, such that $\mathcal{S} = (x_1, x_2, x_3, ..., x_n)$ where $x_i \in \mathbb{R}$ for $1 \leq i \leq n$. The time series classification problem can be formally defined as follows. Given a set of discrete classes Y, a training dataset X associated with the class labels $y(x_i) \in Y$, such that $X = (x1, y(x1)), (x2, y(x2)), (xm, y(xm))$, the goal of a classification task is to find a function $\sigma(X)$ such that $\sigma(X) \simeq (Y)$ at all times. This function, typically referred to as a classifier, may be computed using algorithms, such as neural networks, support vectors, nearest neighbors, etc. The process of finding or calculating this function is known as (model) training.

Dataset

In the current study, the time series signals are categorical sequences of values that represent the operating speed, measured as number of strokes per minute, of nine bodymaker machines in a metal can manufacturing production line. The running speed has three operational settings, represented as categorical values indicating *low*, *medium* and *high*. The values of each class are described as follows. The machine speed is categorised as *low* (labelled as 0) for machine speed values between 0 and 100, *medium* (labelled as 1) for machine speed values between 101 and 300, and *high* (labelled as 2) for machine speed values greater than 300. The machine speed is sensor-collected at a frequency of 1/60 Hz. The dataset contains historical records of the machine-collected speed from nine bodymaker machines operating on the same line, referred to as B_i with $i = 1 \ldots 9$. Table 2 shows the total number of observations of each time series (total) and the number of observations of each class.

Figure 2 presents a graphical extract of the train and test portions of the time series from some of the bodymaker machines, showing the variation in the individual speeds of the machines. As it can be seen from the figure, despite the body-makers operating in the same line, there are variations in speed within the machines, due to semi-automated load balancing and specific settings of each machine. For instance, it can be seen that B_1 mostly operates at medium speed while B_5 has a long period when it was working in low speed. These variations in the speed regimes lead to distributional variations, which hinder the ability to directly use models trained on one machine to predict the speed of other machines.

Baseline network architecture

The 2DConvLSTMAE model for univariate multi-step time series forecasting presented in Essien and Giannetti (2020) was adopted as the baseline model in this current study. Figure 3 shows the model architecture of the baseline model.

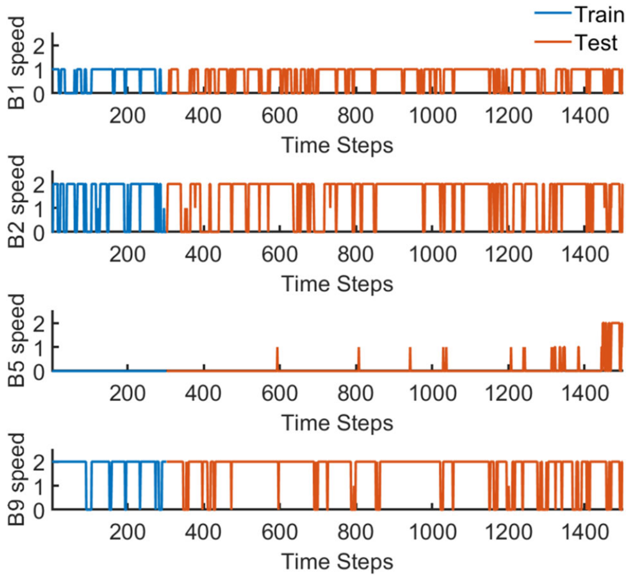


Fig. 2 Extract of time series for body makers B_i $i = 1, 2, 5, 9$. Only a small portion of the train portion is shown

The input is a time series sequence of fixed length ($l = 12$) obtained using a sliding window technique. The encoding layers comprise ConvLSTM layers, with number of filters and sub-sequence length labelled as ($n_{filters} \times len_{filter}$). The output classification layer (softmax) has the three distinct classes - *low*, *medium* and *high*. The end-to-end model has three distinct components: (i) ConvLSTM encoding layers, (ii) bidirectional LSTM decoding layers, and (iii) time-distributed supervised learning (fully connected [FC]) and softmax classification layers. The reader can refer to Essien and Giannetti (2020) for a detailed description of the baseline model architecture. This study adopts a training procedure similar to Essien and Giannetti (2020), applying a similar model configuration and training in a greedy, layer-wise manner on each target dataset using Adam optimizer (Kingma and Ba 2014). In accordance with suggested best-practice as stipulated in Kingma and Ba (2014), the model was trained using the following hyper-parameters: 200 epochs, learning rate $\alpha = 1 \times 10^{-5}$, first-moment exponential decay $\beta_1 = 0.001$, and second-moment exponential decay $\beta_2 = 0.999$.

Transfer learning experiments

The aim of this study is to empirically analyse the application of TL to improve scalability of DL models for CMS, emphasising the potential to promote DL model transfer between different machines in a manufacturing plant. More specifically in the proposed experimental setting, a DL model, $\mathcal{M}_9$, trained on (source) dataset, D_9, is chosen as source model and TL is applied to all other (target) datasets D_i for $i = 1, \ldots, 8$ of the remaining machines B_i $i = 1, \ldots, 8$. Furthermore, the respective TL performances on varying data portions 100%, 50%, 30% and 10% are evaluated for the target datasets (i.e. D_i for $i = 1, \ldots, 8$) to study the effect of training dataset size on a given TL strategy. The experiment reproduces a real-world scenario when predictive models need to be deployed and scaled up to a factory facility that typically consists of many machines/systems in various locations. As explained in Sect. 4.1, $\mathcal{M}_9$ was chosen because it was the model that achieved the best generalisation performance across the different machines, hence representing the best case benchmark against which to compare the performance of TL. In this study, the F-score and training time are used as performance metrics for model evaluation. For each machine, B_i, the model performance is evaluated against the corresponding baseline model trained on the target domain with varying data size, as well as the source model's prediction ($\mathcal{M}_9$). Thus, for each machine, the performance of the TL-based model ($\mathcal{M}_{TL}$) is compared against their respective baseline models $\mathcal{M}_i$ (i.e. training the model from scratch) and the source model $\mathcal{M}_9$, using the F-score as a performance measure. The percentage gain is used to measure the performance of TL, defined as follows:

$$gain_{B_i}(\mathcal{M}_{TL}, \mathcal{M}_k) = 100 * \frac{(f_{score}(\mathcal{M}_{TL}) - f_{score}(\mathcal{M}_k))}{f_{score}(\mathcal{M}_k)}$$
$$k = \{i, 9\} \quad (1)$$

where the f-score is calculated on predictions using the test partition of machines B_i for $i = 1, \ldots, 8$. Two sets of TL strategies form the base of our experimental setup. These are: (i) model weight (parameter) re-using and (ii) model layer fine-tuning respectively, and both are described in the subsequent sub-sections. Each dataset, D_i, is divided into a train and validation dataset (of variable size, depending on which data portion is included and with train and validation ratio of 90:10). The test dataset, which is fixed and chosen to be the last 10,000 time steps of the time series for each machine.

Weight reuse

Training a DL model from scratch is difficult and computationally expensive due to the internal complexity of the models, which comprise of several layers with many parameters (i.e. weights, etc.). These parameters are typically randomly initialized prior to the training process, and iteratively updated through a process of back-propagation using labelled data, an optimizer and a loss (or cost) function. This process of iteratively updating all the weights is extremely time consuming in DL. In the weight reuse strategy, the weights from a source model, $\mathcal{M}_s$, are transferred to initialise the weights of a target model, $\mathcal{M}_t$, as shown

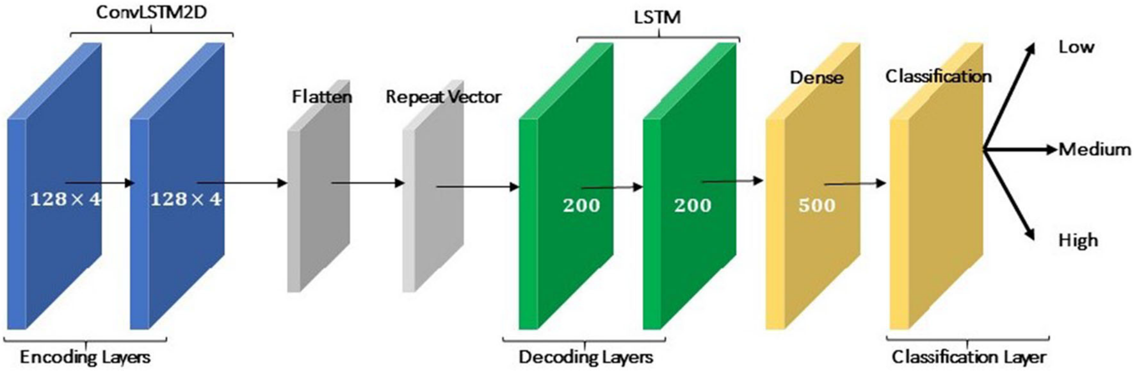


Fig. 3 Baseline model architecture for all models

in Fig. 4a. In this way, the knowledge embedded in the source model's (i.e. $\mathcal{M}_s$) weights are transferred to the target domain. This strategy reduces the time and complexity of the training process comparing to training the model using initial random weights. Furthermore it can improve the learning process when there is shortage of labelled data in the target domain.

Fine tuning

Fine tuning is arguably the most widely-used TL strategy applied to DL models, as it adequately compensates for the shortage of training data, as well as significantly reducing the computational resources required for training DL models from scratch. The concept of layer fine-tuning is straightforward; the learning process commences with a pre-trained source model, $\mathcal{M}_s$, trained on a source domain, $\mathcal{D}_s$. Next, some layers of this source model are retrained/fine-tuned on the target domain $\mathcal{D}_t$. This method is very popular in computer vision and image recognition, where pre-trained models exist that have been trained on online image repositories (for instance, the ImageNet dataset that comprises millions of images). Compared to training a DL model from scratch, fine-tuning has been proven to significantly improve model performance and reduce the requirement for large amounts of labelled target data.

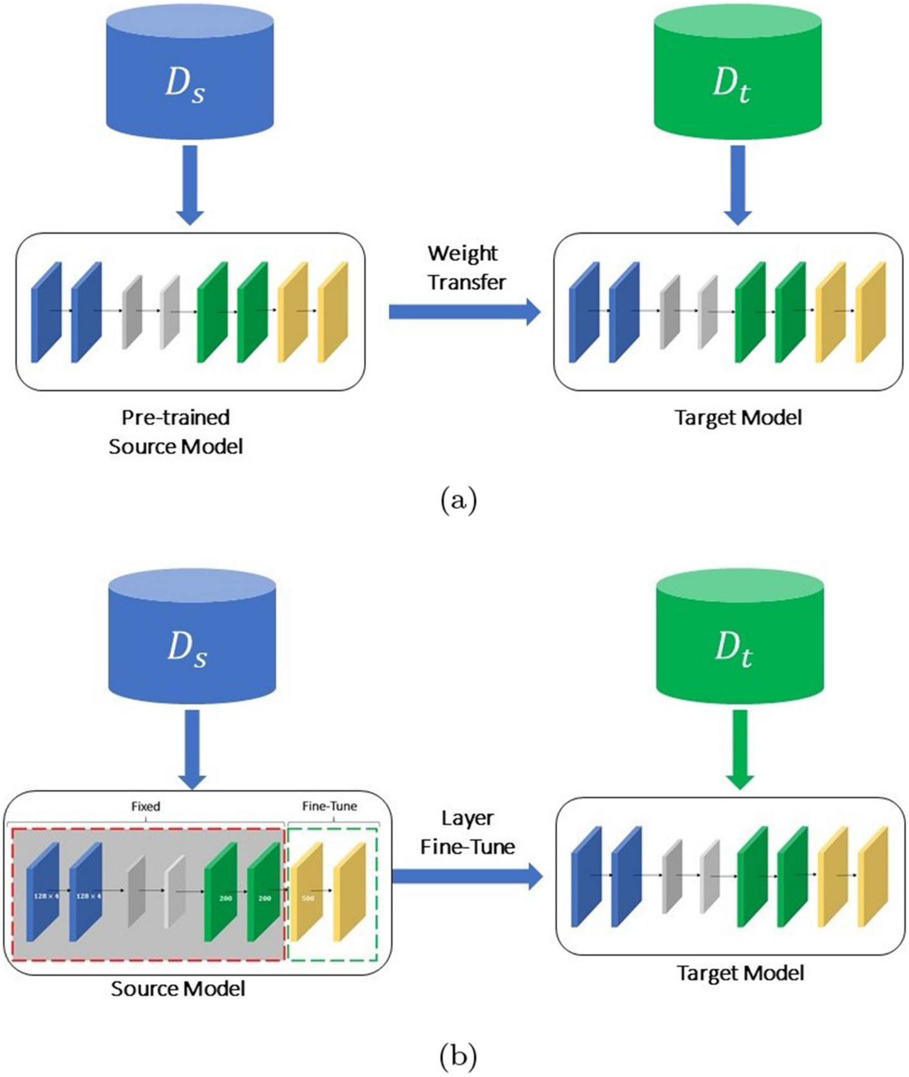
Within a fine-tuning TL implementation, determining the optimal layer(s) to freeze/fine-tune is typically a manual process, and is essential to model predictive performance. For instance, if the target dataset is small, and the model has a large number of parameters, fine-tuning many layers may result in over-fitting (Ng et al. 2015). On the other hand, fine-tuning too few layers can result in an under-fitting model, which negatively impacts model predictive performance. In this present study, TL models were trained and compared at varying numbers of frozen layers on all the test datasets (i.e. B_1 to B_8) using the approach described in Fig. 4b.

Results and discussion

Baseline models

Baseline models were trained for each body maker on different dataset portions using the network architecture described in Sect. 3.4. For baseline models, the set of hyper-parameters used for training were as follows. Two ConvLSTM layers comprised the encoding layer (see Fig. 3). Each ConvLSTM layer had 128 filters, with each filter being of size (1 × 4). Rectified Linear Unit (ReLU) activation was applied to each layer in order to eliminate negative activations. The decoding layers comprised LSTM layers, each having 200 units, and 10% dropout, accompanied by ReLU activation layers. The fully connected dense network had 500 neurons and ReLU activation, while a softmax classification layer is applied to classify the output from the previous FC layer. Figure 5 shows the F-score of each baseline model $\mathcal{M}_i$ trained on varying portions of the respective training datasets. As can be seen, when the training data size reduces below 30%, the performance of the model significantly decreases.

In the experiment set up, the dataset D_9 (collected from machine B_9) was chosen as the source dataset and TL was performed on all other target datasets ($D_i, i = 1, \ldots, 8$). Machine B_9 was selected because the corresponding model, $\mathcal{M}_9$, was the the best performing model among the machine fleet. Figure 6a presents a graphical representation of the performance of the source model, $\mathcal{M}_9$, to directly make predictions (i.e. without any training or TL) using the test portions of the individual datasets (i.e. D_i for $i = 1, 2, \ldots, 8$). However, as can be seen from Fig. 6a, there are variations in the predictive performances across the various target datasets. In particular, the source model, $\mathcal{M}_9$, performs poorly when used to predict the speed of machines B_1 and B_5, respectively. This is due to the fact that these body makers operated at different speeds in this time period, and thereby had different data distributions compared to the B_9 as discussed in Sect. 3.3.



(a)

(b)

Fig. 4 TL strategies adopted in this study. **a** Weight Re-use TL Strategy and **b** Fine-tuning TL Strategy

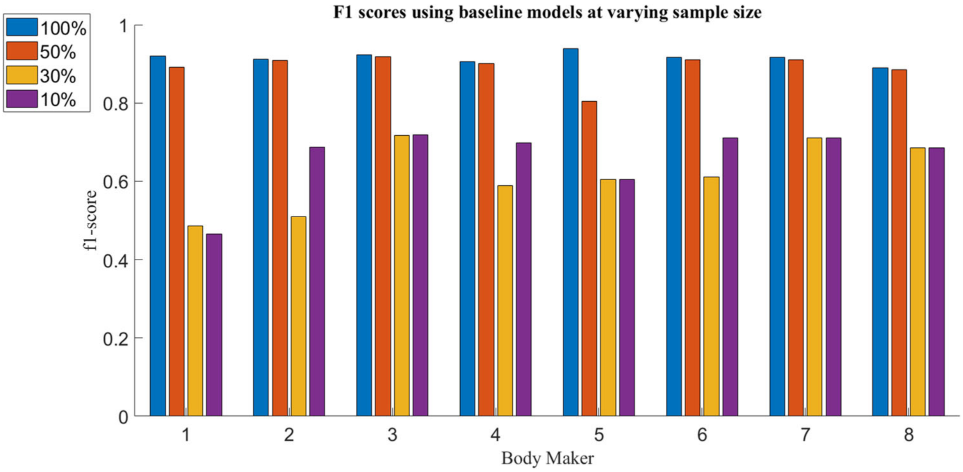
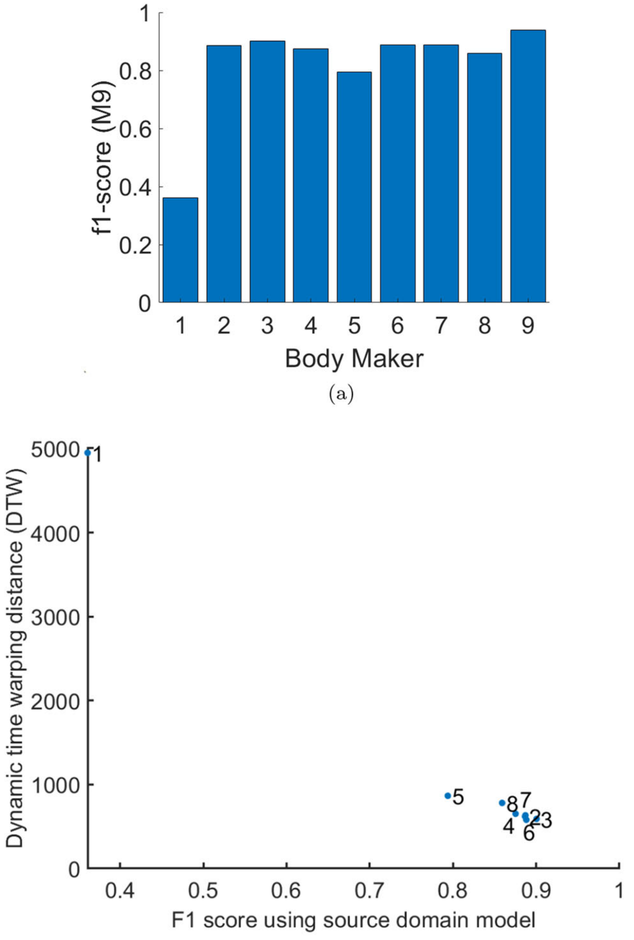


Fig. 5 F-score of baseline models at varying data sizes. The model accuracy significantly decreases for small data samples

This is supported by Fig. 6b that presents a scatter plot of the dynamic time warping (DTW) distance and the F-score of using the model trained on the source domain ($\mathcal{M}_9$) to predict on the respective target datasets (i.e. D_1, D_2, $\ldots D_9$). The DTW distance between two time series represents the optimal alignment or warping path between the time series (Rakthanmanon et al. 2013). Therefore, the larger the distance, the greater the dissimilarity between the two time series. Further



(a)

(b)

Fig. 6 Evaluation of predictive performance using source model, $\mathcal{M}_9$, to predict on target datasets. **a** Comparative evaluation of F-score obtained by using the source model, $\mathcal{M}_9$, to predict on the target datasets. **b** Scatter plot showing relationship between F-score and DTW distance

Table 3 TL strategies adopted in the experimental setup

Strategy	Number of re-trainable layers	Layers retrained
Fine Tune-1	1	Classification layer
Fine Tune-2	2	Dense and Classification layers
Fine Tune-3	3	1xLSTM, Dense and Classification layers
Fine Tune-6	6	2xLSTM, Dense and Classification layers
Reuse	All	All layers

details about DTW are outside the scope of this study and are, therefore, intentionally left out. However, the reader can refer to Rakthanmanon et al. (2013) for more details.

As can be seen from the Fig. 6b, there is a negative correlation between the x and y axes. In other words, the F-score of the target dataset decreases as the dynamic time warping (DTW) distance (between source and target domain) increases, showing that reusing the model trained on the source domain ($\mathcal{M}_9$) is not appropriate if there are dissimilarities between the target and source domain. Furthermore, it can be noted that the predictive performance of machine B_1 (top left point), which had a different data distribution to the source dataset,—and hence, highest DTW distance—is particularly low (≈40%). It can, therefore, be concluded that adopting a simple, conventional ML regime that uses a single model to predict across different machines may be suboptimal, regardless of the fact that the individual machines are all within the same production line. This can be rationalised by the fact that each machine has its own internal behaviour,

Table 4 Summary of experimental results

Target	Data percentage	F-score	Time (s)	TL type
B_1	0.1	0.92938	26.1797	fine-tune1
B_1	0.3	**0.9314**	98.1249	**reuse**
B_1	0.5	0.92931	134.6926	reuse
B_1	1	0.92564	66.3608	fine-tune2
B_2	0.1	0.91126	56.0303	fine-tune6
B_2	0.3	0.91138	38.6265	fine-tune3
B_2	0.5	0.91111	51.1336	fine-tune3
B_2	1	**0.91276**	125.1308	**fine-tune6**
B_3	0.1	0.92165	54.127	fine-tune6
B_3	0.3	0.92319	37.6584	fine-tune3
B_3	0.5	**0.92422**	82.6167	**fine-tune6**
B_3	1	0.9235	1870.6878	base
B_4	0.1	0.90383	53.9838	fine-tune6
B_4	0.3	0.90363	37.3664	fine-tune3
B_4	0.5	0.90444	80.2648	fine-tune6
B_4	1	**0.9052**	1834.4249	**base**
B_5	0.1	0.80879	53.4503	fine-tune6
B_5	0.3	0.94181	65.7033	fine-tune6
B_5	0.5	**0.94226**	79.6871	**fine-tune6**
B_5	1	0.9394	1027.9393	base1
B_6	0.1	0.91702	56.6719	fine-tune6
B_6	0.3	**0.91971**	35.6487	**fine-tune3**
B_6	0.5	0.91963	78.979	fine-tune6
B_6	1	0.91945	122.0456	fine-tune6
B_7	0.1	0.91553	55.1323	fine-tune6
B_7	0.3	0.91593	66.0485	fine-tune6
B_7	0.5	0.91552	253.2316	reuse
B_7	1	**0.9168**	1670.3705	**base**
B_8	0.1	0.88976	55.081	fine-tune6
B_8	0.3	**0.89157**	34.3694	**fine-tune3**
B_8	0.5	0.89061	48.8034	fine-tune3
B_8	1	0.89079	119.8778	fine-tune6

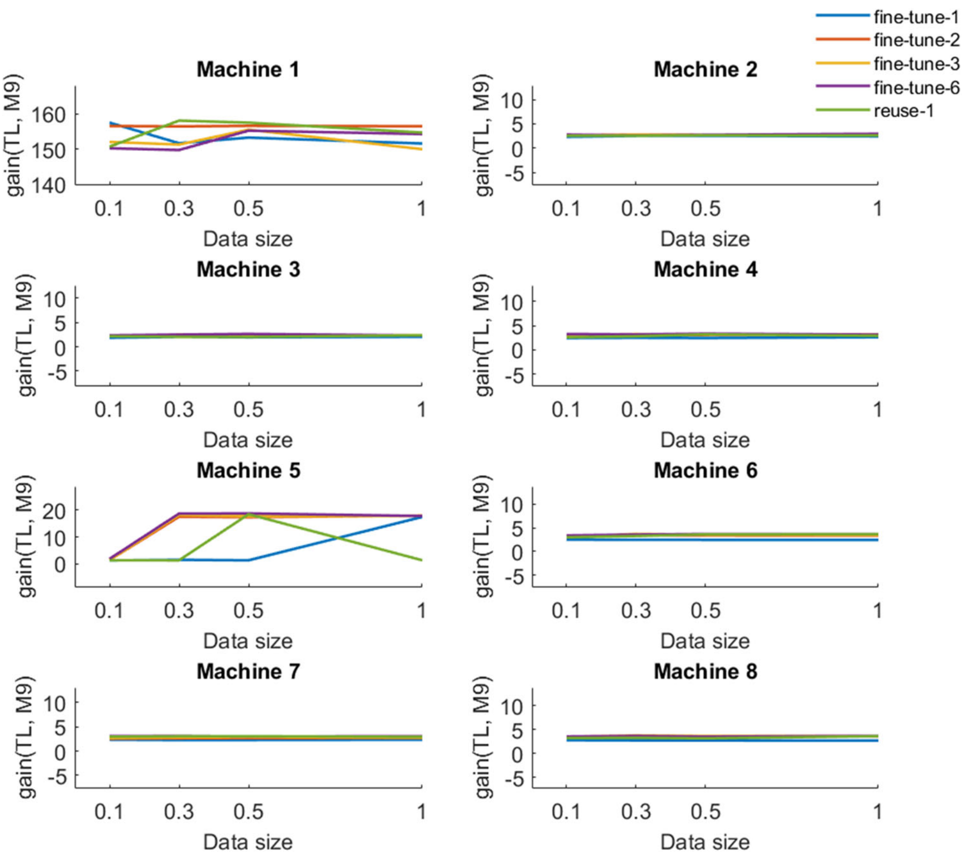


Fig. 7 Comparative performance of TL strategies, at varying data size against source domain model $\mathcal{M}_9$

which is influenced by load balancing, as well as specific operating conditions.

Experiment results

In this study, TL was performed between the source model ($\mathcal{M}_9$) and all other target datasets (i.e. D_1 to D_8). This subsection presents a discussion of the results obtained from the TL experimental strategies—weight re-using (see Sect. 3.5.1) and layer freezing/fine-tuning (Sect. 3.5.2) respectively, which are summarized in Table 3.

Within this study, as described in Sect. 3.5, the TL strategies are applied on varying data portions. Baseline models for all the target domains were also trained with model architecture and training procedures described in 3.4. In the TL experiments, the same hyper-parameters related to the network structure for baseline models were adopted. However, it was decided to fix the training hyper-parameters, adopting $BatchSize = 10{,}000$, $NumEpochs = 50$ and $LearningRate = 1 \times 10^{-4}$. These values vary from the optimal training parameters of the baseline models and were chosen after initial experiments to reduce the training time of the experiment without compromising predictive accuracy.

Table 4 shows the comparative results for the experiments conducted in this study. Only the best performing models for each machine at varying sample data sizes are reported in each row. The model with overall best score for each body maker across the different data sizes is highlighted in bold font. For instance, the first row on Table 4 shows that the best F-score of 0.92938 was obtained by performing TL from source model ($\mathcal{M}_9$) by fine-tuning the last layer only (fine-tune1) and using 10% of training data D_1. Overall, it can be seen that TL was the best strategy in the majority of instances, except in two cases (B_4 and B_7), where the best model is the baseline model trained on the entire dataset. However, there is only a marginal difference in F-score and TL achieves very similar performance with smaller sample sizes, emphasizing the value of TL, as it requires less data to train, decreasing the time required for training, as well as reducing the model complexity.

From the findings presented in Table 4, the optimal TL strategy depends on the target dataset. For instance, fine-tuning six or three layers are optimal strategies for B_i for $i = \{2, 3, 5, 6, 8\}$. The reuse strategy is the best only for B_1, which is the most dissimilar target dataset with respect to the source dataset. Therefore, although the optimal strategy may depend on the target dataset, TL typically resulted in good performances especially with small training data. From the empirical analysis, it is inferred that the fine-tuning TL strategy works well for time series with similar distributions to the source domain (lower DTW distance), while reuse may be more appropriate for more dissimilar target datasets (e.g.

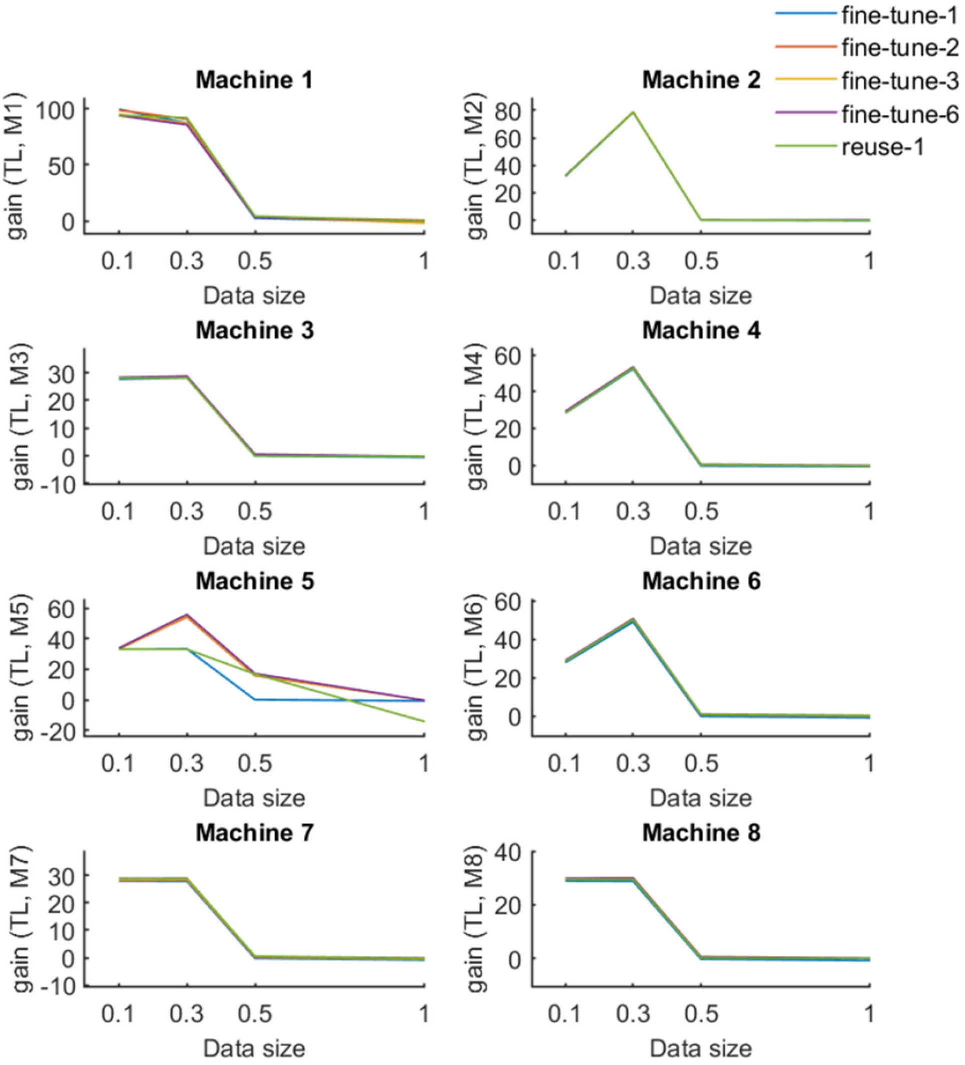


Fig. 8 Comparative performance TL strategies at varying data sizes, against target models $\mathcal{M}_i$ $i = 1, \ldots, 8$

B_1 with higher DTW distance). The empirical analysis did not reveal any strong relationship between the optimal number of layers (for the fine-tuning TL strategy) and predictive performance. Overall, the results support the value of TL to improve generalisation performance, especially for training instances involving limited training data samples. It is also worth mentioning here that the results presented in this set of experiments were achieved with fixed values of hyper-parameters, hence emphasizing the promise of TL towards reducing the complexity of model training, as well as supporting the scalability and reusability of predictive models across fleet of machines.

Comparing predictive performance of transfer learning models

This sub-section presents the performance evaluation of the proposed transfer-learning framework in terms of percentage gain with respect to the source model $\mathcal{M}_9$ and baseline models $\mathcal{M}_i$ for $i = \{1, \ldots, 8\}$ at varying data sizes. The performance gain is measured using Eq. 1. Figure 7 shows the gains realised by different TL strategies and varying data sizes, compared against the source model. In the figure, the x-axis represents the percentage of (target) training data, while the y-axis denotes the percentage gain observed, with negative values indicating negative transfer.

The plots show that for B_i with $i = \{2, 3, 4, 6, 7, 8\}$, TL strategies achieve positive gain in the order 2–3% F-score, with very little variations depending on the data size. The above mentioned machines are those with similar distribution and behaviour to the source domain. This confirms the hypothesis that, when the source and target datasets are highly similar in data distribution (measured via the DTW distance), applying TL to a small data portion of the target dataset will likely result in improvements in the predictive performance.

Conversely, for machines such as B_i with $i = \{1, 5\}$, that present distributional dissimilarity to B_9, TL achieves con-

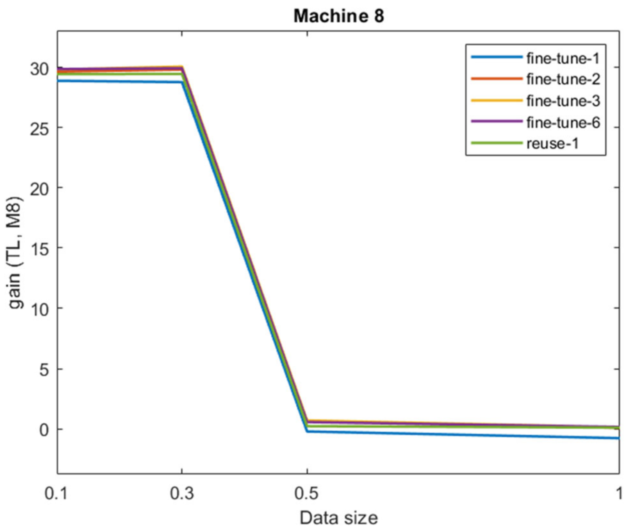
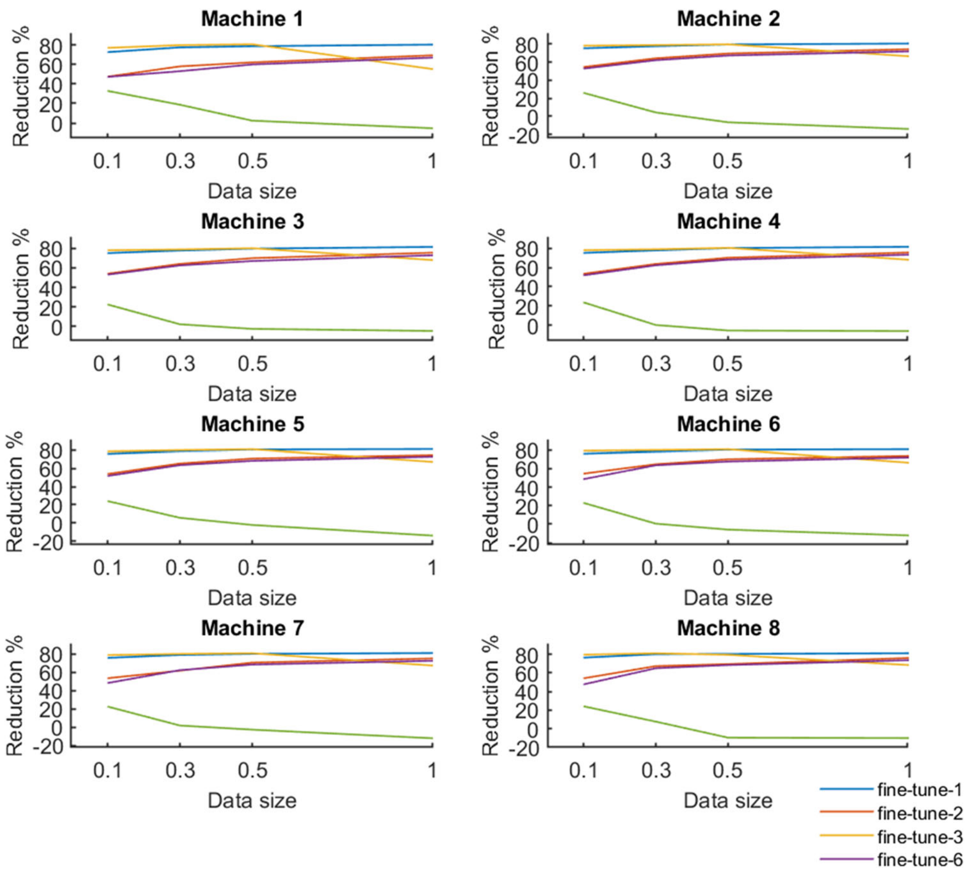


Fig. 9 Gain achieved by TL strategies, compared to baseline models for B_8

siderably higher performance compared to using the source model to predict on the test datasets. The highest TL gains are obtained for the machine having the most dissimilar data distribution (i.e. B_1). In this case, the best performing models are obtained with the fine-tuning strategy (trained on 10% of the data) and reuse strategy (trained on 30% of the data). From the findings obtained in this study, it can be concluded that when the target dataset has large distributional variation, the optimal TL approach depends on the size of the training data. Conversely, when the dataset has small variations in distribution, all the TL approaches work well in cases involving sparse data, while little variations are observed in terms of performance for the individual TL strategies.

Figure 8 shows the percentage gain (or loss) for different TL strategies at varying data sizes when compared to the respective baseline models $\mathcal{M}_i$ for $i = 1, \ldots, 8$. It can be seen that, similar to Fig. 7, machines B_i with $i = \{2, 3, 4, 6, 7, 8\}$ show a consistent pattern of behaviour. Small or negligible gains are achieved with bigger data portions, while large gains are achieved for data portions 10% and 30%. When training with the entire dataset there are some occurrences of negligible negative transfer. An example is shown in Fig. 9, where it can be seen that negative transfer occurs for fine-tune TL strategy with 1-layer. B_1 shows a similar patten to the previous machines, although achieving higher gains (approximately 100% improvement for 30% and 10% data portions of the target datasets). In this case, when a small training sample is available or when there is a need to achieve model training time reduction (such as when there is limited computational resource), the advantage of TL becomes evident. B_5 has a different TL performance

Fig. 10 Time reduction achieved by TL strategies, compared to baseline models at varying sample data size



pattern, with the highest model gain achieved for fine-tune strategies with frozen layers = $\{2, 3, 6\}$ and 30% data portion and reuse with 50% data portion. From the findings obtained from the empirical analysis presented in this section, it can be concluded that fine-tuning may be preferable when training with smaller sample sizes.

Time performance comparison

In real word applications, the ability to reduce the training time is of paramount importance for the development of scalable CMS. In this section, the individual performances of the different TL strategies are compared with respect to their corresponding baseline models using training time as performance metrics. For the purpose of time comparison, all the models were trained using the same hyper-parameters adopted for training the TL models (see Sect. 4.2) to create a fair comparison. Figure 10 shows a summary of percentage reduction of training time achieved by TL with respect to baseline models trained on the same data samples. It can be seen that the fine-tuning strategies always achieved the most significant reduction of training time for all data sizes. For the weight reuse strategy, instead, time reduction is only achieved for smaller data sample below 50%. B_8 was the machine that showed the worst performance of weight reuse in terms of training time when training with 50% and 100% of the data. Fine-tuning with one and three layers were the two best strategies that resulted in the greatest training time reduction. The consistent behaviour across machines supports the hypothesis that fine tuning strategies can effectively lead to significant reductions in training time, whilst improving generalisation performance as discussed in the Sect. 4.2.

Conclusion

In this paper, an empirical study that applies Transfer Learning (TL) to transfer knowledge across a fleet of machines in a manufacturing plant is presented. The results provide the first empirical evaluation of TL to show the potential of TL in building scalable and robust cyber twins models of machines operations. Different TL strategies (weight reuse and fine-tuning) were applied to a time series prediction problem to classify the speed of nine bodymaker machines in a can manufacturing plant. The strategies were compared against baseline models trained using a deep neural network architecture, 2DConvLSTMAE (Essien and Giannetti 2020). The findings from the rigorous empirical analyses successfully demonstrate the transferability of knowledge across machines in the same line, with the most benefit realised when there is limited training data.The fine-tuning TL strategy was found to be the best strategy to achieve the best prediction performance whilst largely reducing the complexity and time required to train self-predictive DL models for machines with similar data distribution to the target machines. In cases when the machine is operating at a different speed regime/setting, the reuse TL strategy has been proven to be most appropriate for realising optimal predictive performance and reduction of training time. When using smaller sizes datasets, TL always achieved the best performance comparing to using baseline models.

The findings from this study support the value of TL to achieve reduction of the time it takes to train models for individual machines as well as improving predictive performance when data is scarce, hence enabling fast and scalable deployments of predictive models for cyber twins in smart factories. These predictive models can contribute to building cyber twins of machines to enable the realisation of smart production systems, showing the value of TL towards the realisation of smart factories as hypothesised by Lee et al. (2020). Future work is planned to extend this framework to transfer of knowledge across different lines and factories as well as demonstrating the value of TL to build predictive models that integrate the behaviour of different machines in the production line. Further studies will also investigate different types of TL, such as multi tasking learning, and the use of TL to develop models that continuously adapt to changes in distributions.

Acknowledgements This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) project EP/S001387/1 and we acknowledge the support of the IMPACT and Supercomputing Wales projects, which are part-funded by the European Regional Development Fund (ERDF) via Welsh Government.

References

Cao, N., Jiang, Z., Gao, J., & Cui, B. (2020). Bearing state recognition method based on transfer learning under different working conditions. *Sensors (Switzerland), 20*(1), 1–12.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

Essien, A. E., & Giannetti, C. (2020). A deep learning model for smart manufacturing using convolutional LSTM neural network autoencoders. *IEEE Transactions on Industrial Informatics, 16,* 6069–6078.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.A. (2019). “Deep Neural Network Ensembles for Time Series Classification,” Proceedings of the International Joint Conference on Neural Networks, vol. 2019

Ferguson, M. K., Ak, R., Lee, Y.-T.T., & Law, K. H. (2018). Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *The ASTM Journal of Smart and Sustainable Manufacturing,2*.

Flynn, J., & Giannetti, C. (2021). Using convolutional neural networks to map houses suitable for electric vehicle home charging. *AI, 2*(1), 135–149.

Giannetti, C., & Ransing, R. (2016). Risk based uncertainty quantification to improve robustness of manufacturing operations. *Computers and Industrial Engineering, 101,* 70–80.

Giannetti, C., Ransing, R., Ransing, M., Bould, D., Gethin, D., & Sienz, J. (2014). A novel variable selection approach based on co-linearity index to discover optimal process settings by analysing mixed data. *Computers and Industrial Engineering, 72*(1), 217–229.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Huang, S., Guo, Y., Liu, D., Zha, S., & Fang, W. (2019). A two-stage transfer learning-based deep learning approach for production progress prediction in IoT-enabled manufacturing. *IEEE Internet of Things Journal, 6*(6), 10627–10638.

Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., & Eschert, T. (2017). Industrial internet of things and cyber manufacturing systems. In S. Jeschke, C. Brecher, H. Song, & D. Rawat (Eds.), *Industrial internet of things* (pp. 3–19). Berlin: Springer.

Jiao, W., Wang, Q., Cheng, Y., & Zhang, Y. (2020). End-to-end prediction of weld penetration: A deep learning and transfer learning based method. *Journal of Manufacturing Processes., 63,* 191–197.

Kashiparekh, K., Narwariya, J., Malhotra, P., Vig, L., & Shroff, G. (2019). *ConvTimeNet: A pre-trained deep convolutional neural network for time series classification*. arXiv preprint arXiv:1904.12546 (2019).

Kingma, D.P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980

Lee, J., Azamfar, M., Singh, J., & Siahpour, S. (2020). Integration of digital twin and deep learning in cyber-physical systems: Towards smart manufacturing. *IET Collaborative Intelligent Manufacturing, 2*(1), 34–36.

Lee, J., Bagheri, B., & Kao, H. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters, 3,* 18–23.

Lee, J. H., Do Noh, S., Kim, H. J., & Kang, Y. S. (2018). Implementation of cyber-physical production systems for quality prediction and operation control in metal casting. *Sensors (Switzerland), 18*(5), 1428.

Li, X., Hu, Y., Li, M., & Zheng, J. (2020). Fault diagnostics between different type of components: A transfer learning approach. *Applied Soft Computing, 86,* 105950.

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., & Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems, 80,* 14–23.

Mao, W., He, J., & Zuo, M. J. (2019). Predicting remaining useful life of rolling bearings based on deep feature representation and transfer learning. *IEEE Transactions on Instrumentation and Measurement., 69,* 1594–1608.

Ng, H.-W., Nguyen, V. D., Vonikakis, V., & Winkler, S. (2015). Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on international conference on multimodal interaction* (pp. 443–449).

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., et al. (2013). Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD), 7*(3), 1–31.

Ransing, R., Batbooti, R., Giannetti, C., & Ransing, M. (2016). A quality correlation algorithm for tolerance synthesis in manufacturing operations. *Computers and Industrial Engineering, 93,* 1–11.

Ransing, R., Giannetti, C., Ransing, M., & James, M. (2013). A coupled penalty matrix approach and principal component based co-linearity index technique to discover product specific foundry process knowledge from in-process data in order to reduce defects. *Computers in Industry, 64*(5), 514–523.

Sun, C., Ma, M., Zhao, Z., Tian, S., Yan, R., & Chen, X. (2019). Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. *IEEE Transactions on Industrial Informatics, 15*(4), 2416–2425.

Tercan, H., Guajardo, A., Heinisch, J., Thiele, T., Hopmann, C., & Meisen, T. (2018). Transfer-learning: Bridging the gap between real and simulation data for machine learning in injection molding. *Procedia CIRP, 72,* 185–190.

Wang, C., & Mahadevan, S. (2011). Heterogeneous domain adaptation using manifold alignment. In *IJCAI international joint conference on artificial intelligence* (pp. 1541–1546).

Wang, J., Zou, B., Liu, M., Li, Y., Ding, H., Xue, K., et al. (2021). Milling force prediction model based on transfer learning and neural network. *Journal of Intelligent Manufacturing, 32,* 1–10.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). *A survey of transfer learning* (Vol. 3). Springer.

Wen, L., Gao, L., Dong, Y., & Zhu, Z. (2019). A negative correlation ensemble transfer learning method for fault diagnosis based on convolutional neural network. *Mathematical Biosciences and Engineering, 16*(5), 3311–3330.

Wen, L., Gao, L., & Li, X. (2017). A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49*(1), 136–144.

Wen, L., Li, X., & Gao, L. (2019). A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Computing and Applications, 32,* 1–14.

Xiao, D., Huang, Y., Qin, C., Liu, Z., Li, Y., & Liu, C. (2019). Transfer learning with convolutional neural networks for small sample size problem in machinery fault diagnosis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233*(14), 5131–5143.

Xu, Y., Sun, Y., Liu, X., & Zheng, Y. (2019). A digital-twin-assisted fault diagnosis using deep transfer learning. *IEEE Access, 7,* 19990–19999.

Yang, B., Lei, Y., Jia, F., & Xing, S. (2019). An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mechanical Systems and Signal Processing, 122,* 692–706.

Zellinger, W., Grubinger, T., Zwick, M., Lughofer, E., Schöner, H., Natschläger, T., & Saminger-Platz, S. (2020). Multi-source transfer learning of time series in cyclical manufacturing. *Journal of Intelligent Manufacturing, 31*(3), 777–787.

Zeng, M., Li, M., Fei, Z., Yu, Y., Pan, Y., & Wang, J. (2019). Automatic ICD-9 coding via deep transfer learning. *Neurocomputing, 324,* 43–50.

Zhang, M., Wang, D., Lu, W., Yang, J., Li, Z., & Liang, B. (2019). A Deep Transfer Model with Wasserstein Distance Guided Multi-

Adversarial Networks for bearing fault diagnosis under different working conditions. *IEEE Access, 7,* 65303–65318.

Zhao, K., Jiang, H., Wu, Z., & Lu, T. (2020). A novel transfer learning fault diagnosis method based on manifold embedded distribution alignment with a little labeled data. *Journal of Intelligent Manufacturing,* 1–15.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., et al. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE, 109,* 43–76.