A Survey on Knowledge Transfer for Manufacturing Data Analytics

[1,3]Seung Hwan Bang, [1]Ronay Ak, [2]Anantha Narayanan, [1]Y. Tina Lee, and [3]Hyunbo Cho

[1]Systems Integration Division, Engineering Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD, 20899, USA

[2]Department of Mechanical Engineering, University of Maryland, College Park, MD, 20742, USA

[3]Department of Industrial and Management Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Gyeongbuk, 37673, Republic of Korea

Abstract

Data analytics techniques have been used for numerous manufacturing applications in various areas. A common assumption of data analytics models is that the environment that generates data is stationary, that is, the feature (or label) space or distribution of the data does not change over time. However, in the real world, this assumption is not valid especially for manufacturing. In non-stationary environments, the accuracy of the model decreases over time, so the model must be retrained periodically and adapted to the corresponding environment(s). Knowledge transfer for data analytics is an approach that trains a model with knowledge extracted from data or model(s). Knowledge transfer can be used when adapting to a new environment, while reducing or eliminating degradation in the accuracy of the model. This paper surveys knowledge transfer methods that have been widely used in various applications, and investigates the applicability of these methods for manufacturing problems. The surveyed knowledge transfer methods are analyzed from three viewpoints: types of non-stationary environments, availability of labeled data, and sources of knowledge. In addition, we categorize events that cause non-stationary environments in manufacturing, and present a mechanism to enable practitioners to select the appropriate methods for their manufacturing data analytics applications among the surveyed knowledge transfer methods. The mechanism includes the steps 1) to detect changes in data properties, 2) to define source and target, and 3) to select available knowledge transfer methods. By providing comprehensive information, this paper will support researchers to adopt knowledge transfer in manufacturing.

Keywords: Data analytics, Manufacturing, Non-stationary environments, Knowledge transfer

1. Introduction

Data analytics (DA), which is a process to discover useful information from data, has been increasingly applied in manufacturing in recent years due to advances in Internet of Things (IoT), sensor technology, and DA (i.e., data mining or machine learning) techniques [1]. DA models have been widely used to guide decision making in manufacturing, such as supporting engineering design, shop floor control, fault detection, machine maintenance, and product quality improvement [2]. DA models often assume that the environment of data generation is stationary, which means that the feature (or label) space or distribution of data does not change over time, because it is difficult to account for changes in the environment before they occur, when training models. However, in the real world, that assumption does not hold [3]. In manufacturing, changes in product design or production processes, adding new sensors on machines, or the effect of aging in sensors (or machine parts) can cause changes in the feature (or label) space or changes in distribution of data [4-7]. These changes in data properties can be characterized as non-stationary (evolving or drifting) phenomena [3].

In non-stationary environments, most DA models need to be retrained using newly-collected data. If a model does not adapt to changes in its environment, the model accuracy will degrade. If the model degrades, its analysis will be inaccurate and unreliable for decision making. A traditional approach to manage the degradation due to changes in environment is to retrain the DA model with data collected from the new environment. However, collecting sufficient data from the new environment for retraining the model is often difficult and time consuming. In highly stabilized and automated manufacturing processes (e.g., semiconductor manufacturing process), data collection from the new environment can be a straightforward task [8]. However, collecting sufficient data is challenging and time consuming when 1) an infrastructure to automatically collect data does not exist [8], 2) the production process is in its commissioning stage, or 3) combining distributed data or time synchronizing of data is complicated [9]. Therefore, when data availability is limited due to the situations that data collection or labelling is expensive or inaccessible, effective and efficient methods for training the new model are needed [10].

Knowledge transfer for DA is particularly good for dealing with degradation in the accuracy of the model in non-stationary environments [3]. When the accuracy of the model is below the accuracy boundary or a new model is needed due to changes in data properties (e.g., changes in the feature space), training a new model should be initiated (see Fig. 1). The traditional approach trains the new model using newly-collected data, which might have different properties compared to the existing data from the old environment. Compared to the traditional approach, knowledge transfer approach trains the new model using the newly-collected data as well as knowledge extracted from the existing data or model(s).

There are two major knowledge transfer approaches: *transfer learning* and *incremental learning* (*online learning*), which are the focus of this paper. Strictly speaking, incremental learning and online learning are different, but incremental learning refers to online learning strategies [11]. Thus, in this survey, we treat both learning approaches as the same concept. Also, there are other terms or knowledge transfer approaches such as *domain adaptation* and *covariate shift*. Transfer learning covers the area of domain adaptation, and covariate shift is a subset of incremental learning; thus, we do not mention model adaptation and covariate shift separately. Transfer learning focuses on maximizing the use of knowledge in data or in similar models with limited information from a new environment. On the other hand, incremental learning focuses on continuously and incrementally adapting to a new environment with knowledge inherent in the existing model and small amount of data collected from the new environment. Many studies have shown that transfer learning [10, 12, 13] and incremental learning [3, 11, 14, 15] can be beneficial to learning in non-stationary environments. Most of these studies have focused on specific applications such as text classification and image recognition.

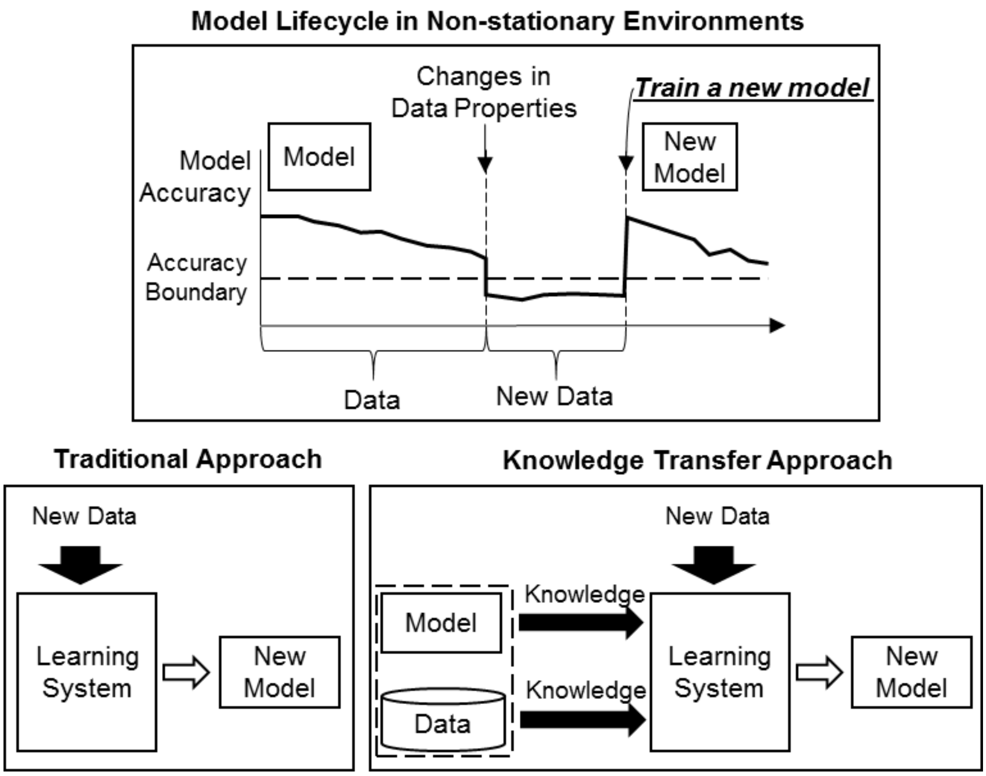

Fig. 1 Concept of knowledge transfer approach compared to traditional approach.

In this paper, we review and analyze transfer learning and incremental learning methods with a focus on supporting knowledge transfer in manufacturing. The result of the analysis could provide great benefit when designing and managing DA models with non-stationary environments in manufacturing.

The rest of the paper is organized as follows. In Section 2, the non-stationary environments are described, and knowledge transfer methods are categorized and discussed. Also, benefits of knowledge transfer in manufacturing are presented. In Section 3, current research on the topic of knowledge transfer, including changes in data properties, availability of labeled data, and knowledge sources is reviewed. Our comprehensive analysis and findings are summarized. In Section 4, we describe events that cause changes in data properties in manufacturing, and present a mechanism to guide practitioners to select appropriate knowledge transfer methods for DA. Finally, Section 5 concludes the paper and presents ideas for future work.

2. Overview of Knowledge Transfer

In this section, we provide an overview of non-stationary environments, and introduce the concept of knowledge transfer. Notations to define non-stationary environments are described. Two representative approaches of knowledge transfer are presented and these approaches are defined in terms of data and model. Also, we discuss the benefits of applying knowledge transfer in manufacturing.

2.1. An Overview of Non-Stationary Environments

In this section, we introduce the definition of *domain*, *task*, and other terms that are frequently used in knowledge transfer. We use the notations and definitions defined by Pan and Yang [10].

Changes in data properties causing non-stationary environments can be defined using a domain $D = \{\chi, P(X)\}$ and a task $T = \{\mathcal{Y}, f(\cdot)\}$. The domain D consists of two components, which are a feature space χ and a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \dots, x_n\} \in \chi$. For example, if the problem is to classify product failure, x_i is the i^{th} feature vector that corresponds to a product failure, n is the number of feature vectors in a particular learning sample X, and χ is the space of all the feature vectors. The task T is defined by a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$, which is learned from training data that consists of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$. Referring to the example of product failure classification, $\mathcal{Y}$ is the set of labels ('good' or 'scrap'), and $f(\cdot)$ can be used to predict a label for a given product. From a probabilistic viewpoint, $f(x)$ is a predicted label for newly-collected data x, and it can be re-written as $P(y|x)$.

In this paper, *source* refers to the existing model or data from which the knowledge is extracted, and *target* refers to a new model or new data of a new environment (see Fig. 1). The source-domain data is defined as $D_S = \{(x_{S1}, y_{S1}), \dots, (x_{Sn}, y_{Sn})\}$, where $x_{Si} \in \chi_S$ is the i^{th} data point of D_S and $y_{Si} \in \mathcal{Y}_S$ is the corresponding class label of x_{Si}. The target-domain data is defined as $D_T = \{(x_{T1}, y_{T1}), \dots, (x_{Tn}, y_{Tn})\}$, where $x_{Ti} \in \chi_T$ is the i^{th} data point of D_T and $y_{Ti} \in \mathcal{Y}_T$. The source task is T_S, the target task is T_T, the source model is $f_S(\cdot)$, and the target model is $f_T(\cdot)$.

Now, we define types of changes in data properties causing non-stationary environments by referring where incremental learning and transfer learning are concentrated on. Incremental learning in non-stationary environments refers to *concept drift*. The meaning of concept drift is that the statistical properties of data change over time. Probabilistic definition of concept drift is divided into two categories depending on 'what' is changed: *real drift* and *virtual drift* [16]. In real drift, $P(y|x)$

changes over time independently from $P(X)$ due to changes in the relationship between features and labels. In virtual drift, $P(X)$ changes without affecting $P(y|x)$ and can happen when the collected data points are not evenly distributed. Transfer learning in non-stationary environments covers both changes in $P(X)$ and $P(y|x)$. In addition, transfer learning focuses on changes in feature and label spaces [10].

Changes in data properties can be categorized into four types:

1) Change in feature space ($\chi_S \neq \chi_T$): the feature space of source χ_S is different from the feature space of target χ_T. The feature space can be changed when changing the variables to be used for DA due to 1) addition of new sensors, 2) changing production machines or processes, 3) adapting an existing DA model to a new production process or new factory, or 4) using a different representation to describe data.

2) Change in marginal probability distribution ($P(X_S) \neq P(X_T)$): the marginal probability distribution of source $P(X_S)$ is different from the marginal probability distribution of target $P(X_T)$. The marginal probability distribution may change due to sensors' and machines' aging effects or changes in a new machine setup.

3) Change in label space ($\mathcal{Y}_\mathrm{S} \neq \mathcal{Y}_\mathrm{T}$): the label space of source $\mathcal{Y}_\mathrm{S}$ is different from the label space of target $\mathcal{Y}_\mathrm{T}$. The label space may change when a label (criterion) is added or removed in the target domain. In the product defects classification example, the label space changes when the categories of product defects are changed from {good, major} to {minor, major, critical}, where 'minor' defect means small and insignificant issues that do not affect function of the product, 'major' defect means considerable issues that could adversely affect performance of the product, and 'critical' defect means critical issues that could render the product unusable.

4) Change in conditional probability distribution ($P(y_S|x_S) \neq P(y_T|x_T)$): the conditional probability distribution of source $P(y_S|x_S)$ is different from the conditional probability distribution of target $P(y_T|x_T)$. The conditional probability distribution might change when there is a change in relationship between features and labels. For example, when a product design being produced does not change, but the existing machine setup does not guarantee the same level of product quality as before, we can speculate that the relationship between the product failure and the machine setup has changed.

In all types of changes in data properties mentioned above, newly-collected data is needed to retrain a new DA model. However, it is not always easy to collect sufficient data for retraining DA model. In this situation, knowledge transfer could play an important role.

2.2. Concept of Knowledge Transfer

Knowledge transfer for DA (see Fig. 1) is an approach that allows training a DA model to a new environment. This approach is particularly useful when limited information available from the new environment. Knowledge transfer can be redefined as transferring knowledge extracted from D_S or T_S when training $f_T(\cdot)$.

As mentioned before, this paper focuses on the two major approaches: *transfer learning* and *incremental learning*. Given D_S with a corresponding T_S, and D_T with a corresponding T_T, transfer learning is an approach of training $f_T(\cdot)$ by exploiting the related knowledge from D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$ [13]. Incremental learning is an approach of updating or improving $f_T(\cdot)$ by referring to T_S [14]. Transfer learning focuses more on knowledge from D_S than knowledge from T_S, whereas incremental learning focuses on knowledge from T_S. From the perspective of 'what to transfer', transfer learning transfers knowledge from data and model, and incremental learning transfers knowledge from model. Thus, we investigate knowledge transfer methods by focusing on both sources of knowledge: data and model.

When transferring knowledge from data (see Fig. 2 (a)), the target model is trained using the target-domain data with the knowledge from the source-domain data. Knowledge transfer methods differ depending on how knowledge is extracted from the source-domain data and utilized. For example, quality assurance of wafers is an important issue in the semiconductor manufacturing process. Virtual metrology (VM) technologies have been developed to monitor the quality of wafers [17]. A new VM model is needed for a new process, however, collecting labeled data is not a straightforward task due to time consuming labelling process. When wafer records (labels) of the target-domain data is not sufficient, previously collected data that involves sets of process variables (e.g. process parameters) and inspection variables (e.g., inspection results) can be used. Thus, data points that have similar data properties to the target-domain data are selected as the source-domain data for training the target model.

When transferring knowledge from model (see Fig. 2 (b)), the source model is transferred as knowledge to train the target model. Sometimes, the source-domain data is also transferred together with the source model to calculate the difference between the source and target domains, but the source-domain data is not always required. The target-domain data is commonly used to evaluate the suitability of the source model to a new environment (target), and the model is modified accordingly. In the above VM model example, the target model can also be trained using knowledge acquired from previously established VM models of other sets of process settings. For example, if the previous VM models were trained based on neural networks (NN), feature representations and model weights (optimal parameters) could be considered as the knowledge. Both types of knowledge are used for model initialization, and the weights are updated using the target-domain data only.

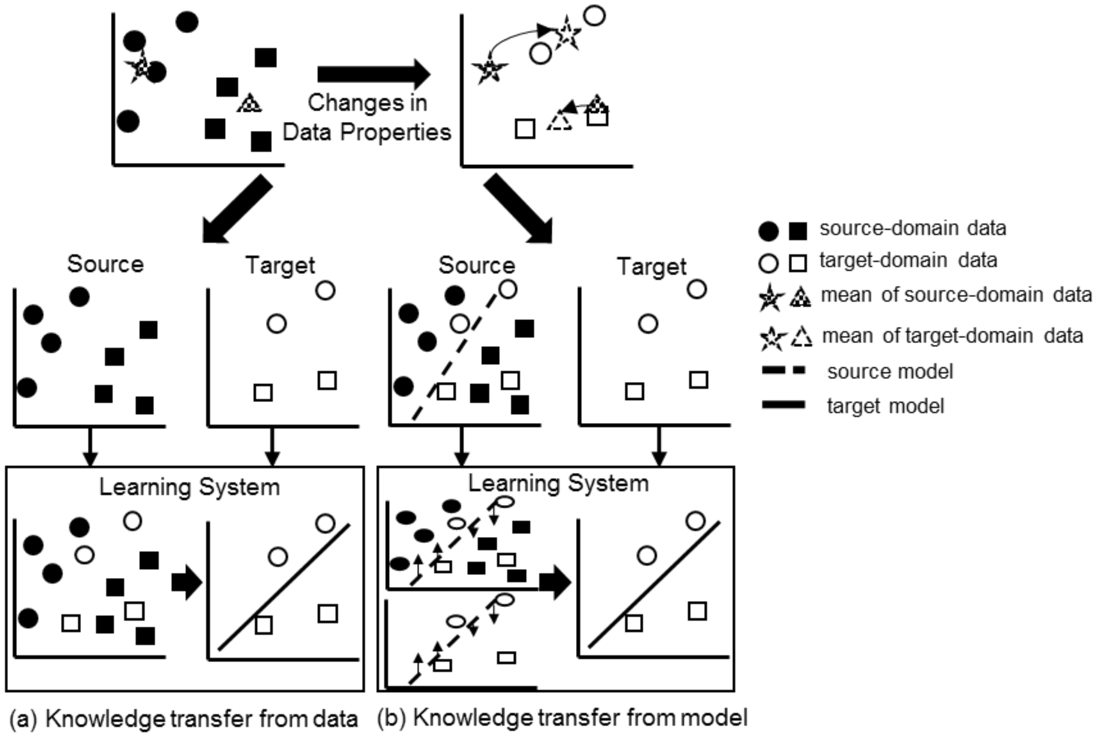

Fig. 2 The role of knowledge sources in knowledge transfer.

Knowledge transfer approaches are different depending on knowledge sources, but all related methods require target-domain data to train the target model. Knowledge transfer can be used when both sufficient and limited labeled target-domain data is available. We use the term 'labeled data' if all target data points are labeled, 'limited labeled data' if the target data points are partially labeled, and 'unlabeled data' if none of the target data points are labeled.

Depending on the availability of the labeled data, machine learning algorithms are categorized as supervised, semi-supervised, and unsupervised learning. Zhu and Goldberg [18] define supervised learning as a task to train a function $f : \chi \rightarrow \mathcal{Y}$ with given input and output data pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{\mathrm{n}}$, which is also called training dataset. The goal is to obtain a function that predicts the true label y on future data $\boldsymbol{x}$. Semi-supervised learning is halfway between supervised and unsupervised learning. The training dataset consists of points and corresponding responses (labels), $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{\mathrm{n}}$, in addition to the points $\{(x_j)\}_{j=n+1}^{\mathrm{m}}$ the labels of which are not known. Unsupervised learning works on data points $\{(x_i)\}_i^{\mathrm{n}}$ without associated responses. The goal is to find patterns in the data without prior information or supervision of correct answers.

Machine learning tasks can also be categorized as supervised learning, unsupervised learning, and reinforcement learning (RL) [22]. Supervised learning uses the ground truth labels or responses, whereas there are no labels available for unsupervised learning tasks. RL allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize their performance. Unlike supervised learning, which trains on labeled datasets, RL achieves its stated objective by receiving positive or negative rewards for the actions taken.

The definition of each category in knowledge transfer can be defined differently with respect to the availability of labeled data, according to Weiss et al. [13]. For example, there are studies [19, 20] that define semi-supervised transfer learning as a learning task with labeled source-domain data and unlabeled target-domain data. Blitzer et al. [21] defined semi-supervised learning as the case of labeled source-domain data and limited labeled target-domain data. Thus, for consistency, we have used the terms *labeled*, *limited labeled*, and *unlabeled* to categorize knowledge transfer methods instead of *supervised*, *unsupervised*, and *semi-supervised (or RL)*.

In most cases of DA applications in manufacturing supervised learning algorithms are dominantly used due to availability of domain experts [23]. However, collecting sufficient labeled data takes time, and often not an easy task [5, 17]. This task might be very labor intensive and prone to human errors. For example, in the product defect classification problem [24], to determine whether products have defects or not, they must be inspected individually through performing a full inspection manually or using sensors. Either way, collecting inspection data is not an easy task. In addition, a very small percentage of parts in a production process are defective in general. It is expected that having sufficient labeled data about defective parts is scarce, especially within a short period. Therefore, knowledge transfer can promise benefits for manufacturers where target data points are limited labeled or unlabeled, and when an existing DA model is needed to be trained quickly to adapt dynamic changes in the manufacturing process.

2.3. Benefits of Knowledge Transfer in Manufacturing

Product quality management [5, 6, 25] and maintenance of machines [26, 27] are two particular manufacturing application areas where knowledge transfer methods have been used. Depending on the application area, different knowledge from data or model(s) can be transferred. In this section, we present benefits of knowledge transfer in the above areas of manufacturing and briefly describe what knowledge can be transferred. Examples of transferred knowledge are summarized in Table 1.

2.3.1. Product Quality Management

These days, product designs frequently change in response to market demands [28]. When a newly-designed product is in the production process, a new DA model might be needed to classify or predict defects (or failures) of the new product.

In response to the changes in product designs, Sankavaram et al. [6] proposed a knowledge transfer framework to detect faults in automotive systems. When a new DA model of a vehicle (or different design of a vehicle) is being produced, it is important to diagnose new types of faults in the early stage of production. However, it is difficult to collect sufficient data in the automotive industry because faults in vehicles rarely occur. Thus, they applied AdaBoost [29] and Learn++.NC [30] to build the new model to help early-stage fault detection in non-stationary environments with limited newly-collected data.

Pulong et al. [25] proposed a method based on incremental learning algorithm for support vector machine (SVM) to recognize faults of a high voltage circuit breaker (HVCB). Their method to update the classifiers is incrementally updating support vectors (SVs) using newly-collected data. Since HVCB faults do not happen very often, it is not easy to obtain sufficient fault samples. By applying knowledge transfer, new faults can be added into the SVM model effectively.

Ramakrishnan and Ghosh [5] proposed a framework called distributed dynamic elastic nets (DDEN) to understand trends of features affecting product conditions in dynamic environments. In the ramp-up phase of an assembly line, dynamic environments are common due to continuous changes in underlying conditions that might lead to defects. The underlying trends of the features are considered to stabilize the fluctuation of parameter weights while optimizing the model, and a stabilized model can increase yield during the ramp-up phase of the production process.

2.3.2. Maintenance

Maintenance in machines plays a key role in reducing manufacturing costs, minimizing downtime of machines, improving product quality, and increasing productivity [31]. DA models can be used to monitor and diagnose the condition of a machine; replacement or repair of the machine can be done by following the results of the DA analysis. The replacement and repair might cause changes in the environment of the data generating process. Aging effects of sensors for condition monitoring may also cause changes in data.

Due to these changes in data, Vilakazi and Marwala [26] applied Learn++ [32] to train a prediction model for fault diagnosis of machines about high voltage bushing condition. The prediction model accommodates newly-collected data or new labels presented in the newly-collected data by adding new classifiers to the existing model. By adopting knowledge transfer, the prediction model can be retrained with a new set of ensemble classifiers, and the machine conditions will be predicted when the environment change.

Yu [27] developed an adaptive hidden Markov model (AHMM) method for condition-based maintenance (CBM). When a new machine health state is detected, the proposed method learns online about such change in its machine health. Thus, through applying AHMM, the model supports recognizing the new type of health degradation at an early stage and allows for timely maintenance service.

2.3.3. Examples of Transferred Knowledge

The benefits mentioned above can be obtained by using the knowledge from data or model. This section presents a set of examples that use knowledge to train the DA models. In Table 1, application area means a field where knowledge transfer is used. We describe the type of changes in data properties, knowledge sources, and transferred knowledge.

Most of the example studies in product quality management and machine maintenance use model as the knowledge source. They focus on continuously updating models when the marginal probability distributions change. In manufacturing, model parameters or structures, which are containing relations between features and labels, are used as the knowledge transferred from model. In case of knowledge transfer from data, relations between features and labels are used to train a new model.

Table 1 Examples of transferred knowledge depending on application areas and changes in data properties.

Application Area	Authors	Changes in Data Properties	Knowledge Source	Transferred Knowledge Used to Train Models
Product Quality Management	Sankavaram et al. [6]	$P(X_S) \neq P(X_T)$	Data	Relations between vehicle operating conditions and faults of vehicle
		$\mathcal{Y}_S \neq \mathcal{Y}_T$	Model	Model parameters for classifying non-changed labels (categories of faults)
	Pulong et al. [25]	$P(X_S) \neq P(X_T)$	Model	Valid classifiers on all source-domain data (Relations between machine conditions and HVCB faults)
	Ramakrishnan and Ghosh [5]	$P(X_S) \neq P(X_T)$	Model	Model parameters (weights) used to classify labels (good or bad) in the source
Maintenance	Vilakazi and Marwala [26]	$\mathcal{Y}_S \neq \mathcal{Y}_T$	Model	Model parameters for classifying non-changed labels (categories of faults)
	Yu [27]	$P(X_S) \neq P(X_T)$	Model	Hidden state and Gaussian components that represent existing failures

Knowledge transfer methods of the above examples are applied not only in manufacturing but also in other domains for various applications. For example, Vilakazi and Marwala [26] applied Learn++ [32] to train the model for fault diagnosis of machines. The training data set involves sets of process settings and inspection results. Learn++ has been also applied in image detection of video events in order to adapt the model to any new class of video events [33]. In other words, knowledge transfer methods are not limited to a certain application, some can be applied for knowledge transfer in manufacturing by various applications in different areas including manufacturing.

In addition, knowledge transfer is commonly used for tasks such as object classification, detection, and text classification. In this regard, similar methods can be applied to solve manufacturing problems. Ferguson et al. [24] used different convolutional neural networks (CNN) architectures to detect and localize casting defects. Instead of solely using a relatively much smaller data set of casting X-ray images, the authors have transferred the knowledge from different CNN architectures that were pre-trained using common object images (e.g., person, bicycle, and car). That allowed the authors to obtain high accuracy in object detection and localization with a smaller data set.

While the above examples show that knowledge transfer is beneficial to manufacturing, there is limited research on this topic for manufacturing applications. The survey discussed in this paper does not solely focus on manufacturing applications, but on various applications of different areas. However, the survey results could support researchers who perform knowledge transfer for manufacturing applications. We also categorize and summarize knowledge transfer methods from the survey to support researchers to easily adopt knowledge transfer in manufacturing applications.

3. Knowledge Transfer Methods in Non-Stationary Environments

In this section, we explore the knowledge transfer methods for classification and regression problems through a literature survey. The methods for transferring knowledge from data or model, presented in the survey are listed in Tables 2 and 3 respectively; the approach, data-related information, and application of each method are also included in the tables. Table 4 summarizes all the surveyed methods in terms of types of changes in data properties, availability of labeled data, and knowledge sources.

3.1. Knowledge Transfer from Data

Knowledge transfer methods with knowledge from data aims at finding "good" data points or feature representations to increase the prediction accuracy and the credibility of the target model. The knowledge transfer methods with knowledge from data are divided into two groups: *instance-based knowledge transfer* and *feature-based knowledge transfer* (see Fig. 3) [10]. Instance-based knowledge transfer [20, 34-38] is an approach that uses weighted source-domain data and target-domain data to train a target model. The weights are used to compensate for the differences between the source and target domains, and they are calculated by comparing the marginal probability distributions or the conditional probability distributions. Feature-based transfer [39-45] is an approach to discover a meaningful feature representation of data points in the source and target domains. By using feature representation, the differences between the source and target domains could be reduced. Like instance-based transfer, feature-based transfer exploits knowledge that is inherent in the source-domain data, but it can be used when the features of the source and target domains are different.

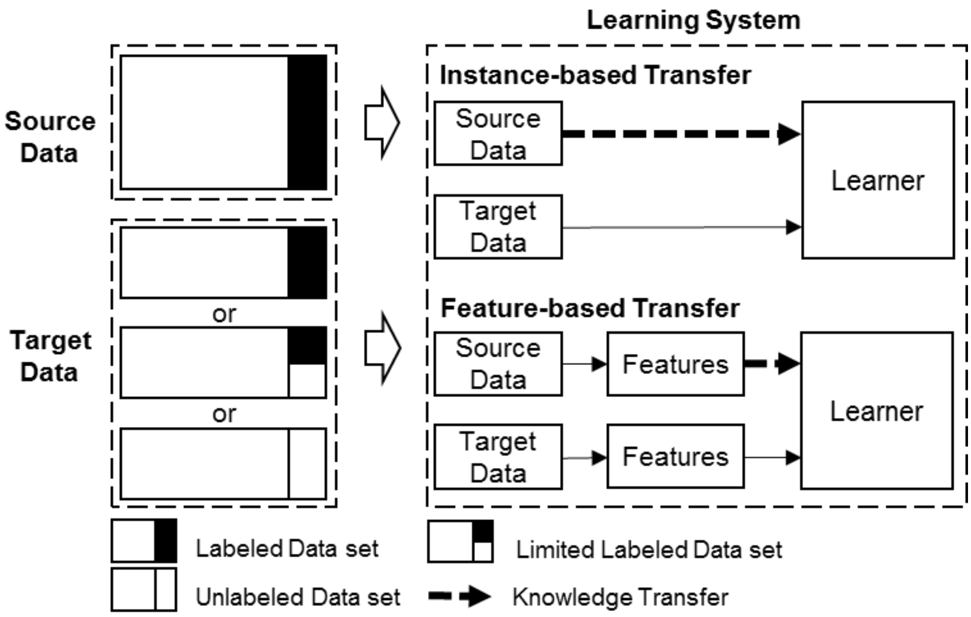

Fig. 3 Two categories of knowledge transfer from data: Instance-based transfer and Feature-based transfer.

3.1.1. Instance-based Knowledge Transfer

Dai et al. [34] proposed an instance-based transfer algorithm, TrAdaBoost, which extends AdaBoost [29]. AdaBoost is a method that improves error of the target predictive model by iteratively giving weights on the training data points. AdaBoost assumes that the distributions of the source domain and the distributions of the target domain are identical. On the other hand, TrAdaBoost assumes that the marginal probability distribution of the target domain is different from the marginal probability distribution of the source domain, even though the source and target have the same feature and label spaces. The classifier is trained using sufficient labeled source-domain data and small amounts of newly-labeled data in the target domain. The key idea is iteratively reweighting the source-domain data to filter out source data points which have different distributions from the target domain. The SVM algorithm was used in this surveyed paper as a basic learner, but any classification algorithm can be used instead of the SVM algorithm.

In the process of extracting and transferring knowledge from a single source, if the source and target domains are not related enough, the dissimilar knowledge can cause a negative impact on the target model. This situation is formally defined as negative transfer [13], i.e., the accuracy of target the model trained only with target-domain data is greater than the accuracy of the target model trained by knowledge transfer. Yao and Doretto [35] proposed Multisource-TrAdaBoost that extends TrAdaBoost [34] to extract knowledge from multiple sources to decrease the negative impact. In the learning system, not only the knowledge extracted from a single-source domain but also the knowledge extracted from relevant multiple-source domains can be used to train the target model ($f_T(\cdot)$) (see Fig. 4). Multisource-TrAdaBoost aims to find a weak classifier from the sources that appears to be the most related to the target in each iteration, and then the final weak classifier is chosen to minimize the target classification error.

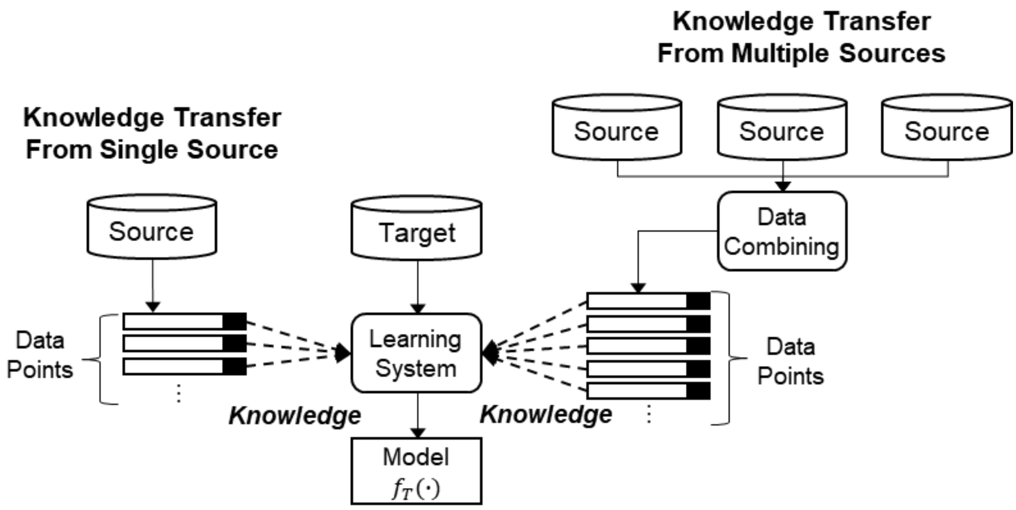

Fig. 4 Comparison of knowledge transfer from a single source and multiple sources.

Pardoe and Stone [36] proposed an instance-based transfer algorithm called TrAdaBoost.R2 that is inspired by ExpBoost [46] and TrAdaBoost [34] for regression on the knowledge extracted from multiple sources. ExpBoost, an extension of AdaBoost [29] is designed for sequential learning tasks. For multiple sources, TrAdaBoost.R2 combines all source-domain data into a single dataset and handles reweighting of each training data point separately. Also, they found that accuracy of TrAdaBoost.R2 decreased when the number of boosting iterations increased. To address this problem, they proposed a two-stage TrAdaBoost.R2 to adjust weights. In the first stage, cross validation is used to adjust the weights of the source data points. In the next stage, the weights of all source data points are frozen while the weights of the target data points are updated using AdaBoost.R2 [47]. They used multivariate regression prediction (M5P) model trees or NN as a basic learner.

Chattopadhyay et al. [37] also proposed a method, conditional probability based multi-source domain adaptation (CP-MDA), for knowledge transfer from multiple sources. The goal of the method is to train the target model using data points from multiple-source domains and a few labeled data points in the target. The key idea is computing the labels of the unlabeled data points in the target, called 'pseudo labels' by integrating multiple-source domains using a set of weights. Finally, the target model is trained from both pseudo and labeled target-domain data. The proposed method is compared with other multi-source knowledge transfer methods: locally weighted ensemble (LWE) [19] and domain adaptation machine (DAM) [48].

Jiang and Zhai [38] proposed a knowledge transfer method for the situation when the conditional probability distribution of the source is different from the conditional probability distribution of the target. The key idea is removing 'misleading' data points from the source, assigning more weights to the labeled data points in the target, and augmenting data points in the source with pseudo labeled data points in the target. Several parameters are introduced, which indicate 1) similarity of conditional probability distributions between the source and target, 2) how to adjust weights of each data point, and 3) how to assign pseudo labels. Also, three approximation methods are proposed, and the target model is trained by controlling the contributions of each approximation method to the target model using introduced parameters. The proposed method is tested with three different natural language processing (NLP) tasks, which are part-of-speech (POS) tagging, entity type classification, and personalized spam filtering.

Hu et al. [20] proposed multi-domain adaptation algorithm based on the class distribution (MACD). The method uses all possible source-domain data to train a new binary classifier. When the distribution of the labels in the target domain is known but the target-domain data is not labeled, MACD is used to train the classifier of target domain. Base classifiers of each source domain are trained using traditional machine learning methods. Relative classifiers of the source domains are selected to obtain pseudo labels of the target by comparing the distribution of the labels. Then, the pseudo labeled target data points are added to the relevant source domains, and the base classifiers of each source are updated. Lastly, a classifier of the target domain is obtained by combining the base classifiers using voting rule [49].

3.1.2. Feature-based Knowledge Transfer

To cope with the change in the marginal probability distributions, Duan et al. [39] proposed a domain transfer multiple kernel learning (DTMKL) method. The method trains a classifier by minimizing the structural risk functional [50] on both the domains. Maximum mean discrepancy measure [51] is used to minimize the difference of marginal probability distributions between the source and target domains. DTMKL can be applied in any kernel method, SVM is used to implement the DTMKL method. The experimental results show that the proposed method can outperform other cross-domain learning algorithms including kernel mean matching [52], cross-domain SVM [53], adaptive SVM [54], and feature replication [55].

Pan et al. [40] proposed a knowledge transfer method to minimize the difference between the source and target domains using a latent space. Maximum mean discrepancy embedding (MMDE), which is a new dimension reduction method, is proposed to learn the latent space. Principal component analysis (PCA) is used to construct low-dimensional representations by selecting the leading eigenvectors. Finally, a traditional machine learning method is used to train the target model using mapping between the target-domain data in the latent space and the labels of the source.

To transfer knowledge when marginal and conditional probability distributions change, Long et al. [41] proposed a Joint Distribution Adaptation (JDA) method. The key idea is jointly adapting changes in both the distributions in a principled dimensionality reduction procedure. A classifier of the source is used to estimate the unlabeled data points in the target. To reduce the difference in the distributions, they adopt maximum mean discovery (MMD) [56] as a distance measure and integrate it with the PCA algorithm that constructs feature representation. The method is tested for image classification using related algorithms including transfer component analysis (TCA) [57] + NN and transfer subspace learning (TSL) [58] + NN.

Long et al. [42] also proposed a knowledge transfer method, which is an adaptation regularization-based transfer learning (ARTL) framework, to reduce the difference in the marginal and conditional probability distributions. In the framework, a supervised classifier finds pseudo labels of the unlabeled target-domain data. Similar to the JDA [41] method, MMD [56] is adopted to measure the difference in the marginal and conditional probability distributions. Joint distributions of the source and target and the structural risk functional on the source data points are minimized, and the manifold consistency is optimized to train an adaptive classifier.

Pan et al. [43] proposed a spectral feature alignment (SFA) method, which is a feature-based knowledge transfer method when the feature spaces of the source and target domains are different. The key idea of the method is finding a new feature representation to reduce the difference between the source and target domains. SFA identifies domain-independent features occurring frequently and acting similar in both domains, and uses the features as a bridge between the source and target. Also, the method identifies domain-specific features that are only used in one specific domain, and a bipartite graph between domain-specific features and domain-independent features is constructed. A spectral clustering on the graph is used to align domain-specific features and domain-independent features to the same cluster, and the target model is trained based on the new representation with clusters.

Shi et al. [44] proposed a heterogeneous spectral mapping (HeMap) to address the differences in the feature and label spaces and the changes in the marginal probability distribution. A common feature space between the source and target domains is constructed using spectral transformation technique to make both data similar. Related data points in the projected space are selected as training data by applying a clustering-based sample selection method [59], and the relationship between the different feature spaces is modeled using a Bayesian-based approach.

Zhou et al. [45] proposed a knowledge transfer method called hybrid heterogeneous transfer learning (HHTL), which allows knowledge transfer across domains even though the corresponding data points are biased across the domains. Once the weights of the source and target data points to hidden representation are calculated, a feature map between the source and target is trained. Then, the distribution bias between the source and target is reduced by discovering a latent representation. The experimental results show that the accuracy of the proposed method is better than the following methods: SVM-source-correspondence (SVM-SC), cross-lingual kernel canonical component analysis (CL-KCCA) [60], HeMap [44], and marginalized stacked denoised autoencoder-CCA (mSDA-CCA).

3.1.3. Summary of Knowledge Transfer from Data

Knowledge transfer methods using knowledge extracted from data are summarized in Table 2. Table 2 presents the name, the approach to extract knowledge from data (transfer approach), the availability of labeled data in the source and target domains (e.g., labeled, limited labeled, and unlabeled), and the applications where the method was demonstrated. All the methods listed in the table need labeled source-domain data. However, knowledge transfer is possible if target-domain data is not labeled. In general, knowledge transfer from data are applied in text classification and image recognition.

3.2. Knowledge Transfer from Model

Knowledge transfer methods with knowledge from model focuses on adapting parameters or structures of models to a new environment, instead of transferring knowledge purely from data. Knowledge transfer methods that transfer knowledge from model can be classified into *single-instance setting* and *batch setting*. They are distinguished based on how many target data points are used for training $f_T(\cdot)$ in the learning system. In a single-instance setting [61-66], the knowledge transfer occurs when training $f_T(\cdot)$ with a single data point (x_{T1}, y_{T1}), whereas in a batch setting [19, 32, 67-71], the knowledge transfer uses multiple data points $\{(x_{T1}, y_{T1}), (x_{T2}, y_{T2}), \dots (x_{Tn}, y_{Tn})\}$. In either setting, knowledge transfer methods are chosen based on 'what to transfer'. SVs of SVM [61, 64, 67, 68, 71] or support vector regression (SVR) [62, 63], prior distributions of Naïve Bayesian models [19, 70] or Gaussian process regression (GPR) [66], model structures of NN [32, 69], and other knowledge from models [65] can all be used as knowledge.

Table 2 Knowledge transfer methods with knowledge from data.

Method	Transfer Approach	Source Data	Target Data	Applications
TrAdaBoost [34]	Instance-based	Labeled	Limited labeled	Classification of news documents and mushrooms
Multisource-TrAdaBoost [35]	Instance-based	Labeled	Limited labeled	Object image classification / Vehicle image detection
TrAdBoost.R2 [36]	Instance-based	Labeled	Labeled	Regression of concrete strength, house price, fuel efficiency, and automobile price
CP-MDA [37]	Instance-based	Labeled	Limited labeled	Fatigue detection
Domain Adaptation for NLP [38]	Instance-based	Labeled	Limited labeled	Natural language processing tasks
MACD [20]	Instance-based	Labeled	Unlabeled	Sentiment classification
DTMKL [39]	Feature-based	Labeled	Unlabeled	Video concept detection / Classification of new documents and spam emails
MMDE [40]	Feature-based	Labeled	Unlabeled	Wi-Fi localizations, Classification of news documents
JDA [41]	Feature-based	Labeled	Unlabeled	Classification of handwritten images, face images, and object images
ARTL [42]	Feature-based	Labeled	Unlabeled	Classification of news documents, handwritten images, and face images
SFA [43]	Feature-based	Labeled	Limited labeled	Sentiment classification of product reviews
HeMap [44]	Feature-based	Labeled	Limited labeled	Drug efficacy prediction / Object image classification
HHTL [45]	Feature-based	Labeled	Unlabeled	Sentiment classification of product reviews

3.2.1. Knowledge Transfer in a Single-instance Setting

Cauwenberghs and Poggio [61] proposed an incremental SVM learning method to update the model on a new data point. The key idea is to retain the Karush-Kuhn-Tucker (KKT) conditions on all previous data points, while updating the SVM model with newly-collected data. When a new data point goes into the learning system, the new data point is checked for meeting KKT conditions. If the new data point is satisfying the current KKT conditions, the existing model would be used continuously without model updating. On the contrary, the new data point is used to update the kernel matrix accordingly. Diehl and Cauwenberghs [72] extended the incremental SVM learning method [61] to a general framework.

Ma et al. [62] proposed an accurate on-line support vector regression (AOSVR) method that extends the method proposed by Cauwenberghs and Poggio [61] for online SVR. The learning strategy of AOSVR is not much different from the incremental SVM learning method [61], but the method is extended for dealing with regression problems. The difference between the incremental SVM learning method [61] and the AOSVR method is a bookkeeping procedure, which is a step to determine the amplitude for changing the category membership of vectors (from reserve to margin/error, from margin to reserve/error, and from error to reserve/margin). The coefficient parameters of the kernel function are being updated until the new data point meets KKT conditions, while ensuring the existing data points also meet the KKT conditions.

Liu and Zio [63] proposed an online learning approach for SVR using the feature vector selection (FVS) method and incremental and decremental learning (Online-SVR-FID) method. The proposed method combines FVS method [73] and the knowledge transfer method proposed by Cauwenberghs and Poggio [61]. The key idea is 1) to judge whether a new data point is a new pattern or a changed pattern, and 2) to modify the model adaptively while retaining the KKT conditions. The new data point is a new pattern if it cannot be represented by existing patterns in the reproducing kernel Hilbert space (RKHS). The new data point is a changed pattern if the data point in RKHS is represented by existing patterns but the predicted value is biased. When the new data point is a new pattern, the new data point (new pattern) is directly used to update the model. If the new data point is a changed pattern, the existing patterns are updated using the new data point, and the changed patterns are used to train the model. Liu and Zio compared Online-SVR-FID with the knowledge transfer method proposed by Cauwenberghs and Poggio [61], Naïve online regularized risk minimization algorithm (NORMA) [74], sparse on-line Gaussian processes (SOGP) [75], and kernel-based recursive least square tracker (KRLS-T) [76]. The results show that the accuracy of Online-SVR-FID is higher and learning time is faster than other methods.

Zheng et al. [64] proposed an online incremental SVM (OI-SVM) method, which does not need to train the model at every new data point. The concept of OI-SVM is similar to Online-SVR-FID [63], but they use prototypes to represent the original data instead of FVs. The prototypes are learned to fit the density of the training data; they are updated when the distance between existing data points and the new data point is larger than thresholds. Then, the model is trained using the prototypes to generate SVs. Both Online-SVR-FID [63] and OI-SVM [64] do not require the original data for knowledge transfer. Also, they can handle large-scale data effectively.

Vijaykumar and Schaal [65] proposed locally weighted projection regression (LWPR), which is a method focusing on finding local projections for local model training. Locally weighted partial least squares (PLS) regression is used to determine the linear model parameters in high-dimensional feature spaces. Next, a nonlinear model is trained by combining local linear models. When a new data point goes to the learning system, the key idea is to find an optimal projection direction to update the model. If the current projections cannot represent the new data point, a new projection is added. Then, the local models are updated according to the projections.

Nguyen-Tuong et al. [66] proposed a local approximation to the standard GPR in a single-instance setting, called local GPR (LGP). The proposed method combines the concept of GPR and LWPR, which is a widely used real-time learning method in high dimensional spaces [77]. To reduce the computational time for model training, the method clusters existing data points. The basic concept is that data points in the same cluster belong to the same local region, and GPR models are trained for those regions. When updating the model with a new data point, the proximity between all available centers of the local region and the new data point is calculated to select the nearest local model. If the proximity to the nearest model is bigger than the threshold, the new data point is allocated to the nearest local region, and the center of the region is updated with the new data point.

If the proximity is below than the threshold, a new local GPR model is trained with the new data point. The prediction is performed by weighted averaging the prediction results of each GPR model, and the weights are obtained by calculating the proximity of the new data point to each GPR model.

3.2.2. Knowledge Transfer in a Batch Setting

If methods suitable for a single-instance setting are used in a batch setting, the methods would be repeated to train the model for each data point, and the computational cost of model training would be increased. Karasuyama and Takeuchi [67] proposed a multiple incremental and decremental SVM (MID-SVM) learning method, which incrementally adapts an SVM-based model to a new environment and works efficiently for multiple data points. Like SVM-based knowledge transfer methods in a single-instance setting [61, 62], each data point is checked for whether it can be classified by using the existing classifier. If the classifier is working, the data point is not used for model update. The key idea of the method is determining the directions and length of changes in Lagrange multipliers, while satisfying KKT conditions. The experimental results show the proposed knowledge transfer method is faster than other knowledge transfer methods [61, 78] suitable for a single-instance setting. The method proposed by Karasuyama and Takeuchi [67] cannot be used when the label space of the source is different from the label space of the target.

Wen and Lu [68] proposed a knowledge transfer method that can be used when the label spaces of the source and target are different. The method, which is called incremental learning by classifier combining (ILbyCC), enables SVM to adapt to a new environment by combining classifiers, while the method does not require access to previously used data points. A new classifier is trained using new data points, and averaged Bayes method [79] is used to combine each classifier. Finally, using the combined classifier, the posterior probability of each data point to all classes is calculated.

Polikar et al. [32] proposed Learn++ that can be used when the source and target have different label spaces, which is a method of incrementally training NN classifiers. Learn ++ does not require access to the original data (source-domain data), while preserving previously acquired knowledge. The method is inspired by the AdaBoost [29], it generates weak hypotheses by training a weak learning algorithm when new data points come into the learning system. Errors of each hypothesis on their training data are calculated, and voting weights are computed based on the errors to combine each hypothesis. Composite hypothesis makes model updating possible when data points with new classes of the label space are introduced. Learn++ was extended to various versions, Muhlbaier et al. [80] proposed Learn++.MT for reducing the number of classifiers while improving the performance of classifiers, Muhlbaier et al. [30] proposed Learn++.NC for learning new classses (NC) of the label space, and Elwell and Polikar [81] proposed Learn++.NSE for change in class definitions over time under non-stationary environments.

Bruzzone and Fernàndez Prieto [69] proposed a knowledge transfer method based on Radial Basis Function (RBF) NN that can be used when the source and target have different label spaces. To train an initial network, the method clusters data points, and prototypes are generated for each partitioned cluster. The prototypes correspond to Gaussian kernel functions which are associated with hidden neurons. The connection between the hidden neurons and the output units are defined by minimizing the sum-of squared errors (SSE). After the initial network is trained, the retraining of the model proceeds using new data points and the prototypes. The most similar prototypes for each data point are selected, and the prototypes are updated using the data point. When new data points cannot be represented by the current prototypes, a new prototype is generated. To update the weights between hidden neurons and output units, SSE before updating the prototype, and SSE after updating the prototype are used.

So far, we have described knowledge transfer methods focusing on how to quickly adapt to a new environment by incrementally and constantly updating models. However, if information of the new environment is not sufficient, models in the similar sources can be used for knowledge transfer. Dai et al. [70] proposed a knowledge transfer method based on the Expectation-Maximization (EM) algorithm and Naïve Bayes classifier, which is called Naïve Bayes transfer classification algorithm (NBTC). The Naïve Bayes classifier is trained using the source-domain data to predict the label of the target when the target-domain data is unlabeled, and the classifier is modified to meet the distribution of the target-domain data based on EM algorithm. Kullback leibler (KL)-divergence is used to measure the distribution difference between the source and target. The experiments are conducted with two semi-supervised methods: transductive SVM (TSVM) [82] and traditional EM-based Naïve Bayes classifier (ENBC) [83]. Like knowledge transfer from data, the knowledge from multiple-source models can also be utilized.

Gao et al. [19] proposed a locally weighted ensemble (LWE) framework that uses knowledge extracted from multiple models via local structure mapping. The key idea of the method is formulating similarity between the multiple-source models (general Bayesian models for classification) and the unknown target model. Clustering is performed on the target-domain data, and neighborhood graphs between the clusters and source models are constructed. The similarity between the source models and the target-domain data is calculated using the graphs, the weights between models and data points of the target is computed.

Tommasi et al. [71] proposed LS-SVM-based model adaptation method for utilizing knowledge from multiple models. The prior knowledge, which is hyperplanes of the classifiers, is transferred by minimizing weighted error rate (WER) based on leave-one-out (LOO) error. LOO error is an unbiased estimator and can be used for model selection [84].

3.2.3. Summary of Knowledge Transfer from Model

Knowledge transfer methods using knowledge extracted from model are summarized in Table 3. Table 3 presents the method name, basic learner, data arrival manner (e.g., single instance and batch), and the applications of the knowledge transfer method. The knowledge transfer methods were developed based on the selected learning algorithms, i.e., basic learners. A basic learner is an algorithm used to train the source model. Data arrival manner means number of target data points used to train $f_T(\cdot)$. Comparison methods are methods used to evaluate the proposed methods. Applications refers to a problem implemented by authors to test proposed methods.

Table 3 Knowledge transfer methods with knowledge from model.

Method	Basic Learner	Data Arrival Manner	Applications
Incremental SVM [61]	SVM	Single instance	-
AOSVR [62]	SVR	Single instance	Regression analysis of fuel efficiency and house price
Online-SVR-FID [63]	SVR	Single instance	Leak flow classification
OI-SVM [64]	SVM	Single instance	Classification of image, text, vehicle ID, cylinder misfire, and income
LWPR [65]	PLS regression	Single instance	Robot Control
LGP [66]	GPR	Single instance	Robot Control
MID-SVM [67]	SVM	Batch	Classification of diabetes, income, forest cover type, and cylinder misfire
ILbyCC [68]	SVM	Batch	Classification of text, type of vehicle, and circular region
Learn ++ [32]	NN	Batch	Classification of text image, type of vehicle, and circular region / Gas sensing
Incremental NN [69]	NN	Batch	Remote sensing image classification
NBTC [70]	Naïve Bayes classifier	Batch	Classification of news documents
LWE [19]	Bayesian model	Batch	Spam filtering / Text classification / Intrusion detection
LS-SVM based Model Adaptation [71]	LS-SVM	Batch	Object image classification

3.3. Analysis and Findings

So far, we have provided an overview of the current knowledge transfer methods that extract knowledge from data or model. Table 4 provides a summary of those methods that is categorized to *availability of labels in target* and *types of changes in data properties*. Availability of labels specifies the availability of labels in the target-domain data. Types of changes in data properties specifies the way in which the source and target domains are different. According to the availability of labels and the types of changes in data properties, available knowledge transfer methods are presented. Knowledge source, as mentioned in Section 2, specifies which type of knowledge is used to train $f_T(\cdot)$.

In summary, knowledge transfer methods differ according to the availability of labels and the type of changes in data properties. The methods are categorized into two types: using knowledge from data and using knowledge from model. The common assumption of methods transferring knowledge from data is that the knowledge is inherent in data. Thus, the methods are different depending on how to extract and transfer the inherent knowledge while reducing the difference (gap) between the source and target. The methods are often limited to specific applications because each method should be able to handle the characteristics of data collected from each domain. Knowledge transfer methods transferring knowledge from model are suitable when data arrives as a continuous stream over time. A few methods can be used when the target-domain data is unlabeled. These methods extract knowledge from the source-domain data in the form of model and transfer knowledge from the model to deal with the limited information of the current environment. The appropriate method for knowledge transfer depends on the basic learner, and the availability of labels in the target-domain data, not the characteristics of data.

Table 4 A summary of knowledge transfer methods.

Availability of Labels in Target	Types of Changes in Data Properties	Knowledge Transfer Methods	Knowledge Source
Labeled	$P(X_S) \neq P(X_T)$	TrAdaBoost.R2 [36]	Data
		Incremental SVM [61]	Model
		AOSVR [62]	Model
		Online-SVR-FID [63]	Model
		OI-SVM [64]	Model
		LWPR [65]	Model
		LGP [66]	Model
		MID-SVM [67]	Model
	$\mathcal{Y}_S \neq \mathcal{Y}_T$	ILbyCC [68]	Model
		Learn ++ [32]	Model
		Incremental NN [69]	Model
		NBTC [70]	Model
	$\mathrm{P}(Y_S\|X_S) \neq P(Y_T\|X_T)$	LS-SVM based Model Adaptation [7]	Model
Limited Labeled	$\chi_S \neq \chi_T$	SFA [43]	Data
		HeMap [44]	Data
	$P(X_S) \neq P(X_T)$	TrAdaBoost [34]	Data
		Multisource-TrAdaBoost [35]	Data
		HeMap [44]	Data
	$\mathcal{Y}_S \neq \mathcal{Y}_T$	HeMap [44]	Data
	$\mathrm{P}(Y_S\|X_S) \neq P(Y_T\|X_T)$	CP-MDA [37]	Data
		Domain Adaptation for NLP [38]	Data
Unlabeled	$\chi_S \neq \chi_T$	HHTL [45]	Data
	$P(X_S) \neq P(X_T)$	MACD [20]	Data
		DTMKL [39]	Data
		MMDE [40]	Data
		JDA [41]	Data
		ARTL [42]	Data
		HHTL [45]	Data
		NBTC [70]	Model
		LWE [19]	Model
	$\mathrm{P}(Y_S\|X_S) \neq P(Y_T\|X_T)$	JDA [41]	Data
		ARTL [42]	Data

4. Knowledge Transfer in Manufacturing

Knowledge transfer can be an effective solution in supporting manufacturing DA. Methodologies [3, 85] to perform knowledge transfer for DA were previously proposed.

Das et al. [85] proposed a methodology for adapting a predictive model to understand the process in unstable environments (e.g., a ramp-up phase in an assembly line) in manufacturing. A probability function (e.g., posterior probability function) is used to detect concept drifts of data. When the concept drifts are detected, it builds an adapted version of the predictive model based on the estimated drifts. The methodology focuses on designing a system for automatic model adaptation, which includes interacting mechanisms between the manufacturing processes, sensors, controllers, and database.

Another methodology which is called *active approach* has been proposed by Dizler et al. [3] for updating DA models in non-stationary environments. It is similar to the methodology proposed by Das et al. [85], but the active approach focuses on algorithms to detect changes in data and to update models in detail. The active approach is divided into three phases: feature extraction, change detection (detector), and adaptation. The feature extraction aims at extracting features to detect changes and to use them for classification or regression. The change detection identifies features affected by the concept drift. The model adaptation phase is activated to update or rebuild DA models.

The above methodologies were designed for certain scenarios; and they have some limitations for knowledge transfer in manufacturing. First, both methodologies only target on the events that cause changes in probability distributions (marginal and conditional). In manufacturing, the feature or label space changes occasionally, but changes in feature or label space are not considered in both methodologies. Second, after the changes in data properties are detected, a source and target to perform knowledge transfer should be defined in advance. In these methodologies, no clear guidance on what can be used as the knowledge source was provided. Lastly, depending on types of changes in data properties and availability of labeled data, available knowledge transfer methods might be different. However, both methodologies only provide learning algorithms that can adapt models to changes in probability distributions.

In this section, we categorize types of events that cause changes in data properties in manufacturing processes, and present additional considerations to overcome these limitations. The considerations are presented within a mechanism by modifying the active approach designed by Dizler et al [3]. The concept of the mechanism is shown in Fig. 5, it includes three steps: detecting changes in data properties, defining source and target with determining difference in data properties, and selecting methods for knowledge transfer in manufacturing. The proposed mechanism does not cover entire steps for training a new model [86], it only describes steps to select knowledge transfer methods in non-stationary environments of manufacturing.

4.1. Events Causing Changes in Data Properties in Manufacturing Process

In manufacturing processes, a large amount of data is generated in the physical systems including machines and sensors [87, 88]. Also, data related to product designs and operation plans can be used together with data generated in the physical system for DA in manufacturing.

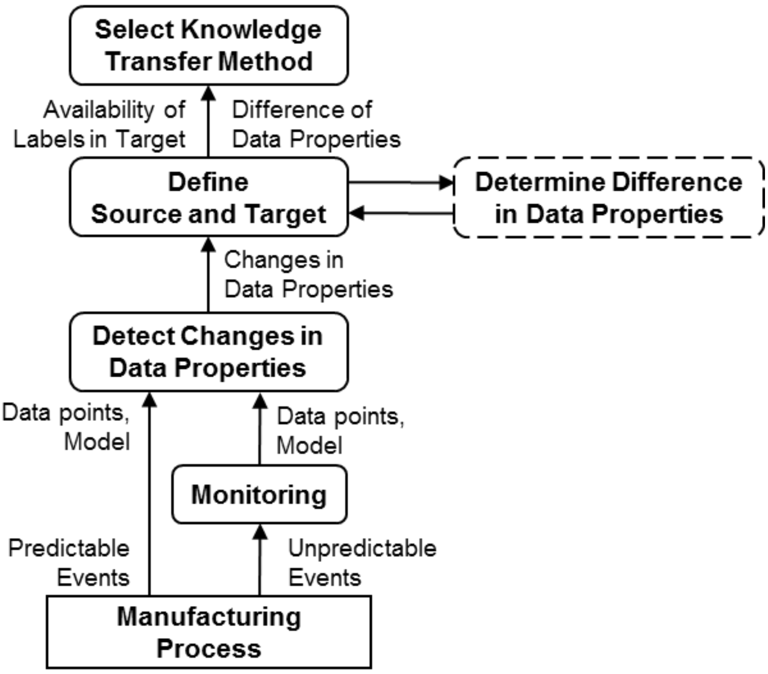

Fig. 5 A mechanism for selecting knowledge transfer methods in a non-stationary environment.

Properties of data generated in the manufacturing process can change when a new event occurs. Events causing changes in data properties can be classified into two types; *predictable* or *unpredictable* according to *whether the occurrence of event is known in advance*. For example, when an engineer adds a new sensor to acquire data which have not been collected in the manufacturing process, the activity *adding a new sensor* is a predictable event that causes changes in the feature space. When a sensor has an aging effect, the aging effect is an unpredictable event because engineers cannot know before changes in probability distributions of data occur.

The definition of the structure in predictable and unpredictable events can be applied to other applications areas (e.g., medical healthcare and finance), but sometimes, only one of the types of events is predominant. For example, in the case of analyzing medical images [89], changes in data properties will be known in advance if different images are used for knowledge transfer. On the other hand, in the case of spam filtering [38], an emergence of a new type of spam can be categorized as an unpredictable event.

Data of which properties often change due to the occurrence of event can be grouped into categories, which are quality management data (e.g., product defect), productivity and maintenance data (e.g., machine states), and traceability data (e.g., raw materials) [90]. Various events can cause changes in data properties in the manufacturing process, we describe examples of the events to help manufacturing engineers to understand their non-stationary environment in Table 5. Table 5 presents the data categories, the type of events, example events according to the data category and type of events, and changed data properties caused by the events.

Table 5 Examples of the events that cause changes in data properties in manufacturing.

Data Categories	Type of Events	Example Events	Changed Data Properties
Quality management	Predictable	A new type of product defect occurs due to a new product design being produced	Probability distributions
		Adding a new sensor to machines for detecting (or classifying) product defects	Label space
	Unpredictable	Constant level of product quality cannot be guaranteed during the same product design being produced without changes in operation settings	Label space
Productivity & maintenance	Predictable	Adding a new sensor to machine for collecting a new data related to machine status	Feature space
		Data collected from machines (or sensors) is shifted due to machine maintenance	Probability distributions
	Unpredictable	Data collected from machines (or sensors) is shifted due to their aging effects	Probability distributions
Traceability	Predictable	Patterns of product defects change due to different raw materials of products being used	Probability distributions

4.2. A Mechanism for Selecting Knowledge Transfer Methods

In this section, we present a mechanism for selecting methods to perform knowledge transfer in manufacturing.

4.2.1. Detecting Changes in Data Properties

Detecting changes in data properties phase focuses on *whether a change has happened* and *what has changed*. Depending on the types of events (predictable or unpredictable), approaches to detect the events causing changes in data properties are different. In case of the predictable events, it is a straightforward task to detect when the changes in data properties occur because engineers have known when the events will occur. To assert changes in the feature or label space, comparing the feature or label space of the previous event with the post event can be used. Changes in probability distributions can be detected by comparing the previous and post events using statistical techniques summarized by Dizler et al. [3].

In case of the unpredictable events, monitoring of the manufacturing process is necessary to detect when data properties have changed. To assert changes in the feature or label space, monitoring only needs to check whether data corresponding to the features or labels is being collected. Changes in probability distributions can be detected using the scoring results of the DA model being used in the manufacturing process. When the results of the model deviate from the predefined accuracy boundary, it means that newly-collected data used for scoring have different data properties compared to data used to train the model. The point at which a new model is needed can be used as the occurrence of the event. If a model being used does not exist, statistical techniques [3] can be used to determine when the new model is needed.

4.2.2. Defining Source and Target

Once changes in data properties are detected, the source and target for knowledge transfer should be defined. The source and target can be identified based on the occurrence of the event. The data or model of the previous events (the old environment) belong to the source-domain data or source model, while data of the post event can belong to the target-domain data.

First of all, differences in the feature and label spaces should be determined because the distributions of data depend on the features and labels. We assume that the label space is determined by a set of labels used when collecting data in the manufacturing process. The difference in the label space can be determined by comparing the labels in the source and target. To define the difference in the feature space, features to be used in the target are determined in advance. Features for training the model can be selected through methods summarized by Guyon and Elisseeff [91]. The authors categorized the methods into three approaches: variable ranking, subset selection, and space dimensionality reduction. However, there are additional considerations to determine the feature space of the target domain. Four possible cases (see. Fig. 6) exist depending on the availability of the source-domain features in the target domain.

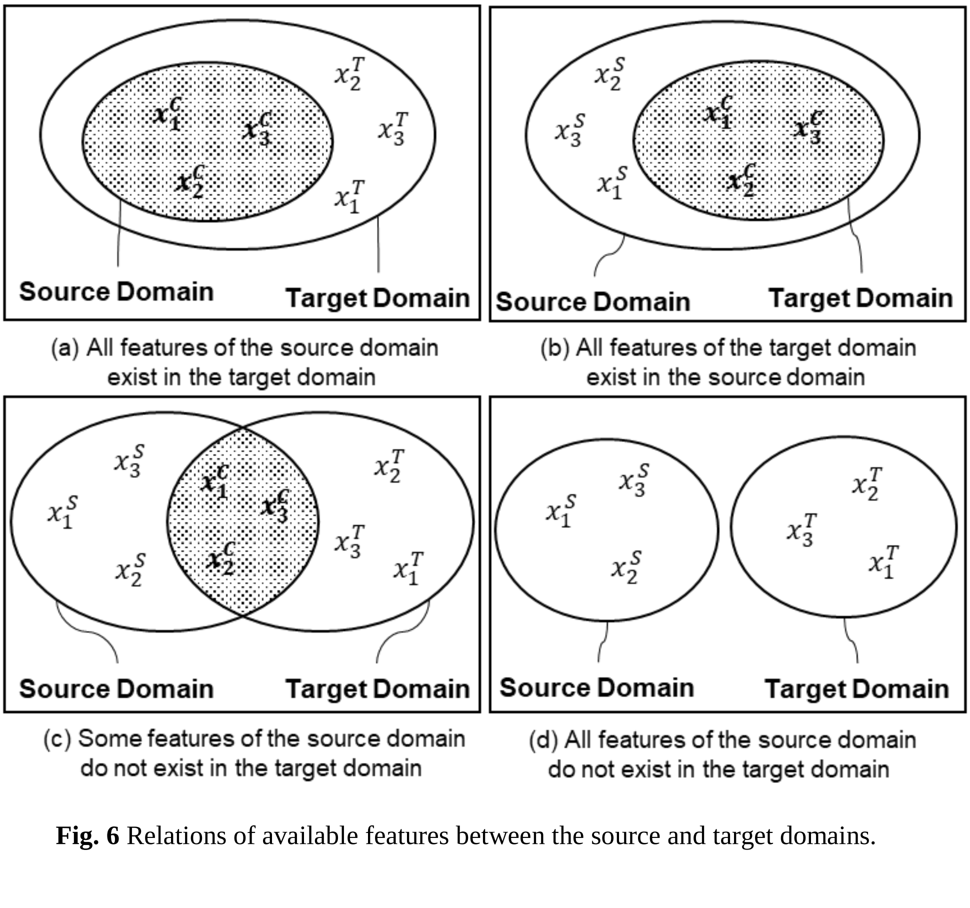

Fig. 6 Relations of available features between the source and target domains.

When all features of the source domain exist in the target domain (see Fig. 6 (a)), the features of the source domain are a subset of the features of the target domain. When the target features are identical to the source features, we can use knowledge transfer methods by adjusting differences in distributions. On the other hand, for example, if the correlation between the feature x_2^{T} and target labels are high enough, the features of the target domain can be $\{x_1^{\mathrm{C}}, x_2^{\mathrm{C}}, x_3^{\mathrm{C}}, x_2^{\mathrm{T}}\}$. When more features that only exist in the target domain are included in the feature space of the target, more data of the target domain can be utilized. However, the difference between the source and target domains will increase.

When some features of the source domain do not exist in the target domain (see Fig. 6 (b) and (c)), the feature spaces of the source and target domains are always different. Thus, in both cases, it is not easy to use models as the knowledge source. Two different types of features exist in the target domain: features which are elements of the intersection of the source and target domains and features which are only elements of the target domain. Similar to the case of Fig. 6 (a), when more features that exist in both domains are used, the source and the target become more similar. On the other hand, when more features that only exist in the target domain are used, more target data can be utilized, but the difference between the source and target domains will increase.

When there is no element in the intersection of the source and target domains (see Fig. 6 (d)), then the selected features of the source and target are completely different. Unlike the above three cases, the features of the source domain do not need to be considered. The feature space of the target domain can be determined in the same way of selecting features in the source domain. In this case, knowledge transfer from source domain to target domain would be more challenging [21].

We assume that the feature space of the source does not change. Target features can be determined by considering four cases above. Once the features of the target are determined, differences in marginal and conditional probability distributions are defined according to the selected target features. In case of the target features that exist in both the source and target, differences in distributions can be defined using methods summarized by Cha [92]. If the target features do not exist in the source, differences in distributions cannot be defined due to the lack of features to be compared.

4.2.3. Selecting Knowledge Transfer Methods

After defining source and target, knowledge transfer method selection is performed by first identifying the availability of labeled data in the target domain. The availability of labeled data in the target domain can be determined by considering the expected profit of DA with labeled data and the expected time (cost) to get the labels. If newly-collected data is not labeled, we should decide whether it is worth collecting labelled data. When the profit is smaller than the loss, using limited labeled or unlabeled target-domain data would be appropriate for avoiding time investment in labelling target-domain data. However, it is often difficult to get the value of profit and loss, the availability of labeled data might rely on domain experts' knowledge and experiences.

The availability of labeled data in the target domain and the difference in data properties are considered to choose appropriate knowledge transfer methods. Table 4 summarizes knowledge transfer methods using the above two major factors, thus referring Table 4 can be a possible way to choose a method for knowledge transfer. Even if the availability of labeled data and difference in data properties are identical, there could be methods transferring knowledge from different knowledge sources. When the time requirement of the model adaptation is important and changes in environments occur frequently, it is appropriate to select methods using a model as the knowledge source to update the model continuously, that is, using model as the knowledge source follows the learning strategy of online (incremental) learning.

When choosing a method from Table 4, Tables 2 and 3 are used to get related information for the method. In some cases, it may be necessary to modify the knowledge transfer method depending on specific needs.

5. Conclusions and Future Work

Since most manufacturers are facing competitive markets, continuous changes in the product design or production process occur to add value to products. Continuous changes in product design and production process also cause changes in data generation, collection, and processing. In this regard, it has become important for DA models to have the capability of responding to continuous and rapid changes in data. In case of degradation in the accuracy of the DA model(s) due to non-stationary data generation and collection environment(s), knowledge transfer could be a good approach to train or update a DA model using knowledge from source data or model, with limited information from the new environment. Knowledge transfer allows a new DA model to be trained efficiently and effectively in non-stationary environments.

In this paper, we have explained the concept of knowledge transfer, and reviewed several knowledge transfer methods that are widely used in various applications. Also, we have introduced benefits of knowledge transfer, and summarized examples of transferred knowledge. Knowledge transfer methods are reviewed considering three factors: knowledge sources, types of non-stationary environments, and availability of labeled data. Among these three factors, we have distinguished knowledge transfer methods based on 'what to transfer' (knowledge sources), and investigated the methods corresponding to each source. From the summary of the methods, we have presented a mechanism to select knowledge transfer methods for manufacturing DA with benefits of reducing or eliminating the degradation in the accuracy of the model in presence of non-stationary environments.

The studies surveyed in this paper have mainly concentrated on problems of classification and regression. Knowledge transfer methods related to clustering problems have not been discussed although clustering algorithms have been used for certain manufacturing problems. Knowledge transfer methods can be selected based on the proposed mechanism, however, a more in-depth theoretical analysis of a generated and extended mechanism to utilize knowledge transfer is required and is under development. The extended mechanism will include a guideline to locate a similar source for knowledge transfer to decrease the negative impact.

Knowledge transfer of DA models is a relatively new paradigm in the manufacturing. This paper can be seen as an initial step towards understanding knowledge transfer and its requirements for manufacturing problems. To utilize knowledge transfer, data and models should be properly stored. In the case of data, storing all generated data points is expensive. It would be cost effective if data is stored in a form that can represent whole data points while reducing the size of data. Also, there could be a difference between scoring engines and communication protocols. If the model presentation is not interoperable, additional time might be required to interpret the model before knowledge transfer. Thus, to store models using a standard format (e.g., Predictive Model Markup Language by Data Mining Group) would enable practitioners to deploy DA models in a faster and easier way. Designing a repository to store data and models appropriately is considered as a future work. Additionally, similarity measures to find appropriate sources whose information is stored in the repository will be considered. To measure the similarity between the source and target in manufacturing, the context (e.g., what and how-to machine) as well as characteristics of data (e.g., feature and label spaces, distributions) should also be taken into consideration. It is necessary to understand the contribution of the features in the manufacturing context. Thus, developing a similarity analysis method to compare manufacturing environments for knowledge transfer would be an important topic of future research.

Disclaimer

Certain commercial systems are identified in this paper. Such identification does not imply recommendation or endorsement by NIST; nor does it imply that the products identified are necessarily the best available for the purpose. Further, any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIST or any other supporting U.S. government or corporate organizations.

References

[1] Lee, J., Lapira, E., Bagheri, B., & Kao, H. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. Manufacturing Letters, 10(1), 38-41.

[2] Harding, J. A., Shahbaz, M., Srinivas, & Kusiak, A. (2006). Data mining in manufacturing: A review. Journal of Manufacturing Science and Engineering, 128(4), 969–976.

[3] Ditzler, G., Roveri, M., Alippi, C., & Polikar, R. (2015). Learning in nonstationary environments: A survey. IEEE Computational Intelligence Magazine, 10(4), 12–25.

[4] He, S.-G., He, Z., & Wang, G. A. (2013). Online monitoring and fault identification of mean shifts in bivariate processes using decision tree learning techniques. Journal of Intelligent Manufacturing, 24(1), 25–34.

[5] Ramakrishnan, N., & Ghosh, R. (2015). Distributed dynamic elastic nets: A scalable approach for regularization in dynamic manufacturing environment. In IEEE 2015 Conference on Big Data (pp. 2752-2761). Santa Clara.

[6] Sankavaram, C., Kodali, A., Pattipati, K. R., & Singh, S. (2015). Incremental classifiers for data-driven fault diagnosis applied to automotive systems. IEEE Access, 3, 407–419.

[7] Seera, M., Lim, C. P., & Loo, C. K. (2016). Motor fault detection and diagnosis using a hybrid FMM-CART model with online learning. Journal of Intelligent Manufacturing, 27(6), 1273–1285.

[8] Moyne, J., Samantaray, J., & Armacost, M. (2016). Big data capabilities applied to semiconductor manufacturing advanced process control. IEEE Transactions on Semiconductor Manufacturing, 29(4), 283–291.

[9] Windmann, S., Maier, A., Niggemann, O., Frey, C., Bernardi, A., Gu, Y., Pfrommer, H., Steckel, T., Krüger, M., & Kraus, R. (2015). Big data analysis of manufacturing processes. Journal of Physics: Conference Series 659(1), 1-12.

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

[11] Gepperth, A., & Hammer, B. (2016). Incremental learning algorithms and applications. In Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning. Bruges.

[12] 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.

[13] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data 3(9), 1-40.

[14] Giraud-Carrier, C. (2000). A note on the utility of incremental learning. AI Communications, 13(4), 215–223.

[15] Joshi, P., & Kulkarni, P. (2012). Incremental learning: Areas and methods – A survey. International Journal of Data Mining & Knowledge Management Process, 2(5), 43–51.

[16] Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Computer Science Department, Trinity College Dublin, 106(2).

[17] Kang, S. (2018). On effectiveness of transfer learning approach for neural network-based virtual metrology modeling. IEEE Transactions on Semiconductor Manufacturing. 31(1). 149-155.

[18] Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. Morgan and Claypool Publishers.

[19] Gao, J., Fan, W., Jiang, J., & Han, J. (2008). Knowledge transfer via multiple model local structure mapping. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 283-291). Las Vegas.

[20] Hu, K., Zhang, Y., & Hu, X. (2012). A multi-domain adaptation for sentiment classification algorithm based on class distribution. In IEEE 2012 Conference on Granular Computing (pp. 179-184). Hangzhou.

[21] Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (pp. 120-128). Sydney.

[22] Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine learning in manufacturing: advantages, challenges, and applications. Production and Manufacturing Research, 4(1), 23-45.

[23] Lu, S. C.-Y. (1990). Machine learning approaches to knowledge synthesis and integration tasks for advanced engineering automation. Computers in Industry, 15(1-2), 105-120.

[24] Ferguson, M., Ak, R., Lee, Y.-T. T., & Law, K. H. (2017). Automatic localization of casting defects with convolutional neural networks. In IEEE 2017 Conference on Big Data (pp. 1726-1735). Boston.

[25] Pulong, G., Jiancheng, S., Chunyu, X., & Yu, Z. (2016). Fault pattern recognition method for the high voltage circuit breaker based on the incremental learning algorithms for SVM. In Proceedings of the 2016 International Conference on Condition Monitoring and Diagnosis (pp. 693-696). Xi'an.

[26] Vilakazi, C. B., & Marwala, T. (2007). Online incremental learning for high voltage bushing condition monitoring. In Proceedings of the 2007 International Joint Conference on Neural Networks. Orlando.

[27] Yu, J. (2017). Adaptive hidden Markov model-based online learning framework for bearing faulty detection and performance degradation monitoring. Mechanical Systems and Signal Processing, 83, 149–162.

[28] Malaca, P., Rocha, L. F., Gomes, D., Silva, J., & Veiga, G. (2016). Online inspection system based on machine learning techniques: Real case study of fabric textures classification for the automotive industry. Journal of Intelligent Manufacturing, 1–11.

[29] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119–139.

[30] Muhlbaier, M. D., Topalis, A., & Polikar, R. (2009). Learn++ .NC: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. IEEE Transactions on Neural Networks, 20(1), 152–168.

[31] Bashiri, M., Badri, H., & Hejazi, T. H. (2011). Selecting optimum maintenance strategy by fuzzy interactive linear assignment method. Applied Mathematical Modelling, 35(1), 152–164.

[32] Polikar, R., Udpa, L., Udpa, S. S., & Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 31(4), 497–508.

[33] Wali, A., & Alimi, A. M. (2010). Incremental learning approach for events detection from large video dataset. In IEEE 2010 Conference on Advanced Video and Signal Based Surveillance (pp. 555-560). Boston.

[34] Dai, W., Yang, Q., Xue, G.-R., & Yu, Y. (2007). Boosting for transfer learning. In Proceedings of the 24th International Conference on Machine Learning (pp.193-200). Corvalis.

[35] Yao, Y., & Doretto, G. (2010). Boosting for transfer learning with multiple sources. In IEEE 2010 Conference on Computer Vision and Pattern Recognition (pp. 1855-1862). San Francisco.

[36] Pardoe, D., & Stone, P. (2010). Boosting for regression transfer. In Proceedings of the 27th International Conference on Machine Learning (pp. 863-870). Haifa.

[37] Chattopadhyay, R., Ye, J., Panchanathan, S., Fan, W., & Davidson, I. (2012). Multi-source domain adaptation and its application to early detection of fatigue. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(4), 18.

[38] Jiang, J., & Zhai, C. (2007). Instance weighting for domain adaptation in NLP. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (pp. 264-271). Prague.

[39] Duan, L., Tsang, I. W., & Xu, D. (2012). Domain transfer multiple kernel learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(3), 465–479.

[40] Pan, S. J., Kwok, J. T., & Yang, Q. (2008). Transfer learning via dimensionality reduction. In Proceedings of the 23th AAAI conference on Artificial Intelligence (pp. 677-682). Chicago.

[41] Long, M., Wang, J., Ding, G., Sun, J., & Yu, P. S. (2013). Transfer feature learning with joint distribution adaptation. In IEEE 2013 Conference on Computer Vision (pp. 2200-2207). Sydney.

[42] Long, M., Wang, J., Ding, G., Pan, S. J., & Yu, P. S. (2014). Adaptation regularization: A general framework for transfer learning. IEEE Transactions on Knowledge and Data Engineering, 26(5), 1076–1089.

[43] Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., & Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In Proceedings of the 19th International Conference on World Wide Web (pp. 751-760). Raleigh.

[44] Shi, X., Liu, Q., Fan, W., Yu, P. S., & Zhu, R. (2010). Transfer learning on heterogenous feature spaces via spectral transformation. In IEEE 2010 Conference on Data Mining (pp. 1049-1054). Sydney.

[45] Zhou, J. T., Pan, S. J., Tsang, I. W., & Yan, Y. (2014). Hybrid heterogeneous transfer learning through deep learning. In Proceedings of the 28th AAAI Conference on Artificial Intelligence (pp. 2213-2220). Quebec.

[46] Rettinger, A., Zinkevich, M., & Bowling, M. (2006). Boosting expert ensembles for rapid concept recall. In Proceedings of the 21th National Conference on Artificial Intelligence (pp. 464-469). Boston.

[47] Drucker, H. (1997). Improving regressors using boosting techniques. In Proceedings of the 14th International Conference on Machine Learning (pp.107-115). San Francisco.

[48] Duan, L., Tsang, I. W., Xu, D., & Chua, T.-S. (2009). Domain adaptation from multiple sources via auxiliary classifiers. In Proceedings of the Annual International Conference on Machine Learning (pp. 289-296). Montreal.

[49] Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3), 226–239.

[50] Vapnik, V. (1992). Principles of risk minimization for learning theory. In Advances in Neural Information Systems 4 (pp. 831-838). Morgan Kaufmann.

[51] Bruzzone, L., & Marconcini, M. (2010). Domain adaptation problems: A DASVM classification technique and a circular validation strategy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(5), 770–787.

[52] Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., & Scholkopf, B. (2007). Correcting sample selection bias by unlabeled data. In Advances in Neural Information Processing Systems 19 (pp. 601-608). Cambridge: MIT Press.

[53] Jiang, W., Zavesky, E., Chang, S.-F., & Loui, A. (2008). Cross-domain learning methods for high-level visual concept classification. In IEEE 2008 Conference on Image Processing (pp. 161-164). San Diego.

[54] Yang, J., Yan, R., & Hauptmann, A. G. (2007). Cross-domain video concept detection using adaptive SVMs. In Proceedings of the 15th ACM International Conference on Multimedia (pp. 188-197). Augsburg.

[55] Daumé III, H. (2007). Frustratingly easy domain adaptation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (pp. 256-263). Prague.

[56] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. J. (2007). A kernel method for the two-sample-problem. In Advances in Neural Information Processing Systems 19 (pp. 513-520). Cambridge: MIT Press.

[57] Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2011). Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks, 22(2), 199–210.

[58] Si, S., Tao, D., & Geng, B. (2010). Bregman divergence-based regularization for transfer subspace learning. IEEE Transactions on Knowledge and Data Engineering, 22(7), 929–942.

[59] Shi, X., Liu, Q., Fan, W., Yang, Q., & Philip, S. Y. (2010). Predictive modeling with heterogeneous sources. In Proceedings of the 10th SIAM International Conference on Data Mining (pp. 814-825). Columbus.

[60] Vinokourov, A., Shawe-Taylor, J. S., & Cristianini, N. (2003). Inferring a semantic representation of text via cross-language correlation analysis. In Advances in Neural Information Processing Systems 15 (pp. 1497-1504). Cambridge: MIT Press.

[61] Cauwenberghs, G., & Poggio, T. (2001). Incremental and decremental support vector machine learning. In Advances in Neural Information Processing Systems 13 (pp. 388-394). Cambridge: MIT Press.

[62] Ma, J., Theiler, J., & Perkins, S. (2003). Accurate on-line support vector regression. Neural computation, 15(11), 2683–2703.

[63] Liu, J., & Zio, E. (2016). An adaptive online learning approach for support vector regression: Online-SVR-FID. Mechanical Systems and Signal Processing, 76–77, 796–809.

[64] Zheng, J., Shen, F., Fan, H., & Zhao, J. (2013). An online incremental learning support vector machine for large-scale data. Neural Computing and Applications, 22(5), 1023–1035.

[65] Vijayakumar, S., & Schaal, S. (2000). Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space. In Proceedings of the 17th International Conference on Machine Learning (pp. 283-293). Stanford.

[66] Nguyen-Tuong, D., Seeger, M., & Peters, J. (2009). Model learning with local Gaussian process regression. Advanced Robotics, 23(15), 2015–2034.

[67] Karasuyama, M., & Takeuchi, I. (2010). Multiple incremental decremental learning of support vector machines. IEEE transactions on neural networks, 21(7), 1048–1059.

[68] Wen, Y.-M., & Lu, B.-L. (2007). Incremental learning of support vector machines by classifier combining. In Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 904-911). Nanjing.

[69] Bruzzone, L., & Fernàndez Prieto, D. (1999). Incremental-learning neural network for the classification of remote-sensing images. Pattern Recognition Letters, 20(11–13), 1241–1248.

[70] Dai, W., Xue, G.-R., Yang, Q., & Yu, Y. (2007). Transferring naive bayes classifiers for text classification. In Proceedings of the 22nd National Conference on Artificial Intelligence (pp.540-545). Vancouver.

[71] Tommasi, T., Orabona, F., & Caputo, B. (2010). Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In IEEE 2010 Conference on Computer Vision and Pattern Recognition (pp. 3081-3088). San Francisco.

[72] Diehl, C. P., & Cauwenberghs, G. (2003). SVM incremental learning, adaptation and optimization. In Proceedings of the International Joint Conference on Neural Networks (pp.2685-2690). Portland.

[73] Baudat, G., & Anouar, F. (2003). Feature vector selection and projection using kernels. Neurocomputing, 55(1–2), 21–38.

[74] Kivinen, J., Smola, A. J., & Williamson, R. C. (2004). Online learning with kernels. IEEE Transactions on Signal Processing, 52(8), 2165–2176.

[75] Csató, L., & Opper, M. (2002). Sparse online Gaussian processes. Neural Computation, 14(3), 641–668.

[76] Vaerenbergh, S. V., Lazaro-Gredilla, M., & Santamaria, I. (2012). Kernel recursive least-squares tracker for time-varying regression. IEEE Transactions on Neural Networks and Learning Systems, 23(8), 1313–1326.

[77] Schaal, S., Atkeson, C. G., & Vijayakumar, S. (2002). Scalable techniques from nonparametric statistics for real time robot learning. Applied Intelligence, 17, 49–60.

[78] Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3), 1–27.

[79] Xu, L., Krzyżak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Systems, Man and Cybernetics, 22(3), 418–435.

[80] Muhlbaier, M., Topalis, A., & Polikar, R. (2004). Learn++.MT: A new approach to incremental learning. In International Workshop on Multiple Classifier Systems (pp. 52-61). Cagliari.

[81] Elwell, R., & Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks, 22(10), 1517–1531.

[82] Joachims, T. (1999). Transductive inference for text classification using support vector machines. In Proceedings of the 16th International Conference on Machine Learning (pp. 200-209). Bled.

[83] Nigam, K., Maccallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2), 103–134.

[84] Cawley, G. C. (2006). Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs. In Proceedings of the International Joint Conference on Neural Networks (pp. 1661-1668). Vancouver.

[85] Das, S., Lade, P., Srinivasan, S., & Ghosh, R. (2017). Model adaption and online learning for unstable environments. United States Patent US 2017/0206469.

[86] Kurgan, L., & Musilek, P. (2006). A survey of knowledge discovery and data mining process models. The knowledge Engineering Review. 21(1). 1-24.

[87] Liu, M., Ma, J., Lin, L., Ge, M., Wang, Q., & Liu, C. (2017). Intelligent assembly system for mechanical products and key technology based on internet of things. Journal of Intelligent Manufacturing, 28(2), 271–299.

[88] Lechevalier, D. Narayanan, A., Rachuri, S., & Foufou, S. (2018). A methodology for the semi-automatic generation of analytics models in manufacturing. Computers in Industry, 95, 54-67.

[89] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis, 42, 60-88.

[90] Cutting-Decelle, A. F., Barraud, J. L., Veenenbaal, B., & Young, R. I. (2012). Production information interoperability over the internet: A standardized data acquisition tool developed for industries enterprises. Computers in Industry, 63, 824-834.

[91] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine

Learning Research, 3, 1157–1182.

[92] Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences, 4(1), 300-307.