

Domain Adversarial Transfer Learning for Generalized Tool Wear Prediction

Matthew Russell¹ and Peng Wang^{1,2}

^{1,2}*Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, 40506, USA*
matthew.russell@uky.edu

²*Department of Mechanical Engineering, University of Kentucky, Lexington, KY, 40506, USA*
Edward.Wang@uky.edu

ABSTRACT

Given its demonstrated ability in analyzing and revealing patterns underlying data, Deep Learning (DL) has been increasingly investigated to complement physics-based models in various aspects of smart manufacturing, such as machine condition monitoring and fault diagnosis, complex manufacturing process modeling, and quality inspection. However, successful implementation of DL techniques relies significantly on the amount, variety, and veracity of data for robust network training. Also, the distributions of data used for network training and application should be identical to avoid the internal covariance shift problem that reduces the network performance applicability. As a promising solution to address these challenges, Transfer Learning (TL) enables DL networks trained on a source domain and task to be applied to a separate target domain and task. This paper presents a domain adversarial TL approach, based upon the concepts of generative adversarial networks. In this method, the optimizer seeks to minimize the loss (i.e., regression or classification accuracy) across the labeled training examples from the source domain while maximizing the loss of the domain classifier across the source and target data sets (i.e., maximizing the similarity of source and target features). The developed domain adversarial TL method has been implemented on a 1D CNN backbone network and evaluated for prediction of tool wear propagation, using NASA's milling dataset. The experimental results indicate that domain adversarial TL can successfully allow DL models trained on certain scenarios to be applied to other scenarios.

1. INTRODUCTION

In manufacturing processes, certain quantities and measurements can reveal critical information about the health

condition of manufacturing machine tools, process efficiency, or product quality. The relevant data collection and analysis, known as condition monitoring, contribute telemetry needed to transform traditional manufacturing into smart, data-driven manufacturing. Deep Learning (DL) provides the necessary tools for handling the big manufacturing data and translating the raw data into information and knowledge that can facilitate process-level and system-level decision makings, such as machine tool predictive maintenance and process optimization (Lee, Jin, Bagheri, & Chao, 2016). For example, one DL variant, the convolutional neural network (CNN), has been investigated to identify machine operating condition, fault detection and diagnosis, tool wear and remaining useful life (RUL) prediction, and product assessment (Wang, Ma, Zhang, Gao, & Wu, 2018; Li, Ota, & Dong, 2018). However, the performance of DL models primarily relies on the amount and variety of training data; DL models are not inherently generalizable. In manufacturing applications, the robustness of DL models is not guaranteed due to process-to-process variation and changes in operating conditions. With enough training examples under certain operating conditions, DL networks can achieve satisfactory performance but will suffer significant adverse effects when utilized outside the strict confines of the original conditions. As a promising solution to overcome this limitation, Transfer Learning (TL) allows DL models trained on a source task or domain to be transferred to a second, related task or input domain (Pan & Yang, 2010).

Within the scope of TL, domain adaptation addresses performing a single task (e.g., RUL prediction) across input domains with differences caused by changing operating conditions or different types of manufacturing machine tools. The change in input data distribution is known as the covariate shift problem, which limits the applicability of DL models (Shimodaira, 2000). To fundamentally address the problem, DL models must adapt to compensate for the differences across the domains through learning common

Matthew Russell et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

latent features shared by the domains. Several methods have been proposed for feature generalization of DL models across the source domain and the target domain. Some methods achieve feature generalization by adding an explicit loss term to the DL network loss function, to penalize the network for generating different feature distributions from source and target inputs. To measure the feature distribution distance, metrics including Correlation Alignment (CORAL), Kullback-Leibler Divergence (KLD), and Maximum Mean Discrepancy (MMD), have been defined or leveraged (Sun & Saenko, 2016; Pan, Kwok, & Yang, 2008; Tzeng, Hoffman, Zhang, Saenko, & Darrell, 2014). Alternative methods for coalescing feature distributions opt for an adversarial approach, in which a domain classifier attempts to identify which input domain, source or target, an input feature came from, while the feature extraction layers compete with it to develop a feature mapping that prevents distinguishability. Several variants of this method include the ReverseGrad approach from Ganin & Lempitsky (2015) and Ganin et al. (2016), Adversarial Discriminative Domain Adaptation (ADDA) introduced by Tzeng et al. (2017), and conditional adversarial domain adaptation discussed by Long et al. (2018) and related work by Hoffman et al. (2017). Domain adversarial transfer learning will be investigated in this paper because of its ability to generalize features without the need for a separately-defined measurement metric.

Recent work in smart manufacturing has explored TL via domain adaptation in connection to condition monitoring. Sun et al. (2018) investigated KLD to promote feature similarity for RUL prediction through an autoencoder DL architecture. Similarly, Wen et al. (2019) examined the benefits of MMD as a feature discrepancy metric for TL in bearing fault classification scenarios. Adversarial approaches have also received attention and demonstrated promising results. Liu et al. (2019) investigated the advantages of applying a GAN loss-based approach in keeping with ADDA from Tzeng et al. (2017) to transfer knowledge from a pretrained source LSTM network to a target domain with a different cutting tool. Guo et al. (2019) combined MMD-based feature discrepancy loss with domain adversarial GAN-style loss to classify machine health conditions, via features extracted from vibration data by 1D CNNs. Zhang et al. (2019) applied the ADDA approach to generalized feature generation for a 2D CNN-based characterization of vibration frequency spectra. More recently, da Cost et al. (2020) successfully applied the ReverseGrad approach that is initially developed by Ganin et al. (2016) to predict RUL of aircraft turbofan engines operating under different altitudes and throttle levels.

This study develops a Domain Adversarial Transfer Learning (DATL) method through ReverseGrad for domain adaptation of tool wear regression and prediction during a milling process. Vibration signals are processed by a 1D CNN, and the extracted features are then correlated to tool wear through fully-connected layers. A domain classifier attempts to

distinguish between features generated from the source domain and target domain data, which were collected under different operating conditions. The entire network can be divided into three parts according to their functions: feature extraction, domain classification, and wear prediction, which are optimized simultaneously with a gradient reversal layer between the feature extractor and the domain classifier. Through the gradient reversal layer, the 1D CNN is forced to extract features that confuse the concurrently optimized domain classifier. For performance evaluation, the performance improvement of the developed DATL versus the availability of target domain data is demonstrated.

The remainder of this paper is organized as follows: Section 2 provides the theoretical background required by the approach, Section 3 outlines the experimental dataset and architecture, Section 4 presents results and analysis, and Section 5 concludes with a discussion of future work.

2. DOMAIN ADVERSARIAL TRANSFER LEARNING (DATL)

2.1. Deep Learning and Convolutional Neural Networks

DL represents a hierarchical learning structure, as a combination of multilayer artificial neural networks and specialized network architectures. Analogous to progressive stages of abstraction, early layers in the networks learn intermediate representations of the input data (i.e., low-level features), while later layers fuse these low-level features to high-level features that better reveal the properties of the objects to be analyzed. Due to the large numbers of free parameters inherent to DL models, several specialized architectures have been developed to leverage known underlying structures in the input data. One highly successful architecture is CNN. Instead of a fully-connected layer that simultaneously learns weights for each input neuron, CNNs take advantage of underlying spatial connections in each input example. For example, images often consist of building blocks such as edges and shapes that constitute high-level motifs. Thus, instead of attempting to directly apply a dense, fully-connected network to processing image pixels, CNNs search for smaller pattern blocks within the larger input by convolving kernels with the input. Each separate kernel produces a new channel of the input image, representing an extracted pattern. The convolutional output can then be compressed by pooling values in adjacent regions, summarizing these regions to suppress local variations in the pixels while at the same time reducing the number of network parameters to be tuned and improving the computational efficiency. Hierarchical stacks of convolutional and pooling layers are employed as feature extraction layers in DL models, which end with a full-connected network for classification or regression on the learned features.

CNNs operating on 2-dimensional input have demonstrated excellent performance in image recognition problems, and 1D CNNs have proved successful on sequential inputs such

as vibration signals (Krizhevsky, Sutskever, & Hinton, 2012; Abdeljaber, Avci, Kiranyaz, Gabbouj, & Inman, 2017). For 1D inputs and kernels, the convolution operation can be expressed as

$$y_c(n) = \sum_{k=0}^{L-1} w(k)x(nS_c + k), \quad (1)$$

where w is the convolutional kernel of length L , x is the input signal of length N , and S_c is the stride or the number of data points that the kernel moves per step across the input. The bounds of n can be set to stay within the original signal length (or including zero-padded regions on sides of the input). The convolutional output $y_c(n)$ can then be passed through an activation function such as the rectified linear unit (ReLU) to introduce nonlinearity:

$$y_a(n) = \max(0, y_c(n)). \quad (2)$$

Next, this activation result $y_a(n)$ is pooled. Max pooling is often used and selects the maximum value in an M -length sliding window across the input:

$$y(n) = \max_{0 \leq k \leq M-1} (y_a(nS_p + k)). \quad (3)$$

Here, S_p is the stride of the pooling window step across the obtained features from the convolutional operations. One or more convolution and pooling layers may be stacked in sequence to construct the feature extraction network. The outputs are then fed into a fully-connected, multilayer network for classification or regression. CNNs can be trained using traditional backpropagation techniques, including stochastic gradient descent (SGD) and RMSProp-based Adam algorithm.

2.2. Transfer Learning

With potentially millions of parameters, DL models are capable of fitting to very complex problems when equally large amounts of training data are available for the desired task. However, due to complexities or physical limitations in some application scenarios, collecting data with proper annotations may not be physically or economically feasible, which creates a challenge for robust model training. With these considerations in mind, Transfer Learning (TL) seeks to discover ways of transferring knowledge learned on a well-posed problem source domain with sufficient training data to a related target domain task. Pan & Yang (2010) outlined several TL variants, which are developed to address different application scenarios, including 1) transferring knowledge obtained from one task to a new task in the same data domain (e.g., transferring vibration analysis from tool wear monitoring to chattering control); 2) transferring from one data domain to another data domain for the same task (e.g.,

machine tool RUL prediction w.r.t. different types of machine tools under different operating conditions); and 3) a mix of the first two scenarios. The second application is also known as domain adaptation and is of particular interest to machine condition monitoring and health management, since common prognosis tasks may encounter challenges from data differences due to process-to-process variation or varying operating conditions. Thus, it is desirable to train a model on a source domain, but maintain its performance on the other domains under the existence of data variances.

With notation from Pan & Yang (2010), the related tasks share a common input set $\mathcal{X}$ and output domain $\mathcal{Y}$. The source domain $\mathcal{D}_s$ and the target domain $\mathcal{D}_t$ can be written with their respective marginal probabilities:

$$\begin{aligned} \mathcal{D}_s &= \{\mathcal{X}; P(\mathcal{X})\} \\ \mathcal{D}_t &= \{\mathcal{X}; Q(\mathcal{X})\} \end{aligned} \quad (4)$$

Although the domain inputs are drawn from the same set $\mathcal{X}$, they do not follow the same distribution due to the variance in operating conditions and other factors, i.e., $P(\mathcal{X}) \neq Q(\mathcal{X})$, and the distribution of the covariate of the desired output has shifted between the two scenarios. Since the DL model learns the conditional probability of the output on the input distribution, this creates an inherent difficulty termed the covariance shift problem (Shimodaira, 2000). Domain adaptation seeks to eliminate the difference in marginal probabilities by developing a feature mapping that generates a consistent feature distribution across both domains. That is, domain adaptation searches for nonlinear feature mapping $\phi(\cdot)$ such that

$$P(\phi(\mathcal{X})) = Q(\phi(\mathcal{X})). \quad (5)$$

Pan et al. (2008) proposed using Maximum Mean Discrepancy (MMD), a reproducing kernel Hilbert space to find a representation in which the marginal probabilities are similar or identical, and Pan et al. (2011) further extended this work to introduce Transfer Component Analysis (TCA) to improve computational efficiency. Deep Domain Confusion (DDC) incorporates a discrepancy metric into the network loss function (to be minimized) to encourage the feature extraction layers to find a single distribution of features (i.e., common latent features) across both the source and target inputs (Tzeng, Hoffman, Zhang, Saenko, & Darrell, 2014). A few metrics have been proposed, including Kullback-Liebler Divergence (KLD), MMD, and Correlation Alignment (CORAL) for deep networks introduced by Sun & Saenko (2016). These features measure the distance or discrepancy between feature distributions statistically or probabilistically. During the network training phase, this distance penalizes feature mappings which separate the source and target features. However, these metrics must be designed and tuned to adequately capture the similarity

between the two feature distributions, and characteristics of the input data may affect how well each metric performs.

2.3. Principles of DATL

Eliminating dependency on these metrics, Ganin & Lempitsky (2015) introduced, and Ganin et al. (2016) further developed, a different approach for encouraging similar source and target feature distributions based on adversarial networks known as DATL. Instead of a discrepancy metric in the network loss function, a new domain classifier network is added in parallel to the original fully connected layers for regression (see Fig. 1). The goal of this domain classifier is to determine which input distribution a given feature came from. Optimizing this classifier improves its ability to differentiate these source and target features.

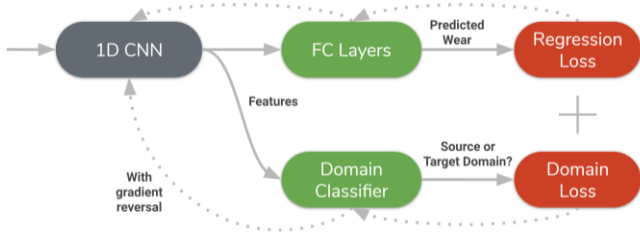


Figure 1. A 1D CNN ReverseGrad domain adversarial approach for tool wear regression with transfer learning

It should be mentioned that once gradients have been backpropagated through the classifier, they are reversed before continuing back through the feature extraction layers (i.e., 1D CNN in Fig. 1). This forces the domain classifier to seek the best parameters to distinguish the source and target domain features, while simultaneously have the feature extraction in 1D CNN to move towards mappings that cause the features of two domains to be indistinguishable. In this adversarial manner, the network is encouraged to converge to feature extraction layer parameters, which map the features of both the source and target domain to the same distribution. The overall loss function can be summarized as

$$J = J_y + \lambda J_d, \quad (6)$$

where J is the total loss, J_y is the loss of the output classifier or regressor, and J_d is the loss of the domain classifier, with a regularization parameter λ . During feedforward prediction, the gradient reversal layer between the 1D CNN and the domain classifier acts as the identity function, but during backpropagation, the gradient from the domain classifier is reversed as:

$$\delta_{FE}^{-1} = -\alpha \delta_{DC}^0, \quad (7)$$

where δ_{DC}^0 is the gradient at the input of the domain classifier, δ_{FE}^{-1} is the gradient at the last layer of the feature extractor, and α is a hyperparameter that controls the impact of the domain classifier on the optimization of the feature extraction layers.

Additional variants of the adversarial approach have been proposed, most notably Adversarial Discriminative Domain Adaptation (ADDA) which modifies the ReverseGrad approach to remove the gradient reversal layer and use two separate and competing optimization processes to pit the domain classifier against the feature extraction layers, more similar to a traditional generative adversarial network (Tzeng, Hoffman, Saenko, & Darrell, 2017; Goodfellow, et al., 2014). Both ADDA and ReverseGrad have been investigated in manufacturing, and this study presents a novel application of ReverseGrad to realize DATL upon a 1D CNN for vibration signal analysis and tool wear regression under two operating conditions (Zhang, Li, Wen, Gao, & Gao, 2019; da Costa, Akçay, Zhang, & Kaymak, 2020).

3. EXPERIMENTAL EVALUATION

3.1. Milling Data Set and Data Preparation

The Milling Data Set published by NASA Ames Research Center is analyzed in this study (Agogino and Goebel, 2007). The data set was collected across 16 milling process test cases with varying materials, feed rates, and cut depths. As these cuts are performed, the rotating tool head experiences progressive flank wear, which was recorded throughout the runs. Each configuration of material, depth of cut, and feed rate was tested twice. Vibration data were collected from two locations, the spindle and the table. Table vibration data was used for this study.

For transfer learning, the source data domain was chosen to be the cast iron material with a 0.25 mm/s feed rate, and the target domain was cast iron with a 0.5 mm/s feed rate, across all cutting depths. Excluding eight of the 16 data set cases which used steel, this choice resulted in using four of the remaining eight cases as source data (i.e., all cast iron runs at 0.25 mm/s) and the final four cases as target data (i.e., all cast iron runs at 0.5 mm/s).

Each case had individual runs consisting of approximately 9000 data points with a 250-Hz sampling rate of the vibration sensor (i.e., a total of 36 seconds). The beginning and ending portions of the run differ significantly from the rest of the signal as the milling process starts and ends, respectively. Hence, the first and last 10 seconds of the signal were removed. To augment the dataset, these cropped 16-second blocks were further split into 2-second (500-sample) sections and given the same flank wear label as the run from which they were taken.

The source and target data sets contained the 2-second windows from the four source and four target cases,

respectively. Within each source and target data set, the collection of 2-second windows was split into 70% train, 15% validation, and 15% test sets.

3.2. DATL Network Architecture

The developed 1D CNN-based DATL network architecture is shown in Fig. 2. A 1D CNN performs feature extraction using three pairs of 1D convolutional and max-pooling layers with ReLU activation functions. This set of feature extraction layers is shared between the source and target domains to generate consistent, similar features from both input distributions.

The flank wear regression output is generated by three fully-connected layers terminating in a sigmoid output neuron. Dropout is used throughout the feature extraction and output layers to combat the overfitting problem. Regression loss is calculated via mean squared error. Domain classification is performed using three fully-connected layers closely resembling the regression layers. However, the number of hidden layer neurons is reduced, and the output is two softmax neurons representing a one-hot vector encoding of the input domain, either source or target. A cross-entropy loss function is used. In addition, a gradient reversal layer is included to invert the backpropagated error as it exits the domain classifier. Thus, while backpropagation makes the domain classifier itself to minimize its loss, the gradient reversal layer will cause the feature extraction layers to seek to maximize the domain classifier's loss, thereby increasing the likelihood that source and target features will not be differentiable.

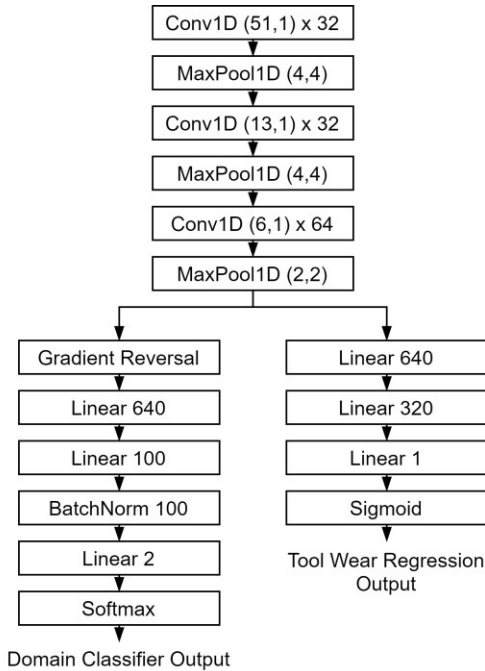


Figure 2. 1D CNN with DA ReverseGrad architecture. ReLU is used as the activation function throughout.

3.3. Training

In each test case, the network was trained for 5000 epochs with early stopping on the source or target validation set performance metric (the coefficient of determination, denoted R^2). The optimizer was Adam with a learning rate of 0.0001. The minibatch size was 20. Experiments were run on an NVIDIA P100 GPU and required between 8 and 15 minutes each to complete.

During transfer learning, the α parameter controls the infusion of domain classifier loss into the optimization of the feature extraction layers. As recommended by the literature, this parameter was defined as

$$\alpha = \frac{2}{1 + \exp(-10p)} - 1 \quad (8)$$

where $p \in [0,1]$ is the training progress (Guo, Lei, Xing, Yan, & Li, 2019). With the α parameter initially close to zero, the optimization process focuses on developing initial features via the labeled source examples. As the parameter increases, the domain classifier loss is gradually incorporated to prevent the features from becoming overly specific to the source domain.

4. RESULTS AND DISCUSSION

Several training cases were performed to evaluate the performance of the developed DATL on the flank wear prediction problem. First, a performance comparison is made between with and without DATL by evaluating the accuracy (quantified by R^2) of the 1D CNN (trained solely on source domain data) on the target domain data. Furthermore, the comparison is expanded to more scenarios where different availability of target domain data can be used for network training, as illustrated in Fig. 3. Without transfer learning, an R^2 of 0.97 was achieved on the source task. It is seen from Fig. 3 that DATL greatly improves the performance of 1D CNN trained on target domain data, when there is no available target domain data to jointly (with source domain data) train the network. Along with the availability of target domain data, the performance without transfer learning gradually catches up with the performance with transfer learning.

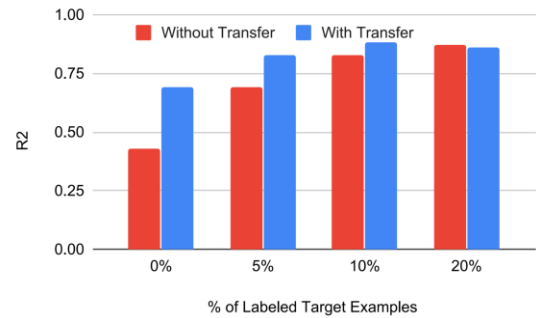


Figure 3. Target domain test set R^2 performance

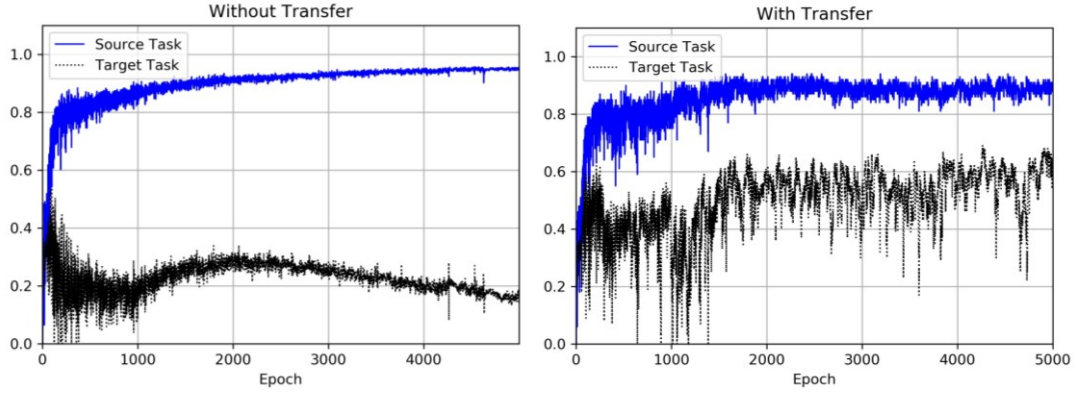


Figure 4. Regression performance (R^2) of 1D CNN on validation data: left) without transfer learning and right) with transfer learning

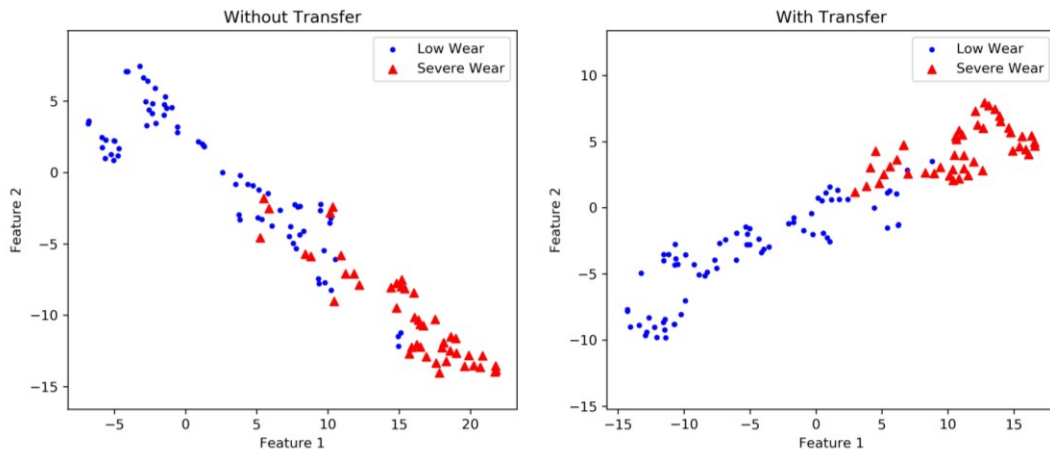


Figure 5. 1D CNN feature distributions vs. wear severity for left) without and b) with transfer learning

The validation R^2 plots for trials without transfer learning demonstrate that as training continues, the target R^2 peaks and begins to decrease (see Fig. 4 left), indicating that the network has begun learning features specific to the source domain instead of those general to the flank wear problem across both operating scenarios. This R^2 divergence in later epochs vanishes when the domain adversarial transfer approach is applied (see Fig. 4b). Without any labeled target domain data, the DATL algorithm immediately increases the target R^2 from 0.43 to 0.69 and achieves a performance of 0.88 when 10% of the labeled target training data is included.

To illustrate how DATL helps with extracting common features from different domains, distributions of features extracted by the 1D CNN on the target domain data (features of the last 1D CNN layer) are shown in Fig. 5 using t-SNE. The distributions are plotted against the severity of the flank wear (mild wear: 0-0.3 mm and severe wear: 0.3-0.8 mm). As shown in Fig. 5, DATL reduces the overlap between feature distributions, makes the boundary between points of two wear severities clearer. Although there are still some mixes among wear points due to either the imperfect

differentiation by the model or the artificially introduced errors in differentiating the wear severities, the results confirm the effectiveness of DATL in generalizing DL models regardless the application domains.

5. CONCLUSION

This study demonstrates the application of domain adversarial transfer learning (DATL) to predict tool flank wear in a milling process. The ReverseGrad model was successfully trained on a source task feed rate and transferred to a second set of process parameters with a different feed rate. DATL enables the network to be simultaneously trained on the source and target tasks, encouraging the feature extraction of the network to generate common features to both input domains. The approach significantly improved the network performance on the target task and demonstrated the ability of DATL to compensate for limited labeled target examples. Future work includes investigating different ways for generating and fusing gradients of domain classification and regression, to further improve the performance.

ACKNOWLEDGEMENT

This paper is based upon work supported by National Science Foundation under Grant No. 2015889.

REFERENCES

- Agogino, A., & Goebel, K. (2007). Milling data set. NASA Ames Prognostics Data Repository, NASA Ames Research Center: Moffett Field, CA; UC Berkeley BEST Lab: Berkeley, CA. Retrieved from <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository>
- Abdeljabber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, 388, 154-170.
- da Costa, P., Akçay, A., Zhang, Y., & Kaymak, U. (2020). Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering and System Safety*, 195.
- Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. *32nd International Conference on Machine Learning, ICML 2015*, 2, 1180-1189.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., . . . Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17, 1-35.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2672-2680.
- Guo, L., Lei, Y., Xing, S., Yan, T., & Li, N. (2019). Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Transactions on Industrial Electronics*, 66(9), 7316-7325.
- Hoffman, J., Tzeng, E., Darrell, T., & Saenko, K. (2017). Simultaneous deep transfer across domains and tasks. In: Csurka, G. (Eds.), *Domain Adaptation in Computer Vision Applications* (pp. 173-187). Springer.
- Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, (1097-1105).
- Lee, J., Jin, C., Bagheri, B., & Chao, J. (2016). Introduction to cyber manufacturing. *Manufacturing Letters*, 8, 11-15.
- Li, L., Ota, K., & Dong, M. (2018). Deep learning for smart industry: Efficient manufacture inspection system with fog computing. *IEEE Transactions on Industrial Informatics*, 14(10), 4665-4673.
- Liu, Y., Hu, X., & Jin, J. (2019). Remaining useful life prediction of cutting tools based on deep adversarial transfer learning. In *Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition* (pp. 434-439). New York: ACM.
- Long, M., Cao, Z., Wang, J., & Jordan, M. (2018). Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems*, 31, 1640-1650.
- Pan, S., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359.
- Pan, S., Kwok, J., & Yang, Q. (2008). Transfer learning via dimensionality reduction. *Proceedings of the National Conference on Artificial Intelligence*, 2, 677-682.
- Pan, S., Tsang, I., Kwok, J., & Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2), 199-210.
- Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(1), 227-244.
- Sun, B., & Saenko, K. (2016). Deep CORAL: Correlation alignment for deep domain adaptation. In: Hua, G., & Jégou, H. (Eds.), *Computer Vision – ECCV 2016 Workshops* (pp. 443-450).
- Sun, C., Ma, M., Zhao, Z., Tian, S., Yan, R., & Chen, X. (2018). Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing. *IEEE Transactions on Industrial Informatics*, 15(4), 2416-2425.
- Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2962-2971). CVPR 2017.
- Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: maximizing for domain invariance. <http://arxiv.org/abs/1412.3474>.
- Wang, J., Ma, Y., Zhang, L., Gao, R., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144-156.
- Wen, L., Gao, L., & Li, X. (2019, 1 1). A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(1), 136-144.
- Zhang, Z., Li, X., Wen, L., Gao, L., & Gao, Y. (2019). Fault diagnosis using unsupervised transfer learning based on adversarial network. *2019 IEEE 15th International Conference on Automation Science and Engineering* (pp. 305-310). IEEE Computer Society.

BIOGRAPHIES

Matthew Russell is a Ph.D. student in Electrical and Computer Engineering at the University of Kentucky, Lexington, KY, USA, and received a B.S. in Computer Engineering from Liberty University, Lynchburg, VA, USA,

in 2018. He is a Graduate Pathways Intern at NASA Johnson Space Center in Houston, TX. His research interests include deep learning and transfer learning for condition monitoring and sustainable manufacturing and machine learning applications for space exploration systems.

Peng Wang, Ph.D., is an Assistant Professor in the Department of Electrical and Computer Engineering and Department of Mechanical Engineering at the University of Kentucky, Lexington, Kentucky, USA. He received his B.S. and M.S. degree in Information Science from Beijing University of Chemical Engineering in 2010 and 2013, and his Ph.D. degree in Mechanical and Aerospace Engineering from Case Western Reserve University in 2017, respectively. His research interests are in the areas of stochastic modeling, machine learning, and data fusion for machine performance prediction under uncertainty, manufacturing process modeling, and product quality assessment.