Deep Transform Learning for Multi-Sensor Fusion

Saurabh Sahu⋆, Kriti Kumar⋆, Angshul Majumdar†⋆, M Girish Chandra⋆
⋆ TCS Research and Innovation, Bangalore, India.
† IIIT Delhi, New Delhi, India.
Email: {sahu.saurabh1, kriti.kumar, m.gchandra}@tcs.com, angshul@iiitd.ac.in

***Abstract*—This paper presents a Deep Transform Learning based framework for multi-sensor fusion. Deep representations are learnt for each of the sensors by stacking one transform after another. Subsequently, a common transform is utilized to fuse the deep representations of all sensors to estimate the output. Restricting to a regression use case, a joint optimization formulation is presented for learning the sensor-specific deep transforms, their coefficients, the common transform, its coefficient and the regression weights together. The requisite solution steps and the derivation of closed form updates for the transforms and associated coefficients are given. The performance of the proposed method is evaluated using two real-life datasets and comparisons with the state-of-the-art dictionary and transform learning techniques for regression are presented. Results show that the deep network has superior performance compared to other methods as it is able to learn the data representation more effectively than the other shallow variants. In addition to the multi-sensor case, estimation results with single sensors alone are also provided to demonstrate the importance of multi-sensor fusion.**

***Index Terms*—Multi-sensor Fusion, Transform Learning, Dictionary Learning, Deep Learning**

I. Introduction

Multi-sensor fusion is a promising technology that combines information from different sensors to obtain more accurate inferences [1]. It has wide range of applications in areas like military, biomedical applications, robotics, image processing and cyber-physical systems [2]. Different fusion architectures exist in literature [3], [4]. The ability to learn a good representation of the sensor data plays an important role in determining the accuracy of these methods. Of late, alternative to hand-crafted feature design based approach, learning the representation directly from the data using representation learning techniques is getting a lot of attention. The data representations learnt by these techniques are used to build classifiers and regressors and are shown to perform better in many scenarios. In signal processing domain, Dictionary Learning (DL) and Transform Learning (TL) are two popular representation learning techniques that provide a compact representation of data in many situations and shown to produce state-of-art in different application domains [5]–[14]. Moreover, between DL and TL, it is shown in [9], [10] that TL based algorithms provide high accuracy with less computational complexity compared to DL counterparts.

Recently, deep version of both DL and TL have been explored to learn more accurate representation of the data. They are shown to perform better as they are able to characterize the data more effectively than the shallow counterparts. Deep DL has been successfully used in hyper-spectral imaging, biometrics and many other areas [15], [16]. Deep TL algorithms also exist in the literature which utilize multiple layers of cascaded transforms connected via non-linear activation function to learn abstract representations of the data [17]. They have been majorly used for unsupervised tasks [18], [19]. As far as fusion tasks are concerned, dictionaries have been used, mainly by the vision community [20]–[22]. To the best of our knowledge, transform learning is not explored for multi-sensor fusion.

Motivated by the advantages of TL and the need for good sensor data representation for improved inference making, this paper presents a Deep Transform Learning framework for multi-sensor Fusion (DTLF). Deep representation for each sensor are learnt using multiple layers of transforms. Subsequently, a common transform is learnt using the deep representations of different sensors to carry out sensor fusion in a supervised setting for regression use case. This proposal consists of a joint optimization formulation for learning the sensor-specific deep transforms, their coefficients, the common transform, its coefficient and the regression weights together. The derivation of the requisite transforms, associated coefficients and their closed form updates are presented. The performance of the method is analyzed using two real-life datasets with Mean Square Error (MSE) and Mean Absolute Error (MAE) as the performance metrics. The results demonstrate the superior performance of DTLF compared to the state-of-the-art dictionary and transform learning techniques.

Towards providing the necessary details of the proposed formulation, the paper is organized as follows. A brief description of TL is provided in Section II along with an extension to deep version. Section III describes the proposed deep transform learning framework for multi-sensor fusion. The results are presented in Section IV and finally, Section V concludes the work.

II. Background on Transform Learning

In this section, a brief background on TL and Deep Transform Learning (DTL) is presented for quick reference.

A. Transform Learning (TL)

Given the data $\boldsymbol{X} \in \mathbb{R}^{m\times L}$ with m features of length L, data-driven transform $\boldsymbol{T} \in \mathbb{R}^{K\times m}$ is learnt using the following basic formulation:

$$\boldsymbol{TX} = \boldsymbol{Z} \tag{1}$$

where $\boldsymbol{Z} \in \mathbb{R}^{K\times L}$ are the coefficients and K is the number of atoms of the transform. In practice, the standard TL

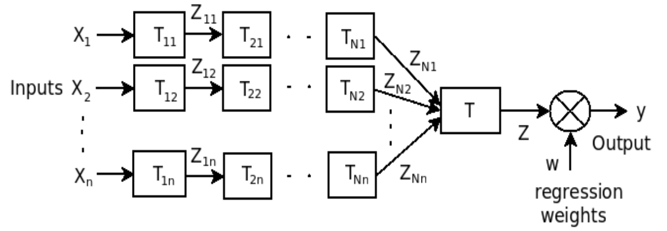

Fig. 1. Block Diagram of Deep Transform Learning Framework for Fusion

formulation is expressed as [9]:

$$\min_{\boldsymbol{T},\boldsymbol{Z}} \|\boldsymbol{TX}-\boldsymbol{Z}\|_F^2 + \lambda(\|\boldsymbol{T}\|_F^2 - \log \det \boldsymbol{T}) + \mu\|\boldsymbol{Z}\|_0 \quad (2)$$

where the additional regularization term prevents the degenerate solution as well as enforces sparsity penalty. The solution to (2) is obtained using alternative minimization framework given in [13] by solving the following sub-problems:

$$\boldsymbol{Z} \leftarrow \min_{\boldsymbol{Z}} \|\boldsymbol{TX}-\boldsymbol{Z}\|_F^2 + \mu\|\boldsymbol{Z}\|_0 \quad (3)$$

$$\boldsymbol{T} \leftarrow \min_{\boldsymbol{T}} \|\boldsymbol{TX}-\boldsymbol{Z}\|_F^2 + \lambda(\|\boldsymbol{T}\|_F^2 - \log \det \boldsymbol{T}). \quad (4)$$

The closed form update for the transform $\boldsymbol{T}$ is obtained using Cholesky decomposition followed by singular value decomposition [13]. The coefficients update is simple; they are updated via one step of soft-thresholding.

B. Deep Transform Learning (DTL)

DTL is the deep version of the basic TL where multiple transform representations are cascaded together to generate the coefficients. For an N-layer network, DTL formulation is expressed as:

$$\boldsymbol{T_N}(\phi...(\boldsymbol{T_2}(\phi(\boldsymbol{T_1X})))) = \boldsymbol{Z} \quad (5)$$

where ϕ denotes the activation function. The greedy solution for (5) is given in [19], where the transforms and coefficients are solved for each layer using standard TL updates (given in [13]) and substituted for the next layer until all the coefficients are estimated. This method is sub-optimal as there is no flow of information from deep to shallow layers, thus, another method is proposed in [18] that solves for the transforms and coefficients of all layers together using a joint optimization framework expressed as:

$$\min_{\boldsymbol{T_i}'s,\boldsymbol{Z}} \|\boldsymbol{T_N}(\phi..(\boldsymbol{T_2}(\phi(\boldsymbol{T_1X}))))-\boldsymbol{Z}\|_F^2 + \lambda\sum_{i=1}^{N}(\|\boldsymbol{T_i}\|_F^2 - \log\det\boldsymbol{T_i}). \quad (6)$$

This method uses variable splitting and ADMM technique for obtaining the requisite updates for the coefficients. The transforms follow the standard updates mentioned in [13].

III. Deep Transform Learning for Fusion (DTLF)

The proposed method makes use of deep transform learning for carrying out multi-sensor fusion. Let $\boldsymbol{X_1}$, $\boldsymbol{X_2}$,..., $\boldsymbol{X_n}$ be the different sensors connected to a system. For each of the sensors, N-layer deep transforms denoted by $\boldsymbol{T_{i1}}, \boldsymbol{T_{i2}},...., \boldsymbol{T_{in}}$ are learnt for $i = 1,..,N$. The deep representations of all sensors are fused together by learning a common transform $\boldsymbol{T}$ in a supervised way utilizing the information of the output $\boldsymbol{y}$. Fig. 1 presents the block diagram of DTLF framework. This method has a training phase where, the sensor-specific deep transforms and the common transform are learnt to predict the output $\boldsymbol{y}$ in the test phase.

Without loss of generality, for $N = 3$ layer deep network and $n = 3$ sensors, the deep representation for the first sensor $\boldsymbol{X_1}$ is given as:

$$\boldsymbol{T_{31}}(\boldsymbol{T_{21}}(\boldsymbol{T_{11}X_1})) = \boldsymbol{Z_{31}}. \quad (7)$$

Similarly, for 3^{rd} sensor it can be expressed as:

$$\boldsymbol{T_{33}}(\boldsymbol{T_{23}}(\boldsymbol{T_{13}X_3})) = \boldsymbol{Z_{33}}. \quad (8)$$

Here, the activation function ϕ in (5) is replaced by a ReLU type non-linearity that is introduced between the layers by putting all the negative values of the coefficients to zero.

DTLF incorporates a ridge regression penalty in the TL formulation for fusing information from different sensors towards a common inference (or output) $\boldsymbol{y}$. A joint optimization is carried out for learning the sensor-specific transforms, their associated coefficients, the common transform, its coefficient and the regression weights together. It is expressed as:

$$\min_{\substack{\boldsymbol{T_{i1}}'s,\boldsymbol{T_{i2}}'s,\boldsymbol{T_{i3}}'s,\boldsymbol{T},\\ \boldsymbol{Z_{i1}},\boldsymbol{Z_{i1}},\boldsymbol{Z_{i3}},\boldsymbol{Z},\boldsymbol{w}}} \|\boldsymbol{T_{31}}(\boldsymbol{T_{21}}(\boldsymbol{T_{11}X_1}))-\boldsymbol{Z_{31}}\|_F^2 +$$
$$\|\boldsymbol{T_{32}}(\boldsymbol{T_{22}}(\boldsymbol{T_{12}X_2}))-\boldsymbol{Z_{32}}\|_F^2 + \|\boldsymbol{T_{33}}(\boldsymbol{T_{23}}(\boldsymbol{T_{13}X_3}))-\boldsymbol{Z_{33}}\|_F^2$$
$$+\lambda\sum_{i=1}^{3}(\|\boldsymbol{T_{i1}}\|_F^2 - \log\det\boldsymbol{T_{i1}}) + \lambda\sum_{i=1}^{3}(\|\boldsymbol{T_{i2}}\|_F^2 - \log\det\boldsymbol{T_{i2}})$$
$$+\lambda\sum_{i=1}^{3}(\|\boldsymbol{T_{i3}}\|_F^2 - \log\det\boldsymbol{T_{i3}}) + \gamma\|\boldsymbol{T}\begin{bmatrix}\boldsymbol{Z_{31}}\\ \boldsymbol{Z_{32}}\\ \boldsymbol{Z_{33}}\end{bmatrix} - \boldsymbol{Z}\|_F^2$$
$$+\lambda(\|\boldsymbol{T}\|_F^2 - \log \det \boldsymbol{T}) + \alpha\|\boldsymbol{y}-\boldsymbol{wZ}\|_2^2$$

$$\text{s.t. } \boldsymbol{T_{31}}(\boldsymbol{T_{21}}(\boldsymbol{T_{11}X_1})) \geq 0, \boldsymbol{T_{21}}(\boldsymbol{T_{11}X_1}) \geq 0, \boldsymbol{T_{11}X_1} \geq 0,$$
$$\boldsymbol{T_{32}}(\boldsymbol{T_{22}}(\boldsymbol{T_{12}X_2})) \geq 0, \boldsymbol{T_{22}}(\boldsymbol{T_{12}X_2}) \geq 0, \boldsymbol{T_{12}X_2} \geq 0,$$
$$\boldsymbol{T_{33}}(\boldsymbol{T_{23}}(\boldsymbol{T_{13}X_3})) \geq 0, \boldsymbol{T_{23}}(\boldsymbol{T_{13}X_3}) \geq 0, \boldsymbol{T_{13}X_3} \geq 0$$

where, the multiple sensors $\boldsymbol{X_1} \in \mathbb{R}^{m_1\times L}, \boldsymbol{X_2} \in \mathbb{R}^{m_2\times L}, \boldsymbol{X_3} \in \mathbb{R}^{m_3\times L}$ are of length L each with feature length of m_1, m_2, m_3 and $\boldsymbol{y}$ is the output. The transforms $\boldsymbol{T_{11}} \in \mathbb{R}^{K_1\times m_1}, \boldsymbol{T_{21}}, \boldsymbol{T_{31}} \in \mathbb{R}^{K_1\times K_1}, \boldsymbol{T_{12}} \in \mathbb{R}^{K_2\times m_2}, \boldsymbol{T_{22}}, \boldsymbol{T_{32}} \in \mathbb{R}^{K_2\times K_2}, \boldsymbol{T_{13}} \in \mathbb{R}^{K_3\times m_3}, \boldsymbol{T_{23}}, \boldsymbol{T_{33}} \in \mathbb{R}^{K_3\times K_3}, \boldsymbol{T} \in \mathbb{R}^{K\times(K_1+K_2+K_3)}$ and their associated coefficients $\boldsymbol{Z_{11}}, \boldsymbol{Z_{21}}, \boldsymbol{Z_{31}} \in \mathbb{R}^{K_1\times L}, \boldsymbol{Z_{12}}, \boldsymbol{Z_{22}}, \boldsymbol{Z_{32}} \in \mathbb{R}^{K_2\times L}, \boldsymbol{Z_{13}}, \boldsymbol{Z_{23}}, \boldsymbol{Z_{33}} \in \mathbb{R}^{K_3\times L}, \boldsymbol{Z} \in \mathbb{R}^{K\times L}$ with K, K_1, K_2, K_3 being the size of the respective transforms. The regression weights $\boldsymbol{w} \in \mathbb{R}^{1\times K}$ for the output $\boldsymbol{y} \in \mathbb{R}^{1\times L}$.

In the ***training phase***, multi-layer transforms and their associated coefficients are learnt for each of the sensors. Subsequently, a common transform is learnt utilizing the representations of the different sensors to arrive at a common inference. The sub-problems to solve for the update of transforms and coefficients corresponding to the deep network of $\boldsymbol{X_1}$ are given as:

$$\boldsymbol{T_{11}} \leftarrow \min_{\boldsymbol{T_{11}}} \|\boldsymbol{T_{11}X_1}-\boldsymbol{Z_{11}}\|_F^2 + \lambda(\|\boldsymbol{T_{11}}\|_F^2 - \log \det \boldsymbol{T_{11}}) \quad (9)$$

$$\boldsymbol{T_{21}} \leftarrow \min_{\boldsymbol{T_{21}}} \|\boldsymbol{T_{21}Z_{11}}-\boldsymbol{Z_{21}}\|_F^2 + \lambda(\|\boldsymbol{T_{21}}\|_F^2 - \log \det \boldsymbol{T_{21}}) \quad (10)$$

$$\boldsymbol{T_{31}} \leftarrow \min_{\boldsymbol{T_{31}}} \|\boldsymbol{T_{31}Z_{21}}-\boldsymbol{Z_{31}}\|_F^2 + \lambda(\|\boldsymbol{T_{31}}\|_F^2 - \log \det \boldsymbol{T_{31}}) \quad (11)$$

$$\boldsymbol{Z_{11}} \leftarrow \min_{\boldsymbol{Z_{11}}} \|\boldsymbol{T_{11}X_1}-\boldsymbol{Z_{11}}\|_F^2 + \|\boldsymbol{T_{21}Z_{11}}-\boldsymbol{Z_{21}}\|_F^2 \quad (12)$$
$$\text{subject to } \boldsymbol{Z_{11}} \geq 0.$$

$$\boldsymbol{Z_{21}} \leftarrow \min_{\boldsymbol{Z_{21}}} \|\boldsymbol{T_{21}Z_{11}}-\boldsymbol{Z_{21}}\|_F^2 + \|\boldsymbol{T_{31}Z_{21}}-\boldsymbol{Z_{31}}\|_F^2 \quad (13)$$

subject to $\boldsymbol{Z_{21}} \geq 0$.

Splitting the common transform into $\boldsymbol{T} = [\boldsymbol{T_1}|\boldsymbol{T_2}|\boldsymbol{T_3}]$, the problem for $\boldsymbol{Z_{31}}$ can be expressed as:

$$\boldsymbol{Z_{31}} \leftarrow \min_{\boldsymbol{Z_{31}}} \|\boldsymbol{T_{31}}\boldsymbol{Z_{21}} - \boldsymbol{Z_{31}}\|_F^2 + \gamma(\|\boldsymbol{T_1}\boldsymbol{Z_{31}} + \boldsymbol{T_2}\boldsymbol{Z_{32}} + \boldsymbol{T_3}\boldsymbol{Z_{33}} - \boldsymbol{Z}\|_F^2). \tag{14}$$

subject to $\boldsymbol{Z_{31}} \geq 0$.

In the similar way, the sub-problems for the transforms and their coefficients for the remaining two sensors can be listed. Let $\boldsymbol{Z'} = \begin{bmatrix}\boldsymbol{Z_{31}}\\ \boldsymbol{Z_{32}}\\ \boldsymbol{Z_{33}}\end{bmatrix}$, then the common transform, its coefficients and the regression weights are updated using the following:

$$\boldsymbol{Z} \leftarrow \min_{\boldsymbol{Z}} \gamma\|\boldsymbol{T}\boldsymbol{Z'} - \boldsymbol{Z}\|_F^2 + \alpha\|\boldsymbol{y} - \boldsymbol{w}\boldsymbol{Z}\|_2^2 \tag{15}$$

$$\boldsymbol{T} \leftarrow \min_{\boldsymbol{T}} \gamma\|\boldsymbol{T}\boldsymbol{Z'} - \boldsymbol{Z}\|_F^2 + \lambda(\|\boldsymbol{T}\|_F^2 - \log\det \boldsymbol{T}) \tag{16}$$

$$\boldsymbol{w} \leftarrow \min_{\boldsymbol{w}} \alpha\|\boldsymbol{y} - \boldsymbol{w}\boldsymbol{Z}\|_2^2. \tag{17}$$

The exact update for the coefficients can be obtained using Forward-Backward method or FISTA [23]. However, these are iterative methods and hence time consuming. Using ADMM technique, a closed form solution of (12), (13) and (14) can be obtained in the form of pseudo-inverse as given below. Here, derivative of the sub-problem is taken with respect to the coefficient to be solved for and equated to zero to obtain the update. An approximate solution for (12), (13) and (14) is obtained by putting all negative values of the coefficients to zero. Following this procedure, the closed form update for $\boldsymbol{Z_{11}}$ is given as:

$$\boldsymbol{Z_{11}} = (\boldsymbol{T_{21}}^T\boldsymbol{T_{21}} + \boldsymbol{I})^{-1}.(\boldsymbol{T_{11}}\boldsymbol{X_1} + (\boldsymbol{T_{21}}^T\boldsymbol{Z_{21}})). \tag{18}$$

In the same way, the updates for the coefficients $\boldsymbol{Z_{21}}$ and $\boldsymbol{Z_{31}}$ are obtained as:

$$\boldsymbol{Z_{21}} = (\boldsymbol{T_{31}}^T\boldsymbol{T_{31}} + \boldsymbol{I})^{-1}.(\boldsymbol{T_{21}}\boldsymbol{Z_{11}} + (\boldsymbol{T_{31}}^T\boldsymbol{Z_{31}})) \tag{19}$$

$$\boldsymbol{Z_{31}} = (\gamma\boldsymbol{T_1}^T\boldsymbol{T_1} + \boldsymbol{I})^{-1}.(\boldsymbol{T_{31}}\boldsymbol{Z_{21}} + \gamma(\boldsymbol{T_1}^T\boldsymbol{Z} - \boldsymbol{T_1}^T\boldsymbol{T_2}\boldsymbol{Z_{32}} - \boldsymbol{T_1}^T\boldsymbol{T_3}\boldsymbol{Z_{33}})). \tag{20}$$

This process is followed for all the sensors. Using (15), the update for the common coefficients $\boldsymbol{Z}$ is obtained as:

$$\boldsymbol{Z} = (\boldsymbol{I} + \alpha\boldsymbol{w}^T\boldsymbol{w})^{-1}.(\boldsymbol{T}\boldsymbol{Z'} + \alpha\boldsymbol{w}^T\boldsymbol{y}). \tag{21}$$

The transforms update for the sensor-specific deep layers and the common transform follow the same standard update as given for (4) in [13] with the input $\boldsymbol{X}$ being replaced by the respective sensor input, their cascaded version and $\boldsymbol{Z'}$ respectively. The closed form update for $\boldsymbol{w}$ is computed as $\boldsymbol{w} = \boldsymbol{y}\boldsymbol{Z}^\dagger$, where '†' denotes the pseudo-inverse. The transforms and coefficients update goes through a lot of iterations till the convergence criteria ($\|\boldsymbol{T_j} - \boldsymbol{T_{j-1}}\| \leq \delta$, where δ is the threshold and j is the iteration number) is satisfied. This completes the training phase.

In the ***test phase***, given new test samples $\boldsymbol{X_1^{test}} \in \mathbb{R}^{m_1 \times t}, \boldsymbol{X_2^{test}} \in \mathbb{R}^{m_2 \times t}, \boldsymbol{X_3^{test}} \in \mathbb{R}^{m_3 \times t}$, the output $\hat{\boldsymbol{y}}_{\boldsymbol{test}}$ is computed using the model learnt in terms of the $\boldsymbol{T_{11}}$, $\boldsymbol{T_{21}}$, $\boldsymbol{T_{31}}$, $\boldsymbol{T_{12}}$, $\boldsymbol{T_{22}}$, $\boldsymbol{T_{32}}$, $\boldsymbol{T_{13}}$, $\boldsymbol{T_{23}}$, $\boldsymbol{T_{33}}$, $\boldsymbol{T}$ and $\boldsymbol{w}$ in the training phase. The transform coefficients are first computed for each sensor using the test data. The coefficients of the first sensor (utilizing the aforementioned ReLU type non-linearity) are given as:

$$\boldsymbol{Z_{31}^{test}} = \boldsymbol{T_{31}}(\boldsymbol{T_{21}}(\boldsymbol{T_{11}}\boldsymbol{X_1^{test}})). \tag{22}$$

Similarly, the coefficients for the remaining sensors are computed to obtain $\boldsymbol{Z^{test}}$.

$$\boldsymbol{Z^{test}} = \boldsymbol{T}\begin{bmatrix}\boldsymbol{Z_{31}^{test}}\\ \boldsymbol{Z_{32}^{test}}\\ \boldsymbol{Z_{33}^{test}}\end{bmatrix}. \tag{23}$$

Using $\boldsymbol{w}$, the final output is estimated as:

$$\hat{\boldsymbol{y}}_{\boldsymbol{test}} = \boldsymbol{w}\boldsymbol{Z^{test}}. \tag{24}$$

IV. Experimental Results and Discussion

The potential of the proposed DTLF technique for multi-sensor fusion is demonstrated using two real-life data sets. The first one is the combined cycle power plant (CCPP) data from the publicly available UCI dataset [24], [25]. The second data is the Building Energy Consumption (BEC) data collected from office buildings. Results of the proposed method are compared against the state-of-the-art transform and dictionary based algorithms used for regression i.e. Transform Learning for Regression (TLR), Kernel Transform Learning for Regression (KTLR) [11], Dictionary Learning for Regression (DLR), Kernel Dictionary Learning for Regression (KDLR) [12] and the shallow (or 1-layer deep) version of Transform Learning for Fusion (TLF). In addition to multi-sensor case, the results with individual sensors are also presented to demonstrate the effectiveness of fusion. KTLR method is employed for the single sensor case since it performs better than other methods. The performance of the different techniques are evaluated using Mean Square Error (MSE) and Mean Absolute Error (MAE) metrics.

Unlike the fusion formulation (TLF & DTLF) where different transforms are learnt for each sensor, the regression based techniques (TLR, KTLR, DLR, KDLR) learn a single transform or dictionary by stacking all the sensor data together. Here, all the kernel methods make use of 'rbf' kernel for fair comparison. The size of the dictionaries and transforms and the values of the hyper-parameters used for solving the optimization problems are tuned appropriately for all techniques. Accordingly, the best configuration of each technique is utilized for learning the data representation. For both the datasets, 5-fold cross-validation is carried out to generate the results. It is observed that the results obtained using all sensors are better than that obtained with single sensors; with DTLF significantly outperforming other methods. More details on the datasets and results are given below.

A. Combined Cycle Power Plant

This dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the power plant was set to work with full load. Features consisting of hourly average ambient variables Temperature (T), Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (V) are considered as different sensors and used to predict the net hourly electrical energy output of the plant.

TABLE I
RESULTS WITH COMBINED CYCLE POWER PLANT

Algorithm	MSE	MAE
DLR	0.0227 ± 0.0006	0.1194 ± 0.0019
TLR	0.0203 ± 0.0009	0.1130 ± 0.0029
TLF	0.0196 ± 0.0011	0.1104 ± 0.0029
KDLR	0.0193 ± 0.0016	0.1057 ± 0.0041
KTLR	0.0184 ± 0.0013	0.1019 ± 0.0043
DTLF	**0.0075 ± 0.0006**	**0.0682 ± 0.0031**
Ambient Temperature	0.0259 ± 0.0035	0.1264 ± 0.0079
Exhaust Vacuum	0.0300 ± 0.0011	0.1378 ± 0.0030
Ambient Pressure	0.0440 ± 0.0009	0.1730 ± 0.0017
Relative Humidity	0.0488 ± 0.0018	0.1829 ± 0.0041

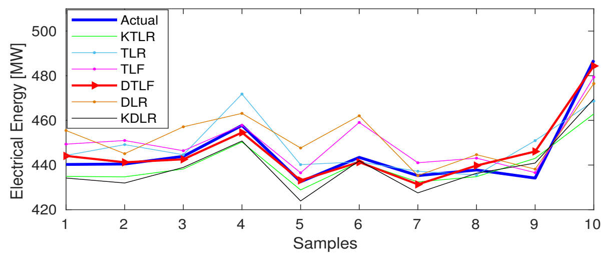

Fig. 2. CCPP Electrical Energy Prediction

The data is normalized and split into windows of 10 samples each, serving as the feature length for all the sensors. In this case, the electrical energy output is a matrix $\boldsymbol{y} \in \mathbb{R}^{10\times L}$ where $L = 956$. Thus, the regression weights are $\boldsymbol{w} \in \mathbb{R}^{10\times K}$ with K being the number of atoms of the transform $\boldsymbol{T}$. Here, a 3-layer DTLF with same atom size for each layer is employed to estimate the electrical energy of the power plant. Table I presents the MSE and MAE values of the energy estimated using different algorithms. Energy estimated using single sensors alone are also presented for reference. It can be clearly seen that the techniques utilizing data from all sensors perform better than the single sensor case. Also, among the kernel and basic variants of the techniques, the former has better estimation accuracy since the kernels methods are able to model the non-linearities in the data well. It can be observed that the proposed DTLF technique outperforms all other learning algorithms as the deep layers are able to learn the data representations in a more effective manner. Fig 2. presents the energy estimation obtained with the proposed DTLF along with other methods. The plot of energy estimation using single sensors versus all sensors employing the DTLF framework is presented in Fig. 3.

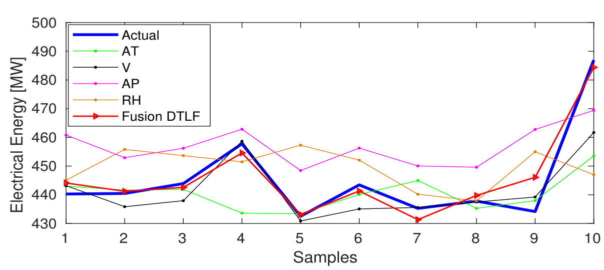

Fig. 3. CCPP Energy Prediction: Single sensor Vs Multi-sensor Fusion

B. Building Energy Consumption Data

This data contains aggregate power consumption measurements collected from office building sampled every 15 minutes over a span of 6.5 months. This data is considered for building load forecasting. Forecasting is framed as a regression problem that uses previous day power (PDay Power) consumption, previous week same day power (PWDay Power) consumption measurements and day ahead forecast temperature to produce a half day ahead load forecast of the building. These input parameters are considered as different sensors and half day data corresponding to 48 samples is taken as feature length for all the sensors.

TABLE II
RESULTS WITH BUILDING ENERGY CONSUMPTION DATA

Algorithm	MSE	MAE
KDLR	0.0251 ± 0.0100	0.0959 ± 0.0199
KTLR	0.0252 ± 0.0100	0.0944 ± 0.0191
DTLF	**0.0124 ± 0.0031**	**0.0765 ± 0.0090**
DLR	0.0350 ± 0.0060	0.1318 ± 0.0109
TLR	0.0382 ± 0.0096	0.1217 ± 0.0064
TLF	0.0254 ± 0.0079	0.1067 ± 0.0159
Temperature	0.0201 ± 0.0038	0.0991± 0.0084
PDay Power	0.0220 ± 0.0039	0.1014 ± 0.0036
PWDay Power	0.0171 ± 0.0035	0.0821 ± 0.0083

TABLE III
DTLF RESULTS WITH BUILDING ENERGY CONSUMPTION DATA

No. of layers (N)	MSE	MAE
TLF (1-layer)	0.0254 ± 0.0079	0.1067 ± 0.0159
2-layer	0.0244 ± 0.0071	0.1140 ± 0.0132
3-layer	0.0138 ± 0.0030	0.0811 ± 0.0093
4-layer	**0.0124 ± 0.0031**	**0.0765 ± 0.0090**
5-layer	0.0127 ± 0.0033	0.0793 ± 0.0088

The data is normalized and 6 months data is considered for training where the transforms and regression weights are learnt. Here again, the output is a matrix $\boldsymbol{y} \in \mathbb{R}^{48\times L}$ where $L = 6\times 30\times 2 = 360$ (total half day instances). The regression weights in this case are $\boldsymbol{w} \in \mathbb{R}^{48\times K}$. A 4-layer deep transform learning framework is employed for load forecasting. Here again, same atom size is considered for each layer of the deep network. Table II presents the day ahead load forecast results in terms of MSE and MAE obtained with different methods for both multi-sensor and single sensor case. Similar to the previous case, DTLF has the best estimation accuracy with the least MSE and MAE. Fig. 4 presents the forecast estimates obtained with the proposed DTLF along with the kernel variants of other methods and TLF. It can be seen that DTLF technique captures the variations in the power consumption much better than other methods. Fig. 5 compares the half day ahead forecast results obtained using single sensors along with all the sensors employing the DTLF technique.

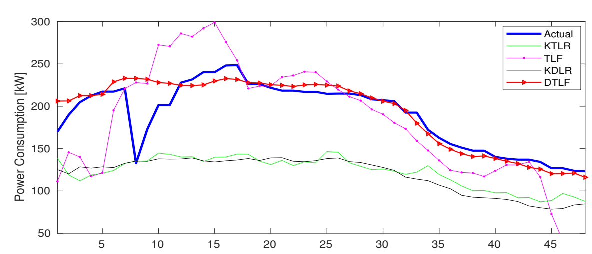

Fig. 4. Building Power Consumption Forecast

It can be observed for both the datasets, the kernel versions of the techniques perform better than the basic or non-kernelized versions as expected. Also, one can observe, the

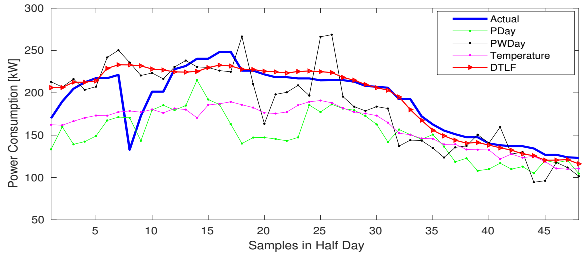

Fig. 5. Building Power Consumption Forecast: Single sensor Vs Multi-sensor Fusion

TLF method has better performance compared to the other basic versions. This is because the dedicated transforms learnt for each sensor capture the data representation more effectively as compared to a single transform learnt using all sensor data together. As we go deep, as in the case of DTLF, the performance further improves and goes beyond the kernel variants as the deep layers are able to learn the data representation much better than shallow versions. However, like any other deep learning based technique, the best accuracy is obtained for a particular number of layers beyond which either the accuracy is saturated or reduces due to insufficient training data. Table III presents accuracy metrics obtained with different number of layers for the BEC data. It can be seen that the accuracy is best with 4-layers and beyond that it reduces due to data insufficiency.

Results obtained using both the datasets show the superior performance of DTLF compared to other algorithms both in terms of MSE and MAE, thus demonstrating the potential of DTLF framework for multi-sensor fusion task. Although detailed formulation with the closed form updates are given for 3-layer deep network with 3 sensors, this technique is generic and can be applied for N-layer deep network with n sensors in a similar way.

V. Conclusion

This paper presents a deep transform learning framework for carrying out multi-sensor fusion. The joint optimization formulation for fusing the information from deep representations of the different sensors is presented along with the requisite closed form updates of the transforms and their coefficients. Experimental results obtained with the different datasets demonstrate the importance of fusion and the applicability of the proposed DTLF technique for multi-sensor fusion task. Since the deep representations learnt for each sensor are able to characterize the sensor data more effectively, the proposed DTLF technique provides superior performance compared to other dictionary and transform variants.

References

[1] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, Jan 1997.

[2] Dong Jiang, Dafang Zhuang, Yaohuan Huang, and Jingying Fu, "Advances in multi-sensor data fusion: Algorithms and applications," *Sensors (Basel, Switzerland)*, vol. 9, pp. 7771–84, 09 2009.

[3] Danilo P. Mandic, Dragan Obradovic, Anthony Kuh, Tülay Adali, Udo Trutschell, Martin Golz, Philippe De Wilde, Javier Barria, Anthony Constantinides, and Jonathon Chambers, "Data fusion for modern engineering applications: An overview," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*. 2005, pp. 715–721, Springer Berlin Heidelberg.

[4] G. Niu, *Data-Driven Technology for Engineering Systems Health Management: Design Approach, Feature Construction, Fault Diagnosis, Prognosis, Fusion and Decisions*, Springer Singapore, 2016.

[5] Bruno A. Olshausen and David J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Research*, vol. 37, no. 23, pp. 3311 – 3325, 1997.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[7] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R. Bach, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems 21*, pp. 1033–1040. Curran Associates, Inc., 2009.

[8] Guangliang Chen and Deanna Needell, "Compressed sensing and dictionary learning," in *Finite Frame Theory, Proceedings of Symposia in Applied Mathematics*, vol. 73, pp. 201–241. Amer. Math. Soc., Providence, RI, 2016.

[9] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, March 2013.

[10] S. Ravishankar and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3088–3092.

[11] K. Kumar, A. Majumdar, M. G. Chandra, and A. Anil Kumar, "Transform learning based functionapproximation for regression and forecasting," in *2019 4th ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, September 2019.

[12] K. Kumar, A. Majumdar, M. G. Chandra, and A. Anil Kumar, "Regressing kernel dictionary learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018.

[13] J. Maggu and A. Majumdar, "Kernel transform learning," *Pattern Recognition Letters*, vol. 98, pp. 117 – 122, 2017.

[14] J. Maggu and A. Majumdar, "Unsupervised deep transform learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 6782–6786.

[15] Vanika Singhal, Hemant K Aggarwal, Snigdha Tariyal, and Angshul Majumdar, "Discriminative robust deep dictionary learning for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5274–5283, 2017.

[16] Ishan Manjani, Snigdha Tariyal, Mayank Vatsa, Richa Singh, and Angshul Majumdar, "Detecting silicone mask-based presentation attack via deep dictionary learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 7, pp. 1713–1723, 2017.

[17] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng, "Multimodal intelligence: Representation learning, information fusion, and applications," 2019.

[18] Jyoti Maggu and Angshul Majumdar, "Unsupervised deep transform learning," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 6782–6786.

[19] Jyoti Maggu and Angshul Majumdar, "Greedy deep transform learning," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1822–1826.

[20] Mansour Nejati, Shadrokh Samavi, and Shahram Shirani, "Multi-focus image fusion using dictionary-based sparse representation," *Information Fusion*, vol. 25, pp. 72 – 84, 2015.

[21] Hui Li and Xiao-Jun Wu, "Multi-focus image fusion using dictionary learning and low-rank representation," *Image and Graphics*, pp. 675–686, 2017.

[22] S. Li, H. Yin, and L. Fang, "Group-sparse representation with dictionary learning for medical image denoising and fusion," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 12, pp. 3450–3459, Dec 2012.

[23] Antonin Chambolle and Charles Dossal, "On the convergence of the iterates of "fista"," 2015.

[24] Pınar Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 126–140, 2014.

[25] Heysem Kaya, Pmar Tüfekci, and Fikret S Gürgen, "Local and global learning methods for predicting power of a combined gas & steam turbine," in *Proceedings of the international conference on emerging trends in computer and electronics engineering icetcee*, 2012, pp. 13–18.