Multi-Sensor Fusion Framework using Discriminative Autoencoders

Arup Kumar Das*, Kriti Kumar*, Angshul Majumdar†*, Saurabh Sahu*, M Girish Chandra*
*TCS Research, India.
†IIIT Delhi, New Delhi, India.
Email: {arup.kdas, kriti.kumar, sahu.saurabh1, m.gchandra}@tcs.com, angshul@iiitd.ac.in

***Abstract*—The paper presents a framework for multi-sensor fusion using discriminative autoencoders. It employs a two stage network where in the first stage, dedicated autoencoders are learnt for each sensor to obtain sensor-specific representations. The corresponding latent representations from all sensors are combined to learn a fusing autoencoder in the second stage. The latent representation of this stage is used to learn a label consistent classifier for multi-class classification. A joint optimization technique is presented for learning the autoencoders and classifier weights together. This framework is tested for two real life scenarios from different domains and comparisons with the state-of-the-art techniques is presented. Additionally, the robustness of the fusion framework is demonstrated in noisy environments. The joint optimization allows discriminative features to be learnt from the different sensors, and hence it displays superior performance than the state-of-the-art methods with reduced complexity.**

***Index Terms*—Sensor fusion, Joint optimization, Supervised learning, Autoencoder, Feature extraction, Classification**

I. Introduction

Multi-sensor fusion aims to combine several data sources or sensors so as to portray a unified picture with improved information [1], [2]. Data fusion exploits the complementary, competitive or cooperative information available from individual sensors to produce a synergistic effect [3]. As a result, it has been extensively used to interpret data from unimodal and multimodal sensors in diverse applications like remote sensing [4], machine health monitoring [5], multi-focus image fusion [6], gesture control and recognition [7], etc. An array of techniques such as geometrical [8] and graphical [9] approaches, multi-modal analysis [10], estimation, classification and inference methods [11] are employed for different data fusion applications.

Lately, the increasing complexity of the sensing environments demand multiple sophisticated sensors to be deployed, with large volume of data being acquired for efficient inference making. Processing of such complex and voluminous data often exceeds human capability [12]. As a result, representation learning with deep learning architectures has profoundly influenced the field of multi-sensor fusion [13]–[16]. Representation learning techniques extract unique signal dependent abstract patterns, and learns useful representations for predictive and prescriptive analytics [17]. Among the different representation learning techniques, autoencoders that are self supervised neural network architectures have been widely used for dimensionality reduction [18], feature learning [19], denoising [20] and image segmentation [21], in addition to multi-sensor fusion.

The authors in [22] use a set of pre-trained autoencoders for multi-channel feature extraction, followed by a multi layer perceptron for spoofing detection in face recognition systems. Pulgar *et al.* propose the AEkNN model for classification tasks [23]. An autoencoder is trained for suitable latent representations of data in the training phase. During the testing or classification phase, kNN algorithm is employed to detect the correct classes. The work in [24] uses SAE-DBN architecture for bearing fault classification. Statistical feature vectors, generated from individual vibration sensors, are fed to sparse autoencoders seperately to generate latent representations. Fault indicators, obtained by fusing the SAE representations, are fed to DBN for classification tasks. It is to be noted that in these works, feature learning and classification are done separately. This does not ensure efficient features to be learnt for the specific application at hand. Hence, in this paper, a joint learning of features and classifier is proposed for multi-sensor fusion scenario. This results in coherent signal representations to be learnt, and helps in robust inference making. The proposed method makes use of discriminative autoencoders for learning representations from different sensors and fusing them together, and thus is referred as F-DiAE (Fusion using Discriminative Autoencoders) in this paper.

F-DiAE uses dedicated autoencoders for each sensor to learn sensor-specific representations. The latent representations from different sensor signals are combined, and used to learn the fusing autoencoder. Subsequently, the representation learnt from the fusing autoencoder is considered as the feature space for multi-class classification. Similar to the discriminative autoencoder work in [25], a discriminative penalty based on the available class information is applied for learning the autoencoders. The work in [25] focused on RGB image classification where feature learning and classification were done separately. However, this work presents a joint learning framework for multi-sensor fusion where sensor representation learning and classification are considered together. A similar work based on joint learning but utilizing transform representation for raw data level fusion of multiple sensors for regression task is presented in [26]. In contrast to [26], the proposed work employs an autoencoder based framework for fusing information from multiple sensors for multi-class classification. To learn the sensor representations, this framework utilizes class label information that helps in learning high level abstract features that are more discriminative compared to the features learnt using the standard autoencoder formulation.

The major contributions of this work are summarized as

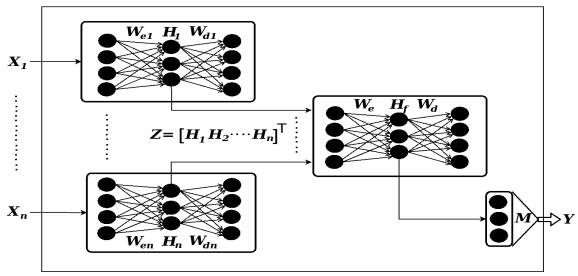


Fig. 1: Block diagram of the proposed F-DiAE framework.

follows:

- A novel discriminative autoencoder based multi-sensor fusion framework (F-DiAE) for classification tasks.
- Joint learning of the autoencoders and classifier weights using an optimization formulation that enables discriminative features to be learnt for individual sensor signals, and aids in classification.
- Improvement in classification accuracy with reduced complexity over state-of-the-art methods.

The rest of the paper is organized as follows. Section II gives an overview on autoencoders followed by a detailed description of the proposed F-DiAE framework. Section III presents the experimental results obtained with different datasets from different domains. Finally, Section IV concludes the work with directions for future work.

II. Multi-Sensor Fusion using Discriminative Autoencoders (F-DiAE)

A. Brief Background on Autoencoders

Autoencoder is a self supervised neural network consisting of an encoder $\boldsymbol{W_e}$ that translates the input data matrix $\boldsymbol{X}$ to a representation $\boldsymbol{H}$ in the latent space, and a decoder $\boldsymbol{W_d}$ to reconstruct the input as $\boldsymbol{X}$. Mathematically, it can be expressed as: $\boldsymbol{H} = \boldsymbol{\phi}(\boldsymbol{W_e X})$ and $\boldsymbol{X} = \boldsymbol{W_d H}$, where $\phi(\cdot)$ is the non-linear activation function like *ReLU*, sigmoid etc. The dimension of the latent representations is mostly under-determined (less than the input dimension), but can be over-determined (more than the input dimension) or the same dimension as the input, depending on the use case considered. Traditionally, the encoder and decoder weights of the autoencoder are learnt using back propagation by minimizing the reconstruction error that is expressed as:

$$\min_{\boldsymbol{W_e}, \boldsymbol{W_d}} \|\boldsymbol{X} - \boldsymbol{W_d}\boldsymbol{\phi}(\boldsymbol{W_e X})\|_F^2. \tag{1}$$

The reconstruction error need not be restricted to Euclidean cost and can accommodate other forms as well. Substituting the latent representation $\boldsymbol{H} = \boldsymbol{\phi}(\boldsymbol{W_e X})$, following from [25], [27], the learning of the autoencoder in an augmented Lagrangian form can be re-written as:

$$\min_{\boldsymbol{W_e}, \boldsymbol{W_d}, \boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W_d H}\|_F^2 + \mu\|\boldsymbol{H} - \boldsymbol{\phi}(\boldsymbol{W_e X})\|_F^2. \tag{2}$$

This formulation is utilized for learning the autoencoders employed in the proposed fusion framework, described in the subsequent section.

B. Proposed Framework

The proposed F-DiAE framework makes use of a two-stage autoencoder based architecture for fusing information from multiple sensors. The block diagram of the proposed framework is presented in Fig. 1. Let $\boldsymbol{X_i}$ denote the raw data or features extracted from the i^{th} sensor data for $i = \{1, 2, \cdots, n\}$. In the first stage, dedicated autoencoder are employed to learn high level abstract features $\boldsymbol{H_i}$ from each $\boldsymbol{X_i}$. Subsequently in the second stage, latent representations from all sensors are stacked together in $\boldsymbol{Z}$ and given as input to fusing autoencoder whose latent representation $\boldsymbol{H_f}$ are fed to a label consistent classifier.

In the training phase, a joint learning is carried out in which the sensor-specific autoencoders, fusing autoencoder and classification weights $\boldsymbol{M}$ are learnt utilizing the knowledge of the output labels $\boldsymbol{Y}$. This configuration allows discriminative features to be learnt from each sensor, thereby exploiting the complementary information shared by them towards deriving the final inference. In the test phase, the learnt sensor-specific encoders, fusing encoder and classification weights are utilized to generate the output corresponding to the test data. These two phases are described in detail below.

Training Phase: For a network of n sensors, utilizing the augmented Lagrangian formulation for autoencoders in (2), the joint optimization framework for F-DiAE is expressed as:

$$\begin{aligned}\min_{\theta} \sum_{i=1}^{n} \|\boldsymbol{H_i} - \boldsymbol{W_{ei}}\boldsymbol{X_i}\|_F^2 + \sum_{i=1}^{n} \|\boldsymbol{X_i} - \boldsymbol{W_{di}}\boldsymbol{H_i}\|_F^2 + \\ \|\boldsymbol{H_f} - \boldsymbol{W_e Z}\|_F^2 + \|\boldsymbol{Z} - \boldsymbol{W_d H_f}\|_F^2 + \lambda\|\boldsymbol{Y} - \boldsymbol{M H_f}\|_F^2 \\ \text{s.t. } \boldsymbol{H_1} \geq \boldsymbol{0}, \cdots, \boldsymbol{H_n} \geq \boldsymbol{0}, \boldsymbol{H_f} \geq \boldsymbol{0}\end{aligned} \tag{3}$$

where the parameter space $\theta = \{\boldsymbol{W_{e1}}, \boldsymbol{W_{d1}}, \boldsymbol{H_1}, \cdots, \boldsymbol{W_{en}}, \boldsymbol{W_{dn}}, \boldsymbol{H_n}, \boldsymbol{W_e}, \boldsymbol{W_d}, \boldsymbol{H_f}, \boldsymbol{M}\}$ and $\boldsymbol{Z} = [\boldsymbol{H_1}, \cdots, \boldsymbol{H_n}]^{\boldsymbol{T}}$. Here, a *ReLU* type non-linearity is applied on the latent representations by enforcing the representations of each layer to be non-negative [25]. The first two terms are associated with learning the sensor-specific autoencoders of the first stage. For the i^{th} sensor, input $\boldsymbol{X_i} \in \mathbb{R}^{L \times N}$ represents N measurements of L dimensions, the associated $\boldsymbol{W_{di}} \in \mathbb{R}^{L \times m_1}$, $\boldsymbol{W_{ei}} \in \mathbb{R}^{m_1 \times L}$ and $\boldsymbol{H_i} \in \mathbb{R}^{m_1 \times N}$ are learnt where m_1 is the dimension of the latent representation. The third and fourth terms are associated with the learning of the fusing autoencoder in the second stage using the stacked input from the first stage $\boldsymbol{Z} \in \mathbb{R}^{nm_1 \times N}$ with $\boldsymbol{W_d} \in \mathbb{R}^{nm_1 \times m_2}$, $\boldsymbol{W_e} \in \mathbb{R}^{m_2 \times nm_1}$ and $\boldsymbol{H_f} \in \mathbb{R}^{m_2 \times N}$ with m_2 as the dimension of the latent representation in this stage. The last term is the label consistency term [25] that maps $\boldsymbol{H_f}$ to the output $\boldsymbol{Y} \in \mathbb{R}^{k \times N}$ using the classifier $\boldsymbol{M} \in \mathbb{R}^{k \times m_2}$ where k denotes the number of classes. The output $\boldsymbol{Y} = [\boldsymbol{y_1}|\boldsymbol{y_2}|\cdots|\boldsymbol{y_N}]$ consists of N binary vectors of length k with the labeled class denoted by 1 and the remaining 0. Please note in this framework, both the encoder and decoder weights of the autoencoders are given equal importance and the hyperparameter associated with these terms is 1 and hence not explicitly shown.

The non-convex optimization problem in (3) is solved

using Alternate Direction Method of Multipliers (ADMM) that guarantee global convergence compared to other gradient methods [28]. ADMM reduces (3) into a number of sub-problems; to solve for one of the parameters at a time while keeping others fixed. For the i^{th} sensor-specific autoencoder of the first stage of F-DiAE, the sub-problems to learn the encoder and decoder weights, and hidden representation are given as:

$$\boldsymbol{W_{ei}} \leftarrow \min_{\boldsymbol{W_{ei}}} \|\boldsymbol{H_i} - \boldsymbol{W_{ei}}\boldsymbol{X_i}\|_F^2 \tag{4}$$

$$\boldsymbol{W_{di}} \leftarrow \min_{\boldsymbol{W_{di}}} \|\boldsymbol{X_i} - \boldsymbol{W_{di}}\boldsymbol{H_i}\|_F^2 \tag{5}$$

$$\begin{aligned}\boldsymbol{H_i} \leftarrow \min_{\boldsymbol{H_i}} &\|\boldsymbol{H_i} - \boldsymbol{W_{ei}}\boldsymbol{X_i}\|_F^2 + \|\boldsymbol{X_i} - \boldsymbol{W_{di}}\boldsymbol{H_i}\|_F^2 + \\ &\|\boldsymbol{H_f} - \boldsymbol{W_e}\boldsymbol{Z}\|_F^2 + \|\boldsymbol{Z} - \boldsymbol{W_d}\boldsymbol{H_f}\|_F^2\end{aligned} \tag{6}$$

The update equations (4) and (5) are straightforward, and can be solved using least squares. For the update of $\boldsymbol{H_i}$, the third and fourth terms in (6) are expanded in terms of $\boldsymbol{H_i}$ as below:

$$\|\boldsymbol{H_f} - \boldsymbol{W_e}\boldsymbol{Z}\|_F^2 = \Big\|\boldsymbol{H_f} - \boldsymbol{W_e^{(i)}}\boldsymbol{H_i} - \sum_{j=1,j\neq i}^{n} \boldsymbol{W_e^{(j)}}\boldsymbol{H_j}\Big\|_F^2$$

$$\|\boldsymbol{Z} - \boldsymbol{W_d}\boldsymbol{H_f}\|_F^2 = \Big\|\boldsymbol{H_i} - \boldsymbol{W_d^{(i)}}\boldsymbol{H_f} + \sum_{j=1,j\neq i}^{n} (\boldsymbol{H_j} - \boldsymbol{W_d^{(j)}}\boldsymbol{H_f})\Big\|_F^2$$

where $\boldsymbol{W_e} = [\boldsymbol{W_e^{(1)}}|\boldsymbol{W_e^{(2)}}|\cdots|\boldsymbol{W_e^{(n)}}]$, and $\boldsymbol{W_d} = [\boldsymbol{W_d^{(1)}}, \boldsymbol{W_d^{(2)}}, \cdots, \boldsymbol{W_d^{(n)}}]^{\boldsymbol{T}}$. Solving for $\boldsymbol{H_i}$ by expanding the formulation (6) in terms of trace, taking derivative with respect to $\boldsymbol{H_i}$ and equating to 0 gives the following closed form update:

$$\begin{aligned}\boldsymbol{H_i} = \Big(\boldsymbol{W_{di}^T}\boldsymbol{W_{di}} + 2I + \boldsymbol{W_e^{(i)}}^T\boldsymbol{W_e^{(i)}}\Big)^{-1}\Big(\boldsymbol{W_{di}^T}\boldsymbol{X_i} + \boldsymbol{W_{ei}}\boldsymbol{X_i} \\ +(\sum_{i=1}^{n}\boldsymbol{W_d^{(i)}})\boldsymbol{H_f} + \boldsymbol{W_e^{(i)}}^T\boldsymbol{H_f} - \sum_{j=1,j\neq i}^{n}\boldsymbol{H_j} \\ -\boldsymbol{W_e^{(i)}}^T(\sum_{j=1,j\neq i}^{n}\boldsymbol{W_e^{(j)}}\boldsymbol{H_j})\Big)\end{aligned} \tag{7}$$

Similarly, one can obtain the updates for all other sensor-specific autoencoders. For the second stage, $\boldsymbol{W_e}$, $\boldsymbol{W_d}$ and $\boldsymbol{M}$ are updated using least squares by solving the following:

$$\boldsymbol{W_e} \leftarrow \min_{\boldsymbol{W_e}} \|\boldsymbol{H_f} - \boldsymbol{W_e}\boldsymbol{Z}\|_F^2 \tag{8}$$

$$\boldsymbol{W_d} \leftarrow \min_{\boldsymbol{W_d}} \|\boldsymbol{Z} - \boldsymbol{W_d}\boldsymbol{H_f}\|_F^2 \tag{9}$$

$$\boldsymbol{M} \leftarrow \min_{\boldsymbol{M}} \|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{H_f}\|_F^2. \tag{10}$$

The update for $\boldsymbol{H_f}$ is obtained by solving for the following sub-problem:

$$\boldsymbol{H_f} \leftarrow \min_{\boldsymbol{H_f}} \|\boldsymbol{H_f} - \boldsymbol{W_e}\boldsymbol{Z}\|_F^2 + \|\boldsymbol{Z} - \boldsymbol{W_d}\boldsymbol{H_f}\|_F^2 + \lambda\|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{H_f}\|_F^2. \tag{11}$$

Similar to solving for (6), the update for $\boldsymbol{H_f}$ is given as:

$$\begin{aligned}\boldsymbol{H_f} = \Big(\boldsymbol{W_d^T}\boldsymbol{W_d} + I + \lambda\boldsymbol{M}^T\boldsymbol{M}\Big)^{-1}\Big(\boldsymbol{W_d^T}\boldsymbol{Z} + \boldsymbol{W_e}\boldsymbol{Z} + \\ \lambda\boldsymbol{M}^T\boldsymbol{Y}\Big).\end{aligned} \tag{12}$$

It is to be noted that the non-negativity constraints are not considered explicitly here. This would have required solving an iterative forward-backward algorithm which, in turn, would make the solution more time consuming. Instead, an approximation is used whereby after solving for each latent representation, the negative values are set to 0 (see [29]). The network parameters of the F-DiAE framework are updated iteratively until the convergence criteria is met after which the model is said to be learnt and the training stops. Convergence is achieved when the loss function (3) is within the predefined threshold or tolerance level.

Test Phase: For test data $\boldsymbol{X_i^{test}}$ corresponding to the i-th sensor, the latent representation $\boldsymbol{H_i^{test}}$ is obtained using the learnt weight $\boldsymbol{W_{ei}}$ expressed as:

$$\boldsymbol{H_i^{test}} = \boldsymbol{W_{ei}}\boldsymbol{X_i^{test}}. \tag{13}$$

This is repeated for all n sensors. All latent representations are stacked as $\boldsymbol{Z^{test}} = [\boldsymbol{H_1^{test}}, \cdots, \boldsymbol{H_n^{test}}]^T$ and are subsequently used to estimate the output label using the following:

$$\boldsymbol{H_f^{test}} = \boldsymbol{W_e}\boldsymbol{Z^{test}}$$

$$\boldsymbol{\hat{Y}^{test}} = \boldsymbol{M}\boldsymbol{H_f^{test}}.$$

III. Results and Discussion

Two datasets from completely different domains have been considered to demonstrate the effectiveness and generalizability of the proposed F-DiAE framework for fusion. The first one is from industrial machines for bearing fault detection and classification. The second one is based on human activity detection using two spatially distributed radars. The performance of F-DiAE is compared against the state-of-the-art methods for both the datasets. F-DiAE employs a *Max Pooling* architecture at the output to determine the class label. The value of hyperparameter λ and hidden layer dimension of all the autoencoders of F-DiAE are tuned using grid search and optimal values are used to present the results.

A. Bearing Fault Detection

The bearing data provided by the Case Western Reserve University (CWRU) Bearing Data Center [30] is used for classifying normal and faulty bearings. The experimental setup includes a motor whose shaft is supported by bearings installed at the drive end and fan end of the motor. Faults of different diameters 7mils, 14mils and 21mils are introduced each in the ball, inner race and outer race of the bearings. Two accelerometers, mounted at the drive end and fan end of the motor respectively, are used to capture the vibration data at a sampling frequency of 12 kHz under four different loading conditions - 0, 1, 2 and 3 hp.

A 4-class classification problem is studied for classifying healthy and faulty bearings. The different classes are healthy, ball defect, inner race defect and outer race defect respectively. This classification is carried out for the following two scenarios: (i) inception faults (faults at its onset) and (ii) faults across all severity levels. A sample duration of 500 milliseconds is considered for these two classification problems. For inception fault detection, faulty data corresponding to 7mils diameter is only considered which amounts to 320 samples for all the 4 classes. For fault detection across all severity levels, faulty

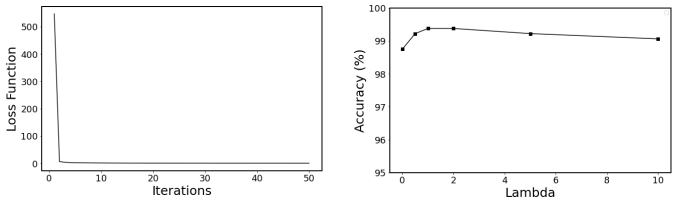
TABLE I: Performance comparison of F-DiAE and other methods for bearing fault detection

	Method	Hidden Nodes	Inception Faults (%)	All Faults (%)
[31]	SVM	-	97.2	-
	KNN	-	95.6	-
	MLP	(20,40)	96.8	-
	SAE-DNN	(20,40)	97.4	-
	RNN	(20,20)	95.6	-
	GRU	(20,20)	98.1	-
	BiGRU	(20,20)	98.5	-
	LFGRU	(20,80)	99.6	-
[32]	ICDSVM	-	**100**	97.75
	F-DiAE	(2*10,20)	**100**	**99.38**

data corresponding to all fault diameters are considered which results in a total of 640 samples for classification.

Similar to the work in [31], different time, frequency and time-frequency domain features are extracted from accelerometer data. Here, features from both the accelerometers mounted at the drive end and fan end are fed to F-DiAE framework for classification. The time domain features include root mean square, variance, data peak, kurtosis, and peak to peak values. The frequency domain features include spectral skewness, kurtosis and energy. Wavelet energy is used as the time-frequency feature. The performance of F-DiAE and comparisons with the state-of-the-art methods averaged over 5-fold cross validation are presented in Table I. The network structure employed for the deep learning models are also mentioned in the table. It can be seen that for inception faults, F-DiAE demonstrates superior performance with relatively simpler structure as compared to deep learning based methods like BiGRU and LFGRU in [31]. For comparing the performance of fault detection across all severity levels, F-DiAE has better accuracy with simple features than the work in [32] that employs permutation entropy and intrinsic mode function based features followed by optimized SVM for classification. Figure 2 presents the performance characteristics of F-DiAE optimization framework with (a) convergence plot and (b) accuracy versus λ plot for this dataset. The algorithm converges within a few iterations. It can be observed that the accuracy is not affected much for different λ values.

The performance of F-DiAE is also tested for noisy environments that emulate a real life scenario. Additive White Gaussian Noise (AWGN) is introduced at different SNR levels, and the corresponding classification accuracies are computed. Similar to the work in [33], 4-class classification for fault



(a) Convergence of loss function (b) Accuracy versus λ plot

Fig. 2: Performance of F-DiAE for bearing fault detection.

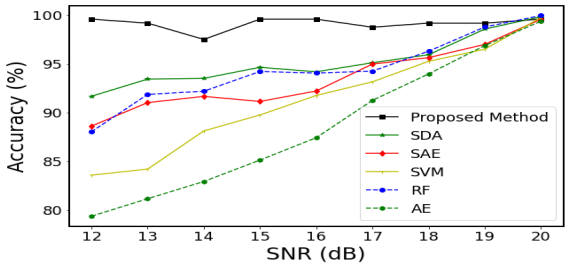


Fig. 3: Performance comparison of F-DiAE and other methods with AWGN at different SNRs.

identification is carried out with noisy bearing signals for 0 loading condition. The accuracy obtained with the proposed algorithm in the presence of noise, averaged over 5 folds cross validation is presented in Fig. 3 along with other methods in [33]. The results exhibit the superior performance of F-DiAE for all SNR values, except at 20 dB where performance is comparable to other methods. This highlights the robustness and effectiveness of the proposed framework for classification that is achieved by fusing information from multiple sensors.

B. Human Activity Recognition

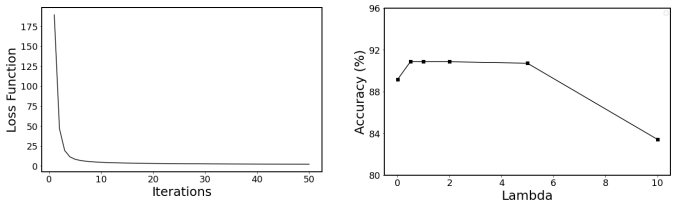
This dataset uses two spatially distributed single channel 10.525 GHz continuous wave radar sensors for human activity recognition [34]. The two radar sensors are placed vertically one below the other separated by a distance of 1 meter. A total of 20 individuals participated to this study, and performed 7 different activities corresponding to hand and leg movements. These activities are - (i) swing right leg, (ii) move hands inwards and outwards, (iii) swinging right leg and right hand, (iv) stamping both feet, (v) swinging hands to and fro, (vi) simulation of walking gesture and (vii) no movement. The individuals perform these activities at a distance of 2 meters from the radar setup stand. Each activity is performed for 20 seconds duration with the data being sampled at 2 kHz.

Different time, frequency and time-frequency features are extracted from 10 seconds data frames. These features include 3 time domain features - kurtosis, variance, zero crossing rate; 1 from frequency domain - spectral kurtosis; and 3 features from spectrogram envelope - mean, variance, skewness. Spectrogram is computed using 1024 point STFT with 256 point Kaiser window having 75% overlap. The average performance metrics of F-DiAE for a 7-class classification with 10-fold cross validation on the radar dataset are presented in Table II, along with results from the state-of-the-art methods utilized for this dataset [34].

The work in [34] employs 10 time domain and 48 time-frequency domain features over a two layer cascaded classifier

TABLE II: Performance comparison of F-DiAE and other methods for activity recognition with radar data

	Method	Hidden Nodes	Precision	Recall	Accuracy (%)
[34]	Multi-class Classifier	-	0.86	0.84	84.3
	2-layer Classifier	-	0.90	0.89	88.6
[25]	DiAE	(10)	0.86	0.88	86.1
	F-DiAE	(2*7,14)	**0.92**	**0.91**	**90.86**



(a) Convergence of loss function (b) Accuracy versus λ plot

Fig. 4: F-DiAE performance for human activity recognition.

network. The first layer determines the action group based on the limb involved, and the second layer consists of dedicated binary classifiers for each action group to determine the corresponding activity. The various classifiers involved require individual training. In Table II, it can be seen that DiAE [25] performs better the multi-class classifier method [34] since it is able to learn more discriminative features. However, the proposed F-DiAE has the best performance. F-DiAE carries out one shot classification using only 7 features mentioned above. This can be attributed to the fact that the joint learning of the autoencoders and classifier weights facilitates discriminative features to be learnt that result in robust classification. Similar to Section III-A, the performance characteristics of the F-DiAE framework with radar data is presented in Fig. 4. It can be seen that the algorithm converges fast, within a few iterations. Also, it can be observed that the accuracy is more for $\lambda \leq 5$, after which it drops considerably.

IV. Conclusion and Future Work

The paper presents a discriminative autoencoder based architecture for multi-sensor fusion. The joint learning of the autoencoders and classifier weights allows discriminative and robust features to be learnt from sensor signals for more reliable inference making. This enables F-DiAE to perform better than state-of-the-art methods for the two datasets considered in this work. Also, the results presented at different SNR levels demonstrate the robustness of the framework to noisy environments. The proposed fusion framework is generic, and can be adopted for other application domains that may involve heterogeneous sensors as well. A direct extension of this work is to explore deeper or stacked versions of F-DiAE at both sensor representation learning and fusion stages for improved inference making. In future, F-DiAE model can also be examined for regression and synthesis tasks.

References

[1] B. Khaleghi *et al.*, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[2] F. Castanedo, "A review of data fusion techniques," *The Scientific World Journal*, 2013.

[3] H. DURRANT-WHYTE, "Sensor models and multisensor integration," *The International Journal of Robotics Research*, vol. 7, no. 6, 1988.

[4] M. Schmitt and X. X. Zhu, "Data fusion and remote sensing: An ever-growing relationship," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 4, pp. 6–23, 2016.

[5] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, 2018.

[6] Y. A. V. Phamila and R. Amutha, "Discrete cosine transform based fusion of multi-focus images for visual sensor networks," *Signal Processing*, vol. 95, pp. 161 – 170, 2014.

[7] S. Jusoh and S. Almajali, "A systematic review on fusion techniques and approaches used in applications," *IEEE Access*, vol. 8, 2020.

[8] M. Tang *et al.*, "Information geometric approach to multisensor estimation fusion," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 279–292, 2018.

[9] J. M. Moura, J. Lu, and M. Kleiner, "Intelligent sensor fusion: A graphical model approach," in *Proceedings of IEEE ICASSP*, vol. 6, 2003, pp. VI–733.

[10] Q. Zhu *et al.*, "Latent correlation embedded discriminative multi-modal data fusion," *Signal Processing*, vol. 171, p. 107466, 2020.

[11] R. C. Luo *et al.*, "Multisensor fusion and integration: Theories, applications, and its perspectives," *IEEE Sensors Journal*, vol. 11, no. 12, 2011.

[12] N. Xiong and P. Svensson, "Multi-sensor management for information fusion: issues and approaches," *Information Fusion*, vol. 3, no. 2, 2002.

[13] C. Zhang, Z. Yang, X. He, and L. Deng, "Multimodal intelligence: Representation learning, information fusion, and applications," *IEEE Journal of Selected Topics in Signal Processing (2020)*, 2020.

[14] J. Ngiam *et al.*, "Multimodal deep learning," in *ICML*, 2011.

[15] L. Wang and Q. Liang, "Representation learning and nature encoded fusion for heterogeneous sensor networks," *IEEE Access*, vol. 7, 2019.

[16] A. Sano *et al.*, "Multimodal ambulatory sleep detection using LSTM recurrent neural networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1607–1617, 2018.

[17] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[18] G. E. Hinton *et al.*, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] H. Shao, H. Jiang, H. Zhao, and F. Wang, "A novel deep autoencoder feature learning method for rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 95, pp. 187–204, 2017.

[20] A. Majumdar, "Blind denoising autoencoder," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, 2018.

[21] J. Yu, D. Huang, and Z. Wei, "Unsupervised image segmentation via stacked denoising auto-encoder and hierarchical patch indexing," *Signal Processing*, vol. 143, pp. 346–353, 2018.

[22] O. Nikisins *et al.*, "Domain adaptation in multi-channel autoencoder based features for robust face anti-spoofing," in *2019 International Conference on Biometrics (ICB)*. IEEE, 2019, pp. 1–8.

[23] F. J. Pulgar *et al.*, "AEkNN: An autoencoder kNN-based classifier with built-in dimensionality reduction," *International Journal of Computational Intelligence Systems*, vol. 12, pp. 436–452, 2019.

[24] Z. Chen and W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 7, pp. 1693–1702, 2017.

[25] A. Gogna and A. Majumdar, "Discriminative autoencoder for feature extraction: Application to character recognition," *Neural Processing Letters*, vol. 49, no. 3, pp. 1723–1735, 2019.

[26] K. Kumar *et al.*, "Transfuse: A transform learning based multi-sensor fusion framework," *IEEE Sensors Letters (2020)*, 2020.

[27] G. Taylor *et al.*, "Training neural networks without gradients: A scalable ADMM approach," in *International Conference on Machine Learning*, 2016, pp. 2722–2731.

[28] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[29] J. Maggu *et al.*, "Deeply transformed subspace clustering," *Signal Processing*, vol. 174, p. 107628, 2020.

[30] Z. Li, "CWRU bearing dataset and gearbox dataset of ieee PHM challenge competition in 2009," 2019. [Online]. Available: http://dx.doi.org/10.21227/g8ts-zd15

[31] R. Zhao *et al.*, "Machine health monitoring using local feature-based gated recurrent unit networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1539–1548, 2017.

[32] X. Zhang *et al.*, "A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM," *Measurement*, vol. 69, pp. 164–179, 2015.

[33] C. Lu *et al.*, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Processing*, vol. 130, pp. 377–388, 2017.

[34] S. Rani *et al.*, "Action recognition using spatially distributed radar setup through microdoppler signature," in *UBICOMM*, 2020, pp. 278–253.