2019 IEEE International Conference on Big Data (Big Data)

Multi-View, Generative, Transfer Learning for Distributed Time Series Classification

Sreyasee Das Bhattacharjee* §, William J. Tolone*, Ashish Mahabal†, Mohammed Elshambakey*‡, Isaac Cho*, Abdullah al-Raihan Nayeem*, Junsong Yuan§, George Djorgovski†
*College of Computing and Informatics, University of North Carolina at Charlotte, Charlotte, United States
Email:{sdasbhat; william.tolone; melshamb; icho1; anayeem}@uncc.edu
‡IRI, SRTA-City, New Borg Elarab, Alexandria, 21934, Egypt
†Department of Astronomy, California Institute of Technology, Pasedona, United States
Email:{aam; george}@astro.caltech.edu
§Department of Computer Sc. & Eng., State University of New York at Buffalo, United States
Email: {sreyasee, jsyuan}@buffalo.edu

Abstract*—In this paper, we propose an effective, multi-view, generative, transfer learning framework for multivariate time-series data. While generative models are demonstrated effective for several machine learning tasks, their application to time-series classification problems is underexplored. The need for additional exploration is motivated when data are large, annotations are unbalanced or scarce, or data are distributed and fragmented. Recent advances in computer vision attempt to use synthesized samples with system generated annotations to overcome the lack or imbalance of annotated data. However, in multi-view problem settings, view mismatches between the synthetic data and real data pose additional challenges against harnessing new annotated data collections. The proposed method offers important contributions to facilitate knowledge sharing, while simultaneously ensuring an effective solution for domain-specific, fine-level categorizations. We propose a principled way to perform view adaptation in a cross-view learning environment, wherein pairwise view similarity is identified by a smaller subset of source samples that closely resemble the target data patterns. This approach integrates generative models within a deep classification framework to minimize the gap between source and target data. More precisely, we design category specific conditional, generative models to update the source generator in order for transforming source features so that they appear as target features and simultaneously tune the associated discriminative model to distinguish these features. During each learning iteration, the source generator is conditioned by a source training set represented as some target-like features. This transformation in appearance was performed via a target generator specifically learned for target-specific customization per category. Afterward, a smaller source training set, indicating close target pattern resemblance in terms of the corresponding generative and discriminative loss, is used to fine-tune the source classification model parameters. Experiments show that compared to existing approaches, our proposed multi-view, generative, transfer learning framework improves time-series classification performance by around 4% in the UCI multi-view activity recognition dataset, while also showing a robust, generalized representation capacity in classifying several large-scale multi-view light curve collections.

***Index Terms*—Generative Model, GAN, LSTM, RNN, Multi-view Classification, Transfer Learning, Distributed Time-Series Analysis, Deep Learning**

I. Introduction

Multi-view time-series data are prevalent in many fields including finance, medicine, security, surveillance, and astronomy. In fact, several specialized metrics like Dynamic Time Wrapping (DTW) [1], edit distance [2], elastic distance [3], and several others, e.g., [4], are proposed to overcome specific challenges associated with time-series classification. However, the basic assumption in most existing literature is the availability of a large collection of labeled data samples. In fact, most of these methods rely exclusively on sophisticated, hand-crafted features to capture the local data patterns. Therefore, the efficiency and precision of these classification approaches are heavily dependent on the availability of large collections of labeled data that capture the entire spectrum of data characteristics and the quality of the hand-crafted features used to define a comprehensive descriptor.

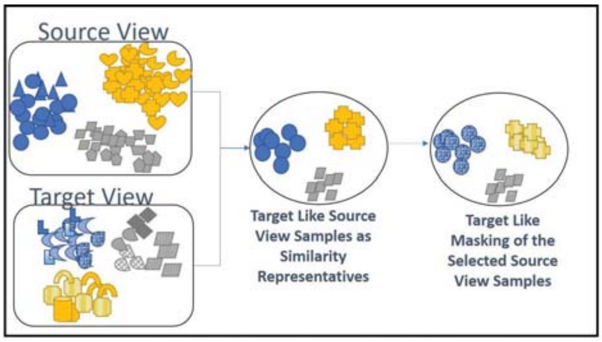

Fig. 1. In a distributed learning environment, the proposed method facilitates a structured view adaptation by defining pairwise view similarity in terms of a smaller subset of source samples that closely resemble the target data patterns, which are then transformed by means of a learned conditional GAN model to appear as target features.

In contrast to the shallow feature-based models [3], [4], deep neural networks recently have shown an outstanding performance over nearest neighbor approaches when coupled with DTW for the Time Series Classification (TSC) problem [5]. However, despite their impressive performance, such models are prone to overfitting, particularly when the number of training samples is not large or when data patterns continuously evolve over time [5].

As a scalable alternative, the use of synthetic data for learning, via Generative Adversarial Networks (GANs) has shown great promise by designing generative models via adversarial training [6]. While GAN models have been very successful in computer vision related tasks such as generating realistic-looking images, there has been limited work to date that proposes a GAN framework for sequential data (e.g. [7], [8]). The initial successes of GAN efforts for realistic complex problem settings in an adversarial fashion, however, are producing encouraging results. In this paper, we propose a multi-view, generative, transfer learning framework to conduct structured view adaptation in a distributed environment for the classification of multivariate time-series data.

While transfer learning has been widely used for various tasks in computer vision, [9], [10], social media analytics [11]–[13], anomaly detection [14], the applicability of transfer learning to time-series classification problems is underexplored. In fact, the development of deep learning models for time-series data – models that adequately address the practical challenges of changing data patterns and the need for sufficiently large annotated data collections where annotation is costly or difficult – is plagued with open research questions.

The proposed method employs an efficient multi-phased learning strategy that combines generative models within a deep classification module to compensate for the dearth of annotated data at a source site, thereby enabling a scheme for smooth knowledge sharing within a distributed, multi-view environment. For each category, an initial view-specific, multivariate GAN is learned at a target site and shared with networked peer-views at a source site to facilitate knowledge transfer as well as effective view adaptation of the source multiclass, deep learning model prior to transfer.

Unlike traditional deep learning architectures, which typically require a large number of annotated samples as well as extensive training, the proposed method offers a more efficient alternative to expedite two-phased learning, wherein target-specific GAN metadata are transfer to a source site, thereby enabling compact knowledge sharing without having to share target samples. In particular, we propose a principled approach to a semi-supervised, view adaptation that transforms the feature maps of the source site so that the samples appear as if they were from the target site, while maintaining their semantic spatial layouts. At every learning iteration, the source generative model (conditioned by the target-view generative model) and the source discriminator compete against each other so that the source generator learns to produce target-like features by successfully fooling the source discriminator. Afterward, a smaller collection of source samples that closely resembles the target site, evaluated in terms of corresponding generative and discriminative losses, is used to fine-tune the baseline source classification model prior to transfer. This produces a more discriminative, multi-view, deep transfer model without requiring each view in the network to be retrained from scratch.

The proposed multi-view, generative transfer learning framework for multivariate time-series data offers important advances that begin to address some of these open questions, referenced above. Unlike the traditional time-series classification models, the conditional GAN-trained discriminators learn to detect the realistic samples for each category across multiple views under consideration. In contrast to the existing multi-view methods that align two domains [15], [16], the proposed transfer learning framework can facilitate accurate knowledge sharing over networked peer-views by means of a comprehensive yet compact set of learned GAN parameters to define target-specific model requirements. By fine-tuning the generic baseline source view multiclass deep classification model on an in-house selected smaller subset that closely resemble the target data patterns, the proposed framework ensures more effective classification performance across multiple independent views. Figure 2 illustrates the overall workflow, which demonstrates an improved classification performance by roughly 4% in the UCI multi-view activity recognition dataset, while also showing a robust, generalized representation capacity in classifying several large-scale multi-view light curve collections.

Figure 1 describes an overview of the proposed method. The rest of the paper is organized as follows: Section II briefly describes the related works. Section III describes the proposed method. Section IV and Section V present the experimental results and concluding thoughts respectively.

II. Related Works

In this section, we describe recent related research from three areas: 1) Generative Learning methods for representing and analyzing sequential data; 2) Deep Learning methods for time-series classification; and, 3) Transfer Learning methods for time-series classification.

Since their inception, Generative Adversarial Networks (GANs) [6] have received significant research attention with much of their application focused on analyzing image data [17]–[19]. In a recent work, Choi et al. [20] propose a GAN to generate synthetic electronic health record (EHR) datasets. Yu et al. [21] design a sequential data generation with GANs trained using Reinforcement Learning. Morgen [7] generates, using a GAN with LSTM generator and discriminator, continuous valued sequences that aim to produce polyphonic music. Conditional GAN models condition the learned model on additional constraints and therefore drive the data generation process in a more customized manner [22], [23]. Li et al. [24] propose a GAN architecture for dialogue generation process. The fundamental difference between our work and these efforts is architectural. First, our GAN-based models are designed for a transfer framework, where a target site generator represents the entire data patterns in a distributed learning environment. Second, we introduce conditional GAN learning to customize the initial baseline source GAN model to produce more accurate target-like samples as a way to compensate the lack of target data during learning.

Deep learning methods for time-series classification also have received significant research attention. [25]. Yi ei al. [25]

have proposed using Multi-channel Deep Convolutional Neural Network (MC-DCNN) for multivariate time-series classification, wherein, input from each variable is used to obtain latent features, which are then fed in a Multi-Layer Perceptron (MLP) to perform classification. Karim et al. [26] augment existing LSTM-FCN and ALSTM-FCN with a squeeze and excitation block for an improved performance. Wang et al. [5] introduce a three layer deep convolutional neural network architecture, accompanied by average pooling for time-series classification. Fawaz et al. [27] propose different data augmentation techniques to avoid overfitting and learn a generalized model. Geng & Luo [28] offer a cost-sensitive learning strategy to modify the temporal time-series learning models. Ziat et al. [30] propose dynamical spatio-temporal model formalized as a recurrent neural network for forecasting time-series of spatial processes, wherein the model learns spatio-temporal dependencies through a structured latent dynamical component, while a decoder predicts the observations from the latent representations. At the intersection of generative methods and time-series classification, in a recent work, Nweke et al. [31] have reviewed classification and evaluation procedures and discussed their applicability on several publicly available datasets for mobile sensor human activity recognition. Che et al. [32] analyze Electronic Health Records (EHRs) sequential data by combining a generative model together with a Convolutional Neural Network (CNN) prediction model to improve risk prediction performance with limited data. A comprehensive review of state-of-the-art methods proposed for time-series classification can be found [29]. A comprehensive review of the state-of-the-art methods proposed for this problem can be found in [29]. Our work is distinguished from these efforts through its unique and effective application of generative models with deep learning models in a distributed context.

In this context, transfer learning methods for time-series data mining tasks have been explored, where models are learned on source data and then transferred to target data to minimize the effect of cross domain discrepancies [33], [34]. Transfer learning methods have been used for anomaly detection [35], time-series forecasting [36] and recognition [37]. However, unlike existing deep learning methods for time-series classification, the proposed framework demonstrates the benefits of fine-tuning from a selected smaller subset of informative source data that present patterns similar to the target data for more precise classification performance upon transfer.

Despite the above advances, the fundamental challenge of learning effectively from large data resources across multiple sites prevails. The ultimate goal remains to identify effective and efficient techniques for time-series classification on large distributed data.

III. Proposed Method

A multivariate time-series representation of a set of time-series signals $\{\mathbf{x}_i\}_i$ is defined in terms of an ordered sequence $\mathbf{f}_i = \{f_{i,1}, ..., f_{i,T}\}$ of T time steps, where each $f_{i,j} = (f^1_{i,j}, ..., f^d_{i,j}) \in \mathbb{R}^d$ is represented using the j^{th} time step response of d ($\in \mathbb{N}$)-streams. For example, the streams of d statistical features represent the same light curve signal at T time steps. Thus, a multivariate time-series $\mathbf{f}_i$ is represented in terms of a $d \times T$ matrix, where d is the number of variables (or feature dimensions) and T is the number of observations (or timestamps). In our multi-view framework, $\mathcal{D}_v = \{(\mathbf{x}^v_i, c^v_i)\}_i$ represents the annotated sample collection available for view $v \in \mathcal{V}$, where $c^v_i \in \{1, ..., C\}$ is the label for the signal $\mathbf{x}^v_i$[1], represented by $\mathbf{f}^v_i$.

Given a collection of annotated samples $\mathcal{D} = \{\mathcal{D}_v\}_{v \in \mathcal{V}}$, the task is to design an effective and efficient transfer learning framework that learns base models on a source view dataset and then transfers the learned models (e.g., network weights) to a target view, for replicating the learning architecture (at least partially), in order to initiate a cost-effective customization with the target data. Such a framework enables a distributed learning environment and enhances the generalization capabilities for multi-view classification. The proposed learning architecture has two important components: 1) Generative Data Representation Module to learn the data patterns in a distributed environment across multiple views; and 2) A Transferable Deep Classification Model.

A. Generative Data Representation Module

Given a labeled dataset in a source (or target) view, our goal is to train a GAN discriminator and generator pair as two Long Short Term Model (LSTM) based Recurrent Neural Network (RNN) to learn the multivariate data patterns from $\mathcal{D}_v$ [38]. In this section, we describe briefly the construction of view-specific GANs and explain of the general design of our proposed conditional GAN model in the context of structured view adaptation in a distributed environment.

1) View-Specific Generative Model: Given $v \in \mathcal{V}$ and the view-specific annotated sub-collection $\mathcal{D}^c_v = \{(\mathbf{x}^v_i, c)\}_i (\subset \mathcal{D}_v)$ representing the specific category $c \in C$, the generator (G^c_v) generates synthetic time series data with sequences from a random latent space as input, and passes the generated sequence samples to the discriminator (M^c_v), which then attempts to distinguish the generated (i.e., 'fake') sequences from the normal (i.e., 'real') training sequences. To capture the latent interactions among feature dimensions within the learned model, the proposed GAN considers the entire feature vector ($\mathbf{f}^v_i$), in contrast to considering each of its d projection components independently.

As in the standard GAN training framework, at every iteration, the parameters of M^c_v and G^c_v are updated based on the outputs of M^c_v from the previous iteration to improve discriminability, while simultaneously preparing G^c_v to capture the hidden multivariate distribution. This enables G^c_v to generate more realistic samples that more closely represent the subtle multivariate data patterns of category c as observed in v^{th} view. Given a collection of multivariate sequences obtained

[1]Please note that for different views, the dimension d of $f_{i,j}$ can differ and therefore the dimension of $\mathbf{f}^v_i$ representing $\mathbf{x}^v_i$ can vary. However, for simplicity and without loss of generality, we hold d consistent across different views in this presentation.

from the latent random space, $\mathcal{Z} = \{\mathbf{z}_i\}_i$ and $\mathcal{D}_v^c$, the GAN model is trained using the corresponding two-player minimax value function defined as:

$$\underset{G_v^c}{min}\ \underset{M_v^c}{max}\ V(G_v^c, M_v^c) = \mathbb{E}_{\mathbf{f}\sim p_{data}(\mathcal{D}_v^c)}[log(M_v^c(\mathbf{f}))] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathcal{Z})}[log(1 - M_v^c(G_v^c(\mathbf{z})))] \quad (1)$$

Both M_v^c and G_v^c are designed using stacked LSTM architectures [38].

2) View-Specific GAN-based Loss Term: In a view-specific GAN model, both the discriminator (M_v^c) and generator (G_v^c) learn the category specific data patterns from two complementary perspectives: representativeness and discriminativeness. While M_v^c learns to evaluate the authenticity of a given test sample $\mathbf{x}_i$, G_v^c learns to generate real-like samples representing category c for view v, viewed as $G_v^c : \mathcal{Z} \rightarrow \mathcal{D}_v^c$ (i.e., representativeness). In fact, for each $\mathbf{z}_i \in \mathcal{Z}$ generator output $G_v^c(\mathbf{z}_i)$ is expected to be very similar to some sample $\mathbf{x}_i \in \mathcal{D}_v^c$.Thus, it is possible to find a corresponding $\mathbf{z}_i^v$ (which we call 'v-invert') to each test sample $\mathbf{x}_i$ represented by $\mathbf{f}_i$ in the test subcollection of $\mathcal{D}_v^c$, such that $Dis(G_v^c(\mathbf{z}_i^v), \mathbf{f}_i)$ is minimized.

Given a test sample $\mathbf{x}_i$, $M_v^c(\mathbf{f}_i)$ and $Dis(G_v^c(\mathbf{z}_i^v), \mathbf{f}_i)$ compute its view specific *Discriminative Loss* and *Generative Loss*, respectively. While these loss values offering some complementary yet insightful evaluations of $\mathbf{x}$'s fitness as an instance of category c from the perspective of view v, the overall *Fitness Loss* is defined as:

$$L(\mathbf{x}_i, c, v) = \lambda M_v^c(\mathbf{f}_i) + (1 - \lambda) Dis(G_v^c(\mathbf{z}_i^v), \mathbf{f}_i) \quad (2)$$

where λ is a user defined parameter.

Given a sample $\mathbf{x}_i$ represented by $\mathbf{f}_i$, a precise estimation of $\mathbf{z}_i^v$ is obtained by formulating a minimization problem $\underset{\mathbf{z}_i \in \mathcal{Z}}{min}[Dis(G_v^c(\mathbf{z}_i), \mathbf{f}_i)]$. Within our framework, starting with a random initialization of $\mathbf{z}_i$, we use gradient descent for optimization.

3) Generative Transfer Learning for Cross-View Data Representation: The category-specific generator at each view aims to generate sample representations that are very similar to the real data to address issues including a shortage of annotated samples for a given category. In fact, the proposed GAN model enables (for transfer) the generation of target-like synthetic data resembling genuine, fine-grained, target data patterns, without having to share the original data in a networked peer view. As such, the representation of the same category will vary across multiple independent views (e.g., light curve representations of a common astronomical object will differ when captured by different telescopes). Therefore, in order to reduce the domain gap, we design a conditional GAN fine-tuning process, that transfers the learned target GAN model parameters (G_{tr}^c, M_{tr}^c) to customize the source view GAN model (G_{sr}^c, M_{sr}^c) as per the target requirement. It is important to note that, while it may be infeasible for the target view to directly share its data repository with other networked peer-views, dissiminating the learned data representation in terms of a compact set of encoded parameters (G_{tr}^c, M_{tr}^c) is simpler, more convenient, and thus more feasible. As shown in Figure 2, given $\mathbf{x}_i^{sr} \in \mathcal{D}_{sr}^c$ from the category $c \in \mathcal{C}$ and its sr-invert $\mathbf{z}^{sr}$, $G_{tr}^c(\mathbf{z}_i^{sr})$ employs a target-like masking to the source sample $\mathbf{x}_i^{sr}$ highlighting critical intraclass data patterns specific to the target view in consideration. Then $G_{tr}^c(\mathbf{z}_i^{sr})$ is combined with $\mathbf{f}_i^{sr}$ (feature descriptor of $\mathbf{x}_i^{sr}$ in the source view) via a nonlinear operator $\mathcal{O}p(,)$. The resulting $\mathbf{g}_i^{tr} = Op(G_{tr}^c(\mathbf{z}_i^{sr}), \mathbf{f}_i^{sr})$ as a transformed feature descriptor for $\mathbf{x}_i^{sr}$ preserves some low-level pattern information of both views (sr and tr), while making the transformed representative appear more like an instance from the target view tr. In this work, we use max-pool operator to define $Op(,)$. Finally, the initial source view GAN model (G_{sr}^c, M_{sr}^c) is fine-tuned by conditioning it on the entire collection $[\underset{c\in\mathcal{C}}{\cup}\{(\mathbf{g}_i^{tr}, c)\}_i]$ of the transformed feature descriptors, and thereby enabling the source system generate a more target-like representations, while retaining sufficient patterns informations specific to source views as well.

4) Multi-View GAN-Based Loss Term: Important to note that transfer learning is performed by moving limited metadata (e.g., learned weights, architecture details, tunable parameters, etc.) between a source view sr and an independent target view tr. Therefore, in order to facilitate knowledge sharing without requiring the transfer of large data, it is critical for the source to gain insights on the subtle yet critical category-specific target data patterns that contribute to the large intraclass variances often observed in such distributed learning environments. To achieve this end, we first identify a subset $\mathcal{D}_0 \subset \mathcal{D}_{sr}$ that closely resembles the data patterns observed at target view tr.

Given the source view collection $\mathcal{D}_{sr} = \underset{c\in\mathcal{C}}{\cup}\mathcal{D}_{sr}^c$ where each $\mathcal{D}_{sr}^c$ represents the subcollection of the source data repository describing each category $c \in \mathcal{C}$, a composite loss term (L_{cr}^{sr}) evaluating the fitness of each $\mathbf{x}_i^{sr} \in \mathcal{D}_{sr}^c$ is defined as:

$$L_{cr}^{sr}(\mathbf{x}_i^{sr}) = \alpha L(\mathbf{x}^{sr}, c, tr) + (1 - \alpha)\mathrm{L}(\mathrm{x}^{sr}, c, sr) \quad (3)$$

where α is a user defined parameter balancing the relative importance of each view-specific loss terms. In our experiments, we set $\alpha = 0.6$.

For each category c in the source view (sr), the top k source samples with least composite loss terms are retained within the selected sample subset ($\mathcal{D}_0 \subset \mathcal{D}_{sr}$); these are samples more similar to the class-specific data patterns observed within the target view (tr). Notice that, although this is a greedy approach with no guarantee of optimality on $\mathcal{D}_0$, with size kC, we found this simple strategy works well in practical cases. Throughout all the experiments reported in the paper, we use $k = 30$ samples per category for fine-tuning prior to transferring to the target view tr. The exprimental results reported in Section IV-D demonstrate the performance stability of the proposed method within a range of neighboring values for k.

B. View-Specific Deep Classification Module

Given $\{\mathrm{f}_i^v\}_i$, a Long Short-Term Memory (LSTM) network model, which is a variant of Recurrent Network Model (RNN), is adopted for learning the time-series descriptor.

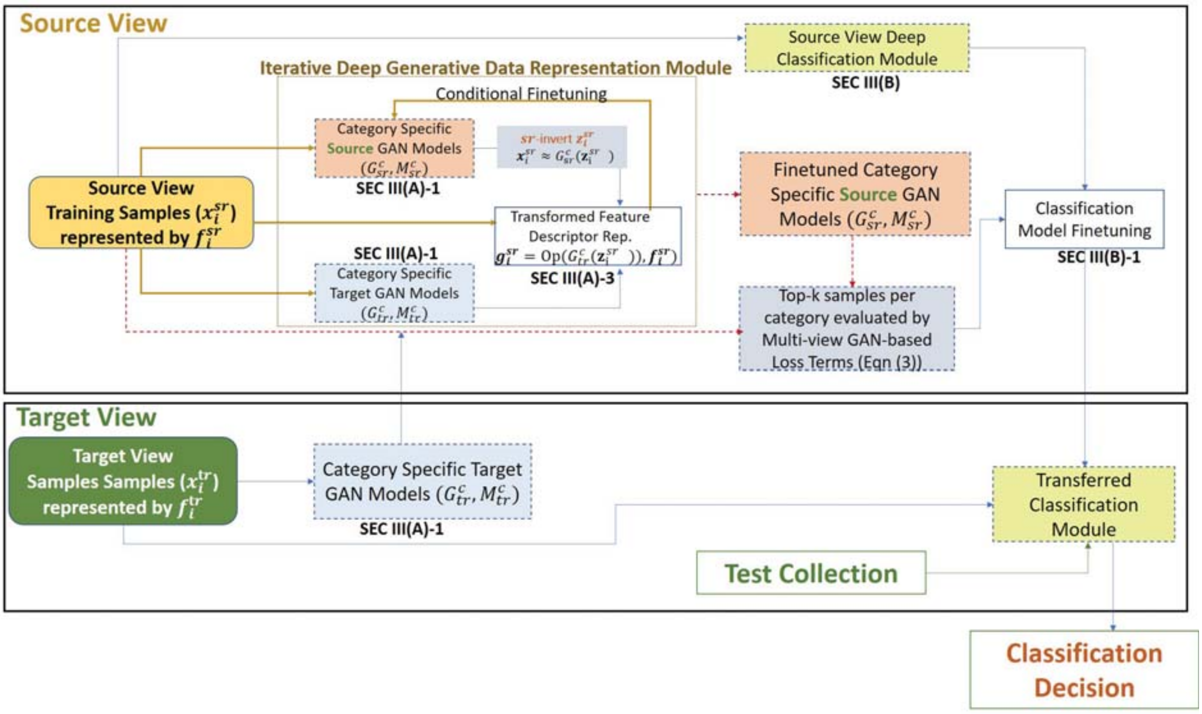

Fig. 2. Overview of the Proposed Generative Transfer Learning Method.

RNNs are a form of neural networks that display temporal behavior through the direct connections between individual layers. Given $\mathbf{x}_i^v$ and its d-dimensional temporal representative sequence $\mathbf{f}_i^v$, RNNs are designed in an iterative learning phase, to propagate historical information via a chain-like neural network architecture that simultaneously takes into consideration of the current input as well as the hidden state at each time step [39]. However, standard RNNs face a vanishing gradient problem and are unable to learn long-term dependencies as time steps become large. To address this challenge, Long Short-Term Memory (LSTM) has emerged as an efficient alternative that integrates the gating functions into its state dynamics [38]. In this work, stacked LSTM models are used as the feature extraction modules to obtain view-specific representations, which are then fed into a stack of Fully Connected (FC) layers for view-specific classification.

The proposed deep classification model uses three FC layers. While adding more layers makes the network more expressive, it becomes harder, at the same time, to train due to increased computational complexity, vanishing gradients, and model overfitting. In order to address the issue of overfitting, dropout-based regularization is employed, which randomly chooses a percentage κ of hidden units during the forward backpropagation step. This technique is used to cancel the contribution of some randomly chosen weight vectors in the network. A scaled version of the learned weight ($wt_{sc} = \kappa \cdot wt$) without applying the dropout, is used at the inference step. The standard back propagation algorithm is employed to update FC layer weight parameters in $\mathbf{W}^v$. More specifically, if F denotes the loss function defined as follows:

$$F(\mathbf{W}^v) = -\frac{\sum_c \sum_{i=1}^{|\mathcal{D}_v|} \mathbf{N}\{c_i = c\} \log\, p(c_i = c|\mathbf{x}_i^v; \mathbf{W}^v)}{|\mathcal{D}_v|} \tag{4}$$

where $\mathbf{N}\{.\}$ is the indicator function, $\mathbf{W}^v$ represents the CNN weight parameters, and the $prob(c_i^v = c|\mathbf{x}_i^v; \mathbf{W}^v)$ computes the probabilistic score of the sample $\mathbf{x}_i^v$ for the class c. The task is formulated as solving a minimization problem defined as: $\underset{\mathbf{W}^v}{min}\ F(\mathbf{W}^v)$. The activation of the last FC layer is fed into a softmax layer to obtain the probabilistic class membership scores.

Finally, prior to transferring, the pre-trained model $\mathbf{W}^{sr}$ learned at the source view sr, is fine-tuned with $\mathcal{D}_0$ to customize it for the target view tr that can produce a more discriminative classification performance at transfer.

IV. Experiments

A. Dataset

The performance of the proposed method is evaluated by analyzing the results of several experiments conducted for two types of tasks: (1) Light Curve Classification, and (2) Daily Activity Recognition. These testbed choices are influenced by their unique application-specific challenges, which make the corresponding classification task more complex.

In order to classify large collections of periodic light curves [40] generated from multiple independent surveys, one of the major challenges is the variance in the measurements frequently observed for similar light curves obtained from different telescopes. For our experiments, the first collection of light curves is taken from Catalina Real-Time Transient Survey (CRTS). A set of $\sim 50k$ periodic variables from the CRTS North (CRTS-N) survey builds one view-specific sub-collection of $\mathcal{D}$. Other view-specific sub-collections of $\mathcal{D}$ include $\sim 37k$ samples from CRTS-South [41], $\sim 15k$ samples from Palomar Transient Factory (PTF) [42], and $\sim 17k$ samples from the 2018 Gaia Data Release2 [43]. The fourth view-specific collection, Gaia Data Release2 (Gaia DR2), contains synthetically generated samples describing the celestial positions and the apparent brightness at three different bands: White-Light 'G'; Blue Prism ('BP'); and, Red Prism ('RP'). A separate set of multi-view experiments with this dataset uses subsets of each of these samples as a single view data collection; thus, the entire multi-view collection $\mathcal{D}$ contains the Gaia DR2 data at all three different bands. The released PTF and Gaia DR2 collection have 5 and 6 such classes, respectively with at least 500 representative samples to constitute the training collection. CRTS-N contains samples from 17 classes in the entire sample collection. Any class with fewer than 500 samples is added entirely to the training collection. There are 7 such small classes. The proposed method is tested on the remaining collection of 10 classes. In contrast to excluding samples from these 7 classes completely from the experimental settings, by adding them to the training collection we create a more challenging multiclass learning environment for the system. This also ensures learning of an effective model, also capable of classifying future samples from those minority classes for which number of samples was less in the present version of the data release, without needing a complete model update. CRTS-S uses the same asteroid-finding cadence as CRTS-N and also has an open filter. In order to evaluate the proposed transfer learning framework, the classes, which are common to CRTS-N and CRTS-S, are identified from the whole collection of CRTS-S to build the CRTS-S test collection.

The second dataset used in this work is the UCI Daily and Sports Activity Dataset [44], which contains motion sensor data of 19 daily and sports activities (listed as $A1, ..., A19$) such as sitting, standing, walking, running, jumping, etc. Each activity is performed by 8 subjects (4 males and 4 females) within the age range $[20, 30]$ for 5 minutes. The subjects were asked to perform the activities freely in their own styles, resulting in considerable intraclass variations observed within each activity type in terms of speed and amplitude, which created additional challenges for precise classification. This dataset contains data from 9 sensors, placed at each of 5 different units: torso, right arm, left arm, right leg, and left leg (45 sensors in total). Each sensor is calibrated to acquire data at a 25 Hz sampling frequency. The 5-minute time-series collected from each subject is divided into 5-second segments. Therefore, each segment has in total 125 samples, from which 50 random samples are chosen to define a single database segment. While there are $\binom{125}{50}$ such choices, we select 20 of them to build a sufficiently large set of representative samples per activity. Samples from similar activities (like *standing* and *standing in an elevator*) are treated as the same activity class. Therefore, the dataset has 11 classes: *Sitting* (A1), *Standing* (A2, A7), *Lying* (A3, A4), *Going up and down the staircase* (A5, A6), *Walking* (A8, A9), *Walking on a treadmill* (A10, A11) *Running* (A12), *Exercising* (A13, A14), *Cycling* (A15, A16), *Rowing* (A17) and *Jumping* (A18, A19). The activity samples obtained from 4 subjects are used to build the training collection, while the samples obtained from the remaining 4 subjects constitute the test collection.

B. Feature Processing

The proposed transfer learning framework is generic. Its efficacy is independent of feature selection. While more sophisticated features are expected to improve classification performance, in this work our primarily goal is to evaluate the framework. Therefore, for the reported experiments we define each light curve using a lower dimensional ($d = 8$) descriptor consisting of a small set of computationally efficient, statistical measures. We adopt the feature processing scheme proposed by Mahabal et al. [40] to represent each $\mathbf{x}_i$ in terms of an ordered sequence $\mathbf{f}_i$ with $T = 27$ time steps. For example, the light curves are represented in terms of brightness variations (expressed here in the traditional inverse logarithmic scale - Mags) as a function of time (expressed here in days - MJD). While the timestamps in these raw data are different for different light curves, the proposed feature processing step is initiated by computing the difference curve of length $\binom{p_i}{2}$ for each $\mathbf{x}_i$ of length p_i. For this dataset, we have $\mathbf{x}_i = [\mathbf{x}_i^{MAG}, \mathbf{x}_i^{MJD}]$ and its corresponding difference curve is represented as $d\mathbf{x}_i = [d\mathbf{x}_i^{MAG}, d\mathbf{x}_i^{MJD}]$. $d\mathbf{x}_i^{MJD}$ is aggregated within a binned window $\mathbf{B} = [\frac{1}{145}, \frac{2}{145}, \frac{3}{145}, \frac{4}{145}, \frac{1}{25}, \frac{2}{25}, \frac{3}{25}, 1.5, 2.5, 3.5, 4.5, 5.5, 7, 10, 20, 30, 60, 90, 120, 240, 480, 720, 960, 2000, 3000, 4000, 5000]$, which can be represented as:

$$\begin{aligned} d\mathbf{x}_{i,j}^{MAG} &= \left[k,\ s.\ t.\ d\mathbf{x}_i^{MAG}[k] \in \mathbf{B}_j\right] \\ d\mathbf{x}_{i,j}^{MJD} &= \left[d\mathbf{x}_i^{MJD}[k],\ s.\ t.\ k \in d\mathbf{x}_{i,j}^{MAG}\right] \end{aligned} \tag{5}$$

where $\mathbf{B}_j = [\mathbf{B}[j-1], \mathbf{B}[j]]$, a window ranged within two consecutive entries $\mathbf{B}[j-1]$ and $\mathbf{B}[j]$. For example, $\mathbf{B}_1 = [\frac{1}{145}, \frac{2}{145}]$ and $B_7 = [\frac{3}{25}, 1.5]$. Then, we compute $f_{i,j} = [f_{i,j}^1, f_{i,j}^2, f_{i,j}^3, f_{i,j}^4, f_{i,j}^5, f_{i,j}^6, f_{i,j}^7, \mathrm{f}_{i,j}^8]$, where 8 statistical measures including mean, min, max, standard deviation, range cumulative sum, kurtosis, skew and mean absolute deviations are respectively computed for $d\mathbf{x}_{i,j}^{MJD}$ This represents each $\mathbf{x}_i$ in terms of an ordered sequence $\mathbf{f}_i$ with $T = 27$ time steps. At each time step j, we have a $d = 8$ dimensional response $f_{i,j}$.

In the case of the UCI Daily Sports and Activity dataset analysis, each activity is represented in terms of 60 segments spanning over 5 minutes. Therefore, in this case, the difference curve computation followed by binning formalization (as described in Equation (5)) is not required for the feature processing. As such, we directly compute the statistical features describing each segment.

The stacked LSTM model representing a specific view $v \in \mathcal{V}$, has $L = 3$ LSTM layers. Each layer is followed by an immediately by a drop-out layer. The number of hidden units in each of the LSTM layers is set to be 128, while the drop-out ratio for each of the corresponding dropout layers is set to be 0.2. Each FC layer of the multi-view deep classification module, is designed with 128 units and defined with Rectified Linear Unit (ReLU) activation. In order to reduce the risk of overfitting, each FC layer is followed by a dropout layer with its dropout ratio fixed as $\kappa = 0.5$. The learning of each view-specific stacked LSTM model occurs with 80 epochs and 20% of the training samples are used for validation at every learning epoch. In case of UCI Daily Sports and Activity dataset analysis, each activity is represented in terms of 60 segments spanning over 5 minutes. We, compute the same statistical features describing each segment.

C. Implementation in a Distributed Learning Environment

The design of our distributed, multivariate deep learning framework for time-series data is motivated in part by the cost and associated challenges of moving data for analysis. More specifically, our objective is to promote frameworks for analysis that make previously distributed, fragmented, and un-shareable data more accessible without requiring the movement or direct exposure of raw data.

Traditional data fabrics require data to be moved to central locations for analysis. When data are large or distributed, transfer costs can be prohibitive; when data are un-shareable, analysis may be completely unfeasible. The proposed multi-variate deep transfer learning framework can efficiently utilize virtual information fabrics like that proposed by Talukder et al. [45]. Within such an information fabric, deep learning algorithms are encapsulated inside reusable, lightweight containers that can be seamlessly deployed and executed in a distributed fashion. Thus, to support transfer learning, analysis is distributed across the network (as enabled by the virtual information fabric) to the source and target data locations (rather than having to transfer source and target data). Only limited data are shared under this approach, the size of which, in this instance, is further optimized by our compact context descriptor representation.

The Virtual Information Fabric Infrastructure [45] (VIFI) allows researchers to perform analyses using a transfer learning workflow. Transfer learning workflows allow analysis across distributed datasets (possibly separated by great distances) to facilitate an iterative deep learning process on distributed data. Each workflow step executes inside a container that is transmitted to the location of a specified dataset. Under a transfer learning workflow, a model is learned on source data at one location, transferred to a target location, where it is refined on target data. We have implemented the proposed deep learning framework within VIFI environment to validate the efficacy of our distributed framework within a virtual information fabric.

Test Band \ Training Set Bands	CRTS-N	PTF	Proposed Framework sr=CRTS-N, tr=PTF	Proposed Framework sr=PTF, tr=CRTS-N
CRTS-N	0.764	0.73	0.751	0.784
PTF	0.644	0.848	0.836	0.845
Average	0.704	0.789	$\mathbf{0.794}$	0.815

TABLE I

COMPARATIVE PERFORMANCE OF THE PROPOSED TRANSFER CLASSIFICATION FRAMEWORK IN CRTS/PTF LIGHT CURVE COLLECTION AGAINST SINGLE VIEW CLASSIFIERS OVER ALL CLASSES OF CRTS-N AND PTF SURVEYS USING AVERAGE AUC SCORES AS THE PERFORMANCE METRIC.

Test Band \ Training Set Bands	CRTS-N	CRTS-S	Proposed Framework sr=CRTS-N, tr=CRTS-S	Proposed Framework sr=CRTS-S, tr=CRTS-N
CRTS-N	0.764	0.442	0.797	0.78
CRTS-S	0.412	0.856	0.877	0.84
Average	0.588	0.649	$\mathbf{0.837}$	0.81

TABLE II

COMPARATIVE PERFORMANCE OF THE PROPOSED TRANSFER CLASSIFICATION FRAMEWORK IN CRTS LIGHT CURVE COLLECTION AGAINST SINGLE VIEW CLASSIFIERS OVER ALL CLASSES OF CRTS-N AND CRTS-S SURVEYS USING AVERAGE AUC SCORES AS THE PERFORMANCE METRIC.

D. Results

1) Light Curve Classification: In order to handle the large variances in sample populations representing different classes, the same set of experiments is performed multiple times and the average performance details are reported in Table IV and Table V. In this work, we use the Receiver Operating Characteristic (ROC) curve for evaluation. Unlike overall accuracy scores for pairwise binary classification performances reported by Mahabal et. al. [40], which is dependent on one specific cut-point, the ROC curve investigates the performance of the multiclass classification task at a broader range, trying several cut-points to analyze the pattern of changes observed for the False Positive Rate with a varying True Positive Rate. The Area Under Curve (AUC) scores computed for these ROC curves, therefore, are found to be more insightful and useful as the evaluation metric.

The proposed cross-view transfer learning strategy was investigated in several different experimental settings. As seen in Tables I, II (please see Column 2 and 3), and Table III, while single-view classifiers perform well in identifying test samples from its corresponding view, they are not equivalently robust in classifying samples across views. However,

Test Band \ Training Set Bands	'G'	'BP'	'RP'
'G'	$\mathbf{0.91}$	0.87	0.86
'BP'	0.81	$\mathbf{0.89}$	0.84
'RP'	0.81	0.83	$\mathbf{0.88}$
Average	0.84	0.86	0.86

TABLE III

COMPARATIVE PERFORMANCE OF THE PROPOSED TRANSFER CLASSIFICATION FRAMEWORK IN GAIA DR2 LIGHT CURVE COLLECTION AGAINST SINGLE VIEW CLASSIFIERS OVER ALL CLASSES OF GAIA DR2 SURVEYS USING AVERAGE AUC SCORES AS THE PERFORMANCE METRIC.

Learning Configuration	v_s:Gaia-'RP' v_r: Gaia-'G'	v_s:Gaia-'BP' v_r: Gaia-'G'	v_{sr}:Gaia-'G' v_{tr}: Gaia-'RP'	v_{sr}:Gaia-'BP' v_{tr}: Gaia-'RP'	v_{sr}:Gaia-'G' v_{tr}: Gaia-'BP'	v_{sr}:Gaia-'RP' v_{tr}: Gaia-'BP'
AUC	0.895	0.88	0.82	0.851	0.82	0.879

TABLE IV

PERFORMANCE SUMMARY OF THE PROPOSED TRANFER LEARNING FRAMEWORK IN GAIA DR2 LIGHT CURVE COLLECTION.

Learning Configuration	sr: CRTS-N tr: CRTS-S[#]	sr: CRTS-N tr: CRTS-S[$\star$]	sr: CRTS-N tr: CRTS-S[¶]	sr: PTF tr: CRTS-S[#]	sr: PTF tr: CRTS-S[$\star$]	sr: PTF tr: CRTS-S[¶]
AUC	0.841	0.792	0.837	0.855	0.824	0.872
Learning Configuration	sr: CRTS-S tr: CRTS-N[#]	sr: CRTS-S tr: CRTS-N[$\star$]	sr: CRTS-S tr: CRTS-N[¶]	sr: PTF tr: CRTS-N[#]	sr: PTF tr: CRTS-N[$\star$]	sr: PTF tr: CRTS-N[¶]
AUC	0.806	0.787	0.811	0.82	0.806	0.815
Learning Configuration	sr: CRTS-N tr: PTF[#]	sr: CRTS-N tr: PTF[$\star$]	sr: CRTS-N tr: PTF[¶]	sr: CRTS-S tr: PTF[#]	sr: CRTS-S tr: PTF[$\star$]	sr: CRTS-S tr: PTF[¶]
AUC	0.81	0.808	0.794	0.836	0.811	0.857

TABLE V

PERFORMANCE SUMMARY OF THE PROPOSED TRANFER LEARNING FRAMEWORK IN CRTS-N, CRTS-S AND PTF LIGHT CURVE COLLECTION. EACH COLUMN REPORTS AN AVERAGE AUC SCORE OVER THE ENTIRE TEST COLLECTION OF sr AND tr FROM ALL CLASSES.[#] COLUMN (COLUMN 2 AND 5) REPRESENT A TEST SCENARIO IN WHICH THE BASELINE MODEL IS LEARNED AT THE SOURCE VIEW sr, THE MODEL IS SHARED WITH TARGET VIEW tr, AND FINE-TUNING IS PERFORMED BY THE ENTIRE TRAINING COLLECTION $\mathcal{D}_{tr}$ OF tr. [$\star$] COLUMN (COLUMN 3 AND 6) REPORT THE PERFORMANCE OF THE EXPERIMENTS, WHEN A RANDOM SUBSET (WITH k SAMPLES PER CATEGORY) OF $\mathcal{D}_{sr}$ IS SELECTED TO BUILD $\mathcal{D}_0$, WHICH WAS THEN USED FOR FINE-TUNING AT THE SOURCE VIEW PRIOR TO TRANSFER TO THE TARGET VIEW.[¶] COLUMN (COLUMN 4 AND 7) SHOW THE PERFORMANCE OF THE PROPOSED METHOD WHERE THE TARGET DATA PATTERN INFORMATION IS SHARED IN A COMPACT MODEL FORMAT WITH tr. AS DESCRIBED IN SECTION III-A4, A SELECTED SUBSET $\mathcal{D}_0 \subset \mathcal{D}_{sr}$ IS USED FOR CUSTOMIZE THE MODEL AT v_{tr} PRIOR TRANSFERRING.

a cross-view deep classifier learned utilizing the proposed generative transfer learning strategy is proved to be equally competitive in categorizing samples across different surveys (i.e. views). As observed in Table I, the proposed generative transfer learning strategy offers about 6% improvement (please compare column 2 with 4, and column 3 with 5) over the corresponding single baselines of CRTS-N and PTF. Another promising performance improvement was observed in Table II, where the proposed method ensures an average of 19% performance gain over its corresponding single view baselines.

As shown in the Table V, compared to traditional transfer learning approaches, the proposed framework utilizes a smaller source sample collection $\mathcal{D}_0 \subset \mathcal{D}_{sr}$ that closely resembles the class specific data patterns observed within v_{tr} for customizing the generic baseline source model as a part of pre-processing prior to transfer. In fact, as seen by comparing the column 2 with column 4 (and also column 5 with column 7), with a small set of selected samples in $\mathcal{D}_0$ (note that we have $|\mathcal{D}_0| = 0.1 * |\mathcal{D}_{sr}|$), the proposed method achieves a nearly equivalent average classification performance across multiple experimental settings. As observed, comparing the pairwise performances, as reported in columns $(3, 4)$ (and columns $(5, 6)$), we can see that the proposed method shows significant promise by reporting an impressive AUC score (which on average is 3% higher). We also observe by pairwise comparison of the corresponding rows of column 4 with column 7 that the proposed transfer framework presents a consistently robust generalization capacity of the learned models irrespective of the choice of the baseline source view. Figure 3 demonstrates the performance stability of the proposed generative transfer learning strategy against different values of $k \in [20, 30, 40, ..., 100]$.

Table III reports the performance of the single view classifier. While each of these classifiers shows good discriminative

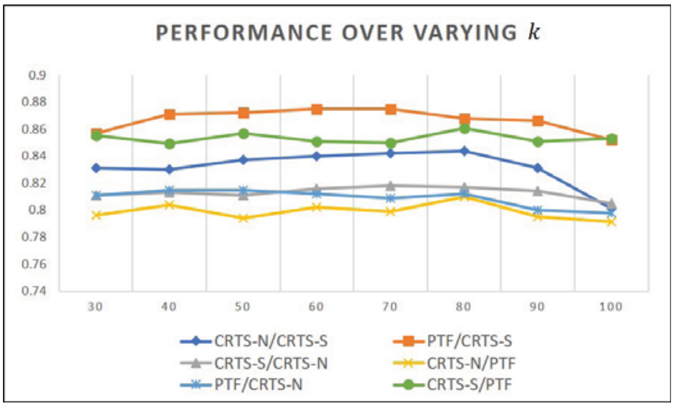

Fig. 3. Performance variance observed against different values of k in the range $[20, 30, 40, ..., 100]$. Each curve (legended as sr/tr) represents the results over the range of k values, where the proposed method uses the source view sr and transfers the learned classification module to the target view tr.

behavior in terms of classifying the samples from its own band, the performance deteriorates when identifying samples from other bands. In order to check the performance of the proposed deep transfer learning strategy using the Gaia DR2 collection, we adopt multiple experimental settings, where each of Gaia-'G', Gaia-'BP', and Gaia-'RP' collections, constitutes an independent view of the system. Table IV uses mean AUC scores (computed over all classes available in the test collection) to summarize the average performance. In order to minimize the effect of any bias due to the specific choice of training/test collection, the same set of experiments is performed 10 times and the average scores are reported in the table. As seen in Table IV, the proposed deep transfer learning framework remains consistently stable across multiple experimental settings and offers a nearly equivalent performance, while introducing

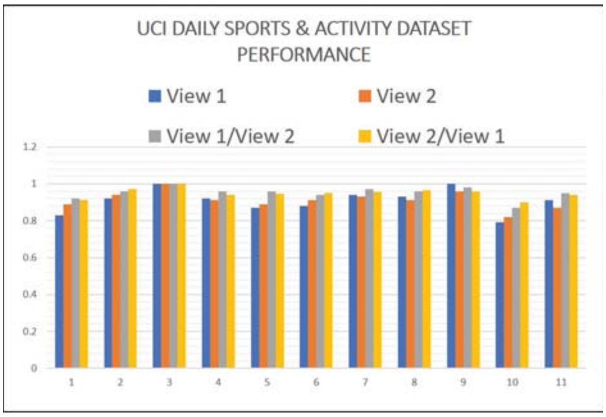

Fig. 4. Summarized Performance study of the proposed multi-view learning framework in UCI Daily and Sports Activity Dataset. The activities indexed along x-axis are as follows: 1. *Sitting* (A1), 2. *Standing* (A2, A7), 3. *Lying* (A3, A4), 4. *Going up and down the staircase* (A5, A6), 5. *Walking* (A8, A9), 6. *Walking on a treadmill* (A10, A11), 7. *Running* (A12), 8. *Exercising* (A13, A14), 9. *Cycling* (A15, A16), 10. *Rowing* (A17) and 11. *Jumping* (A18, A19). A bar legended as sr/tr reports the performance of the proposed method in a setting which uses the source view sr and transfers the learned classification module to the target view tr.

an effective fine-tuning scheme utilizing a small set of selected samples of v_{tr} that bear close resemblance with the data patterns at target view v_{sr}. In fact, the proposed method presents an efficient and precise transfer learning approach for those use-case settings, where data-sharing in a distributed environment can be a significant concern for multiple reasons like data volume, data security, etc.

2) Daily Activity and Sports Recognition: In order to investigate the performance of the proposed deep multi-view framework, we follow [46] to design a two-view experimental setting on the UCI Daily and Sports Activity dataset. Specifically the first 27 sensors on torso, right arm and left arm are treated as View-1, while the remaining 18 sensors on right leg and left leg as View-2. In this application setting, the activities are observed from two distinct views (i.e., two groups of sensors) simultaneously. The training set consists of 400 samples representing each activity type from 4 subjects. A test collection is built using the activity samples collected similarly from the other 4 subjects. The AUC scores summarized in the bar graph shown in Figure 4, prove the effectiveness of the proposed multi-view approach over the single-view classifiers by displaying an effective classification performance across all the classes, except a few specific activities like *Cycling*, for which single view proved to be more discriminative over multiple views. The same experiment is repeated 10 times by selecting a different set of 4 subjects in the training set that consists of a different set of random 50 samples per segment and subject for each activity. Table VI reports the average accuracy scores for comparing the performance of the proposed approach in the UCI Daily Sports and Activity dataset, against several state-of-the art results reported by Li et al. [46]. Accuracy is an evaluation metric that computes the ratio of the correct predictions over all the predictions made by a classifier. By demonstrating an improvement of around 4% in the average accuracy, the proposed method shows significant promise compared to the state-of-the-art methods in classifying the activity patterns in this dataset. Also, it is important to note that, in contrast to [46], where the authors learn the optimized latent subspace for designing a discriminative representative by utilizing the entire multi-view data repositories, the proposed method enables learning within a more distributed context, which makes it more easily adaptable for several practical application settings.

Method	CCA	MvDA	MDBP	Proposed Method
Accuracy	0.601	0.859	0.913	**0.953**

TABLE VI
COMPARATIVE STUDY ON UCI DAILY SPORTS AND ACTIVITY DATASET, WHERE THE PROPOSED METHOD IS COMPARED AGAINST CCA [47] MVDA [48], AND MDBP [46].

V. CONCLUSION

In this paper, we present an effective, multi-view, multivariate deep transfer learning framework for time-series data. The proposed framework introduces the concept of pairwise view-similarity by identifying a smaller set of source data samples that closely resemble the specific target data under consideration. The proposed two-phased learning model learns a generic baseline, which is later fine-tuned utilizing a selected view-specific similarity representative collection for expediting the task of knowledge sharing in a distributed environment. A conditional generative adversarial architecture combined with deep classification framework minimizes the gap between the source and target data. Our proposed generative, deep transfer learning framework demonstrates an improved classification performance of 4% over the state-of-the-art methods, while enabling a more efficient distributed learning capacity (e.g., the ability to operate more efficiently within virtual information fabrics) and a significant capacity for generalization to a wide range of applications

ACKNOWLEDGMENT

Funding for this research was provided by the National Science Foundations (NSF) Data Infrastructure Building Blocks (DIBBs) Progam under award $\#1640818$.

REFERENCES

[1] S. Seto, W. Zhang, and Y. Zhou, "Multivariate time series classification using dynamic time warping template selection for human activity recognition," *2015 IEEE Symposium Series on Computational Intelligence*, pp. 1399–1406, 2015.
[2] P.-F. Marteau and S. Gibet, "On recursive edit distance kernels with application to time series classification," *IEEE transactions on neural networks and learning systems*, vol. 26, June 2015.
[3] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Min. Knowl. Discov.*, vol. 29, pp. 565–592, May 2015.
[4] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," *SIGMOD Rec.*, vol. 45, pp. 69–76, June 2016.
[5] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," *CoRR*, vol. abs/1611.06455, 2016.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[7] O. Mogren, "C-RNN-GAN: continuous recurrent neural networks with adversarial training," *CoRR*, vol. abs/1611.09904, 2016.

[8] D. Li, D. Chen, J. Goh, and S. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," *CoRR*, vol. abs/1809.04758, 2018.

[9] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *CoRR*, vol. abs/1702.05374, 2017.

[10] S. D. Bhattacharjee and A. Talukder, "Graph clustering for weapon discharge event detection and tracking in infrared imagery using deep features," 2017.

[11] S. Das Bhattacharjee, B. V. Balantrapu, W. Tolone, and A. Talukder, "Identifying extremism in social media with multi-view context-aware subset optimization," in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 3638–3647, 2017.

[12] S. Das Bhattacharjee, A. Talukder, and B. V. Balantrapu, "Active learning based news veracity detection with feature weighting and deep-shallow fusion," in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 556–565, 2017.

[13] S. Das Bhattacharjee, V. S. Paranjpe, and W. Tolone, "Identifying malicious social media contents using multi-view context-aware active learning," in *Future Generation Computer Systems, Elsevier*, 2017.

[14] S. Das Bhattacharjee, J. Yuan, Z. Jiaqi, and Y. Tan, "Context-aware graph-based analysis for detecting anomalous activities," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1021–1026, 2017.

[15] S. Das Bhattacharjee, W. J. Tolone, M. Elshambakey, I. Cho, A. Mahabal, and G. Djorgovski, "Context-aware deep sequence learning with multi-view factor pooling for time series classification," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 959–966, 2018.

[16] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," *CoRR*, vol. abs/1701.01370, 2017.

[17] Y. Yu, Z. Gong, P. Zhong, and J. Shan, "Unsupervised representation learning with deep convolutional neural network for remote sensing images," in *Image and Graphics* (Y. Zhao, X. Kong, and D. Taubman, eds.), (Cham), pp. 97–108, Springer International Publishing, 2017.

[18] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016.

[19] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the Computer Vision and Pattern Recognition*, June 2018.

[20] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete electronic health records using generative adversarial networks," *CoRR*, vol. abs/1703.06490, 2017.

[21] *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[22] G. Antipov, M. Baccouche, and J. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 2089–2093, Sep. 2017.

[23] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1335–1344, 2018.

[24] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *CoRR*, vol. abs/1701.06547, 2017.

[25] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *Web-Age Information Management* (F. Li, G. Li, S.-w. Hwang, B. Yao, and Z. Zhang, eds.), pp. 298–310, Springer International Publishing, 2014.

[26] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *CoRR*, vol. abs/1801.04503, 2018.

[27] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller, "Data augmentation using synthetic data for time series classification with deep residual networks," *CoRR*, vol. abs/1808.02455, 2018.

[28] Y. Geng and X. Luo, "Cost-sensitive convolution based neural networks for imbalanced time-series classification," *CoRR*, vol. abs/1801.04396, 2018.

[29] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, pp. 11–24, 2014.

[30] A. Ziat, E. Delasalles, L. Denoyer, and P. Gallinari, "Spatio-temporal neural networks for space-time series forecasting and relations discovery," in *IEEE International Conference on Data Mining (ICDM)*, pp. 705–714, 2017.

[31] H. Nweke, T. Wah, M. Ali Al-garadi, and U. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, vol. 105, 04 2018.

[32] Z. Che, Y. Cheng, S. Zhai, Z. Sun, and Y. Liu, "Boosting deep learning risk prediction with generative adversarial networks for electronic health records," *CoRR*, vol. abs/1709.01648, 2017.

[33] M. Baktashmotlagh, M. Faraki, T. Drummond, and M. Salzmann, "Learning factorized representations for open-set domain adaptation," *CoRR*, vol. abs/1805.12277, 2018.

[34] M. Long and J. Wang, "Learning transferable features with deep adaptation networks," *CoRR*, vol. abs/1502.02791, 2015.

[35] H. Sivaraks and C. Ratanamahatana, "Robust and accurate anomaly detection in ecg artifacts using time series motif discovery," in *Comp. Math. Methods in Medicine*, 2015.

[36] "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83 – 95, 2016.

[37] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," *CoRR*, vol. abs/1805.03908, 2018.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.

[39] R. Pascanu, Ç. Gülçehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *CoRR*, vol. abs/1312.6026, 2013.

[40] A. Mahabal, K. Sheth, F. Gieseke, A. Pai, S. G. Djorgovski, A. Drake, M. Graham, and the CSS/CRTS/PTF Collaboration, "Deep-learnt classification of light curves," *IEEE Symposium Series on Computational Intelligence (SSCI)*, vol. arXiv:1709.06257, 2017.

[41] A. J. Drake, S. G. Djorgovski, M. Catelan, M. J. Graham, A. A. Mahabal, S. Larson, E. Christensen, G. Torrealba, E. Beshore, R. H. McNaught, G. Garradd, V. Belokurov, and S. E. Koposov, "The catalina surveys southern periodic variable star catalogue," *Monthly Notices of the Royal Astronomical Society*, vol. 469, no. 3, pp. 3688–3712, 2017.

[42] N. Law, S. Kulkarni, R. Dekany, and E. e. Ofek, "The palomar transient factory: System overview, performance, and first results," *PUBLICATIONS OF THE ASTRONOMICAL SOCIETY OF THE PACIFIC*, vol. 121, no. 886, pp. 1395–1408, 2009.

[43] A. Brown, A. Vallenari, T. Prusti, J. de Bruijne, C. Babusiaux, and C. Bailer-Jones, "Gaia data release 2. summary of the contents and survey properties," *AA Special Issue on Gaia DR2*.

[44] B. Barshan and M. C. YÃ¼ksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Computer Journal*, vol. 57, no. 11, pp. 1649–1667, 2014.

[45] A. Talukder, M. Elshambakey, S. Wadkar, H. Lee, L. Cinquini, S. Schlueter, I. Cho, W. Dou, and D. J. Crichton, "Vifi: Virtual information fabric infrastructure for data-driven discoveries from distributed earth science data," in *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation*, 2017.

[46] S. Li, Y. Li, and Y. Fu, "Multi-view time series classification: A discriminative bilinear projection approach," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pp. 989–998, 2016.

[47] H. Hotelling, *Relations Between Two Sets of Variates*, pp. 162–190. New York, NY: Springer New York, 1992.

[48] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, pp. 188–194, Jan. 2016.