IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 May 2024, accepted 3 June 2024, date of publication 11 June 2024, date of current version 3 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3412653

RESEARCH ARTICLE

Entropy and Memory Aware Active Transfer Learning in Smart Sensing Systems

UMAIR ALAM [1], **ASMA AHMAD FARHAN** [1], **SUMMRINA KANWAL** [2], **AND NASSER ALLHEEIB** [3]

[1]FAST School of Computing, National University of Computer and Emerging Sciences, Lahore 54770, Pakistan
[2]School of Electrical Engineering and Computer Science, KTH University, 114 28 Stockholm, Sweden
[3]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

Corresponding author: Nasser Allheeib (nallheeib@ksu.edu.sa)

This work was supported in part by the Researcher Supporting Project, King Saud University, Riyadh, Saudi Arabia, under Grant RSPD2023R609.

ABSTRACT Automated Human Activity Recognition (HAR) stems from the requirement to seamlessly integrate technology into daily life, to enhance user experience, improve healthcare, provide improved operations, ensure safety, deliver data-driven insights, and address various real-world challenges. However, unscripted Human activity faces challenges that must be understood, and require advances in sensor technology and machine learning models. This paper presents an Active Transfer Learning (ATL) approach for real-time HAR using mobile sensor data. Unlike traditional methods, our approach accounts for both the physical and habitual constraints of individuals. Existing works make an unrealistic assumption of an omniscient oracle while using the same datasets for both training and testing of the models, which makes them impractical for industry requirements. Our proposed approach addresses challenges in existing HAR algorithms, proposing a methodology to adapt models to the real-world datasets while training and testing on cross datasets. We have tailored an existing Entropy and Memory Maximization algorithm to work in a real-time environment while considering user constraints. Primarily trained in a well-labeled controlled environment dataset, we introduce noise injection to prevent the model from overfitting and enhance its generalization for scarcely labeled real-world datasets. Evaluations on publicly available datasets demonstrate our approach achieves 80% - 90% accuracy, outperforming the base algorithm accuracy of 12% - 14%. Importantly, our proposed technique outperforms with limited labeled data, making it valuable for real-time scenarios where labeling is sparse. This research advances HAR in real-world settings, offering improved accuracy and adaptability.

INDEX TERMS Active transfer learning (ATL), Human activity recognition (HAR), Internet of Things (IoT), Machine learning (ML), Ubiquitous computing, Wearable sensor devices.

I. INTRODUCTION

The world is transitioning towards the automation of almost every domain of life by developing smart homes [1], smart grids [2], autonomous vehicles [3], smart security surveillance [4], intensive care units (ICU) and healthcare systems [5], [6], [7]. Hence, the world is converging into smart cities [8], where everything would be controlled and operated by automated computer systems. All these systems are required to be efficient in human-computer interaction to facilitate the people and to ensure the safety of the users. These systems require the ability to recognize and classify human activities to operate efficiently. For instance, if a driver-less vehicle is unable to identify the activities of the pedestrians around it, then it would be very dangerous for the passengers as well as the other people on the road. Therefore, for the successful deployment of automated systems, their ability of Human Activity Recognition (HAR) is indispensable. To achieve the ability of HAR, these systems depend on the sensor devices to record the activity data and then use Machine Learning (ML) models to recognize and classify different activities from that received dataset.

The associate editor coordinating the review of this manuscript and approving it for publication was Angelo Trotta.

In recent years, the Internet of Things (IoT) sector has significantly evolved, which has resulted in a vast application of sensor devices and ubiquitous computing in various domains of life [9]. As these sensor devices have become an important part of the HAR applications, together they form the concept of ubiquitous computing, where it is very important to provide the real-time identification and classification of the data being recorded. We can understand the importance of HAR from the fact that these sensor devices, connected as pervasive computing, and these ML models of activity recognition have become part of almost all the major domains. For instance, in automated manufacturing units [10] of Industry 4.0 for decision making by ML, military applications [11] for real-time surveillance using automated HAR [12], context recognition [13], indoor navigation [14] as well as navigation systems for self-driving vehicles using sensor devices and automated HAR [15], attention management systems [16], smart homes [17], smart cities [18], [19], and intensive health care [20] domains. As we can see the applications of HAR by using sensor devices and automated systems in our lives are rapidly increasing, the need for ML models that can provide efficient results in real-time is also very important to enable the successful deployment of these systems in practical life.

These sensor devices [21] have enabled us to record different aspects of HAR [22] by using different data collection methods [23], mainly through the images/videos [24], ambient sensors [25], and wearable sensors [26]. Each of these types then further has various categories of receptors to record the activity data, such as, the ambient sensors may record the radar waves [27], [28], infrared motion [29], temperature change [30], and/or RFID based [31] data from the surroundings. Similarly, wearable sensors may collect data from accelerometers [32], magnetometers [33], and gyroscopic measures [34] to record human activities. The development of various types of wearable sensor devices has paved the way for the practical implementation of IoT network-based smart sensing devices. One of the main objectives of these devices is to improve human well-being [35], and for this purpose, these devices record a huge data regarding human activities.

While these sensor devices have partially solved the problem by gathering real-time data on daily activities, the issue lies in the lack of proper labels for the observations recorded through these devices in datasets collected in the natural setting. And for the analysis of these data, and detection of important characteristics regarding human activities, it requires trained models that can recognize the individual activity performed. As the collection of data is growing rapidly with the help of increasingly available sensing devices [36], large datasets are now available. However, for the training of the ML models, properly labeled datasets are required, which is costly as well as effort-intensive to collect perfectly labeled real-time datasets.

As it is very expensive and effort-intensive to collect labeled in-the-wild datasets on a very large scale, spanning different types of activities. There are always some pros and cons for each dataset type. The collection of data in a controlled environment is comparatively less effort-intensive and cost-efficient, but this type of data is based on certain unrealistic assumptions that are made to collect useful data. These unrealistic assumptions are related to the user/oracle, such as the perfect memory of the users, their ability to respond immediately to the queries correctly, and the user's performance of all the activities is always for equal duration and in equal number of iterations. However, these assumptions are not realistic when dealing with real-life events in day-to-day routine activities. There seems no feasible solution to this problem where neither the collection of perfectly labeled real-time datasets is practically affordable, nor the lab datasets that are produced in control environments are useful for training the efficient models that can operate in real-time. Hence, there is a need to optimize the ML models for performance improvement, when applied in real life, by making them capable of adapting themselves to new environments.

In ML, there is a technique called Active Learning (AL) in which the labels are queried by the user, and based on those responses the ML models are trained. There is another technique called Transfer Learning (TL) in which an ML model, already trained on a similar dataset, is used to predict the labels, and based on those labels the model retrains/fits itself. This technique does not require responses from the user. A hybrid technique based on both AL and TL also exists, that queries only specific observations from the user that can provide the maximum gain, and for the rest of the dataset the model adapts itself based on the technique of the TL approach. We have used this Active Transfer Learning (ATL) technique in our methodology. In this technique, the concept of entropy is used, which means lack of information, disorder, or randomness. Based on the entropy level, the observations are sorted and then the observation that has maximum entropy is selected to ask the label from the user, as this particular observation would provide the maximum information gain to the model when the correct label would be provided by the user. After one query from the user, the model retrains itself and predicts the labels, as well as the probability, for the remaining unlabeled dataset. Based on that probability, again the entropy calculation is repeated until the model perfectly predicts the labels and no observation is below the threshold to be selected for the query. In this way, the model reduces the large number of queries that would not require to be asked from the user. In our proposed methodology, we have introduced a technique in which the ML model is first trained on the lab dataset by using the information gained along with the AL approach. Then the model is used to predict the labels for unlabeled in-the-wild dataset. For the next subject's activity recognition, the model is trained using the new lab dataset as well as the previous subject's in-the-wild dataset.

This allows the model to gradually adapt itself according to the in-the-wild dataset.

A real user does not perform the activities like a machine that always performs activities on a pre-defined pattern. Due to this reason, the models do not perform well when trained on controlled datasets and tested in the wild environment. The proposed model in this work aims to overcome these unrealistic assumptions of physical and behavioral constraints by adding the effect of realism in datasets of controlled environments, hence, resulting in better performance in the real-time execution of the algorithm. This model is applied and has better performance to correctly identify human activities in scenarios where either we do not have an in-the-wild dataset to train the model first, or the users have very versatile patterns of performing activities based on their routines. The proposed model can learn/train from the controlled dataset in the beginning and can adapt itself later by repeatedly training itself based on each person's individual habit and physical ability to perform the activities. In this way, the model can have better results when applied in the real-time scenarios.

To overcome the above-discussed issues, we have tailored the existing Entropy and Memory Maximization (EMMA) [37] approach to make it work with the supervised but incrementally informed datasets collected in the wild environment. By implementing our proposed methodology, we have:

- Enabled the model to partially fit itself repeatedly as many times as it would be required during the iterations of the algorithm.
- Replaced the existing Support Vector Machine (SVM) [38] model with a modified version of incremental SVM model. As the incremental SVM model does not exist itself, we have used the Stochastic Gradient Descent (SGD) [39] model with the log loss function which allows us to achieve the characteristics of an incremental SVM model.
- The SGD model allowed us to have probability calculations as a byproduct when the log loss function is used. This probability is later used for calculating the entropy of the dataset observations.
- The overall performance of the model is improved by resolving the inconsistency issue regarding the probability calculation caused by the separate probability function of the SVM model.
- We have eliminated the requirement of a complete preloaded target dataset. As the model can now partially fit itself, it does not require the target dataset to be available and loaded before the execution of the algorithm.
- Further eliminated the requirement of a pre-learned expert model. For this purpose, the existing AL approach is modified into the ATL [40] technique, which has enabled our model to adapt itself according to the target dataset characteristics while incrementally training/fitting the model.
- This hybrid technique of ATL allows the model to get the benefits of both approaches and hence provides a better accuracy performance by adapting itself according to the patterns/variations of values in the target dataset.

Besides being able to work in a real-time environment, the model is also able to acknowledge and reflect the realistic oracle's physical and habitual constraints of irregular activity patterns during initial training from the source dataset collected in the controlled environment. For this purpose, the technique of adding the noise [41] is used which allows generalization while training the model on lab-generated datasets. Hence, our proposed model has maintained performance accuracy between 80% to 90% when trained on the controlled environment dataset and then applied to the real-time dataset collected in the wild environment.

The rest of the paper is organized into the following sections. Section-II is the literature review of the related work that we have explored during our research and also discusses the problem statement that is dealt with in our work. Section-III is covers our proposed methodology. The section-IV discusses the datasets used and implementation details of the proposed techniques. Section-V is the results and discussion section which explains the results we have achieved through our proposed approaches. At last section-VI discusses the conclusion and possible suggestions for future research.

II. LITERATURE REVIEW

There are various approaches that have been proposed regarding the efficient and optimized algorithms for HAR systems based on various dataset types and environmental circumstances. Activity recognition can be studied from three different aspects which include approaches, label availability, and feature space. There are various approaches to achieving the activity recognition objective, but the most popular, regarding the datasets that we intend to use, are the AL [37] and TL [42] approaches. And then there is a third category of ATL [43] that combines the characteristics of both and achieves much better performance results in terms of accuracy and reduced annotation efforts.

Furthermore, we can categorize the activity recognition process based on the availability of the labels for dataset instances. The availability or unavailability of labels in the source dataset is termed as Supervised or Unsupervised activity recognition and the availability or unavailability of labels in the target dataset is termed as Informed or Uninformed activity recognition respectively [42]. In this way, it can be categorized into four types as Informed Supervised, Informed Unsupervised, Uninformed Supervised, and Uninformed Unsupervised activity recognition. And lastly, we can differentiate the activity recognition based on the feature space, i.e., the type of dataset being used for activity recognition. The most common types are Video or Image frames, Ambient Sensors, and Wearable Sensors' generated datasets. Video stream is a completely different dataset type that requires algorithms for image frame processing and identifying the objects in those frames. On the other hand, ambient sensors and wearable sensors generate numeric datasets based on the sensor readings in different intervals. The ambient sensors are fixed in a place, like inside a building or at an area where the activity is to be observed. They only record the information when a person comes within the range of the sensor., Wearable sensors are always carried by the person in the form of a smartwatch or smartphone and hence keep recording the user's activity all the time. Now as we have learned about different aspects by which activity recognition can be categorized, we can mention here the particular domain of this study being conducted. The area of the paper is ATL based Informed Supervised Activity Recognition using Wearable Sensor datasets. In this section, an overview of some of the major approaches, related to the one under discussion, is provided. A comparison matrix of the major works is also presented in Table. 1 below, which provides an overall observation of the related work.

A. RELATED WORK

In a work proposed by [37], the authors have dealt with the cognitive constraints of human beings while considering the compliance of the oracle with the activity recognition system. They have proposed the EMMA algorithm for activity recognition based on dataset received from wearable sensing devices. EMMA deals with the cognitive constraints of the oracle while performing AL by repeatedly asking queries from the oracle and labeling the dataset. This process keeps repeating until the assumed budget ends. This budget represents the cognitive constraints as well as the physical limit of an oracle and hence imitates the real human's capacity to correctly answer the number of queries in real life. Furthermore, the memory strength of the users is reflected in the process of selecting the queries for labeling. Although this parameter is also set as presumed values representing different memory strengths of the people, it reflects the effect of memory constraints of an oracle. The authors in this paper have used three datasets known as HART [44], DAS [45], [46], [47], and AReM [48] which are publicly available. From their model, they have achieved 21% to 91% accuracy range for activity recognition based on the memory strength and other variable factors of the model. They report an average of 13.5% higher accuracy by their proposed model. However, there are limitations in the working of the algorithm, as it never completely implies the characteristics of the dataset collected in the wild. The duration of activities is fixed as it works on the dataset collected in the controlled environment, which is not the same for natural datasets. Moreover, the model used in this paper for classification is the SVM which operates as a supervised learning method. But this model does not retrain itself if more labeled data is available and works only if all the datasets are available in advance, hence, it does not operate in a real-time environment.

The work [49] has proposed an OptiMapper algorithm for activity recognition systems in mobile sensing systems. The OptiMapper algorithm classifies different activities of the oracle in real time based on the technique of TL. It requires a source dataset to train the model and then keeps updating the model by learning from the target dataset while labeling the target data in each iteration. Through this repeated process the model is retrained from the updated labels of the target dataset and hence transforms the model into a better version. As the labeling and training occur as a repeated process, the algorithm can work in the partially available data. Hence, this algorithm achieves the ability to work in a real-time environment where data is not completely available in hand, and can train the model from the partially available dataset in an iterative process. In this study, the authors have used three datasets known as DAS [46], PAMAP2 [50], and SmartSock [51] which are wearable sensor datasets. Through their proposed model they report up to 22.5% improvement in the accuracy for activity recognition. Although it seems that the algorithm achieves a better performance theoretically, in reality, the constraints of the oracle are not dealt with in the working of the algorithm. In a controlled environment scenario, the performance of the algorithm seems promising. But as in a natural setting, the datasets collected pose certain human physical and cognitive constraints, as discussed earlier, and these aspects of oracle constraints must be considered while designing a model for activity recognition.

Nawal et al. [52] have presented an incremental SVM based algorithm for HAR. They have tried to solve the problem of large-scale dataset processing in real-time while training and deploying the model on target datasets. For this purpose, they have introduced two different versions of incremental SVM in the paper. At first, the last reported event of sensors is used in the algorithm for determining certain aspects of the dataset while the incremental process takes place. In the second version, the clustering-based approach is used to boost the learning performance of the algorithm. In this version of the incremental SVM, the relationships between the clusters of datasets and SVM chunks are established, hence increasing the learning performance. To implement this proposed model, they have used Aruba [53] and Tulum [54] datasets which are available publicly. According to them, their model is 5 to 9 times faster than the existing models and has achieved at least 5% accuracy improvement. This approach does work in a real-time environment but lacks the ability to deal with datasets collected in the wild as it never takes into consideration the aspects of natural settings.

In this paper, Hasan et al. [57] have implemented a different approach toward event recognition while taking into account the contextual aspects of the target object. The major difference between this paper from our domain of research is the type of dataset being used for event recognition, as in this paper image frames are used for event recognition. The thing to be discussed here is the method

TABLE 1. Comparison matrix of literature review papers.

Paper	Approach	Class	Supervised (Source)	Informed (Target)	AL	TL	Real Time	Dataset Type	Datasets	Limitations / Differences
Ashari et al. [37]	SVM – EMMA	AR	Yes	Yes	Yes	No	No	Wearable Sensor	HART [44], DAS [45]–[47], AReM [48]	Does not operates on in-the-wild data. No cross dataset training/testing.
Fallahzadeh et al. [49]	OptiMapper	AR	No	No	No	Yes	Yes	Wearable Sensor	DAS [46], PAMAP2 [50], Smart Sock [51]	Does not considers human constraints. Controlled dataset. No cross dataset training/testing.
Nawal et al. [52]	Incremental SVM	AR	Yes	No	No	Yes	Yes	Ambient Sensor	Aruba [53], Tulum [54]	Dataset of fixed area. No cross dataset training/testing.
Alam et al. [55]	HALT	AR	No	No	Yes	Yes	No	Ambient Sensor	RCC [55], TU Darmstadt [56]	Dataset of fixed area. Not applicable in real time. No cross dataset training/testing.
Hasan et al. [57]	Personalized CRF	Event Recog.	Yes	Partially	Yes	Yes	Yes	Video	UCF50 [58], VIRAT [59], UCLA [60], AVA [61]	Applied on video data. No cross dataset training/testing.
Hasan et al. [62]	CRF	AR	Yes	Partially	Yes	Yes	Yes	Video	UCF50 [58], VIRAT [59], UCLA [60], MPII [63]	Applied on video data. No cross dataset training/testing.
Wang et al. [43]	TL	Image Classific.	Yes	Partially	Yes	Yes	No	Image	Riesling [43], Traminette [43]	Not AR, so human constraints are not studied in it. Not applicable in real time.
Qu et al. [64]	PCTL - SAS	Image Classific.	Yes	No	No	Yes	Yes	Image	PaviaU - PaviaC [65], Houston [66], [67]	Not AR, so human constraints are not studied in it.
Ahmed et al. [68]	Semantic Vector	Behavior Classific	Yes	Partially	Yes	No	No	Text	Online Forum [69]	Not AR, so human constraints are not studied in it. Not applicable in real time. No cross dataset training/testing.

Active Learning (AL), Transfer Learning (TL), Support Vector Machine (SVM), Hierarchical Active Transfer Learning (HALT), Conditional Random Field (CRF), Physically Constrained Transfer Learning (PCTL), Shared Abundance Space (SAS), Activity Recognition (AR).

used for this purpose, i.e., AL for query selection. They have proposed a technique of the Conditional Random Field (CRF) model for this purpose but took AL into account while making the queries and training the model. Instead of burdening the oracle to answer a large number of queries, this algorithm predicts a label for the selected activity for the query, and based on the oracle's response it builds a teacher that allows it to skip a certain number of queries based on the weights it assigns to its predicted label in case of correct predictions. Hence, it reduces a great burden by decreasing the number of queries to be asked from the oracle. They have used six different datasets for their experiment out of which, AVA [61], VIRAT [59], and MPII [63] are mainly discussed. Not providing an average measure, they have achieved various accuracy rates on different datasets, mostly achieving superior performance rates.

In another work [62], HAR from image frames obtained from video data is explored, which is a different type of dataset than ours of sensor-based datasets. However, it aims to reduce the oracle's manual efforts for annotating the queries, using the CRF algorithm. For this purpose, the authors in this paper have designed an algorithm that takes into account the contextual information of the object and hence develops the interrelationship between the knowledge of the context and the object, and tries to recognize the activity of the oracle. In this way, it resembles our work of calculating the entropy to determine which queries to raise for annotation. Here in this paper, this concept is discussed as maximum gain and tries to determine queries that can help obtain maximum gain. The authors in this paper have used AVA [61], VIRAT [59], UCLA [60], and UCF50 [58] datasets to check their model's performance, and have reported a significantly less amount of effort required for labeling and have superior performance rate. This type of implementation is not done for the datasets collected from the smart sensor devices to recognize human activity.

There is another category of dealing with the activity recognition process. In this paper [55], the authors have devised an algorithm that even does not require the labels of the source dataset to train the model. It can train the model from an unseen dataset and use that model to classify the activities on another dataset. For this purpose, it uses novel techniques to classify data in the source dataset to build the model from that classified dataset. It achieves this objective by using a network clustering method used in bipartite graphs to establish boundaries between certain types of data features. They have tested their model on two smart home systems and have reported it as achieving a significant performance improvement. Although, this technique of dealing with this problem is novel, but it also has limitations as it requires all the data to be available in advance to establish relationships between two datasets based on their resembling features. This logic of the algorithm makes it impossible for the algorithm to work in a real-time environment where data would be partially available over time.

The authors in [43] have dealt with the large amount of data required for the case of TL technique for activity recognition. This proposes the technique of using both AL and TL algorithms merged to avail benefits from vast datasets while improving the performance of the algorithm by making choices for the data to be presented for query from the user. In this way, the algorithm takes benefit of the large dataset while keeping the performance on a better scale. The authors of this study have used two image datasets, one consisting of riesling grapes and the other containing traminette grapes images, for evaluating their model and have found overall better performance for activity recognition using their proposed image classification model. In this approach, the author has only considered one problem to be handled while all the other aspects discussed above regarding the constraints and environment adaptation for activity recognition are not discussed in the paper.

In a survey paper [42], an in-depth survey on algorithms and techniques used in the TL approach is discussed. It provides details about the major categories of the TL approach and how those categories are helpful depending on certain environmental and data availability conditions. Largely it divides TL algorithms into four major categories based on the modality of the sensor, types of environments, availability of data, and type of information transferred. There are further sub-divisions of the algorithms in each category, e.g., in the category of modality of sensors, the algorithms can be differentiated based on the type of sensor they work with i.e., the video sequences, wearable sensors, and ambient sensors. And if talking about the algorithms that work with the wearable sensors, which is also the domain of algorithm that we are going to work on, they have further divisions based on the availability of the data, as well as the type of environment where these algorithms can work also differs. However, the main part of the paper is the discussion on limitations, which is described to be a great constraint for the algorithms to be generalized. Because these algorithms cannot be generalized, or there exists no algorithm that can provide good performance for different scenarios, there is a great need for improvements in the existing algorithms, so that they can achieve better results in real-life implementation where these factors may change and the models should be able to adapt to the new scenarios.

In this paper [70] the SVM classifier is used for activity recognition from video. The objective of the paper is to reduce the long training time and the large size of the feature vector. The model proposed here achieves its objectives by using the technique of SVM to provide the labels for the activities and learn the model from fewer number of examples. Weizmann [71] and UIUC1 [72] are the two datasets used in this research paper to train and test the proposed model. According to the authors, the overall performance of the model is slightly better than the existing algorithms, but if observed with minute details, it is mostly better in the case of the Weizmann dataset and had at par or slightly lower performance in the case of the UIUC1 dataset. The results they have provided are not of the performance percentage, but of the rate of recognition of the activity in comparison with other existing models. In this perspective, their model has a better rate of recognition of activities as compared to most of the existing models, or it is at par with a few of the existing models. But as we know the standard SVM requires an informed dataset and hence the model cannot perform well in a real-time environment.

The authors in this paper [68] have proposed an approach for the classification of human emotions based on the linguistic dataset. It uses the AL approach with the incorporation of deep entropy for clustering the dataset and subset selection for annotation. This approach is implemented on a linguistic dataset as it endeavors to classify whether a person is in depression or not by using Natural Language Processing to process words to identify the behavior. The use of entropy for determining the feature labels of clusters of datasets is perfectly embedded in the system to improve the performance of the system. In this study, the dataset used is the textual data, collected from an online forum, website, and social media site. The data used here is collected by another study conducted by Mukhiya et al. [69]. The authors have applied their own proposed model on a blind test set and, according to them, have achieved 0.85 Receiver Operating Characteristic (ROC curve) which has improved the detection rate as compared to earlier algorithms. But as it differs in dataset type and environmental settings, it does not involve the factors of human memory and cognitive constraints that affect while executing the algorithm for activity recognition. Although the algorithm has some similarities with respect to entropy and the AL approach, it differs completely in terms of activity recognition. Moreover, it aims to solve the problem of Internet-Delivered Psychological Treatment and is not focused on real-time identification, hence it has a different scope of discussion in a different environmental setting.

Qu et al. [64] have proposed a Hyper-spectral image (HSI) classification algorithm that uses a TL approach to train the model. As the authors stated almost all of the known algorithms for image classification do not perform well if training and testing datasets are from two different domains. This can cause differences in the number of feature classes and even the types of classes. To resolve this issue, the TL approach is used which covers the gap between the source dataset and target dataset variations and helps generalize the model to perform better in various environments in real-time. The model proposed by the authors in this study is called Physically Constrained Transfer Learning through Shared Abundance Space (PCTL-SAS) which is applied and tested on three datasets named PaviaU-PaviaC [65], Houston [66], [67], and Hanghzou–Shanghai [73]. They have achieved 71.8% performance rate on the Houston dataset, while the previous one was 67% recorded on that dataset. As this paper revolves around the classification of images, it never resembles our problem statement, although the approach is similar, it has no concerns regarding the cognitive and memory constraints of the oracle while annotating regarding the activities performed.

According to the authors of this paper [74], their proposed algorithm TransNet is based on a supervised TL approach for activity recognition from wearable systems. The collection and labeling process of a large amount of data for training the models is expensive and requires a lot of hectic effort to make it possible. To cope with these issues, authors have proposed TransNet which is designed to adapt according to the new users' events and add them to its learning vocabulary. To test their model's performance, they have used four datasets consisting of WISDM [75], OPPORTUNITY [76], Sport and Daily Activity [46], and the fourth one collected by themselves from wearable devices during their work. Through their experiments, they have achieved on average from 88.1% to a maximum of up to 92.7% performance based on different hyper-parameters. As mentioned earlier, their main objective is to enable the model to adapt itself according to the unseen events of a new user. They do not have considered and dealt with the cognitive, physical, and behavioral constraints of the real-life oracles in their research. And all the datasets that they have used are the ones collected in the controlled environments.

The field of study in the following paper [77] is based on sentiment analysis and their research objective is to analyze the opinions of the people for different products. Although it differs from our research objectives, the methodology they have proposed for achieving their objective is considerable for understanding the concepts of TL while building a model to perform in a cross-domain environment. They have proposed a model for the classification of opinions by using four different product domains. They have proposed a model that is trained on the source domain and can be applied to the target domain to classify the opinions. Their research is categorized as a semi-supervised approach based on Entropy Maximization. To test the performance of their model, they have used the dataset of SentiWordNet [78] which is a lexical resource often used for research in opinion mining. According to the authors, their model has correctly classified an average of 72.6% domain-specific words, and an average of 88.4% domain-independent words. This type of model is also required to be designed for the domain of activity recognition that can adapt to real-life environments.

In this paper [79], the authors have presented an approach by combines the techniques of AL and TL for the classification of medical image data. In their proposed approach, they have designed a model that can distinguish between the queries from the target dataset and iteratively asks the oracle to label only those queries that can provide maximum information gain for the model to train, and remove the source samples that are not fit for training the adapted model. Hence, this approach reduces a great number of manual annotations for the oracle. In order to test their model, they have used ten different datasets mainly from the University of California Irvine (UCI) Machine Learning Repository [80] and The Cancer Imaging Archive (TCIA) [81]. The three major datasets include p53, eye, and Soft-tissue-Sarcoma publicly available in above mentioned repositories. According to the authors, their model has shown 95% effectiveness in the tests. Our objective is to implement a similar combination of ATL approach of classification for the wearable sensor datasets in activity recognition.

Huang et al. [82] have presented a study on an AL approach based on the oracle epiphany model. In this theoretical analysis, the authors have dealt with the issue of an omniscient oracle which is often assumed with its preset capabilities in the traditional models proposed earlier. The major reason for assuming such oracles is the convenience while analyzing the models. According to them, there is a huge difference between a real-world oracle in terms of providing the labels, as real-world humans tend to restrain on certain queries that they find difficult to answer, or the answers might be influenced by the order of previous queries and hence result in providing an incorrect label. The authors have not designed any new model but have performed analysis on the variants of two existing algorithms named EPICAL derived from CAL [83] and Oracular-EPICAL derived from Oracular-CAL [84] to test their proposed approach of oracle epiphany. For their testing purpose, they have used the website classification dataset to distinguish between basketball sports websites or any other website. The term epiphany here is explained as the delay or wait

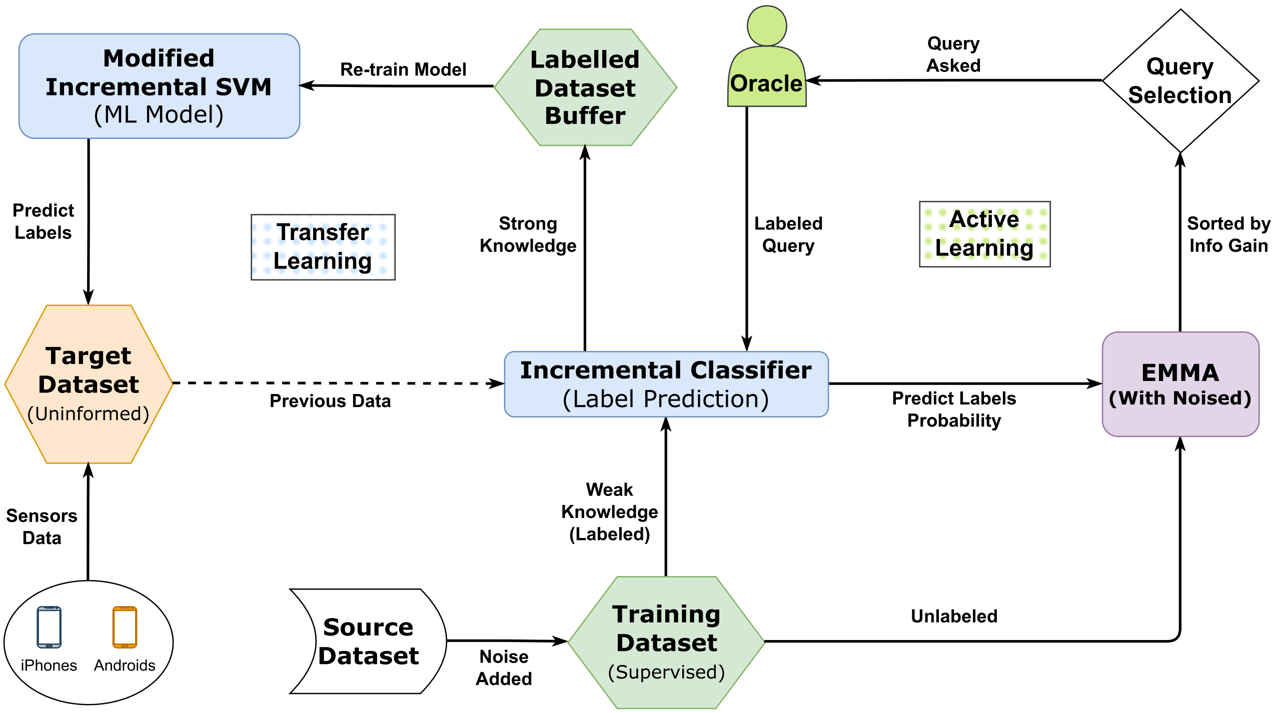

FIGURE 1. Proposed High Level Architecture (HLA).

associated with a query that the oracle is not currently sure about and it is put into the wait, after experiencing similar queries, if the oracle suddenly decides the label for those queries, it then assigned to the queries of the same category. This wait for the epiphany to happen does cause the penalty and adds up the extra cost, but according to the authors, the results have shown that AL is possible to implement with oracle epiphany which resolves the issues of an assumed omniscient oracle. This paper highlights the importance of a realistic oracle that can behave in the same way as a normal human would behave in a natural setting.

B. PROBLEM STATEMENT

From the above study of related work, it is evident that there are multiple models proposed by different people to deal with the activity recognition problem, based on data collected from sensor devices. But as discussed above, all of them have some limitations regarding the dataset collection in the controlled environment, mostly models designed to work only on preloaded datasets, making assumptions about the capacity of Oracle to deal with the number of queries, Oracle's ability to perform the daily activities on the prescheduled pattern, i.e., considering the oracle as a machine, or assuming an omniscient oracle which is infallible. To the best of our knowledge, there does not exist any model that can work on partially available unseen target datasets, operating in the wild environment, and considering the oracle's physical and habitual constraints into account.

We have designed an algorithm that can:

- Perform initial training of the model from source dataset, collected in the controlled environment.
- The model is still able to perform better on incrementally available target datasets, collected in the wild, where data is fetched in chunks.

For this purpose, the existing entropy and memory-aware AL model is tailored:

- To work with supervised but incrementally informed datasets.
- By incorporating the abilities of TL with the existing AL algorithm.
- While being able to operate in real-time, it is also able to acknowledge and reflect the realistic oracle's constraints of irregular activity patterns during initial training from source dataset collected in the controlled environment.

In the next section, a detailed overview of our proposed methodology is covered.

III. PROPOSED METHODOLOGY

Before deep diving into the explanation of our proposed methodology, there are a few terms and approaches of ML that are related to our work and are used in the implementation of the model. Regarding the activity recognition approaches, there is a technique to train the model by iteratively raising the query to the user to provide the labels of the activity observations. The model repeatedly selects/filters a subset of observations from the unlabeled dataset and queries the user/oracle to provide the labels, and based on the user query responses, the model is trained to classify similar activities. In this approach, the model is trained on the current source dataset available at hand and is known as AL. There is another approach for the activity recognition problem in ML named TL. In the TL approach, an existing model already

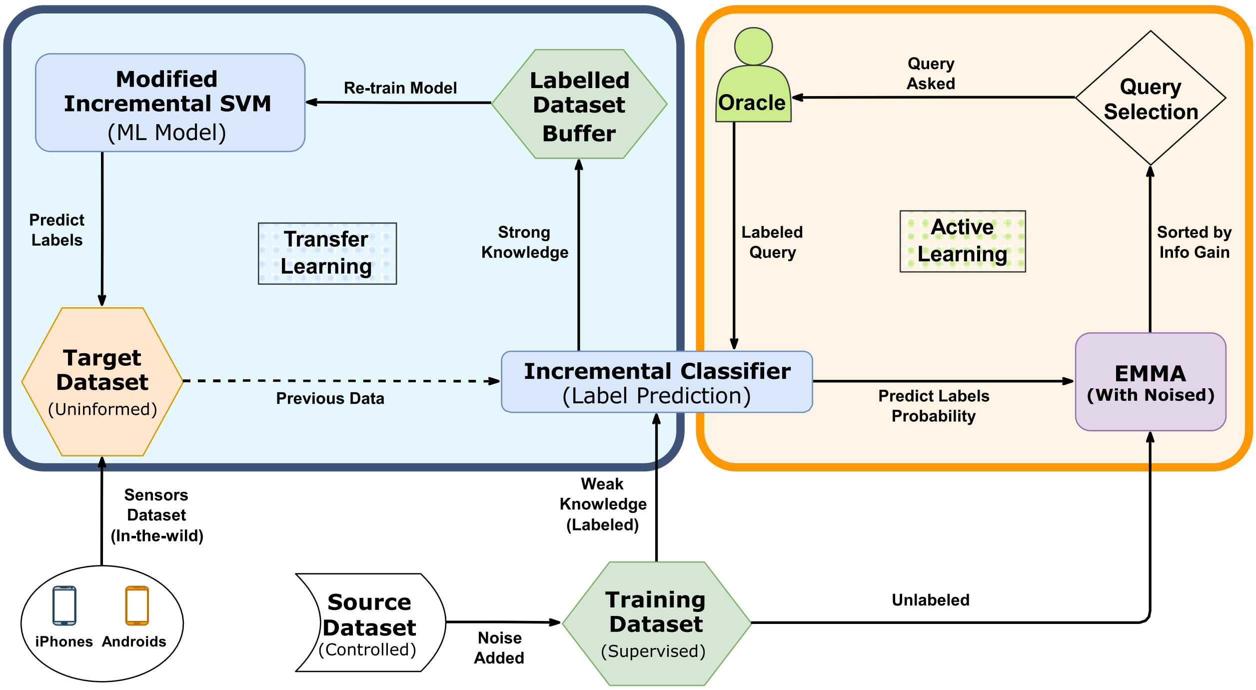

FIGURE 2. Active Learning and Transfer Learning approach in HLA.

trained on any relevant dataset similar to the current source dataset is used, and gradually the model is adapted according to the current dataset being processed. In this approach, the user is not asked to provide the labels for the activity observations, instead, the model uses its existing knowledge to predict the labels, and using those predicted labels the model adapts/refits itself for the new dataset.

In ML, there is also a hybrid implementation of both the above-discussed approaches, known as ATL. In this approach, the model also has pre-learned knowledge and works according to the TL technique, as well as it also asks the user to provide the labels for certain important observations that the model finds hard to classify or where the model has the least information available according to its existing knowledge. This hybrid technique helps the model to rapidly learn and adapt to the new dataset as well as enables the model to operate with partial datasets and improve itself incrementally.

A. OUR PROPOSED VS EXISTING APPROACH

In our work, we have implemented/merged the TL technique with the existing AL technique to achieve the incremental/partial fit ability of the model. The high-level architecture (HLA) of our proposed methodology is given in Fig. 1. The unit called "EMMA (with Noised)" in the above HLA is derived from our base paper by Ashari et al. [37], in which they have presented their EMMA approach. The technique of entropy maximization involves finding that particular case/observation for which it is harder for the model to classify, as its information is least available, and the annotation of that particular case can provide maximum information gain for the model while training. In their work, they have dealt with the cognitive constraint of the user, but not have considered the physical and habitual constraints. Moreover, their model is based on SVM and does not operate in real-time.

B. HIGH-LEVEL ARCHITECTURE

We have used the noise in the source dataset to modify the effect of consistent behavior of the oracle, i.e. behavior to perform each activity on prescheduled intervals and for a fixed amount of duration as if the oracle is a machine. By using the random noise in the source dataset, it would be possible to reflect the habitual constraints of the real user in the dataset and the effect of a stereotype omniscient oracle can be reduced from the dataset collected in the controlled environment. In Fig. 2, the orange highlighted section on the right side of HLA represents this AL part of the algorithm, where the user is asked to label the query selected from this noised dataset by using the EMMA unit, and then it is added to the training dataset. This approach improves the performance of the model when applied to the wild dataset.

Moreover, in the Fig. 2, the blue highlighted section on the left side of HLA represents the implementation of TL and shows our methodology for making the algorithm operate in real-time, and for this purpose, the Modified Incremental SVM model is used. It has enabled the model to update/refit itself, based on the technique of TL as it helps the model to build a strong knowledge by learning from the previous user's labeled dataset of the observations and by adapting to this in-the-wild oracle's pattern of sensor values. Hence, it also reduces the annotation efforts of the oracle as it improves itself rapidly instead of being dependent on user responses only.

According to the above HLA, the algorithm begins with loading the Source Dataset of first Subject/user which is collected in the controlled environment and scales it to normalise the values, and then follows the below steps:

- After loading dataset, the algorithm adds the Random noise into the source dataset and prepares it as the Training dataset.
- On the next step, a list of 8 observations is selected from the Training dataset. This list, know as Weak Knowledge, contains the labelled observations for each one of the 8 activities.
- After selecting 8 observations, these are used to partially fit the Incremental Model. The rest of the unlabelled Training dataset is sent to EMMA for label prediction. For the first time, this Model would be trained only with the source dataset of the current Subject. But for the subsequent Subjects, the labeled target dataset of previously predicted Subjects would also be used along with the new Weak Knowledge of next Subject.
- EMMA uses the partially trained model to obtain label prediction probabilities and calculates the entropy for these observations.
- Using these probabilities, the observations are sorted based on the concept of Information Gain.
- The Information Gain is maximum where it is hard for the model to predict the correct label, and that observation would be selected for the query from the oracle to provide the actual label of that observation.
- Oracle's provided label would be sent to the Model as Strong Knowledge, and the Model would be partially fit/trained again.
- Now this process of Active Learning would keep repeating itself until the Budget is consumed.
- Budget represents the ability of a user to answer the maximum number of queries with correct labels, which is affected by the memory strength of each user/subject. To imitate the memory ability of a real user, we have used 5 different Budget sizes for each subject/oracle. After repeating this process for all the Budget sizes, the algorithm then proceeds on next steps.
- On the next step, the Strong knowledge is saved in a buffer which is later used for partially fit/training the Modified Incremental SVM Model.
- After training, the Incremental Model is used to predict the Labels of the Target Dataset which is collected from in-the-wild environment through various heterogeneous devices.
- This labeled target dataset is then sent to the previous Model for partial training for the next Subject.

C. UNDERSTANDING THE ALGORITHM

The basic algorithm structure of the above-discussed approach can be seen in the algorithm 1. This algorithm is composed of 4 nested iterations/loops, each according to the list of subjects (S), list of different budget ranges (B), number of iterations for calculating the averages of performance measures (I), and the remaining budget range (R) respectively. These four loops can be seen in the algorithm on line number 3, 6, 7, and 17 in the same order.

In this implementation of the algorithm, we have an activity dataset of 8 subjects, which means the first loop iterates 8 times based on the list of 1-8 subjects shown at line number 3 in the algorithm. The second loop iterates 5 times, as it is based on the 5 different budget range values provided in the algorithm. In our implementation, we have used 5, 20, 60, 120, and 200 budget ranges provided in a list *budget range* to test the algorithm on different budget sizes to observe the effect of the number of user query responses on the performance of the model. The third loop is used to repeat the training and testing process at certain times to calculate an average value of model performances against each particular subject and budget range combination. The number of iterations for calculating this average value can be determined manually in the algorithm, and we have set it to 6 iterations to repeat each testing and training process. The fourth loop is based on the value of the remaining budget (R) which is calculated according to the following (1) where *"B"* is the value of the current budget range and the *"B2"* is the previous value in budget range list.

$$R = B - B2 \tag{1}$$

To calculate the range of the remaining budget, the difference between *"B"* and *"B2"* is calculated in order to start the training and testing process after the previous budget value to avoid redundancy. For instance, if assuming that the third loop does not affect the total number of iterations, in the first phase this fourth loop will run 5 times as *"B"* would be 5 and *"B2"* would be zero initially. But in the next phase, when *"B"* would be 20, then *"B2"* would be 5, this fourth loop will iterate 15 times as calculated in (1) above.

$$R_i = B_i - B2_{i-1} \tag{2}$$

This process would be repeated with each budget loop and a new value of *"R"* would be calculated. Let's present the second loop with *"i"* which iterates *"n"* times, so the value of *"R"* would be computed n-times and can be expressed as *"R_i"* where "i" represents a particular iteration of the loop. In this way the values of *"B"* and *"B2"* would be *"B_i"* and *"$B2_{i-1}$"* respectively, except that the *"B2"* would always be zero for the first iteration. Hence, we get the following (2) above.

The third loop of the algorithm controls the number of iterations we want to perform for each combination of subject and budget range. The performance results produced during these iterations are averaged at the end of the loop execution. In this loop, the values of datasets are also loaded if it is the first iteration of the loop, or accessed from the previous iterations already loaded and scaled data. Moreover, line number 11 of the algorithm makes sure that the predicted data of the previous subject is also included with the training dataset of the current model, which allows the model to

Algorithm 1 EMMA With Active Transfer Learning

```
Result: Performance of the model
subjects = [1, 2, 3, 4, 5, 6, 7, 8];
budget range = [5, 20, 60, 120, 200];
for subject in subjects do
    B2 = 0;
    Avg calculation iterations = 6;
    for number of values in budget range do
        for range in Avg calculation iterations do
            if Avg calculation iterations == 1 then
                Datasets = Load and scale (from Files);
                if subject != 1 then
                    Datasets = Append predicted data of previous subject with training data of current subject;
                end
                Confidence Error = Calculate by memory strength of subject;
            else
                Datasets = Use already loaded datasets for subsequent budget ranges (Data Reference Sets);
            end
            for range in remaining budget do
                Model = Partially fit (on Training dataset);
                Predictions = Model predictions (on Test dataset);
                PredictProb = Calculate probabilities (from Predictions);
                Entropy = Calculate entropy (from PredictProb);
                Evaluate Factor = Calculate from (Entropy * Confidence Error);
                Re-sorted Dataset = Sort (Test dataset, using Evaluate Factor);
                Query Label Assignment = Calculate from (Re-sorted Dataset, using Confidence Error);
                Training dataset = Append queried observation (Query Label Assignment, in Training dataset);
            end
            Model = Partially fit (on Training dataset);
            Predictions = Model predictions (on Test dataset);
            Performance Measures = Calculate using (Predictions);
            Data Reference Sets = Save to avoid reload in subsequent Budget ranges (Datasets);
        end
        Average Performance Results = Compute for this budget iteration and Append (Performance Measures);
    end
    Predictions Reference Set = Save for next iterations, calculated on Test dataset of current subject (Predictions);
    Current Subject Average Performance Results = Save (Average Performance Results);
end
```

adapt according to this target dataset gradually by repeatedly refitting/retraining itself. The iterations of the subjects are presented by *"j"*, the current subject is represented with *"S_j"*, and the previous subject is expressed with *"S_{j-1}"*. The training/labeled dataset is presented with *"L"* and the testing/unlabeled dataset is presented with *"U"*. When the model predicts the labels of the testing dataset (U), this dataset including its labels is presented with *"P"*.

For the first subject, no previous subjects dataset is available, this is why the testing is performed on the source/training dataset and testing is performed on the target dataset of the same subject only. Whereas for the subsequent subjects, the previous subject's predicted dataset, which is presented as *"P_{j-1}"*, is available. For each subsequent iteration, all the predicted observations of the previous subjects are carried forward for the next subjects. Hence we can rewrite it as (3) below.

$$\sum_{k=1}^{n} P_k \tag{3}$$

Subject to:

$$n < j \tag{4}$$

Here in (4), *"j"* represents the number of elements/subjects in *"S"* list. Hence, this collection of datasets is combined with the current subject's training dataset presented as *"L_j"* and we get a combined dataset represented with *"T_j"* that is used to train the model for the current subject. We have computed the performance results based on two different variations of appending these two datasets. In the first implementation of the TL approach, the following (5) is used to get the *"T_j"* dataset. For this purpose, we simply took the union of both the *"L_j"* and the *"P_{j-1}"* datasets.

$$T_j = L_j \cup \sum_{k=1}^{n} P_k \tag{5}$$

This approach has achieved very promising results, while initially training the model on a lab dataset and gradually adapting the model according to the target dataset. However, the rate of adaptation is slow in this approach due to the combined datasets which include complete observations of the lab dataset. As the observations in the lab dataset are significantly more than the observations in the datasets collected in-the-wild, the process of model adaptation is slow even if the datasets are combined. In our second approach, we have replaced the existing observations in the lab dataset with the new observations of the in-the-wild dataset. For instance, if the number of observations in the *"P_k"* dataset is *"OP"*, as shown in (7) below, we have removed the *"OP"* number of observations from *"L_j"* dataset, and then have combined both the datasets to get *"T_j"* dataset which now has less number of *"L_j"* observations. This process would be repeated with each iteration and gradually the observations of the *"L_j"* dataset would be replaced by the *"P_k"* dataset. This produces the following (6) below.

$$T_j = (L_j - OP) \cup \sum_{k=1}^{n} P_k \tag{6}$$

Subject to:

$$OP = Number\ of\ Observations\ of\ \sum_{k=1}^{n} P_k \tag{7}$$

In this method, as the number of observations in the *"P_k"* dataset increases, the same number of existing observations from the *"L_j"* dataset are reduced. Hence, the number of observations of *"L_j"* dataset are gradually reduced in the *"T_j"* dataset for training the current model. This allows the model to adapt more rapidly as compared to the earlier approach of TL.

IV. IMPLEMENTATION

A. DATASETS

In this work, we have used two datasets for the training and testing of our proposed model. As Table. 2 shows, these datasets include the Daily and Sports Activity (DAS) [46] dataset which is used in the base paper and the second is ExtraSensory (ES) [85], [86] dataset. According to our proposed methodology, we have trained the model on the DAS dataset which is a lab/controlled environment dataset, and the testing of the model has been done on the ES dataset which is an in-the-wild dataset collected in a real-time environment. From both datasets, eight activities are selected consisting of sitting, standing, laying on the back, ascending stairs, descending stairs, running/running on a treadmill, exercising, and cycling. After determining these eight similar activities in both datasets, the available features/columns for these activities are compared. The features in datasets consist on the values provided in columns that are recorded through the sensor devices used for collecting that dataset. Each feature/column in the dataset represents a particular type of values in that column. Because we have to perform the cross dataset analysis by training the model on DAS and testing it on ES, it is necessary to match the columns/features for each type of sensor values being used in both datasets. From both datasets, we have selected three sensors' data, and from each sensor' data we have selected three features/columns of values titled as X, Y, and Z. We have insured that the features are ordered correctly to avoid the wrong comparison, i.e. training the model on Accelerometer sensor data from source dataset and testing it on the Magnetometer sensor data in the target dataset. The works that are performed on the same datasets do not require consideration of the feature's similarity while training and testing of the models. The authors of the base paper have used the processed form of the DAS dataset which is obtained by performing Principal Component Analysis (PCA) on the large feature size and hence reducing it to their required feature size. They first processed the raw values of the dataset to obtain the particular characteristics of the data, i.e. min, max, mean, kurtosis, skewness, frequency values, and autocorrelation from those raw values of the sensors, and then by performing PCA on that resulting 1170 columns dataset, they converted it into 30 columns dataset. This processed form of the DAS dataset is not useful for our research as it does not correspond to the actual features when compared with other datasets. For our work, we have found three types of sensors that are available in both datasets and separated the columns of accelerometer, gyro, and magnetometer features. Each sensor meter has three types of raw values named as X, Y, and Z for each

TABLE 2. Datasets used in our work.

DAS	Extra Sensory
01) Sitting	01) Sitting
02) Standing	02) Standing
03) Laying on Back	03) Lying Down
04) Ascending Stairs	04) Stairs - Going Up
05) Descending Stairs	05) Stairs - Going Down
06) Running on Treadmill	06) Fix Running
07) Exercising on Stepper	07) Exercise
08) Cycling	08) Bicycling
09) Rowing	09) Fix Walking
10) Jumping	10) Sleeping
…	11) Lab Work
19)	12) In Class
	13) In A Meeting
	14) Loc Main Workplace
	15) Or Indoors
	16) Or Outside
	…
	51)

observation. Hence, the total number of features/columns selected is nine, three of each sensor type from both datasets.

B. EMMA

The base paper implementation of the EMMA uses the SVM as the classifier model for the prediction of labels. In EMMA, the entropy means randomness or disorder of information and memory means the oracle/user's cognitive ability to provide the correct labels. The objective of EMMA is to maximize both the entropy and the memory of the user for selecting the observation for the query. This approach is based on AL where the model asks the user to provide the labels for the unseen dataset and then trains itself on that labeled dataset. The authors of our base paper have considered the cognitive constraint in their implementation of the EMMA approach to creating the environment of a real user while taking the labels from query responses generated through the Oracle algorithm, the machine itself. They have used the memory retention and budget to calculate the memory strength of the user, and have implemented it to represent different user capacities to answer labels.

The budget represents the number of queries that a real user can respond to in a certain time, and it has different capacities according to the different users. Similarly, memory retention represents the rate of a user's ability to provide correct labels. The product of budget and memory retention defines the memory strength of a user for consideration in EMMA. The second value used in EMMA is the maximum entropy of the observations. The observation with maximum entropy is selected for the query as it helps the model to gain maximum information after getting its label from the user. Observation with maximum entropy means it is hard for the model to classify this observation, and when the model gets the label for this particular observation from the user, it helps the model to know most disordered observations, hence the information gained for the model is also maximum. In the base paper, they have also implemented the clustering of the dataset sample to increase the possibility that each time the observation for query is selected from a well-distributed space of the dataset. However as that algorithm code with clustering implemented is not shared by the authors, the code we have used in our research is the one publicly shared by them, and have rest of the implementation same as discussed above. Moreover, to proceed with further alterations and implementation of our proposed methodology, the aspect of memory retention is considered static for all the users, although the memory strength is still being separately calculated the same way as in the base paper implementation.

The results of the base paper provide the performance of their model for the DAS dataset. On different budget sizes, their model has an average accuracy between 35% to 98% corresponding to the increase in budget. From their results, we can see that the performance is not so good when the budget size is small, in other words, when the user can provide the labels only for a limited number of queries, the performance of the model is low. But when the budget size is 200 queries (the actual number of queries asked differs due to previous budget size subtraction and the number of iterations determined before executing the algorithm) there can be a great improvement in the performance. At the 200-budget size, the accuracy performance is 98% achieved in the base paper implementation of EMMA.

C. PRELIMINARY TESTS

As discussed earlier in the datasets section above, due to the requirement of our proposed methodology and because of the research gap that we have identified in the training and testing methods, used for the existing activity recognition models, we have to use two different datasets for the training and testing of the model respectively. The reason behind this is because of the large size availability of the lab or controlled environment datasets which can be used for the initial training of the model and then with the gradual increase in the availability of real-time datasets that are collected in the wild, the model adapts itself according to that new dataset. This transfer of model learning source during the real-time implementation allows the model to adapt itself better according to the user-specific dataset being collected in real time. Because of this purpose, we have to use only such features of the dataset that can be found in both datasets being used. Hence, we could not use the processed or manipulated form of the dataset for our model. During the preliminary tests, we have attempted to replicate the original work, and for this reason, these tests are performed only on the DAS dataset for training and testing of the model. From the DAS dataset, we have selected the data of three sensors, i.e. accelerometer, gyro meter, and magnetometer sensors data for feature training of the model. The results of the existing EMMA algorithm can be seen in Fig. 3 when we applied the EMMA on this raw form of the DAS dataset. On the X-axis of the above Fig. 3 are the different budget sizes and on the Y-axis is shown the prediction accuracy performance of the model according to those budget sizes. It can be seen that the performance of the algorithm is quite similar to the original work on the maximum budget size. At this point, we do not have preprocessed the dataset by making the changes in the base paper algorithm to avoid scaling of the data, and we have replicated similar results of 95% performance accuracy on the same 200 budget size.

In the next step, because we need both datasets to be normalized to improve the model's prediction accuracy in later stages of our work, we used the original EMMA which scaled both the training and testing datasets. The results of this second test can be seen in Fig. 4 where data is normalized before the training and testing process. It can be seen that the performance of the model has reduced to around 45% accuracy after enabling the preprocessing of the dataset. These are the results of the original EMMA algorithm while implemented on the DAS dataset only, for both training and testing of the model. Although currently, the scaling of the dataset shows a bad impact on the performance of the algorithm, the normalization is required later during

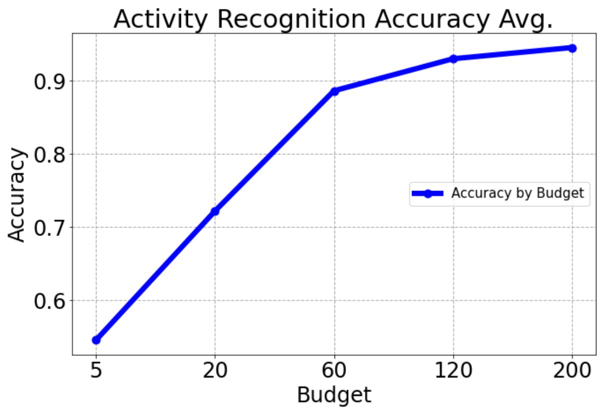

FIGURE 3. DAS without Normalized.

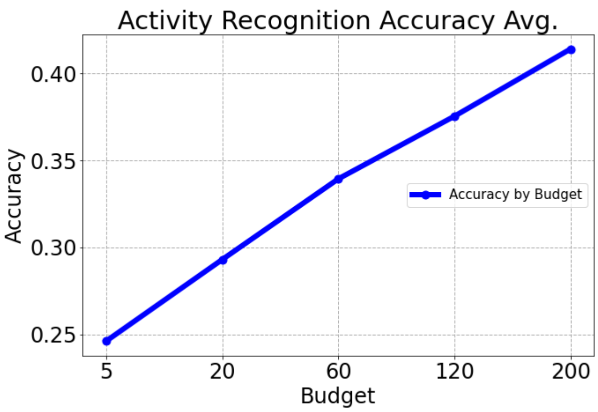

FIGURE 4. DAS Normalized.

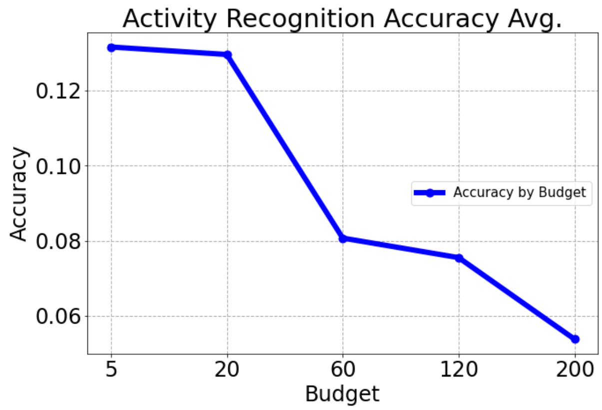

FIGURE 5. DAS, ES without Normalized.

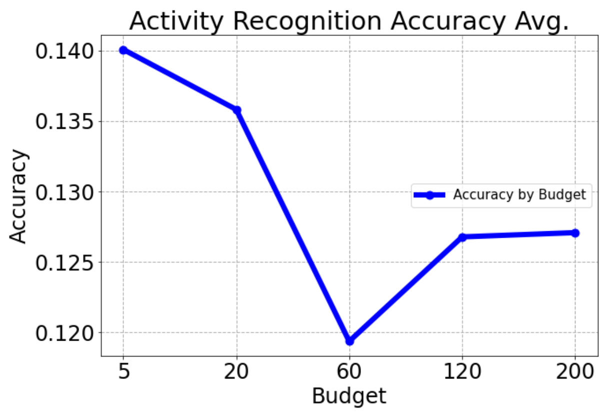

FIGURE 6. DAS, ES Normalized.

training and testing of the model with two different datasets respectively.

In the next phase of preliminary tests, we have just changed the dataset for testing the model. When the original model is trained on the DAS dataset and tested on the ES dataset when the datasets are not normalized, the performance of the model can be seen in Fig. 5 which shows that the performance accuracy has drastically dropped to 5% on the 200-budget size, whereas the number of queries being asked from the user has been increasing. Increasing the number of queries to be asked from users means that more observations of the DAS dataset are provided to the model for its training. And we can see that as the model trains more on the DAS dataset, its performance of prediction decreases on the ES dataset. This same test is performed again, but this time we have normalized the datasets before training and testing of the model on DAS and ES datasets respectively. In the Fig. 6 we can see the performance of the model which shows a minor improvement in the accuracy of the model. Although the accuracy has increased from 5% to 12.5% on the 200-budget size, still this accuracy of the model for activity recognition is very poor, which makes the model useless when trained on a lab dataset and implemented in the real-time environment on an unseen dataset collected in-the-wild when data is not available in hand as preloaded but would be available in chunks gradually.

D. OUR PROPOSED IMPLEMENTATIONS

1) ADDING THE NOISE

In real life, different users have different types of constraints while performing their daily life activities. For instance, a young person's daily activity routine would be very different from that of an elderly person. There would certainly exist some activities that a young person would be doing more frequently than an elderly person who even might not be doing those activities for many days. To develop a better understanding of this concept, consider a household environment where a young child might be performing ascending and descending stairs activity more frequently, even daily, to bring stuff from the first-floor attic, whereas the grandparent in that house might not use the stairs for weeks. These types of behavioral constraints of the users would result in generating different sets of values from the sensor devices. Similarly, the same activity done by the two different users would result in producing different sensor values due to their physical constraints. These constraints might occur due to their age factors, disabilities, or any other circumstances. For instance, the activity of running done by an athlete and an overweight person would certainly result in different sensor values. Similarly, the activity of climbing the stairs done by a young child and an elderly person with knee pain would result in producing different values from sensor devices.

Moreover, as we know the datasets of controlled or lab environments are collected by using the same type of sensors, i.e. the devices being used for data collection are more likely to be the same for all the subjects doing the activities. Due to this factor, the collected dataset tends to have very similar sensor values for similar activities. The same is the case with the DAS dataset, for which the Xsens MTx units were used to collect data from all the participants performing the activities. Whereas, the datasets collected in the wild tend to have different brands of sensor devices. These different devices cause variations in the values to be produced by the sensors for a particular activity. For instance, the ES dataset, that we have used for testing the model, is collected from various heterogeneous smartphones including different models of iPhone as well as various brands of Android smartphones and their different models. These heterogeneous devices do possess different sensor devices embedded in them and hence produce the sensor data with varying calibration that results in slightly different values being produced for the same activity being performed.

This difference in lab and in-the-wild datasets needs to be taken care of while being used for the training and testing of the same model. For this purpose, the technique of adding the noise [41] in the source or lab dataset is used to help the model generalize instead of overfitting according to the lab environment dataset. There are different approaches to adding noise, such as Gaussian noise, Salt and Pepper, Poisson noise, and Random noise. The approach that we have used in our work is random noise, in which a certain noise value or values are used to replace the dataset values by selecting those values at random. Our intention to use Random noise is based on the fact that it avoids any biasness by adding noise values on random places in the dataset. Whereas, in Gaussian noise, random noise values are normally distributed in the entire dataset. It may produce an almost evenly distributed interval at which the values are added in the dataset. The Salt and Pepper noise uses two values, one extreme minimum and other extreme maximum value, to replace with the dataset values in order to add the noise. This approach was not required in our case as it would result in adding such values in the dataset which are useless for the model training. The Poisson noise is used for image processing which involves datasets in matrix form. Whereas, feature values in our case are in single columns. In our work, we computed the median of each column/feature and then used that median value wherever the value would be replaced from a random place in that column. This adds a certain level of distortion in the dataset, which helps improve the performance of the model, not always but sometimes, by reducing overfitting and increasing the generalization ability. The rate of noise that we choose is 0.10 which would replace only 10% values in each feature column. Choice of this rate is selected after generating and testing multiple noised dataset variants, and we found this rate to be ideal with DAS dataset. We found no set rule for determining the rate of noise, which would be different for each dataset, and left it for further studies in future work.

2) INCREMENTAL CLASSIFIER

The authors in the base paper have implemented the Support Vector Classifier (SVC) from the sklearn library of Python, to classify the activities. This SVC classifier belongs to the SVM ML models, that use the hinge loss function and hence do not calculate the probability by default. The probabilities of each observation are used in the EMMA algorithm to calculate the entropy of that observation as discussed in the above section. To compute the observation probabilities of being classified into a particular class, the extra parameter of probability is manually set to true in SVM model parameters. But this parameter slows down the working of the model because of the five-fold cross-validation being used to compute those probabilities. Hence, the "predict_proba" function of the SVC linear model may be inconsistent with the "predict" function [87]. Moreover, the SVM classifier does not work with incrementally available datasets, as it cannot partially fit itself repeatedly. Due to this limitation, this model needs all the datasets to be used to be available in hand before starting the training and testing process. As a result, this fails to be able to work in a real-time environment where data is to be fetched in chunks.

Whereas, we have replaced the SVM classifier with the SGD Classifier. The SGD classifier with the same hinge loss function can be referred to as the incremental SVM because it is also a linear classifier that works the same way as the SVM classifier but can partially fit itself with the new dataset. But similar to the SVM, it also never computes the probabilities with the hinge loss function. Due to this reason, we have made a variation and used the log loss function in the SGD classifier which has the probabilities computed by default as a byproduct of the loss function. Hence, this modified incremental SVM, or in other words, the SGD classifier with log loss, allows us to partially fit the model each time the new part of the dataset becomes available. This characteristic of the model enables it to be used in a real-time environment where all the data is not available in advance.

3) TRANSFER LEARNING

As we have discussed the technique of TL in the methodology section above, we know that it is used to adapt a pre-learned model according to the newly available dataset. As part of our implementation of the proposed methodology, we have used this technique to adapt a model that is pre-trained onto the dataset but as the new ES dataset becomes available after initial predictions, the model partially transfers its learning source toward the ES dataset. Because the SGD Classifier model can partially fit itself repeatedly, in the subsequent iterations of the algorithm, we have also partially trained the model on the ES dataset which is a real-time dataset. For the

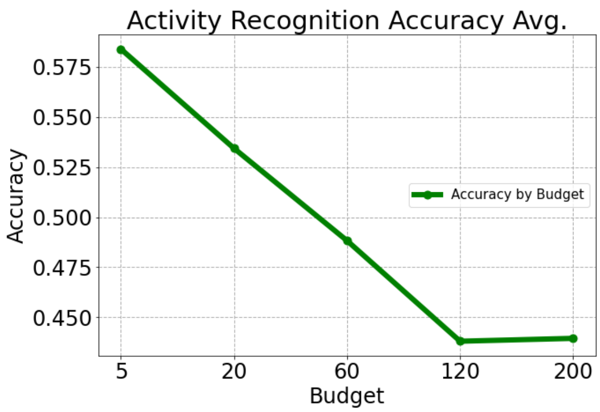

FIGURE 7. Trained on Processed DAS, and Tested on ES.

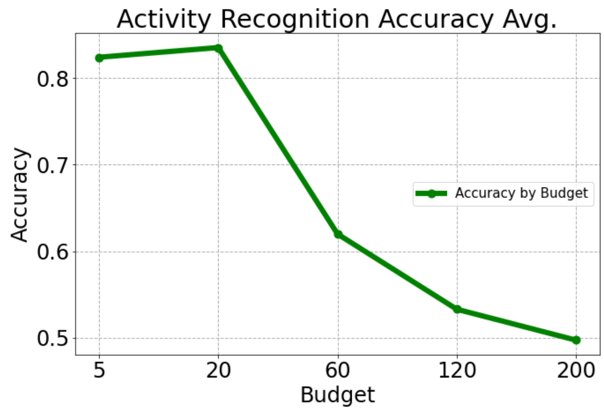

FIGURE 8. Processed DAS, Incremental Classifier, Tested on ES.

first subject/user training and testing, only the DAS dataset is used for the training purpose and tested on the ES dataset. But for the subsequent users/subjects, the model has trained on DAS as well as the previous subject's ES dataset. In this way, the model gradually adapts itself according to the ES dataset which is increasing for each next subject. Hence, the model carries its learning from the previous subjects' ES dataset as a weak model for the next iterations and improves itself by adapting to the new user's dataset after predicting the labels for that user. Previous user's data is used to adapt the model because the same user's subsequent days' data is not available in this dataset. In a real-time environment, the model can learn from as many increments of the user's new data being collected daily.

In the next step of our implementation of this technique, instead of appending the previous user's ES dataset with the current user's DAS dataset for the training of the model, we have replaced the existing observations of the DAS dataset with the observations of the ES dataset from previous user/subject. This allows the model to rapidly adapt itself to the new ES dataset by learning the pattern of values in the ES dataset. It helps to increase the process of TL for the model.

V. RESULTS AND DISCUSSION

A. THE EFFECT OF NOISE

After adding the noise in the DAS dataset for training the model, the results of the model's classification prediction on the ES dataset can be seen in Fig. 7, where we can see a major improvement in the performance of the model. The accuracy of the model has increased up to 44% for the 200-budget size. On average, the performance of the model lies between 60% to 40% accuracy. Although, the overall performance of the model still tends to decrease with the increase in the DAS dataset observations for the training of the model, the model's ability to predict the correct labels for the ES dataset has increased due to the generalization during the model training which is achieved due to the noise being added in the source dataset. This noise prevents the model from overfitting itself for the DAS dataset only and hence enables it to have better predictions on the dataset collected in-the-wild environment. The decrease in the performance due to the increase in the DAS dataset is still proof that the model overfits itself according to the source dataset as the number of observations increases with the query budget.

B. USE OF INCREMENTAL CLASSIFIER

By implementing the SGD classifier for the incremental training of the model, the average performance of the model has increased maximum of up to 85% accuracy. In Fig. 8 we can see the performance difference between the SVM classifier, which slows down due to the five-fold cross-validation method used when the prediction probability is computed which also affects the consistency of the predict function, and the performance of the SGD classifier with log loss function which automatically computes the probabilities by default. The SGD classifier's ability to perform better with large datasets allows it to have better accuracy as compared to the SVC when trained and tested on two different datasets. But even though the overall performance has improved, still we can see the issue of overfitting is still present, and as the model's training on the DAS dataset increases, its ability to predict labels in the ES dataset gradually starts decreasing.

C. THE TRANSFER LEARNING TECHNIQUE

In Fig. 9 we can see that the implementation of the TL technique for model training and adaptation has helped improve the performance of the model. After the first implementation in which the real-time dataset is appended with the existing lab dataset, the average performance has improved up to 82% accuracy on 200-budget size. Due to the ability to adapt the model through TL, the issue of overfitting the model has been finally resolved. In the next step of implementing the TL technique, we have replaced the existing DAS dataset observations with the new ES dataset. This allows the model to adapt to the new dataset faster as compared to the previous approach. In Fig. 10 we can see the performance of the model has further improved and lies on average between 80% to 90% accuracy. This is the

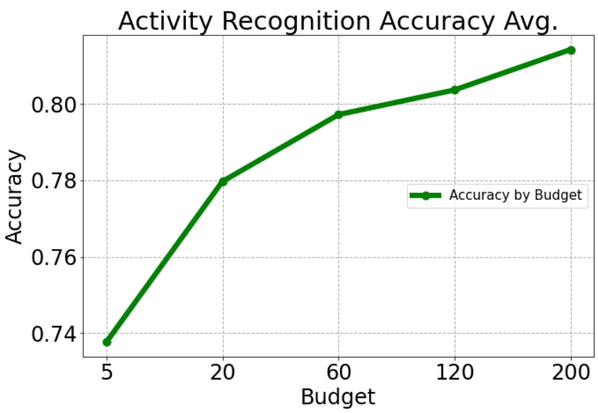

FIGURE 9. Trained on DAS and ES, Incremental Classifier, Tested on ES.

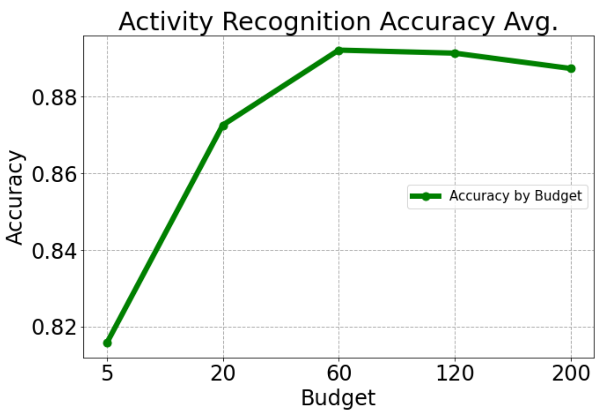

FIGURE 10. Trained on DAS and ES, Incremental Classifier, Tested on ES.

highest possible accuracy that we have achieved through the implementation of our proposed methodology to overcome the identified research gaps.

Moreover, we can see the comparison of other performance measures for our proposed algorithm as well as the existing implementation in the Table. 3 below, including the Precision, F1-Score, and Recall measures. With the changes in budget, the performance variations can be noticed. In precision measures of our proposed implementation, the results show that the algorithm has very promising performance as the results can be seen with up to 90% performance. Similarly, the results with recall measures have achieved 88% performance. Only in the case of the f1-score, the performance of the algorithm show less improvement of up to 85%, which is still much improved as compared to the existing approach. The existing model of base paper resulted up to 6% only in the F1-Score when that model is trained and tested on the cross datasets. We can see that the overall performance of our model has improved and can classify all the activities with a minimum to maximum performance difference of 10% only. It means that even if the user has provided only a limited query response and the budget is very low, even then the performance of the model would not fall drastically as it was the case with the existing algorithm.

TABLE 3. Comparison of performance measures.

Implementation	Budget	Precision	Recall	F1-Score
Our Proposed	5	0.78	0.79	0.73
	20	0.89	0.86	0.83
	60	0.90	0.88	0.85
	120	0.87	0.86	0.82
	200	0.87	0.86	0.83
Existing	5	0.18	0.14	0.05
	20	0.15	0.11	0.06
	60	0.15	0.11	0.05
	120	0.16	0.13	0.06
	200	0.15	0.11	0.06

D. DISCUSSION

From the above-discussed results of the model performance, it is evident that our proposed approach has improved the accuracy of the existing model regarding the factors caused by the real-time implementation of the EMMA algorithm. The reason behind this huge performance difference between the existing methodologies and our proposed methodology in real-time implementation is because of the difference between the training and testing methods used for the model building. The Fig. 11 shows the difference in technique that is used for selecting the dataset for the model training and testing. To better understand the concept, only three activities done by three users are shown in that Fig. 11, where the upper three users' activities represent how the datasets are used for building the model in controlled environments. To the best of our knowledge, all the sensor-based HAR models that are trained and tested on lab/controlled environment datasets are trained and tested for the model's performance on the same dataset observations. In other words, if a model is trained on the DAS dataset, it would also be tested on the DAS dataset. Even if those models were tested on multiple datasets, still the training and testing of the models would have been done individually on each dataset separately.

The upper half of the Fig. 11 shows the same thing as if the controlled environment model trains on the feature set of users A, B, and C, it is also tested on the observations of users A, B, and C. Moreover, in this training and testing process, all the features are equally split into a training set and testing set, most often found is the 70% and 30% split ratio between the training and testing dataset. Whereas, the lower half of that Fig. 11 shows our implementation of the training and testing process, where if the model is trained on the feature set of users A, B, and C, it is then tested to predict the same activities but by different users X, Y, and Z's activity observations. Moreover, the split ratio of the dataset for training and testing of the model is different for each activity by each user. The number of observations can be the same in controlled datasets, but in-the-wild datasets are collected in a real-time environment and do not have the same number of observations for each activity as the users have different physical constraints as well as habits

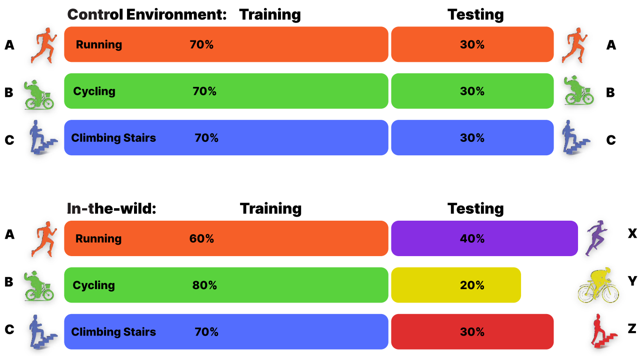

FIGURE 11. Lab vs Real-time Environment Implementation.

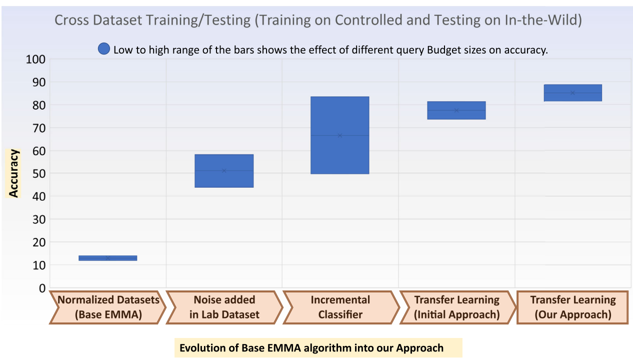

FIGURE 12. Evolution of Base EMMA [37] algorithm into our Approach.

of doing certain activities vary according to their routines. The reason for our proposed approach is that the models have to deal with the same scenario when implemented for real-time execution. To improve the performance of the model in this particular situation, as discussed above, we have implemented our proposed methodology which has resulted in promising performance of the model in real-time implementation.

This new modified model is incrementally informed and does not require the complete target dataset to be available in advance for training of the model, and also eliminates the requirement of a pre-learned expert model for label prediction as the new approach allows it to improve itself gradually. Moreover, the new approach allows the model to operate with in-the-wild datasets collected in the real-time environment. It is also able to overcome the target dataset variations due to users' physical and behavioral constraints by reflecting them in the source dataset using noise. Hence, the model can execute in a real-time environment and have better performance as compared to the original algorithm that can neither operate in the real-time environment nor handle the differences between lab datasets and in-the-wild datasets. Moreover, in the case of the cold start of the algorithm, our model has significantly better performance as compared to the base paper implementation. In the original EMMA approach the model is trained separately for each new user/subject's activities prediction. Whereas, in our discussed approach above, the model carries its previous learning for the next users' prediction process and hence its performance for a new unseen user's activity prediction is comparatively better than the base EMMA approach. Moreover, if the new user's activity data were available in a couple of chunks for activity prediction, the model would be able to rapidly improve its performance by adapting to that user's previous activity patterns through TL.

We can see the accuracy improvement of the model in Fig. 12, while the model evolves from the base EMMA into our proposed approach. The accuracy difference between the base model and our final modified model is huge. As we can see, the base model performs drastically poor when trained and tested on the cross dataset. Whereas, our proposed approach enables the model to evolve and provide higher accuracy when trained and tested in cross dataset environment. The Fig. 12 summarises all the accuracy results for each step of the modification process of the model and shows that our final approach has achieved upto 88% accuracy by modifying the model to work in real-time environment.

With the implementation of our proposed methodology, although the model's performance accuracy has significantly improved, still there is a 10% to 12% deficiency in the prediction of all the activities perfectly. We could not obtain above the 88% accuracy with our implementation of the algorithm. One of the factors that might be the reason is the fact that there is a possibility of errors by the users in providing the labels for the ES dataset while performing their daily activities., The base paper implementation of EMMA includes the factor of human cognitive ability while training the model, which calculates the possibility of a user's ability to remember the label correctly for the activity done, based on the time difference between when the activity is done and when the query is asked to provide the label. In the base paper implementation, the model was then tested on the lab environment dataset which did not have wrong labels against activity observations. Whereas, the ES dataset we have used for testing the model is collected in a real-time environment and very much has the possibility of wrong labels provided by the users. Hence, this could be one factor for not being able to achieve further higher accuracy, unlike those HAR models that have achieved higher results but in controlled environment implementations.

VI. CONCLUSION AND SUGGESTIONS

A. CONCLUSION

After implementing our proposed methodology to overcome the research gap, identified in the EMMA approach for activity recognition, we have successfully trained the model on the controlled dataset and applied it to the dataset collected in the wild environment while dealing with the variable activity patterns of the oracle. This modified model have maintained on average between 80% to 90% performance accuracy. Secondly, we have designed this model that is now capable of incremental model adaptation on incrementally available target datasets, and it eliminates the requirement of a pre-learned expert model. Also, by adding the noise in the source dataset we have reflected the characteristics of in-the-wild dataset. Finally, with the help of TL, the model can learn better even if the user's annotated dataset is not available in large size while applying the model for real-time activity recognition.

B. SUGGESTIONS

These modifications have enabled the model to perform better in several situations while implemented in a real-time environment. Still, there can be certain limitations of the model that are not yet encountered and might be observed in the future. One such future improvement might be the optimum control on the rate of adding noise in the source dataset by understanding the most effective rate of noise to make the learning of the model more generalized for the activities done by different humans. The model might also be improved if somehow the noise could be added on run-time and the values to be replaced could be computed from the target dataset, instead of the source dataset, which would better help the model to adapt to the target sensor device variations of the values. The largest in-the-wild human activity dataset, collected with wearable sensor devices, available in open source, that we could arrange for our research is ES. Because of the feature space limitations, to be matched in source and target datasets, the number of datasets that we could use was limited. We aim to conduct further analyses in our future work by finding other datasets that have the same features for cross dataset training and testing of the model. In-the-wild datasets are considered prone to have a certain rate of wrong labels provided by the users against activities. If there could be the datasets, collected in-the-wild but annotated by expert observers by watching the camera feed of the users. Such datasets can result in better training and performance of the model. Moreover, further experiments of the model can be performed on low-powered devices that are actually deployed in the wild, to measure the performance in terms of latency and power consumption.

ACKNOWLEDGMENT

The authors would like to thank the King Saud University, Riyadh, Saudi Arabia for supporting this research under the Researcher Supporting Project number (RSPD2023R609).

REFERENCES

[1] N. Thakur and C. Y. Han, "An intelligent ubiquitous activity aware framework for smart home," in *Human Interaction, Emerging Technologies and Future Applications III*, T. Ahram, R. Taiar, K. Langlois, and A. Choplin, Eds. Cham, Switzerland: Springer, 1007, pp. 296–302.

[2] O. S. Neffati, S. Sengan, K. D. Thangavelu, S. D. Kumar, R. Setiawan, M. Elangovan, D. Mani, and P. Velayutham, "Migrating from traditional grid to smart grid in smart cities promoted in developing country," *Sustain. Energy Technol. Assessments*, vol. 45, Jun. 2021, Art. no. 101125. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213138821001351

[3] A. Gazis, E. Ioannou, and E. Katsiri, "Examining the sensors that enable self-driving vehicles," *IEEE Potentials*, vol. 39, no. 1, pp. 46–51, Jan. 2020.

[4] M. Tortonesi, A. Morelli, M. Govoni, J. Michaelis, N. Suri, C. Stefanelli, and S. Russell, "Leveraging Internet of Things within the military network environment—Challenges and solutions," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 111–116.

[5] M. M. Rodgers, V. M. Pai, and R. S. Conroy, "Recent advances in wearable sensors for health monitoring," *IEEE Sensors J.*, vol. 15, no. 6, pp. 3119–3126, Jun. 2015.

[6] J. E. Bardram and A. Matic, "A decade of ubiquitous computing research in mental health," *IEEE Pervasive Comput.*, vol. 19, no. 1, pp. 62–72, Jan. 2020.

[7] L. S. Kondaka, M. Thenmozhi, K. Vijayakumar, and R. Kohli, "An intensive healthcare monitoring paradigm by using IoT based machine learning strategies," *Multimedia Tools Appl.*, vol. 81, no. 26, pp. 36891–36905, Nov. 2022, doi: 10.1007/s11042-021-11111-8.

[8] H. Zhang, M. Babar, M. U. Tariq, M. A. Jan, V. G. Menon, and X. Li, "SafeCity: Toward safe and secured data management design for IoT-enabled smart city planning," *IEEE Access*, vol. 8, pp. 145256–145267, 2020.

[9] G. Suciu, I. Hussain, and G. Petrescu, "Role of ubiquitous computing and mobile WSN technologies and implementation," in *Proc. Int. Conf. Electr., Commun., Comput. Eng. (ICECCE)*, Jun. 2020, pp. 1–6.

[10] S. Mohsen, A. Elkaseer, and S. G. Scholz, "Industry 4.0-oriented deep learning models for human activity recognition," *IEEE Access*, vol. 9, pp. 150508–150521, 2021.

[11] A. Mukherjee, S. Misra, P. Mangrulkar, M. Rajarajan, and Y. Rahulamathavan, "SmartARM: A smartphone-based group activity recognition and monitoring scheme for military applications," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2017, pp. 1–6.

[12] M. A. Khan, K. Javed, S. A. Khan, T. Saba, U. Habib, J. A. Khan, and A. A. Abbasi, "Human action recognition using fusion of multiview and deep features: An application to video surveillance," *Multimedia Tools Appl.*, vol. 83, no. 5, pp. 14885–14911, Mar. 2020, doi: 10.1007/s11042-020-08806-9.

[13] D. Roggen, G. Tröster, P. Lukowicz, A. Ferscha, J. del R. Millán, and R. Chavarriaga, "Opportunistic human activity and context recognition," *Computer*, vol. 46, no. 2, pp. 36–45, Feb. 2013.

[14] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han, "Joint activity recognition and indoor localization with WiFi fingerprints," *IEEE Access*, vol. 7, pp. 80058–80068, 2019.

[15] M. Tammvee and G. Anbarjafari, "Human activity recognition-based path planning for autonomous vehicles," *Signal, Image Video Process.*, vol. 15, no. 4, pp. 809–816, Jun. 2021, doi: 10.1007/s11760-020-01800-6.

[16] C. Anderson, I. Hübener, A.-K. Seipp, S. Ohly, K. David, and V. Pejovic, "A survey of attention management systems in ubiquitous computing environments," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 2, pp. 1–27, Jul. 2018, doi: 10.1145/3214261.

[17] L. Babangida, T. Perumal, N. Mustapha, and R. Yaakob, "Internet of Things (IoT) based activity recognition strategies in smart homes: A review," *IEEE Sensors J.*, vol. 22, no. 9, pp. 8327–8336, May 2022.

[18] Y. Zhou, C. Xie, S. Sun, X. Zhang, and Y. Wang, "A self-supervised human activity recognition approach via body sensor networks in smart city," *IEEE Sensors J.*, vol. 24, no. 5, pp. 5476–5485, Jun. 2023.

[19] A. R. Javed, R. Faheem, M. Asim, T. Baker, and M. O. Beg, "A smartphone sensors-based personalized human activity recognition system for sustainable smart cities," *Sustain. Cities Soc.*, vol. 71, Aug. 2021, Art. no. 102970. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670721002560

[20] R. Liu, A. A. Ramli, H. Zhang, E. Henricson, and X. Liu, "An overview of human activity recognition using wearable sensors: Healthcare and artificial intelligence," in *Internet of Things—ICIOT 2021*, B. Tekinerdogan, Y. Wang, and L.-J. Zhang, Eds. Cham, Switzerland: Springer, 2021, pp. 1–14.

[21] Z. Hussain, Q. Z. Sheng, and W. E. Zhang, "A review and categorization of techniques on device-free human activity recognition," *J. Netw. Comput. Appl.*, vol. 167, Oct. 2020, Art. no. 102738, doi: 10.1016/j.jnca.2020.102738.

[22] O. C. Ann and L. B. Theng, "Human activity recognition: A review," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2014, pp. 389–393.

[23] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107561. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320320303642
[24] D. R. Beddiar, B. Nini, M. Sabokrou, and A. Hadid, "Vision-based human activity recognition: A survey," *Multimedia Tools Appl.*, vol. 79, nos. 41–42, pp. 30509–30555, Nov. 2020, doi: 10.1007/s11042-020-09004-3.
[25] D. Bouchabou, S. M. Nguyen, C. Lohr, B. LeDuc, and I. Kanellos, "A survey of human activity recognition in smart homes based on IoT sensors algorithms: Taxonomies, challenges, and opportunities with deep learning," *Sensors*, vol. 21, no. 18, p. 6037, Sep. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/18/6037
[26] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.
[27] Z. Li, J. Le Kernec, Q. Abbasi, F. Fioranelli, S. Yang, and O. Romain, "Radar-based human activity recognition with adaptive thresholding towards resource constrained platforms," *Sci. Rep.*, vol. 13, no. 1, p. 3473, Mar. 2023.
[28] Y. Qian, C. Chen, L. Tang, Y. Jia, and G. Cui, "Parallel LSTM-CNN network with radar multispectrogram for human activity recognition," *IEEE Sensors J.*, vol. 23, no. 2, pp. 1308–1317, Jan. 2023.
[29] C. Yin, J. Chen, X. Miao, H. Jiang, and D. Chen, "Device-free human activity recognition with low-resolution infrared array sensor using long short-term memory neural network," *Sensors*, vol. 21, no. 10, p. 3551, May 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/10/3551
[30] K. Kumari, P. H. Chandankhede, and A. S. Titarmare, "Design of human activity recognition system using body sensor networks," in *Proc. 6th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2021, pp. 1011–1016.
[31] T. Wang, S. Dai, Y. Liu, and T. T. Ye, "Battery-less sensing of body movements through differential backscattered RFID signals," *IEEE Sensors J.*, vol. 22, no. 9, pp. 8490–8498, May 2022.
[32] J. Lu and K.-Y. Tong, "Robust single accelerometer-based activity recognition using modified recurrence plot," *IEEE Sensors J.*, vol. 19, no. 15, pp. 6317–6324, Aug. 2019.
[33] A. K. M. Masum, E. H. Bahadur, A. Shan-A-Alahi, M. A. U. Z. Chowdhury, M. R. Uddin, and A. A. Noman, "Human activity recognition using accelerometer, gyroscope and magnetometer sensors: Deep neural network approaches," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–6.
[34] I. Koo, Y. Park, M. Jeong, and C. Kim, "Contrastive accelerometer–gyroscope embedding model for human activity recognition," *IEEE Sensors J.*, vol. 23, no. 1, pp. 506–513, Jan. 2023.
[35] S. Nasiri and M. R. Khosravani, "Progress and challenges in fabrication of wearable sensors for health monitoring," *Sens. Actuators A, Phys.*, vol. 312, Sep. 2020, Art. no. 112105.
[36] S. Zhang, Y. Li, S. Zhang, F. Shahabi, S. Xia, Y. Deng, and N. Alshurafa, "Deep learning in human activity recognition with wearable sensors: A review on advances," *Sensors*, vol. 22, no. 4, p. 1476, Feb. 2022.
[37] Z. E. Ashari, N. S. Chaytor, D. J. Cook, and H. Ghasemzadeh, "Memory-aware active learning in mobile sensing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 181–195, Jan. 2022.
[38] A. Y. Chervonenkis, *Early History of Support Vector Machines*. Berlin, Germany: Springer, 2013, pp. 13–20, doi: 10.1007/978-3-642-41136-6_3.
[39] L. Bottou, *Stochastic Gradient Descent Tricks*. Berlin, Germany: Springer, 2012, pp. 421–436, doi: 10.1007/978-3-642-35289-8_25.
[40] Z. Peng, W. Zhang, N. Han, X. Fang, P. Kang, and L. Teng, "Active transfer learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1022–1036, Apr. 2020.
[41] S. S. Raju, B. Wang, K. Mehta, M. Xiao, Y. Zhang, and H.-Y. Wong, "Application of noise to avoid overfitting in TCAD augmented machine learning," in *Proc. Int. Conf. Simulation Semiconductor Processes Devices (SISPAD)*, Sep. 2020, pp. 351–354.
[42] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 537–556, Sep. 2013.
[43] X. Wang, T.-K. Huang, and J. Schneider, "Active transfer learning under model shift," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, vol. 32, no. 2, pp. 1305–1313. [Online]. Available: https://proceedings.mlr.press/v32/wangi14.html
[44] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21st Int. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, 2013, pp. 437–442.
[45] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, Oct. 2010.
[46] B. Barshan and M. C. Yüksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *Comput. J.*, vol. 57, no. 11, pp. 1649–1667, Nov. 2014.
[47] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *Proc. Int. Workshop Hum. Behav. Understand.* Berlin, Germany: Springer, 2010, pp. 38–51.
[48] F. Palumbo, C. Gallicchio, R. Pucci, and A. Micheli, "Human activity recognition using multisensor data fusion based on reservoir computing," *J. Ambient Intell. Smart Environ.*, vol. 8, no. 2, pp. 87–107, Mar. 2016.
[49] R. Fallahzadeh, Z. E. Ashari, P. Alinia, and H. Ghasemzadeh, "Personalized activity recognition using partially available target data," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 374–388, Jan. 2023.
[50] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
[51] R. Fallahzadeh, M. Pedram, and H. Ghasemzadeh, "SmartSock: A wearable platform for context-aware assessment of ankle edema," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, pp. 6302–6306.
[52] Y. Nawal, M. Oussalah, B. Fergani, and A. Fleury, "New incremental SVM algorithms for human activity recognition in smart homes," *J. Ambient Intell. Hum. Comput.*, vol. 14, no. 10, pp. 13433–13450, Oct. 2023.
[53] D. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intell. Syst.*, vol. 27, no. 1, pp. 32–38, Jan. 2012.
[54] M. Schmitter-Edgecombe and D. J. Cook, "Assessing the quality of activities in a smart environment," *Methods Inf. Med.*, vol. 48, no. 5, pp. 480–485, 2009.
[55] M. A. U. Alam and N. Roy, "Unseen activity recognitions: A hierarchical active transfer learning approach," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 436–446.
[56] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly supervised recognition of daily life activities with wearable sensors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2521–2537, Dec. 2011.
[57] M. Hasan, S. Paul, A. I. Mourikis, and A. K. Roy-Chowdhury, "Context-aware query selection for active learning in event recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 554–567, Mar. 2020.
[58] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Mach. Vis. Appl.*, vol. 24, no. 5, pp. 971–981, Jul. 2013.
[59] S. Oh et al., "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. CVPR*, Jun. 2011, pp. 3153–3160.
[60] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing video events with goal inference and intent prediction," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 487–494.
[61] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik, "AVA: A video dataset of spatio-temporally localized atomic visual actions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6047–6056.
[62] M. Hasan and A. K. Roy-Chowdhury, "Context aware active learning of activity recognition models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4543–4551.
[63] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1194–1201.
[64] Y. Qu, R. K. Baghbaderani, W. Li, L. Gao, Y. Zhang, and H. Qi, "Physically constrained transfer learning through shared abundance space for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 12, pp. 10455–10472, Dec. 2021.
[65] M. Graña, M. A. Veganzons, and B. Ayerdi. *Hyperspectral Remote Sensing Scenes*. Accessed: Jan. 24, 2023. [Online]. Available: https://ehu.eus/ccwintco/index.php
[66] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. van Kasteren, W. Liao, R. Bellens, A. Pižurica, S. Gautama, W. Philips, S. Prasad, Q. Du, and F. Pacifici, "Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2405–2418, Jun. 2014.

[67] B. Le Saux, N. Yokoya, R. Hansch, and S. Prasad, "2018 IEEE GRSS data fusion contest: Multimodal land use classification [technical committees]," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 1, pp. 52–54, Mar. 2018.

[68] U. Ahmed, S. K. Mukhiya, G. Srivastava, Y. Lamo, and J. C.-W. Lin, "Attention-based deep entropy active learning using lexical algorithm for mental health treatment," *Frontiers Psychol.*, vol. 12, Mar. 2021.

[69] S. K. Mukhiya, U. Ahmed, F. Rabbi, K. I. Pun, and Y. Lamo, "Adaptation of IDPT system based on patient-authored text data using NLP," in *Proc. IEEE 33rd Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Jul. 2020, pp. 226–232.

[70] K. G. M. Chathuramali and R. Rodrigo, "Faster human activity recognition with SVM," in *Proc. Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, Dec. 2012, pp. 197–203.

[71] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Oct. 2005, pp. 1395–1402.

[72] D. Tran and A. Sorokin, "Human activity recognition with metric learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, Oct. 2008, pp. 548–561.

[73] M. Ye, Y. Qian, J. Zhou, and Y. Y. Tang, "Dictionary learning-based feature-level domain adaptation for cross-scene hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1544–1562, Mar. 2017.

[74] S. A. Rokni, M. Nourollahi, P. Alinia, I. Mirzadeh, M. Pedram, and H. Ghasemzadeh, "TransNet: Minimally supervised deep transfer learning for dynamic adaptation of wearable systems," *ACM Trans. Design Autom. Electron. Syst.*, vol. 26, no. 1, pp. 1–31, Jan. 2021.

[75] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.

[76] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. D. R. Millàn, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.

[77] J. S. Deshmukh and A. K. Tripathy, "Entropy based classifier for cross-domain opinion mining," *Appl. Comput. Informat.*, vol. 14, no. 1, pp. 55–64, Jan. 2018.

[78] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meet. Assoc. Comput. Linguist.*, 2007, pp. 440–447.

[79] X. Tang, B. Du, J. Huang, Z. Wang, and L. Zhang, "On combining active and transfer learning for medical data classification," *IET Comput. Vis.*, vol. 13, no. 2, pp. 194–205, Mar. 2019.

[80] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[81] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *J. Digit. Imag.*, vol. 26, no. 6, pp. 1045–1057, Dec. 2013.

[82] T.-K. Huang, L. Li, A. Vartanian, S. Amershi, and X. Zhu, "Active learning with Oracle epiphany," in *Proc. 30th Conf. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, May 2016, pp. 2820–2828. [Online]. Available: https://www.microsoft.com/en-us/research/publication/active-learning-oracle-epiphany/

[83] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Mach. Learn.*, vol. 15, no. 2, pp. 201–221, May 1994.

[84] T.-K. Huang, A. Agarwal, D. J. Hsu, J. Langford, and R. E. Schapire, "Efficient and parsimonious agnostic active learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2755–2763.

[85] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 62–74, Oct. 2017.

[86] Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel, "ExtraSensory app: Data collection in-the-wild with rich user interface to self-report behavior," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2018, pp. 1–12.

[87] *SVC—scikit-learn.org*. Accessed: Aug. 12, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

UMAIR ALAM received the degree in computer science from COMSATS University Islamabad (CUI), Sahiwal, in 2019, and the Master of Science in Computer Science (MSCS) degree from the Department of Computer Science, FAST National University of Computer and Emerging Sciences (NUCES), Lahore, Pakistan, in 2023. He also has experience of teaching CS courses, from 2019 to 2020, as a College Teaching Intern. His main research interests include machine learning and smart sensing devices.

ASMA AHMAD FARHAN received the Ph.D. degree in CSE from the University of Connecticut, USA, in 2016. Currently, she is an Assistant Professor with the National University of Computers and Emerging Sciences, Lahore, Pakistan. Her research interests include human behavior modeling, sensors, machine learning, and ubiquitous computing. She received a Taylor L. Booth Fellowship while working as a Ph.D. Student.

SUMMRINA KANWAL is currently an Academician and a Researcher with many years of well-honed expertise, she is committed to innovation and resourcefulness. Her research and development revolve around designing intricate algorithms based on deep-dive statistical analysis and predictive data modeling for developing efficient decision support systems (DSS) using machine learning, deep learning, image processing, pattern recognition, and optimization techniques.

NASSER ALLHEEIB received the master's degree in information technology from the School of Information Technology, Rochester Institute of Technology, Rochester, NY, USA, in 2014, and the Ph.D. degree from the Clayton School of Information Technology, Monash University, Australia, in 2021. He is currently an Assistant Professor with the College of Computer and Information Sciences, King Saud University. His research interests include artificial intelligence (AI), spatial databases, and location-based services.