HARNet: Towards On-Device Incremental Learning using Deep Ensembles on Constrained Devices

Prahalathan Sundaramoorthy
Solarillion Foundation
Chennai, India
prahalath27@gmail.com

Gautham Krishna Gudur
Solarillion Foundation
Chennai, India
gauthamkrishna.gudur@gmail.com

Manav Rajiv Moorthy
SSN College of Engineering
Chennai, India
manav15057@cse.ssn.edu.in

R Nidhi Bhandari
SSN College of Engineering
Chennai, India
nidhi@cse.ssn.edu.in

Vineeth Vijayaraghavan
Solarillion Foundation
Chennai, India
vineethv@ieee.org

ABSTRACT

Recent advancements in the domain of pervasive computing have seen the incorporation of sensor-based Deep Learning algorithms in Human Activity Recognition (HAR). Contemporary Deep Learning models are engineered to alleviate the difficulties posed by conventional Machine Learning algorithms which require extensive domain knowledge to obtain heuristic hand-crafted features. Upon training and deployment of these Deep Learning models on ubiquitous mobile/embedded devices, it must be ensured that the model adheres to their computation and memory limitations, in addition to addressing the various mobile- and user-based heterogeneities prevalent in actuality. To handle this, we propose HARNet - a resource-efficient and computationally viable network to enable on-line Incremental Learning and User Adaptability as a mitigation technique for anomalous user behavior in HAR. Heterogeneity Activity Recognition Dataset was used to evaluate HARNet and other proposed variants by utilizing acceleration data acquired from diverse mobile platforms across three different modes from a practical application perspective. We perform Decimation as a Down-sampling technique for generalizing sampling frequencies across mobile devices, and Discrete Wavelet Transform for preserving information across frequency and time. Systematic evaluation of HARNet on User Adaptability yields an increase in accuracy by ~35% by leveraging the model's capability to extract discriminative features across activities in heterogeneous environments.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile devices**; **Empirical studies in ubiquitous and mobile computing**; • **Computing methodologies → Neural networks**; *Ensemble methods*;

KEYWORDS

Activity Recognition; Deep Learning; Incremental Learning

EMDL'18, June 15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5844-6/18/06…$15.00
https://doi.org/10.1145/3212725.3212728

ACM Reference Format:
Prahalathan Sundaramoorthy, Gautham Krishna Gudur, Manav Rajiv Moorthy, R Nidhi Bhandari, and Vineeth Vijayaraghavan. 2018. HARNet: Towards On-Device Incremental Learning using Deep Ensembles on Constrained Devices. In *EMDL'18: 2nd International Workshop on Embedded and Mobile Deep Learning, June 15, 2018, Munich, Germany.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3212725.3212728

1 INTRODUCTION

The ubiquitous proliferation of low-cost mobile devices with embedded sensors has spawned a growing research in extracting contextual information from sensor data, particularly for HAR, owing to its applications in healthcare, physical activity monitoring, fitness tracking, behavioral analysis, etc. [3][16]. The contemporary evolution of Machine Learning has provided a convenient way for exploiting these raw sensor data to abstract meaningful information. However, employing Machine Learning algorithms generally requires extensive domain knowledge for feature engineering, which is often limited by the competency of the human designing the model.

Recently, the emergence of sophisticated Deep Learning techniques has greatly alleviated the problem of crafting shallow features that have questionable generalizability. Powerful Deep Learning methods aid in automatic extraction of discriminative features by exploring hidden correlations within and between data, thereby capturing intricate details that are crucial for achieving high-level classification efficacy and robustness. However, learning complex features generally involves training models that require extensive resources, thereby making them unfriendly for real-world deployment on lightweight wearables. Furthermore, the device- and user-related diversities, such as different sensor types, device orientations, varied user-behavior, CPU loads etc., oftentimes hamper the real-world performance of the model [17]. This brings about a growing necessity for developing resource-friendly and robust HAR systems that leverage mutual interaction between the model and data, optimized to achieve state-of-the-art accuracies.

In this paper, we intend to address the following two prominent challenges in HAR.

On-device Incremental Learning

Most deep learning HAR systems are often trained on remote servers off-line or via cloud. To facilitate *User Adaptability* through *Incremental Learning* - a technique to enhance the performance of these

models by catering to each user independently, the raw inertial data needs to be transmitted to the servers from the device. However, communication between the server and device is often compromised due to latency issues (Round Trip Time taken between the server and device), and overheads in synchronization of data. One possible approach to achieve User Adaptability is to ensure training can be performed on the resource-constrained mobile/embedded systems, provided that the model is optimized.

Heterogeneity

When a HAR system is tested on multiple smartphones in real-world, the performance across various users is generally sub-optimal when compared to its simulated environment. This is due to the presence of various mobile-sensing heterogeneities prevalent during deployment. These heterogeneities predominantly include varying sampling rates, sampling rate instability due to different OS types, CPU load conditions and varied user characteristics among others [17].

In this paper, we focus on systematic minimization of resources to develop a generic HAR model in heterogeneous conditions that can be effectively trained and deployed on a Mobile/Embedded platform, whilst achieving on-par accuracies compared to state-of-the-art recognition models.

2 RELATED WORK

Modeling Deep Learning architectures for HAR has been an extensive area of research. Many researchers predominantly use Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Restricted Boltzmann Machines (RBMs) and Deep Belief Networks (DBNs) or a combination of these to build their recognition models [18].

Jiang et al. [8] effectively used Deep CNN to learn local features across dimensions from a synthesized activity image. Although this approach yields appreciable results, creating and analyzing an activity image is laborious and memory consuming, which might not be well-suited for deployment on mobile and embedded platforms. Ronao et al. [15] successfully demonstrated the usage of two-dimensional CNN for efficient classification of activities. However, the model was tested on a dataset that contains data from a single smartphone, thereby failing to showcase its generalizing capabilities across devices. Ravi et al. [14] performed temporal convolutions on Short-Time Fourier Transform (STFT) spectrogram of the input signals. Though the system was deployed on low-cost wearable devices, its capability to perform well for unseen users is not pronounced.

To learn hierarchial features, Guan et al. [4] proposed an ensemble of LSTM learners for building a robust recognition system. RBMs and multi-layer RBMs [13] have also been used to capture local and multi-modal interactions in HAR. Ordóñez et al. [12] and Hammerla et al. [5] exploit their own convolutional and recurrent network architectures, but fail to illustrate the performance of the same under real-world heterogeneous environments. Implementing hybrid models using a combination of CNNs and RNNs has been proposed by Yao et al. [19] in DeepSense, which fuses data from multiple sensor modalities while also incorporating temporal relations. Although these works achieve impressive results in terms of accuracy and classification time, the feasibility of incremental learning seems to be debatable.

The rest of the paper is organized as follows: Section 3 and Section 4 discuss the dataset and various preprocessing steps to achieve dataset reduction for reducing memory overhead in devices. We elucidate our proposed model in Section 5, followed by systematic evaluation and demonstration under heterogeneous and resource-constrained scenarios in Sections 6 and 7.

3 DATASET

We utilize the Heterogeneity dataset (D_H) proposed by Allan et al. [17] to design our model. D_H consists of inertial values recorded from accelerometer and gyroscope present in eight smartphones across nine users performing six daily activities (Table 1). To ensure uniformity, each activity was performed for five minutes by all users across all phones. The real-world sensing diversities are reflected by data from different phones operating with varying Sampling Frequencies (F_S, ranging from 50-200 Hz) in D_H.

Table 1: Heterogeneity Dataset (D_H) characterized by their respective attributes

Activities	Devices	F_S	Users
['Biking', 'Sitting', 'Standing', 'Walking', 'StairsUp', 'StairsDown']	Nexus 4 Samsung S3 Samsung S3 Mini Samsung S+	200 150 100 50	[a, b, c, d, e, f, g, h, i]

4 DATASET PREPROCESSING

In order to handle the varying sampling frequencies of different devices and to obtain a rich yet sparse representation of the signal components, we perform the following preprocessing steps.

4.1 Windowing and Decimation

We initially segment raw inertial data into non-overlapping two-second activity windows (w_a). These segmented chunks of data have non-uniform lengths due to varying sampling frequencies of the devices (F_s). This disparity may impede the performance of the model, particularly when F_s of smartphones are not identical during training and testing. To handle this issue, *Up-sampling* or *Down-sampling* of data can be performed to ensure that each window w_a has a fixed size. Up-sampling is likely to induce noise in the data and increase memory requirements [17]. Hence, a better approach would be to down-sample the signals to a common sampling frequency, as the characteristics of the input signals are likely to be retained, while resulting in data size reduction.

The authors perform *Decimation* - a technique that down-samples a signal, by applying an 8^{th} order Chebyshev type-I filter without any phase shift. In D_H, we choose the lowest F_s - 50 Hz as the common sampling frequency for down-sampling. Decimation is performed on all signals from mobile devices whose F_s is greater than 50 Hz, thereby ensuring consistency in size of each window w_a. Decimation results in data reduction upto 75% for phones with highest F_s - 200Hz.

4.2 Discrete Wavelet Transform

A better representation of the raw inertial signals can be obtained by capturing both temporal and frequency information, which retain locally well-defined temporal characteristics in the frequency domain [11].

DWT convolves the incoming signal *x(n)* with a wavelet ψ by using multiple filter banks to achieve decomposition into high- and low-frequency components. These components are represented by *Detail* (C_D) and *Approximation* (C_A) coefficients. By discarding C_D and utilizing C_A, we get a smoothened version of *x(n)* [2]. This process yields a sparse representation of the signal, thereby compressing the size of the data (~50%).

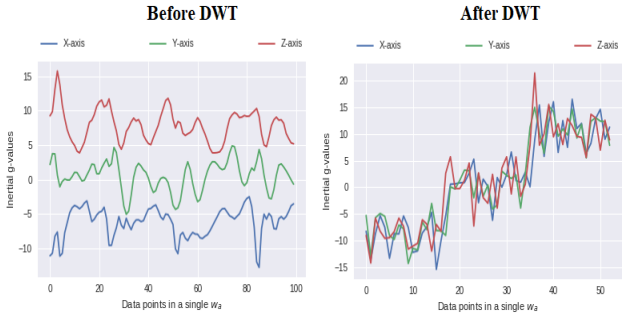

Figure 1: Inertial accelerometer signals of the three axes before and after DWT

From Figure 1, we can visually interpret the enhanced correlation between inertial *g*-values of the three axes after performing DWT by utilizing the temporal and frequency information captured.

5 MODEL

HAR in real-time requires identification of discriminative sets of features for effective classification. Deep Learning facilitates automatic extraction of such distinctive features, that otherwise do not generalize across datasets. Comprehensive analysis of correlations across various axes is essential to learn such features efficiently. Hence, in this paper, we study various architectures to extract the intra-axial and inter-axial feature dependencies.

INTRA-AXIAL DEPENDENCIES

Each activity window w_a is constituted of $\{w_a^X, w_a^Y, w_a^Z\}$ denoting the input vectors across axes X, Y and Z respectively. These vectors are represented by the frequency sub-bands of C_A. To extract the local information within each vector in $\{w_a^X, w_a^Y, w_a^Z\}$, we systematically evaluate the following intra-axial variants as shown in Figure 2.

Conv-1D. Convolutional Neural Networks (CNNs) are used as powerful feature learning tools in many Deep Learning classification tasks. Each convolutional kernel analyzes and extracts local characteristics within each input axis. We combine the learned features of all axes for achieving distinctive representation of the incoming sensor signal for classification.

The intra-axial CNN takes an input vector at the first layer L_1^C from $\{w_a^X, w_a^Y, w_a^Z\}$. Each layer L_n^C provides a feature-map $f_{L_n^C}$ as

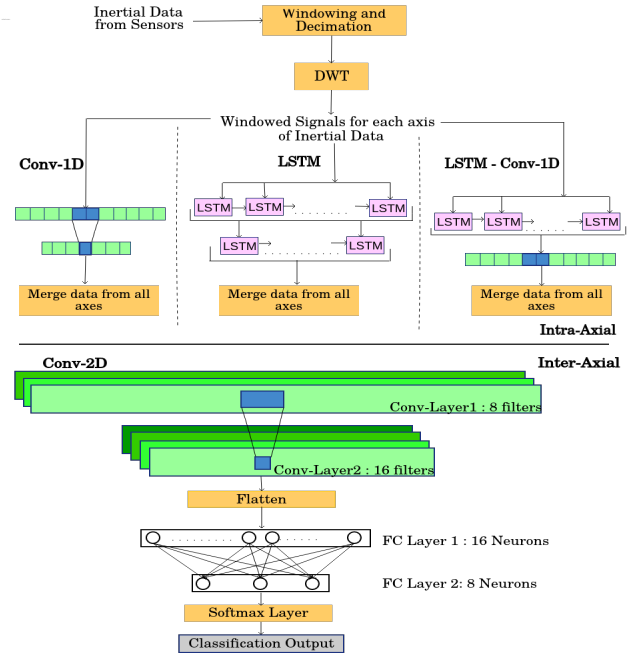

Figure 2: Model Architecture Variants

the input for every subsequent layer L_{n+1}^C in the network. Each feature map is obtained using convolutional filters applied throughout the feature map $f_{L_n^C}$ and is provided as input to the hidden layer L_{n+1}^C.

In this variant, we utilize a two-layer stacked convolutional network with a receptive field size (kernel size) of 2. To regularize each mini-batch and reduce the internal covariate shift, a Batch Normalization layer is also used in each stack [7]. Each batch-norm layer is followed by a Max-pooling layer of pool size 2x2.

LSTM. The Long Short-Term Memory (LSTM) units are one of the extended variants for vanilla recurrent networks. LSTMs have proven to be successful in capturing pattern information in time-series data, as they have the potential to model dynamic temporal behavior [6].

Recurrent units of the first layer utilize the input vectors $\{w_a^X, w_a^Y, w_a^Z\}$ to learn the local temporal characteristics. The input of the each following hidden recurrent layer L_n^R is of the form $d = \bigcup_{i=0}^{N} d_i$, where N is the number of units of the previous recurrent layer L_{n-1}^R. These sequences will be modeled for each timestep d_t by remembering the states for the previous d_{t-1} timesteps, $\forall t \in [0, N]$.

We use a two-layer stacked LSTM network in this variant, comprising of 32 and 20 output cells each. A Hyperbolic Tangent (*tanh*) activation function is used for the same.

LSTM → Conv-1D. By using LSTMs for modeling complex temporal relations and 1-D convolutions for extracting the most salient features from these functions pertaining to each axis, we could obtain a well-learned representation of the input signal.

Table 2: Total parameters, Accuracies, F1-Scores and Time taken for Classification of a single window for D_H across all models in mode M_U

Model	Params	Accuracy	F1-Score	Time (in ms)
HAR-CNet	31,806	95.68	0.9619	**10.9**
HAR-LNet	29,910	95.42	0.9573	850.2
HAR-LCNet	40,094	**96.79**	**0.9651**	68.9

This proposed framework utilizes a combination of layers from both LSTM and Conv-1D. Inputs $\{w_a^X, w_a^Y, w_a^Z\}$ are initially fed into a recurrent network from which a modeled sequence $f_{L_n^R}$ is obtained, which is further used to generate feature maps using 1-D convolutions. The Convolutional layer L_n^C is stacked over the final recurrent layer L_n^R, and takes the input $f_{L_n^R}$ to provide a feature-extracted vector $f_{L_n^C}$ per axis.

In this variant, we propose a one-dimensional convolutional layer comprising of 8 filters and a kernel size of 2 over an LSTM layer similar to the aforementioned variant, with a Batch Normalization regularizer and a 2x2 pooling layer.

INTER-AXIAL DEPENDENCIES

A two-dimensional CNN can effectively learn distinctive characteristics across spatial dimensions [10]. We aim to capture the interactions between data from the three axes, using convolutional layers.

The outputs of the intra-axial models for all three input vectors $\{w_a^X, w_a^Y, w_a^Z\}$ are concatenated to form a feature matrix F, which provides a sophisticated representation from which inter-axial dependencies can be easily correlated.

In this paper, we propose an inter-axial model - a two-layer stacked 2-D CNN with convolutional layers comprising of 8 and 16 filters each and a receptive field of size 3x3. Each convolutional layer is followed by a Batch Normalization and a Pooling layer of size (3x2). This stacked network is followed by two Fully-Connected (FC) layers constituting 16 and 8 neurons each with Rectified Linear Unit (ReLU) activations. The Dropout regularization technique is applied after each Fully-Connected layer with a probability of 0.25. Negative log-likelihood (Softmax) probability estimations are used for classification of activities.

The intra-axial patterns and inter-axial interactions together will enable extensive analysis and modeling of activities. Using deep ensembles of the intra-axial variants with the inter-axial model provide an all-encompassing and rich representation of the input signals.

In this work, we thus propose the following *HARNet* variants:

•***HAR-CNet*** : $\{Conv\text{-}1D \rightarrow Conv\text{-}2D\}$

•***HAR-LNet*** : $\{LSTM \rightarrow Conv\text{-}2D\}$

•***HAR-LCNet*** : $\{LSTM \rightarrow Conv\text{-}1D \rightarrow Conv\text{-}2D\}$

Parametric optimization of convolutional filters and kernel size, recurrent cells and FC neurons drastically reduces the memory and time complexities. Significant reduction of such parameters in each layer enables efficient memory management on a resource-constrained platform. We recursively prune the model parameters to systematically arrive at the optimal proposed variants, while not compromising on recognition accuracy. Introducing Dropout between FC layers further reduces the number of parameters, thereby enabling successful on-device training and deployment on constrained devices.

We formalize our ensembled deep model using the TensorFlow module [1]. The network is trained with a learning rate of $2e^{-4}$ to minimize the *categorical cross-entropy loss* (ρ) as shown below.

$$\rho = -\sum_{k=1}^{K} y_{i,k} \log(x_{i,k}) \tag{1}$$

where x are the predictions, y are the target values, i denotes the data points from w_a across K classes. This loss ρ is optimized during back-propagation for each mini-batch using the Adam optimizer [9].

6 EXPERIMENTS AND RESULTS

The evaluation modes of HAR algorithms are crucial in quantifying its extensibility and generalizability during real-world deployment. We evaluate the performance of our proposed models across the three modes stated below.

•*Mixed User Mode* (M_U)

This is one of the most commonly used evaluation modes in HAR. In this mode, the whole dataset is split into stratified samples of 80% train and 20% test data.

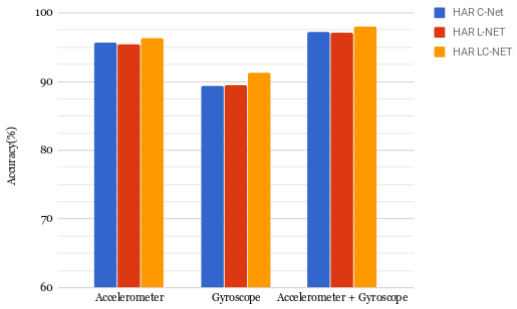

Figure 3: Sensor Minimization: A Comparative Analysis of Accuracies for Accelerometer and Gyroscope

Sensor Minimization: We analyze the classification accuracies of our model variants using data from different combinations of both accelerometer and gyroscope. From Fig. 3, we observe that the accuracies obtained when using data from both accelerometer and gyroscope do not significantly exceed those obtained by using just the accelerometer data (~ 1.5%). We thus perform sensor minimization to address the challenge of On-Device Incremental Learning by forgoing data from gyroscope, which substantially reduces memory requirements by 50% and computational cost for each input vector. We hence use the data from accelerometer alone for further analysis of our models.

The results for mode M_U on dataset D_H are showcased in Table 2. We observe that HAR-LCNet outperforms the other two variants in terms of accuracy and F1-score, as it exploits a combination of recurrent and convolutional networks. Each recurrent unit preserves the observed patterns in accelerometer data over time across each axis by utilizing a common weight matrix W, which encapsulates the diversity in instances of the same activity. Using convolutional filters over these modeled sequences then provides a rich feature-set from which the model can learn effectively.

	'Stand'	'Sit'	'Walk'	'Stairsup'	'Stairsdown'	'Bike'
'Stand'	99.28	0.72	0	0	0	0
'Sit'	0.12	99.88	0	0	0	0
'Walk'	0	0	90.19	3.76	6.05	0
'Stairsup'	0	0	4.48	87.75	6.92	0.85
'Stairsdown'	0	0	4.16	5.27	90.57	0
'Bike'	0.73	0	0	0.73	0.48	98.06

Figure 4: Confusion Matrix for *HARNet* in Mode M_U

Upon comparing HAR-LCNet with the next-best performing model HAR-CNet, we observe that HAR-CNet is ~7x faster than HAR-LCNet in terms of inference time per sample, with a ~1% difference in accuracy and F1-score. Considering the resource-constrained nature of mobile/embedded platforms, we narrow down to HAR-CNet as our final ensembled deep framework - ***HARNet***, which gives high accuracy with least classification time. The confusion matrix of *HARNet* is shown in Figure 4. Majority of the misclassification occurs between the classes : 'StairsUp', 'StairsDown' and 'Walking', which can be attributed due to the lack of orientation details of the smartphone, that is traditionally captured by the gyroscope.

•*Device Independent Mode* (D_I)

This mode aims at evaluating the model's performance across various devices, thereby reflecting its capability to deal with various mobile-sensing heterogeneities when deployed in a real-world scenario. A stratified k-fold cross validation technique is employed for a *Leave-One-Device-Out* approach. The average accuracy and F1-score obtained are 89.5% and 0.887 respectively. Figure 5 illustrates the accuracy of our model across devices, thereby showcasing its generalizing capabilities.

•*User Independent Mode* (U_I)

In this mode, we attempt to classify activities performed by a previously unseen user through the *Leave-One-User-Out* approach. This mode hence provides a strong measurement of generalizability of the model across diverse users. A similar cross validation technique as mentioned above is used. Testing in this mode yields an F1-score of 0.80 using the accelerometer data alone which is higher than the F1-scores presented in [20] and [17], which use data from both accelerometer and gyroscope.

We analyze the relation between the number of epochs and classification accuracies for two specific users: 'b' and 'c', for whom the best and least accuracy are observed. We can infer that user 'b'

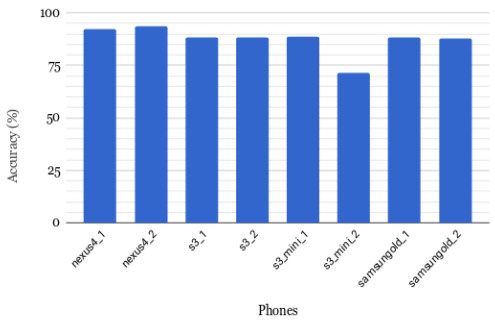

Figure 5: Mode D_I: Comparative Analysis of Accuracies across various devices

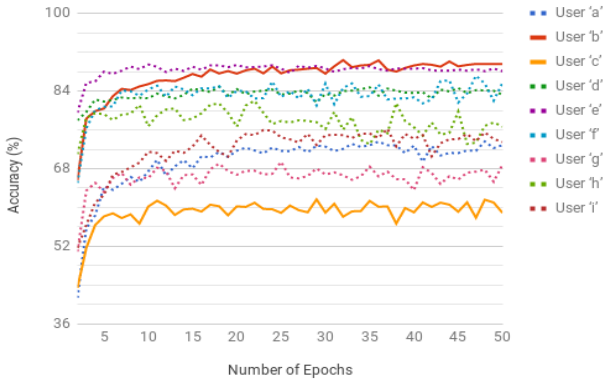

Figure 6: Mode U_I: Comparative Analysis of Accuracies vs number of Epochs for Users 'a' through 'i'

performs activities similar to the general trend as high accuracy is observed for the same. However, user 'c' achieves least accuracy which can be attributed to the user's unique physical build, posture and execution of activities. It is evident from Figure 6 that even though the number of epochs is increased during training phase, the model does not yield better accuracies for user 'c'. To enhance the efficiencies of such users, the model should adapt to the user's unique behavioral pattern. We hence perform *Incremental Learning* by using a portion of the data from the unseen test user to update the weights of the previously trained model, thereby adapting to even the least-performing users.

7 ON-LINE INCREMENTAL LEARNING

We experiment user-based Incremental Learning using *HARNet* for the users 'b' and 'c' by deploying the system on a Raspberry Pi 3 Model B. The model is initially trained in mode U_I, and the trained weights and parameters are stocked on Raspberry Pi. The portion of the unseen user data included for Incremental Learning is governed by the adaption factor λ. We first assign λ=0.25 for both users and observe the accuracy change. As shown in Figure 7, the accuracies

of the users increased after performing Incremental Learning, particularly for user 'c', where there is a substantial increase in accuracy of ~35%.

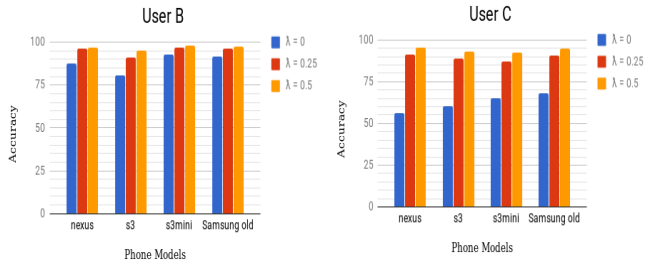

Figure 7: Incremental Learning for User with Best and Least Accuracy

When there is an influx of a stream of data (λ increases) for a particular user, the model adapts itself well to the user's behavioral pattern, thus leading to higher accuracies. Table 3 illustrates the time taken for preprocessing and testing phases per activity window on the Raspberry Pi. The user-based incremental learning on Raspberry Pi takes 3 seconds per epoch. It is evident that inference time per activity window w_a is attributed to the size of the model (~0.5 MB). Furthermore, the time taken for preprocessing and testing together ensures the computational viability of the proposed methodology on embedded and mobile platforms.

Table 3: Time taken for Execution per activity window (w_a)

Process	Computational Time
Inference time	17 ms
Discrete Wavelet Transform	0.5 ms
Decimation	4.8 ms

8 CONCLUSION

In this paper, we proposed *HARNet* - a Deep Learning framework with capabilities to handle various mobile-sensing and user-based heterogeneities while being resource-friendly on low-cost embedded and mobile platforms. By systematically optimizing the data preprocessing and model design phases, we were able to achieve remarkable accuracies using *HARNet*, which has a size of ~0.5 MB. Thus, the authors were able to perform Incremental Learning on Raspberry Pi 3 to facilitate User Adaptability, which proves beneficial for anomalous users. Notably, an increase in accuracy of ~35% was achieved signifying the feasibility of *HARNet* on embedded and mobile devices.

9 ACKNOWLEDGEMENT

The authors would like to thank Solarillion Foundation for its support and funding of the research work carried out.

REFERENCES

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Borovykh, A., Bohte, S., and Oosterlee, C. W. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691* (2017).

[3] Buttussi, F., and Chittaro, L. Mopet: A context-aware and user-adaptive wearable system for fitness training. *Artif. Intell. Med. 42*, 2 (Feb. 2008), 153–163.

[4] Guan, Y., and Plötz, T. Ensembles of deep lstm learners for activity recognition using wearables. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1*, 2 (June 2017), 11:1–11:28.

[5] Hammerla, N. Y., Halloran, S., and Plötz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (2016), IJCAI'16, AAAI Press, pp. 1533–1540.

[6] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Comput. 9*, 8 (Nov. 1997), 1735–1780.

[7] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37* (2015), ICML'15, JMLR.org, pp. 448–456.

[8] Jiang, W., and Yin, Z. Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM International Conference on Multimedia* (New York, NY, USA, 2015), MM '15, ACM, pp. 1307–1310.

[9] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[10] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (USA, 2012), NIPS'12, Curran Associates Inc., pp. 1097–1105.

[11] Najafi, B., Aminian, K., Paraschiv-Ionescu, A., Loew, F., Bula, C. J., and Robert, P. Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on Biomedical Engineering 50*, 6 (June 2003), 711–723.

[12] Ordóñez, F. J., and Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors 16*, 1 (2016).

[13] Radu, V., Lane, N. D., Bhattacharya, S., Mascolo, C., Marina, M. K., and Kawsar, F. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (New York, NY, USA, 2016), UbiComp '16, ACM, pp. 185–188.

[14] Ravi, D., Wong, C., Lo, B., and Yang, G. Z. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)* (June 2016), pp. 71–76.

[15] Ronao, C. A., and Cho, S.-B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl. 59*, C (Oct. 2016), 235–244.

[16] Shoaib, M., Bosch, S., Incel, O. D., Scholten, H., and Havinga, P. J. M. Fusion of smartphone motion sensors for physical activity recognition. *Sensors 14*, 6 (2014), 10146–10176.

[17] Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., Sonne, T., and Jensen, M. M. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems* (New York, NY, USA, 2015), SenSys '15, ACM, pp. 127–140.

[18] Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L. Deep learning for sensor-based activity recognition: A survey. *arXiv preprint arXiv:1707.03502* (2017).

[19] Yao, S., Hu, S., Zhao, Y., Zhang, A., and Abdelzaher, T. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, Switzerland, 2017), WWW '17, International World Wide Web Conferences Steering Committee, pp. 351–360.

[20] Yao, S., Zhao, Y., Shao, H., Zhang, A., Zhang, C., Li, S., and Abdelzaher, T. Rdeepsense: Reliable deep mobile computing models with uncertainty estimations. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1*, 4 (Jan. 2018), 173:1–173:26.