Studienarbeit Final Report

Enhancing Activity Recognition

Hruday Kumar Kolla

1654541

Guide:

Mr. Marius Bock

University of Siegen

Faculty IV: School of Science and Technology

Department of Computer Science

Ubiquitous Computing Group & Computer Vision Group

January 2, 2024

Confirmation

I hereby confirm that this report is entirely my own work and that I have not used any additional assistance or resources other than those indicated. All quotations, paraphrases, information and ideas that have been taken from other sources (including the Internet as well as other electronic sources) and other persons' work have been cited appropriately and provided with the corresponding bibliographical references. The same is true of all drawings, sketches, pictures and other illustrations that appear in the text. I am aware that the neglect to indicate the used sources is considered as fraud and plagiarism in which case sanctions are imposed that can lead to the suspension or permanent expulsion of students in serious cases.

Place, Date	Signature

Contents

1. Abstract

Human Activity Recognition (HAR) plays a crucial role in understanding and interpreting human behaviour, especially in dynamic and real-world settings. HAR involves the use of sensor data to identify and classify different activities performed by individuals, contributing to applications such as health monitoring, and sports analysis. In their recent work, Bock et al. introduced the WEAR dataset, a groundbreaking resource designed for outdoor sports activity recognition using wearable devices and egocentric vision [2]. Leveraging this dataset, they demonstrated the effectiveness of a Deep Convolutional LSTM network (DeepConvLSTM) with inertial data. Furthermore, Bock et al. explored Temporal Action Detection (TAD) vision networks, including ActionFormer and TriDet, incorporating various combinations of inertial and Inflated 3D (I3D) features [1, 13, 11, 4, 8].

This report builds upon Bock et al.'s work, presenting a novel approach to further enhance Human Activity Recognition methodologies. Three significant contributions are outlined. Firstly, the report introduces a unique methodology for leveraging pre-trained Convolutional and LSTM inertial features derived from the DeepConvLSTM network in the training of the ActionFormer and the TriDet models. Secondly, meticulous benchmarking and performance analysis are conducted on both the WEAR dataset and the Opportunity Daily Activity runs dataset (Opportunity ADL) [10], exploring various combinations of inertial features. Lastly, a comprehensive methodology for preprocessing HAR datasets is proposed, underscoring its pivotal role in refining model performance. The results demonstrate that the integration of pre-trained Convolutional and LSTM inertial features significantly enhances the performance metrics of the ActionFormer and TriDet models. Notably, the TriDet model consistently outperforms the ActionFormer model across diverse inertial feature types, showcasing improvements in F1 scores and mean Average Precision (mAP). Additionally, the preprocessing methodology contributes to the overall effectiveness of the models, emphasizing the importance of thoughtful data preparation in HAR tasks. The Code to reproduce experiments is publicly available via: `https://github.com/hrudaykolla/wear`.

2. Introduction

Human activity recognition (HAR) uses technology, such as sensors and machine learning algorithms, to identify and interpret human actions and behaviours. HAR has diverse applications across various sectors, revolutionizing industries and applications for enhancing user experiences. In healthcare, HAR is utilized for monitoring patient movements and detecting anomalies, facilitating remote patient care and rehabilitation. In the field of sports and fitness, it helps track and analyze athletic performance, providing valuable insights for training optimization. In smart homes, HAR contributes to the development of intelligent systems that adapt to occupants' behaviours, optimizing energy consumption and enhancing security. In the workplace, HAR enhances safety protocols by monitoring employee activities and identifying potential hazards. The versatility of human activity recognition underscores its potential to reshape and improve various aspects of our daily lives across different domains.

The **WEAR dataset** by Bock et al.[2] is designed for outdoor sports Human Activity Recognition (HAR) and encompasses a diverse range of workout activities undertaken by 18 participants. These individuals wore inertial sensors on their wrists and ankles, alongside a head-mounted camera capturing egocentric vision with a wide field of view, as illustrated in Figure 1.

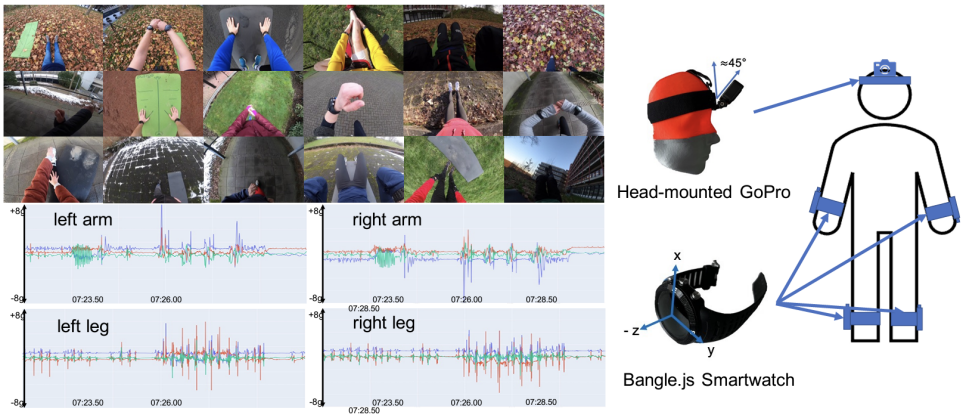

Figure 1.: Overall setup and sample data captured by sensors featured in the WEAR dataset. Participants were equipped with four Bangle.js Version1 open-source smart watches(one on each limb) and a head-mounted GoPro Hero8. The camera was tilted at a 45-degree angle and the axis orientation of the smartwatches was fixed across all subjects.

On a parallel note, the **Opportunity dataset** serves as a benchmark for HAR using data from Wearable, Object, and Ambient Sensors [10]. Comprising four subjects, this dataset includes drills and daily activity runs (ADL). The Opportunity dataset also presents a challenging scenario with its diverse set of sensors and activities. In summary, both the WEAR and Opportunity datasets offer valuable insights into the realm of Human Activity Recognition, providing researchers with rich and challenging datasets to assess and enhance their algorithms.

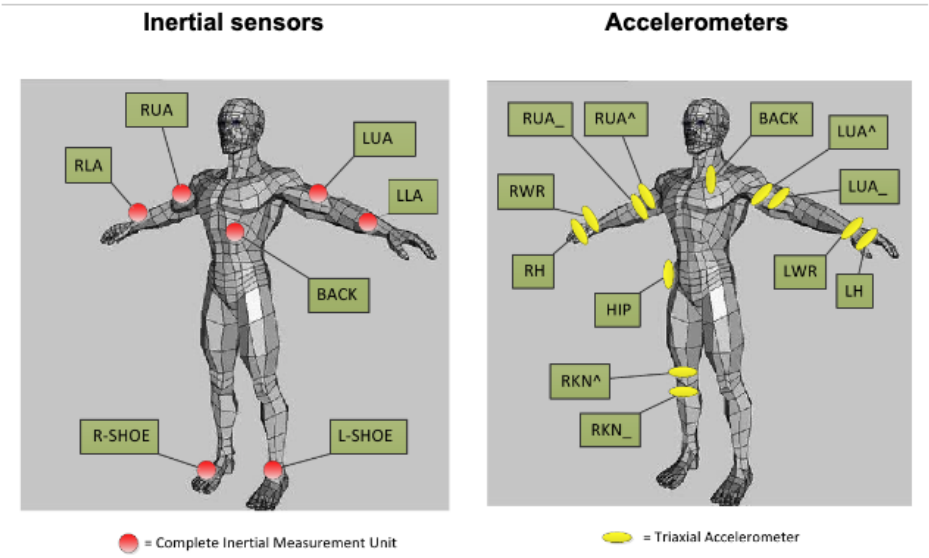

Figure 2.: The image above shows the location of the on-body sensors used in the Opportunity dataset. The subject wore a custom-made motion jacket composed of 5 commercial RS485-networked XSense inertial measurement units. In addition to 12 Bluetooth acceleration sensors on the limbs and commercial InertiaCube3 inertial sensors located on each foot. [10]

The WEAR dataset is a comprehensive resource featuring inertial data and egocentric videos. Bock et al. delve into action classification using inertial features, achieved through Inertial-based Human Activity Recognition (HAR) models. Additionally, Temporal Action Detection (TAD) is explored using vision-based models, incorporating both inertial data and egocentric videos. The innovative aspect of the paper lies in proposing the utilization of inertial data and the fusion of inertial data with I3D features[1] for enhancing vision-based models.

The motivation driving this report is grounded in the intriguing question: If the video features extracted from a pre-trained network yield advantages, can a similar advantage be harnessed by employing pre-trained inertial features? Inspired by the success demonstrated in utilizing

[1]**I3D features** [4] is extracted from a pre-trained model on Kinetics 400 from the second-to-the-last layer, providing two tensors with 1024-dimensional features for RGB and flow streams. Notably, these features are generated over 0.5, 1, and 2-second windows, corresponding to 20, 60, and 120 frames for a 60 FPS (Frames Per Second) video. The windows are generated with a fifty per cent overlap, contributing to a more robust and comprehensive analysis of human activities.

pre-trained video features like I3D, this investigation seeks to explore whether an analogous approach with pre-trained inertial features could enhance the performance of Human Activity Recognition (HAR) models. By delving into this comparison, the report aims to shed light on the potential benefits and implications of leveraging pre-trained inertial features, contributing to a deeper understanding of their efficacy in activity recognition.

The report outlines contributions that build upon the groundwork in three ways:

1. The report introduces an innovative methodology for acquiring pre-trained Convolutional and LSTM inertial features from the DeepConvLSTM network, and utilization of these features in the TAD vision networks.

2. Attainment of benchmark scores on two prominent HAR datasets, namely the WEAR dataset and the Opportunity Daily Activity runs dataset (Opportunity ADL). The report evaluates the effectiveness of pre-trained inertial features, as well as a fusion of these features with I3D features, using the ActionFormer and TriDet models. This benchmarking process provides valuable insights into the applicability and performance of the proposed methodology across diverse datasets.

3. The report outlines a systematic methodology for preprocessing Human Activity Recognition (HAR) data. This step-by-step approach ensures the optimization of data quality and relevance, laying the groundwork for improved model training and evaluation. The detailed methodology contributes to the broader understanding of best practices in preparing HAR datasets for effective analysis.

3. Related Work

3.1. Inertial-based HAR

In this report, the category of networks specializing in activity recognition based on the inertial data from sensors is referred to as Inertial-based Human Activity Recognition (HAR). Ordonez et al. pioneered this field by introducing the Deep Convolutional LSTM network for HAR using inertial data, featuring 4 Convolution Layers and 2 LSTM layers [8]. This network set a benchmark, surpassing all existing HAR networks at the time.

Building on Ordonez et al.'s work, Bock et al. introduced a modified architecture known as the shallow Deep Conv LSTM. Through empirical evidence, they demonstrated that a configuration of 4 Convolution Layers and 1 LSTM layer is often sufficient for effective HAR across various datasets [1]. Subsequently, Bock et al. applied the proposed shallow Deep Conv LSTM architecture to the WEAR dataset, further validating its efficacy in the context of outdoor sports activity recognition [2]. This series of advancements underscores the evolution of Inertial-based HAR networks, from the groundbreaking Deep Convolutional LSTM to more streamlined architectures tailored for improved efficiency and performance.

3.2. Vison-based Temporal Action Localization

Identifying Action instances in time and classifying their categories is known as Temporal Action Localization(TAL). ActionFormer and TriDet are state-of-the-art Single Stage TAL models for video data. They have published results on the action recognition video datasets such as Thumos14 [12], Activity net 1.3[6], EPIC-Kitchens 100 [5] and HACS[14].

ActionFormer: As described above, ActionFormer [13] is a single-stage Temporal Action Localization (TAL) model that uses a multi-scale feature representation. It employs a local self-attention-based transformer and a lightweight decoder to classify each moment in time and estimate the corresponding action boundaries. The model utilizes local self-attention to model temporal context in input untrimmed videos. To capture actions at different temporal scales, feature maps are downsampled between the layers of the transformer.

TriDet: Temporal Action Detection with Relative Boundary Modeling [11] is also a one-stage convolution-based framework for temporal action detection. This model introduced a novel trident head for modelling action boundaries via an estimated relative probability distribution of the boundary. This model proposed an efficient scalable-Granularity Perception(SGP) layer instead of a transformer model as in ActionFormer. TriDet model also utilizes a feature pyramid to model a longer context through downsampling the output features of the video backbone network several times via max-pooling (with a stride of 2). The SGP layer used is the replacement of the self-attention module with the fully convolutional module. This resultant SGP-based feature pyramid layer achieves better performance than the transformer-based feature pyramid while being much more efficient.

4. Methodology

4.1. Obtaining Pretrained Inertial Features

This section outlines the methodology employed to acquire pretrained inertial features. Given the absence of readily available pre-trained inertial networks, we opted to train an inertial HAR model—utilizing the Deep Convolutional LSTM architecture in this study—on raw inertial data to obtain both convolutional and LSTM inertial features.

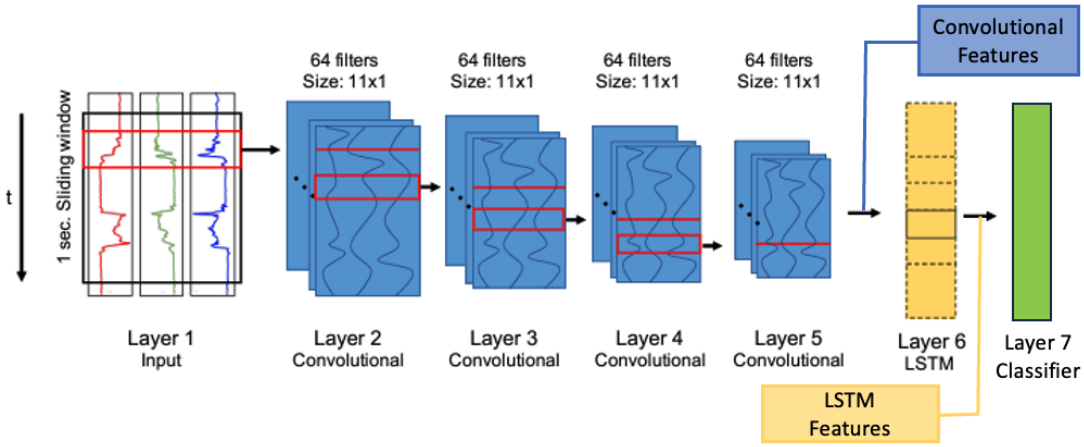

Figure 3.: Convolution and LSTM Features from Deep Convolution Network

The DeepConvLSTM network adopts a sliding window approach, applying 1D convolutions along the temporal axis for each sensor. The convoluted features are then processed through an LSTM layer, followed by classification using a dense network. This architecture ensures the capture of both spatial and temporal features within the defined windows. Figure 3 illustrates the extraction process, where convolution inertial features are obtained by excluding the LSTM and dense layers, and LSTM inertial features are obtained by excluding the dense layers.

For convolution features, the output of layer 5 is reshaped into a 1D array, resulting in convolution feature lengths as shown in equation 1. On the other hand, LSTM inertial features have a constant feature length equal to the number of LSTM units (1024 in the model), irrespective of window size. The impact of feature lengths on runtime and their influence on model performance

are detailed in Tables 8 and 10.

$$\textbf{Convolution Feature Length} = (\mathbf{W} - 4 \times (\mathbf{K} - 1)) \times \mathbf{f} \times \mathbf{n} \tag{1}$$

Where W is the number of sensor readings in Window Size, K is 1D kernel length, f is the number of filters in convolution, n is the number of sensor axes in data. For the WEAR dataset, the convolution feature lengths are 46,080, 7680, and 3840 for window sizes of 2, 1, and 0.5 seconds, respectively.

A notable limitation of the DeepConvLSTM network is its inability to account for context beyond the temporal window, posing a significant disadvantage in action recognition tasks. During training, Cross Entropy loss with class weights is employed, and F1 score, accuracy serve as metrics.

4.2. Inertial features in Vision networks

For training the networks cross-validation is employed, with subjects divided into groups. In this report, the validation is termed as cross-split validation. In the WEAR dataset, 18 subjects are available and they are divided into 3 splits, i.e. 6 subjects are used for validation in each split leaving the remaining 12 subjects for training. The model was then trained using Cross-validation 3 times. To explain in detail, in the first run subjects 0 to 5 are used for validation, subjects 6 to 17 for training, in the second run subjects 6 to 11 are used for validation and subjects 0 to 5, 12 to 17 for training and in the third run subjects 12 to 17 for validation, subjects 0 to 11 in training as in table 1.

Split	**Subjects**																	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	X	X	X	X	X	X												
2							X	X	X	X	X	X						
3													X	X	X	X	X	X

Table 1.: Cross-Validation Splits with Subjects of WEAR Data('X' marked subjects are used for validation and the remaining subjects for training.

Both the inertial and the vision network are trained using cross-validation with the same splits as shown in table 1. Usage of the same splits avoids data leakage to the vision model from the inertial features. This ensures the training subjects in the inertial network are also training subjects for the vision network during particular splits. To sum up, the features obtained from split 1 of the inertial network are used for split 1 of the vision network. Similar to the WEAR

dataset the Opportunity ADL dataset is divided into 3 splits as shown in table 2. Subject 1 is always used in training as it has almost all missing values in some axes and also to limit the splits to 3.

Split	**Subjects**			
	1	2	3	4
1		X		
2			X	
3				X

Table 2.: Cross-Validation Splits with Subjects of Opportunity ADL Data('X' marked subjects are used for validation and the remaining subjects for training, Subject 1 is always used in Training

The ActionFormer model uses Focal loss [7] for Classification and DIOU(Distance Intersection Over Union) loss [15] for distance regression of start and end time of actions. Focal loss naturally handles imbalanced samples when there are many more negative samples than positive ones. Soft-NMS(Non-maximum suppression) [3] is applied for the deduplication of predicted instances. Similar to ActionFormer, TriDet also used Focal Loss for classification and IOU Loss for regression of boundaries and also uses Soft-NMS.

4.3. Methodology for HAR Data Preprocessing

Data preprocessing is an essential step in any machine learning or Deep learning project, and Human Activity Recognition (HAR) is no exception. Most of the inertial data published are imbalanced with a high percentage of null data and small percentages of actual activity data. It is challenging to preprocess such datasets due to their imbalance and range of activities. In this section, we will discuss the preprocessing steps required to prepare the HAR dataset for machine learning for activity detection. The preprocessing will be explained with the help of the *Opportunity dataset* [10] as it has missing values and outliers in the dataset.

Figure 4 shows the statistics (mean, maximum and minimum) of the opportunity ADL dataset for all subjects. Each column has different scales as shown in the figure(Accelerometers and IMUs have different scales and different units). To give the same importance to all sensor axes, normalization along the axes is to be performed. Refer to section A.1 for more details on the Opportunity ADL dataset. Also, table 3 shows the percentage of missing values of more than 10% per column. The missing values in the dataset were due to disconnections in the sensors and these were to be handled.

Methodology for Filling Missing Values: In this section, we will discuss the methodology followed for filling in missing values. In the first step, Interpolation is done, if less than 20 missing values are in between two data points. The remaining missing values can be filled by the mean or median of the whole column data but as stated above the mean of data was influenced by null data and filling the mean of columns will bias the data. Instead, the mean/ median of each activity can be filled respectively for the missing value of that particular activity. Since data in some of the columns is a mixture of Gaussians filling the median would be optimum. So the missing values are filled in with the median of that particular column and activity. Refer to Algorithm 1 for the pseudo-code.

Methodology for Inertial Data Outliers Removal and Normalization: Figure 5 shows the Box plots of activity data in Column 18 in opportunity ADL data with and without outliers. If

Table 3.: Columns and Their Percentage of NaN in Opportunity ADL data of all Subjects (Columns above 10% missing values)

Column	Percentage of NaN Values
14	21.37%
15	21.37%
16	21.37%
35	44.11%
36	44.11%
37	44.11%

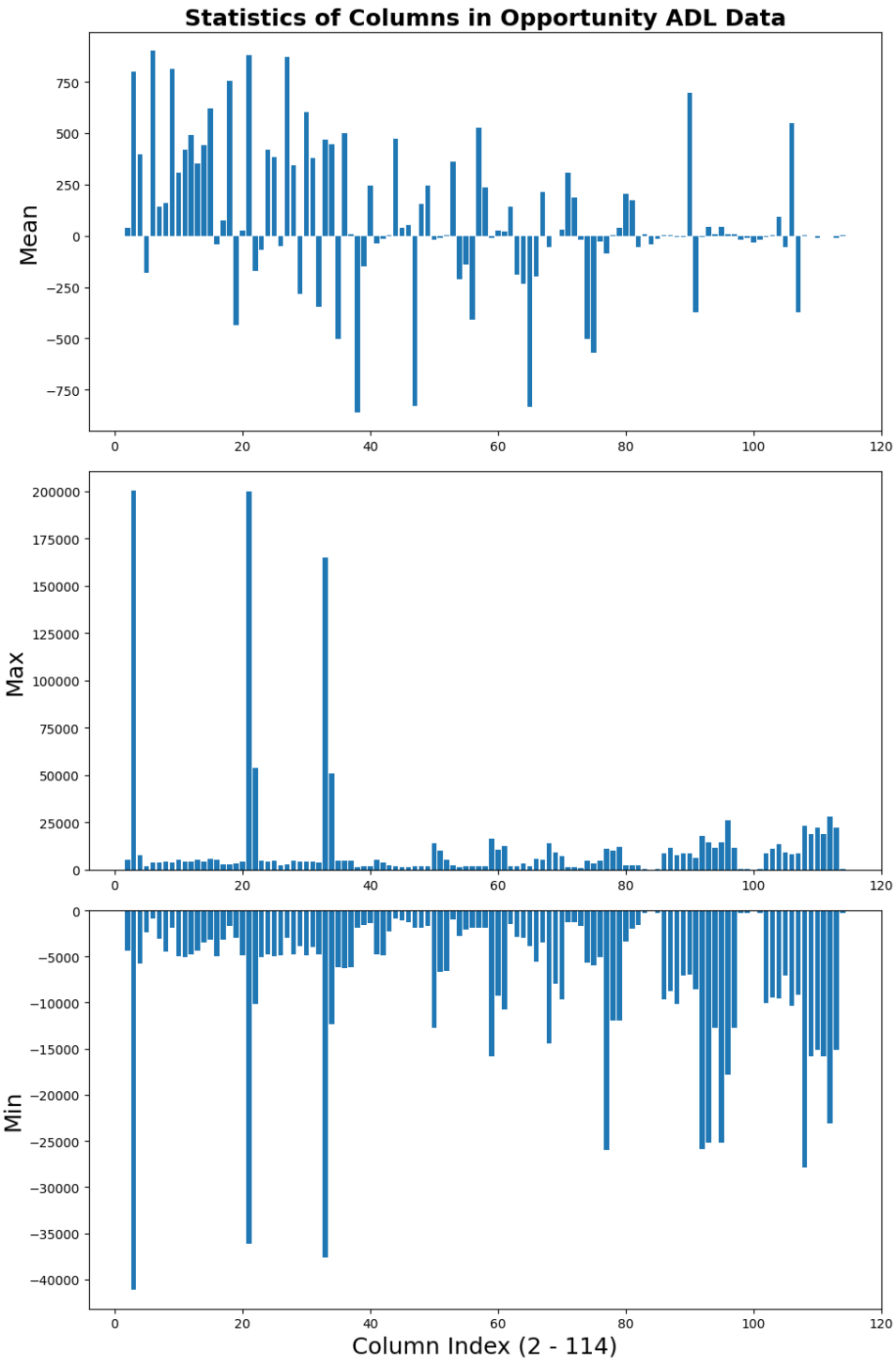

Figure 4.: Statistics of Opportunity ADL Data, Mean, Max, Min - Mean, Maximum, and Minimum for each Column

outliers were not removed in a particular column the scale of importance of this column data would come down compared to the column without the outliers. Normalization helps the model to treat all columns equally.

As mentioned earlier, Human Activity Recognition (HAR) datasets often exhibit imbalance, as

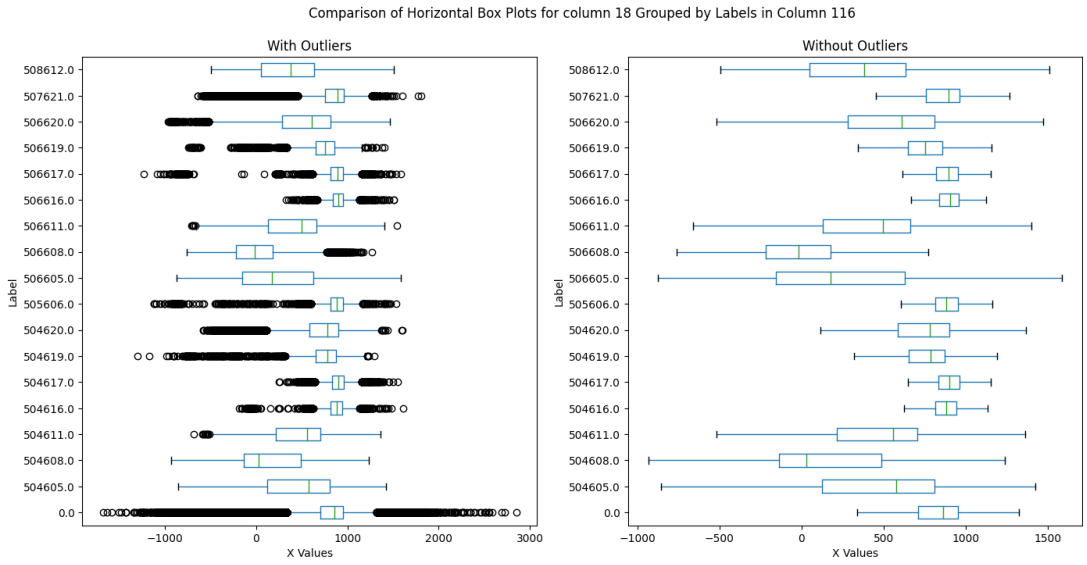

Figure 5.: Column 18 of Opportunity ADL data: With and Without Outliers

Table 4.: Activity, Label, Simplified Label, and % of data in Opportunity ADL data of Subject 1

Activity	Label	Simplified Label	Percentage of data
null	0	0	80.98%
Drink Cup	507621	16	7.13%
Open Door1	506616	1	1.37%
Open Door2	506617	2	1.21%
Open Fridge	506620	5	1.16%
Close Fridge	504620	6	1.14%
Close Door2	504617	4	1.05%
Close Door1	504616	3	1.00%
Open Drawer3	506608	13	0.72%
Close Drawer3	504608	14	0.61%
Toggle Switch	505606	17	0.58%
Close Drawer2	504611	12	0.51%
Open Drawer2	506611	11	0.46%
Close Dishwasher	504605	8	0.46%
Close Drawer1	504619	10	0.44%
Clean Table	508612	15	0.41%
Open Drawer1	506619	9	0.38%
Open Dishwasher	506605	7	0.36%

illustrated by the imbalanced percentages in the Opportunity dataset shown in the table 4. The null data is almost around 80% of the total data and because of this most of the actual activity data pose as outliers as the mean of data will be almost the same as the mean of null data. To handle this, outliers are removed considering each activity in a column. Z-score of 3 is calculated for each activity and the z-score of extreme ends (Lower Bound and Upper Bound) is chosen to perform outlier detection as shown in figure 6.

Figure 6 shows the statistics of columns 18 in the Opportunity ADL dataset. It was evident that the whole data distribution was highly influenced by the null data. If the outliers are removed based on the column z-score, as can be seen in Figure 6 actions like open fridge, Close fridge, open drawer 2, close drawer 2, open drawer 3, close drawer 3, open dishwasher, close dishwasher

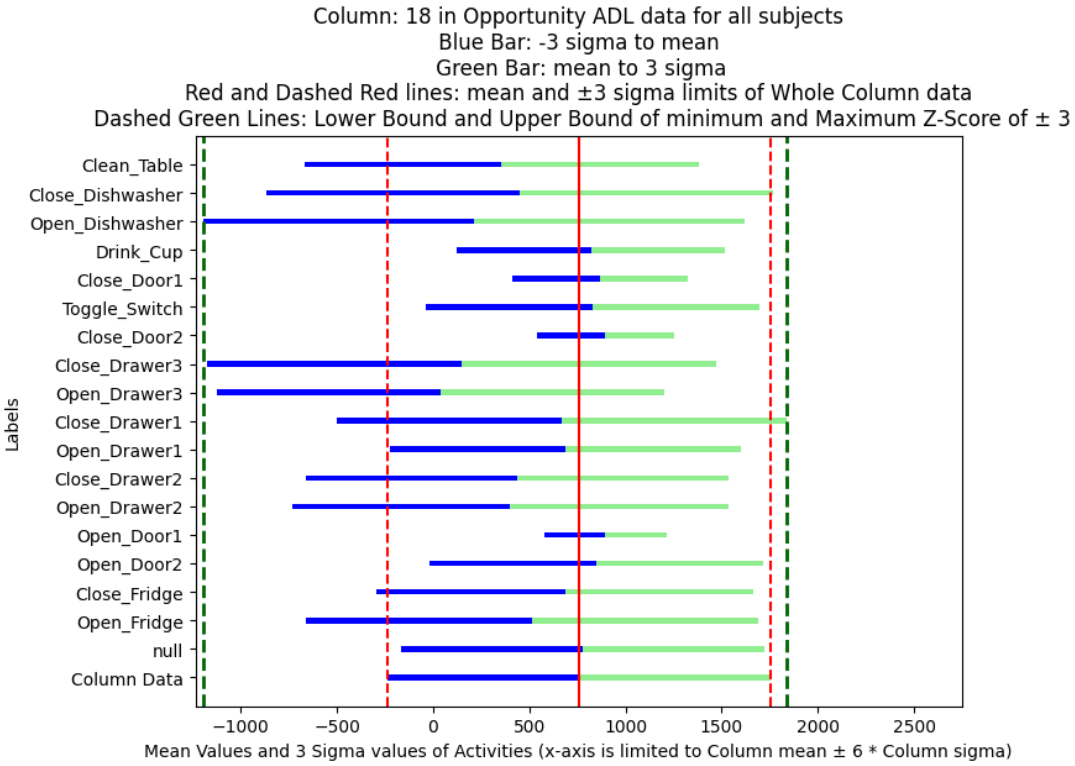

Figure 6.: Column 18 of Opportunity ADL Data of all subjects

and clean table has signal data below the z-score of -3 which will be lost. The data below the minimum of the z-scores of all the activities (lower bound) and above the maximum of all the z-scores of all the activities (upper bound) were considered outliers, this prevents the loss of activities data and effectively normalizes the columns using Min-Max normalization [1] as shown in equation 3. Refer to algorithm 1,2 and 3 for the pseudo-code.

Table 5, shows the Lower and Upper Bound of z-scores (see figure 6 for column 18) and Box plots (see figure 5 for column 18) for columns 12 to 18 in Opportunity ADL data. As we can see both of them have different Upper and Lower values. Considered Lower and Upper bounds of z-scores for the algorithm instead of box plots' lower and upper bounds.

The left plot of Figure 7, shows the distribution of Colum 18 data, and the middle plot shows the clipped data based on the above minimum and maximum z-scores of all activities, the right

[1] **Min-Max normalization**, also known as feature scaling, is a common technique used to scale and transform a dataset's values to a specific range, typically between 0 to 1 or -1 to 1.
The formula for **Min-Max normalization** for $[0, 1]$ Range:

$$X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{2}$$

The formula for **Min-Max normalization** for $[-1, 1]$ Range:

$$X_{\text{normalized}} = \frac{2(X - X_{\min})}{X_{\max} - X_{\min}} - 1 \tag{3}$$

Table 5.: Extremes of Z-Scores and Box Plots

Columns	Lower Bound of Z-Scores	Lower Bound of Box plot	Upper Bound of Z-Scores	Upper Bound of Box plot
12	−1179.57	−1025.50	1769.75	1934.00
13	−878.42	−652.00	1543.50	1636.50
14	−1209.07	−1471.50	2358.51	2428.50
15	−1801.35	−1957.87	1605.46	1907.75
16	−969.98	−1003.00	1298.54	1365.00
17	−1194.45	−1337.00	1836.99	1836.50
18	−1752.41	−1385.50	589.38	259.50

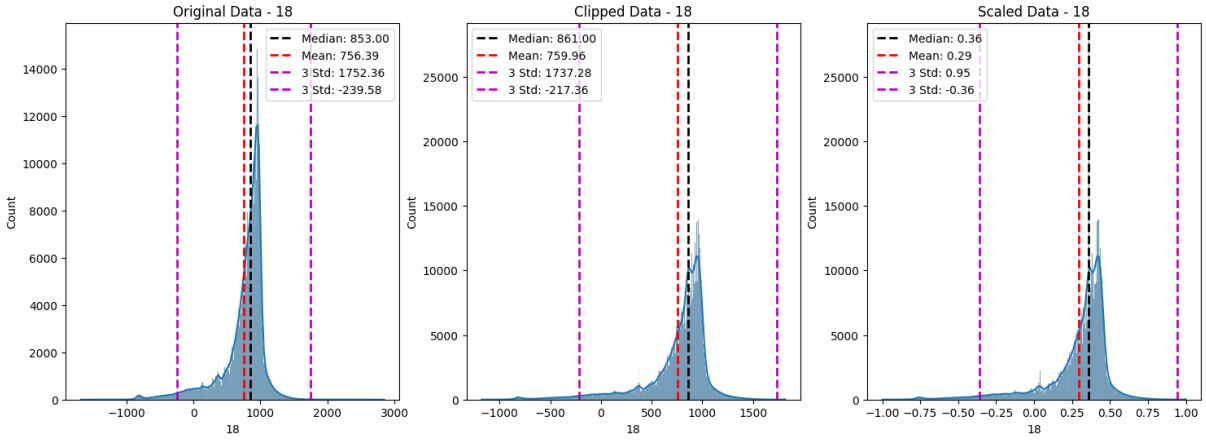

Figure 7.: Plots of Column 18 Data, Actual vs Clipped vs Scaled

plot is min-max normalization (see Equation 3) of the clipped data. This method ensures the prevention of loss of activity data in the process of normalizing the columns of HAR data. Refer to Section A.3 for preprocessing on the WEAR dataset. Refer to Section A.1 for details on WEAR and Opportunity ADL datasets and their data.

5. Experiments and Evaluation

5.1. Evaluation Criteria

Confusion Matrix and its terms for multi-class Classification: A confusion matrix is a useful tool to assess the effectiveness of a classification algorithm. It presents the results of a classification problem in a table format, with four key metrics: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Using these metrics, one can calculate several performance measures, such as accuracy, precision, recall, and F1 score. It is a valuable tool for evaluating the performance of a classification model, especially when dealing with imbalanced datasets. It provides a detailed breakdown of how well the model performs for each class and helps understand where the model might be making errors. Table 6 shows the Confusion matrix for binary classification.

	Actual Class 1	Actual Class 0
Predicted Class 1	True Positive (TP)	False Positive (FP)
Predicted Class 0	False Negative (FN)	True Negative (TN)

Table 6.: Confusion Matrix for a binary classifcation

Activity detection has multiple activities and the classification of those activities comes under multi-class classification. In multi-class classification, TP, TN, FP, and FN are calculated as below,

True Positive (TP): The number of instances, the predicted label is equal to the class and the ground truth label is equal to the class.

True Negative (TN): The number of instances, the predicted label is not equal to the class and the ground truth label is not equal to the class.

False Positive (FP): The number of instances, the predicted label is equal to the class and the ground truth label is not equal to the class.

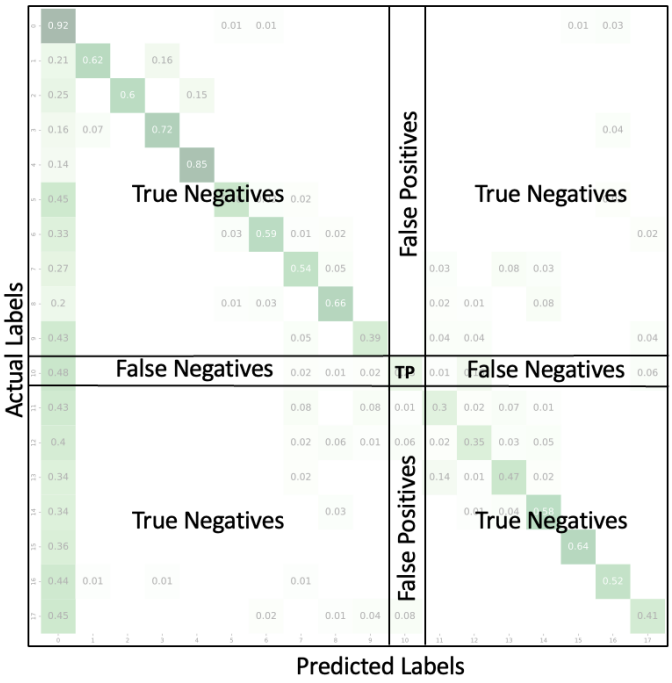

Figure 8.: Confusion Matrix for multi Class Classification

False Negative (FN): The number of instances, the predicted label is not equal to the class and the ground truth label is equal to the class. Figure 8 shows the TP, TN, FP, and FN for a multiclass classification.

Accuracy, Precision, Recall and F1 score for multi-class Classification:

Accuracy: It measures the overall correctness of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Precision: It measures the accuracy of the positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

Recall: It measures the ability of the model to capture all the positive instances. It is also known as Sensitivity, True Positive Rate.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

F1 Score: It is a balance between precision and recall.

$$\text{F1 Score} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \tag{7}$$

Intersection Over Union (IOU): This is the metric that is used to evaluate the object detection models. It is the ratio of the overlap area of predicted and ground truth bounding boxes to the union of predicted and ground truth bounding boxes.

$$\text{IOU} = \frac{\text{Predicted} \cap \text{Ground truth bounding boxes}}{\text{Predicted} \cup \text{Ground truth bounding boxes}} \tag{8}$$

In the context of action detection, the IOU is used temporally and termed **Temporal IOU (tIOU)**. It is the ratio of the overlap area of predicted and ground truth action segments to the union of predicted and ground truth action segments.

$$\text{tIOU} = \frac{\text{Predicted} \cap \text{Ground truth action Segments}}{\text{Predicted} \cup \text{Ground truth action Segments}} \tag{9}$$

Mean Average Precision (MAP): Mean Average Precision (MAP) is a metric commonly used to evaluate the performance of object detection models. It provides a single scalar value that summarizes the precision-recall curve. MAP is used as a measure for Activity detection (detecting actions temporally) as it is similar to object detection. MAP leverages metrics like Confusion Matrix, Intersection Over Union(IOU), Precision, and Recall. The below discussion is summarized from the paper [9], and refer to the same for a detailed discussion on mAP.

In the case of detection, it is important to note that, a **true negatives (TN)** result does not apply, as there are an infinite number of action segments that should not be detected within any given video or signal. The definitions of TP, FP, and FN change for the detection models as follows.

True positive (TP) : A correct detection of a ground-truth action Segment.

False positive (FP) : An incorrect detection of a nonexistent action segment or a misplaced detection of an existing Action.

False negative (FN) : An undetected ground-truth Action segment.

An action segment detection can be classified as correct or incorrect based on the Intersection Over Union (IOU) and threshold (t) as follows:

- If IOU $\geq t$, then the detection is considered correct.
- Else If IOU $< t$, the detection is considered incorrect.

As previously stated, the term **true negatives (TN)** is not applicable in the context of detection frameworks. Consequently, metrics associated with TN, such as True Negative Rate (TNR), False Positive Rate (FPR), and ROC curves, cannot be employed. Detection frameworks primarily rely on the concepts of precision and recall.

Precision is the ability of a model to identify only relevant action segments. It is the percentage of correct action segment predictions.

$$\text{Precision for detection frame works} = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \tag{10}$$

Recall is the ability of a model to find all relevant cases (all ground-truth action segments). It is the percentage of correct positive predictions among all given ground truths.

$$\text{Recall for detection frame works} = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truths} \tag{11}$$

The precision $\times$ recall curve is a trade-off between precision and recall for different confidence values associated with the action segments generated by a detector. If the confidence of a detector is such that its FP is low, the precision will be high. However, in this case, many positives may be missed, yielding a high FN, and thus a low recall. Conversely, if one accepts more positives, the recall will increase, but the FP may also increase, decreasing the precision. However, a good action detector should find all ground-truth action segments (FN = 0 = high recall) while identifying only relevant objects (FP = 0 = high precision). Consequently, an effective action detector should maintain a high level of precision while increasing its recall. This implies that even when adjusting the confidence threshold, both precision and recall should remain high. Therefore, a substantial area under the curve (AUC) generally signals a combination of high precision and high recall. Unfortunately, in practical cases, it often results in a precision $\times$ recall plot resembling a zigzag pattern, making it difficult to accurately measure the Area Under the Curve (AUC). To address this issue, the precision $\times$ recall curve is manipulated to eliminate the zigzag behaviour before estimating the AUC.

In the 11-point interpolation, the shape of the precision $\times$ recall curve is summarized by

averaging the maximum precision values at a set of 11 equally spaced recall levels [0, 0.1, 0.2, ..., 1], as given by

$$\text{AP @ IOU} = \frac{1}{11} \sum_{R \in \{0, 0.1, \ldots, 0.9, 1\}} \text{P}_{interp}(\text{R}) \tag{12}$$

where,

$$\text{P}_{\text{interp}}(R) = \max_{R' : R' \geq R} P(R') \tag{13}$$

In this definition of AP, instead of using the precision P(R) observed at each recall level R, the AP is obtained by considering the maximum precision Pinterp(R) whose recall value is greater than R.

$$\text{mAP @ IOU} = \frac{1}{N} \sum_{i=1}^{N} (\text{AP @ IOU})_i \quad \textit{where } N \textit{ is the number of classes} \tag{14}$$

$$\text{mAP} = \frac{1}{K} \sum_{i=1}^{K} (\text{mAP @ IOU})_i \quad \textit{where } K \textit{ is the number of IOU thresholds} \tag{15}$$

5.2. Performance Metrics on WEAR Dataset

Experiments are performed on the 1-second window (60 frames 30 strides) as it performs best in the WEAR paper [2] for the WEAR Inertial (**WI**), WEAR Inertial Data Concatenated (**WIF**) and Combined features (**WIF+I3D**). Table 7 shows the performance metrics of the inertial model (DeepConvLstm model) on the WEAR Inertial(**WI**) and WEAR Inertial Normalized (**WIN**) data. Convolutional and LSTM features are saved as described in section 4.1 during these runs. The WEAR Inertial (**WI**) data performs better than the WEAR Inertial Normalized (**WIN**) data. The results in table 7 are processed (using Majority filter 1251) and run on the Deep Convolution LSTM model (4 Convolution layers and 1 LSTM layer) with the same Hyperparameters for 300 epochs. The best results in each column are highlighted with **bold** numbers. Refer to table 17 for abbreviations for the WEAR data.

Table 7.: Performance Metrics (Precision(P), Recall (R), F1-score (F1) and mean Average Precision (mAP) at different tIOU) of DeepConvLstm model on **WI** Data and **WIN** data, (1) - seed 1

Data	**Model**	**P**	**R**	**F1**	**mAP**					
					0.3	0.4	0.5	0.6	0.7	Avg
WI (1)	DeepConvLSTM	**79.59**	**81.62**	**79.52**	**55.52**	**54.07**	**52.15**	**50.05**	**46.47**	**51.65**
WIN (1)	DeepConvLSTM	75.25	79	75.61	50.69	48.64	46.55	43.28	41.13	46.06

Table 8.: Model Characteristics and Training Time for different types of WEAR Inertial features with ActionFormer(AF) and Tridet Models.
(1,2): Seed 1 & Seed 2, FL: Features Length, LP: Learnable Parameters, and T: Time Taken for (1,2), Refer to table 17 for abbreviations for the WEAR data

Data	**FL**	**Model**	**LP**	**T**
WEAR Inertial Features (**WIF**)	600	AF(1,2)	27,024,410	1h 54m, 1h 55m
		TriDet(1,2)	15,585,886	1h 48m, 1h 49m
WEAR Normalized Inertial Features (**WNIF**)	600	AF(1,2)	27,024,410	1h 56m, 1h 54m
		TriDet(1,2)	15,585,886	1h 50m, 1h 50m
LSTM Features from **WI** (**WLF**)	1024	AF(1,2)	27,675,674	1h 52m, 1h 53m
		TriDet(1,2)	16,237,150	1h 45m, 1h 45m
LSTM Features from **WIN** (**WNLF**)	1024	AF(1,2)	27,675,674	1h 51m, 1h 50m
		TriDet(1,2)	16,237,150	1h 46m, 1h 45m
Convolutional Features from **WI** (**WCF**)	7680	AF	37,899,290	2h 24m
		TriDet	26,460,766	2h 18m

Table 8 shows the Feature lengths, model learnable parameters and time taken for the cross-split validation for two different seeds (1, 2). The TriDet model has 42 per cent fewer parameters than the ActionFormer model, which indicates the run time of Tridet is also less compared to ActionFormer. The LSTM processed features (**WLF & WNLF**) take a similar runtime to that of Wear Inertial Features (**WIF**) as they have feature lengths of 1024, and 600 respectively. The Convolutional processed features take a higher runtime than that of **WIF** data as they have a

higher feature-length of 7680.

Table 9.: Performance Metrics (Precision(P), Recall (R), F1-score (F1) and mean Average Precision (mAP) at different tIOU) for the ActionFormer(AF) and the TriDet Models on different types of WEAR inertial Features, for score threshold 0.2, Refer to table 17 for abbreviations for the WEAR data

Data	**Model**	**P**	**R**	**F1**	**mAP**					
					0.3	0.4	0.5	0.6	0.7	Avg
WIF	AF(1,2)	81.35	75.3	76.62	72.85	71.23	68.03	64.32	56.34	66.55
	TriDet(1,2)	83.67	74.15	77.23	73.41	72.1	70	67.18	62.39	69.02
WNIF	AF(1,2)	78.42	72.3	73.63	70.02	68.41	64.9	60.81	53.6	63.55
	TriDet(1,2)	81.52	72.4	75.22	71.46	70.1	68.27	65.15	58.76	66.75
WLF	AF(1,2)	84.08	78.95	80.19	77.51	76.16	73.6	69.9	65.09	72.45
	TriDet(1,2)	**85.38**	78.58	**80.68**	77.61	76.73	74.92	71.38	67.32	73.59
WNLF	AF(1,2)	82.07	78.02	78.86	76.57	75.35	73.14	69.91	63.49	71.69
	TriDet(1,2)	84.64	77.71	79.96	76.53	75.81	73.7	71.06	67.51	72.92
WCF	AF	79.21	73.24	74.76	77.85	73.69	60.19	46.59	42.21	60.11
	TriDet	81.53	**79.9**	79.26	**79.2**	**77.98**	**76.12**	**73.39**	**69.85**	**75.31**

Table 9 shows the performance metrics of ActionFormer and Tridet models for different types of WEAR Inertial features. General observations from the table 9, for score threshold 0.2 are the TriDet model performs better than the ActionFormer model for all the different types of features. The WEAR Inertial Features (**WIF**) performed better than the WEAR Normalized Inertial Features (**WNIF**) on both the ActionFomer and the Tridet models. LSTM Features from **WI** (**WLF**), LSTM Features from **WIN** (**WNLF**) and Convolutional Features from **WI** (**WCF**) outperform the baseline **WIF** data in both models except in the case of **WCF** data in the ActionFomer model. **WLF** data with the TriDet model have the highest Precision and F1 score compared to all other features. **WCF** data with the TriDet model have the highest maP at all tIOU compared to all other features.

Compared to Baseline(**WIF**) data,

1. WLF data outperforms by **4.65 % increase** in F1 and **8.86 % increase** in mAP for the ActionFormer model and also outperforms by **4.46 % increase** in F1 and **6.62 % increase** in mAP for the TriDet model.

2. WNLF data outperforms by **2.92 % increase** in F1 and **7.72 % increase** in mAP for the ActionFormer model and also outperforms by **3.53 % increase** in F1 and **5.65 % increase** in mAP for the TriDet model.

3. WCF data underperform by **2.42 % decrease** in F1 and **9.67 % decrease** in mAP for the ActionFormer model and outperforms by **2.62 % increase** in F1 and **9.11 % increase** in mAP for the TriDet model.

Table 10 shows the Feature lengths, model learnable parameters and time taken for the cross-split validation for two different seeds (1, 2) for the combined features. The WLF+I3D and WNLF+I3D data take a similar runtime to that of raw WIF+I3D as they have feature lengths of 3072 and 2648 respectively. The WCF+I3D data take a higher runtime than that of WIF+I3D as the WCF+I3D data have a higher feature length of 9278 features.

Table 10.: Model Characteristics and Training Time for different types of WEAR combined features with ActionFormer(AF) and Tridet Models.
(1,2): Seed 1 & Seed 2, FL: Features Length, LP: Learnable Parameters, and T: Time Taken for (1,2), Refer to table 17 for abbreviations for the WEAR data

Data	**FL**	**Model**	**LP**	**T**
WIF concatenated with **WI3D** (**WIF+I3D**)	600 + 2048 = 2648	AF(1,2) TriDet(1,2)	30,170,138 18,731,614	2h 1m, 2h 1m 1h 54m, 1h 52m
WNIF concatenated with **WI3D** (**WNIF+I3D**)	600 + 2048 = 2648	AF(1,2) TriDet(1,2)	30,170,138 18,731,614	2h 2m, 2h 2m 1h 56m, 1h 55m
WLF concatenated with **WI3D** (**WLF+I3D**)	1024 + 2048 = 3072	AF(1,2) TriDet(1,2)	30,821,402 19,382,878	2h 4m, 2h 1h 54m, 1h 55m
WNLF concatenated with **WI3D** (**WNLF+I3D**)	1024 + 2048 = 3072	AF(1,2) TriDet(1,2)	30,821,402 19,382,878	2h 2m, 1h 58m 1h 57m, 1h 55m
WCF concatenated with **WI3D** (**WCF+I3D**)	7680 + 2048 = 9728	AF TriDet	41,045,018 29,606,494	 2h 33m

Table 11.: Performance Metrics (Precision(P), Recall (R), F1-score (F1) and mean Average Precision (mAP) at different tIOU) for the ActionFormer(AF) and the TriDet Models on different types of WEAR Combined Features, for score threshold 0.2, Refer to table 17 for abbreviations for the WEAR data

Features	**Model**	**P**	**R**	**F1**	**mAP**					
					0.3	0.4	0.5	0.6	0.7	Avg
WIF+I3D	AF(1,2)	82.94	80.16	80.28	77.51	75.79	73.21	69.99	64.1	72.12
	TriDet(1,2)	85.8	80	81.78	78.6	77.42	75.64	73.86	67.8	74.66
WNIF+I3D	AF(1,2)	79.95	78.17	77.3	75.5	74.11	72.41	69.17	62.29	70.7
	TriDet(1,2)	83	77.18	78.12	76.52	76.01	74.06	71.85	67.5	73.19
WLF+I3D	AF(1,2)	88.91	76.16	80.7	75.09	72.08	68.88	60.95	54.13	66.23
	TriDet(1,2)	**89.47**	77.04	81.67	74.59	71.01	66.9	61.44	55.49	65.89
WNLF+I3D	AF(1,2)	84.52	80.44	81.41	79.27	78.35	75.77	73.16	68.28	74.97
	TriDet(1,2)	85.45	80.76	**82.08**	79.03	78.22	77.03	73.55	70.46	75.66
WCF+I3D	AF	81.54	82.2	80.73	80.96	79.48	77.06	72.53	68.76	75.76
	TriDet	82.52	**81.62**	81.04	**80.45**	**80.04**	**78.58**	**74.53**	**71.44**	**77.01**

Table 11 shows the performance metrics of the ActionFormer and the Tridet models on the WEAR Combined Features data. General observations from the table 11, for score threshold 0.2 are the TriDet model performs better than the ActionFormer model for all the different types of features. **WIF+I3D** data performed better than the WNIF+I3D data on both the ActionFomer and the Tridet models. **WLF+I3D** data run on the TriDet model has the highest Precision compared to all other features. **WNLF+I3D** data run on the TriDet model has the highest F1 score compared to all other features. **WCF+I3D** data with the TriDet model have the highest

mAP at all tIOU compared to all other features.

Compared to the Baseline data (**WIF+I3D**) in combined features,

1. **WLF+I3D** data underperforms by **0.52 % increase** in F1 and **8.16 % decrease** in mAP for the ActionFormer model and also underperforms by **0.13 % decrease** in F1 and **11.74 % decrease** in mAP for the TriDet model.

2. **WNLF+I3D** data outperforms by **1.4 % increase** in F1 and **3.95 % increase** in mAP for the ActionFormer model and also outperforms by **0.36 % increase** in F1 and **1.33 % increase** in mAP for the TriDet model.

3. **WCF+I3D** data outperforms by **0.56 % increase** in F1 and **5.04 % increase** in mAP for the ActionFormer model and also outperforms by **0.9 % decrease** in F1 and **3.15 % increase** in mAP for the TriDet model.

Refer to Section A.4 for loss and metrics curves, and confusion matrices of the WEAR data.

5.3. Performance Metrics on Opportunity ADL Dataset

Similar to the WEAR dataset experiments are performed on the 1-second window. Table 12 shows the performance metrics of the inertial model (DeepConvLstm model) on the Opportunity ADL (**OA**) and Opportunity ADL Normalized (**OAN**) data. Only LSTM features (**OALF & OANLF**) are saved as described in section 4.1 during these runs. Convolutional features are not a good option when there are high sensor axes. For the Opportunity ADL dataset, the convolutional feature lengths are 7232, 43392, and 260352 for 0.5, 1, and 2-second windows respectively. This is not good for both memory and run time. LSTM features are fixed in feature lengths (Feature length -1024) and **69.79 % fewer features** than the concatenated features (Feature length -3390). The results in table 12 are unprocessed and run on the Deep Convolution LSTM model (4 Convolution layers and 2 LSTM layers) with the same Hyperparameters for 35 epochs. The model is overfitting after 35 epochs. The **OAN** data performs better than the **OA** data.

Table 12.: Performance Metrics (Precision(P), Recall (R) and F1-score (F1) of DeepConvLstm model on **OA** and **OAN** data, (1) - seed 1

Data	**Model**	**P**	**R**	**F1**
Opportunity ADL (**OA**) (1)	DeepConvLSTM	18.62	36.97	21.59
Opportunity ADL Normalized (**OAN**) (1)	DeepConvLSTM	**21.07**	**42.14**	**25.77**

Table 13 shows the Feature lengths, model learnable parameters and time taken for the cross-split validation for two different seeds (1, 2) for the Opportunity ADL data for the ActionFomer

and TirDet models. The TriDet model has lower learnable parameters and lower runtime compared to the ActionFormer model. The LSTM features (**OALF & OANLF**) have **11.6%** and **18.2%** reduced parameters for the ActionFormer and Tridet Models respectively and **69% reduced feature lengths** than the **OAIF** and **OANIF** data.

Table 13.: Model Characteristics and Training Time for different types of Opportunity ADL data Features with ActionFormer(AF) and Tridet Models.
(1,2): Seed 1 & Seed 2, FL: Features Length, LP: Learnable Parameters, and T: Time Taken for (1,2), Refer to table 18 for abbreviations for the Opportunity ADL data

Data	**FL**	**Model**	**LP**	**T**
Opportunity ADL Inertial Features (**OAIF**)	3390	AF(1,2) TriDet(1,2)	31308313 19866715	41m 3s, 41m 16s 36m 1s, 35m 2s
Opportunity ADL Normalized Inertial Features (**OANIF**)	3390	AF(1,2) TriDet(1,2)	31308313 19866715	40m 40s, 40m 28s 36m 5s, 36m 14s
LSTM Features from OA (**OALF**)	1024	AF(1,2) TriDet(1,2)	27674137 16232539	35m 1s, 34m 14s 32m 4s, 32m 4s
LSTM Features from OAN (**OANLF**)	1024	AF(1,2) TriDet(1,2)	27674137 16232539	35m 4s, 35m 35s 32m 28s, 32m 1s

Table 14.: Performance Metrics (Precision(P), Recall (R), F1-score (F1) and mean Average Precision (mAP) at different tIOU) for the ActionFormer(AF) and the TriDet Models on different types of Opportunity ADL inertial Features, for score threshold 0, Refer to table 18 for abbreviations for the Opportunity ADL data

Data	**Model**	**P**	**R**	**F1**	**mAP**					
					0.3	0.4	0.5	0.6	0.7	Avg
OAIF	AF(1,2)	34.27	59.66	38.66	59.35	55.45	50.62	43.4	31.2	48.0
	TriDet(1,2)	32.45	59.07	37.73	60.99	58.24	53.37	47.22	36.89	51.34
OANIF	AF(1,2)	**51.85**	51.13	45.96	51.31	48.86	45.14	38.85	28.77	45.29
	TriDet(1,2)	35.91	60.06	40	61.43	59.68	54.6	48.66	36.73	52.22
OALF	AF(1,2)	36.69	59.19	41.94	57.28	53.26	46.83	38.84	28.61	44.96
	TriDet(1,2)	35.55	59.14	40.98	61.93	59.59	55.44	47.94	36.87	52.35
OANLF	AF(1,2)	38.24	**62.58**	41.95	63.65	60.48	55.38	47.85	36.99	52.87
	TriDet(1,2)	38.23	61.39	**42.11**	**64.85**	**61.85**	**57.54**	**49.83**	**37.45**	**54.31**

Table 14 and 15 shows the performance metrics of the ActionFormer and the Tridet models on the different types of Opportunity ADL inertial Features. General observations from the table 14, for score threshold 0 are **OANIF** data outperforms the baseline **OAIF** data in metrics of Preciosn and F1. **OALF** outperforms the baseline **OAIF** in metrics of Precision and F1 score. **OANLF** outperform the baseline concatenated data in all the metrics.
Compared to Baseline(**OAIF**),

1. **OANLF** data outperforms by **11.6 % increase** in F1 and **5.78 % increase** in mAP for the TriDet model and also **8.51 % increase** in F1 and **10.14 % increase** in mAP for the ActionFomer model.

2. **OALF** data outperforms by **8.61 % increase** in F1 and **1.96 % increase** in mAP for

the TriDet model and also **8.48 % increase** in F1 and **6.33 % decrease** in mAP for the ActionFomer model.

General observations from the table 15, for score threshold 0.2 are the TriDet performs better than ActioFormer for all the different types of features. **OALF** obtained from the Deep Convolutional LSTM model outperforms the baseline concatenated data in Precision and F1 scores. **OANLF** have the highest Precision and Recall compared to all other features with Action Former.

Compared to Baseline(**OAIF** data),

1. **OANLF** data outperforms by **9.55 % increase** in F1 and **8.55 % increase** in mAP for the TriDet model and also **9.5 % increase** in F1 and **12.08 % increase** in mAP for the ActionFomer model.

2. **OALF** data outperforms by **6.43 % increase** in F1 and **2.39 % decrease** in mAP for the TriDet model and underperforms by **3.27 % increase** in F1 and **12.32 % decrease** in mAP for the ActionFomer model.

Table 15.: Performance Metrics (Precision(P), Recall (R), F1-score (F1) and mean Average Precision (mAP) at different tIOU) for the ActionFormer(AF) and the TriDet Models on different types of Opportunity ADL inertial Features, for score threshold 0.2, Refer to table 18 for abbreviations for the Opportunity ADL data

Data	**Model**	**P**	**R**	**F1**	**mAP**					
					0.3	0.4	0.5	0.6	0.7	Avg
OAIF	AF(1,2)	52.8	50.97	47.02	51.15	47.95	44.17	38.21	27.41	41.78
	TriDet(1,2)	53.03	50.18	47.87	50.6	48.95	44.95	40.45	31.82	43.35
OANIF	AF(1,2)	54.62	47.82	45.72	47.73	45.47	42.31	36.43	26.94	39.78
	TriDet(1,2)	61.82	46.23	47.06	46.84	45.5	42.17	39.04	30.13	40.74
OALF	AF(1,2)	59.41	45.64	48.56	46.47	43.09	38.05	31.71	23.82	36.63
	TriDet(1,2)	**61.83**	47.17	50.95	49.22	47.36	44.66	39.42	30.91	42.31
OANLF	AF(1,2)	56.88	**55.77**	51.49	**56.1**	**53.17**	49.02	42.66	33.18	46.83
	TriDet(1,2)	58.72	54.37	**52.44**	55.58	53.07	**49.85**	**43.57**	**33.21**	**47.06**

Refer to Section A.5 for loss and metrics curves, and confusion matrices of the Opportunity ADL data. All the experiments are conducted on the OMNI cluster at the University of Siegen, specifically utilising nodes 2 and 4. These nodes are part of the larger cluster infrastructure and are equipped with NVIDIA Tesla V100 GPUs, enhancing the computational capabilities for tasks involving the vectorization of double-precision floating-point numbers. The GPUs on these nodes contribute to accelerated processing and efficient handling of parallelizable workloads.

6. Conclusion

The multifaceted exploration presented in this report significantly advances the performance of the ActionFormer and TriDet models when coupled with pre-trained Convolutional and LSTM inertial features. The benchmarking results on the WEAR and Opportunity ADL datasets underscore the efficacy of the proposed methodology. Particularly, the TriDet model consistently outperforms the ActionFormer model across different feature types, showcasing its efficiency in HAR tasks. Moreover, the performance gains achieved by combining inertial features with I3D features highlight the potential for leveraging diverse data modalities in HAR.

The computational characteristics, including feature lengths, model parameters, and runtime, provide valuable considerations for model selection and implementation. The reduced parameters and runtime of the TriDet model, coupled with its competitive performance, position it as a promising choice for real-world applications where efficiency is crucial.

In conclusion, this work advances the state-of-the-art in HAR by introducing a robust methodology, benchmarking results, and insightful preprocessing practices. The demonstrated improvements in model performance, along with considerations for computational efficiency, contribute to the ongoing progress in HAR research.

Bibliography

[1] Marius Bock, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven. Improving Deep Learning for HAR with shallow LSTMs. In *2021 International Symposium on Wearable Computers*, 2021. doi: 10.1145/3460421.3480419. URL http://arxiv.org/abs/2108.00702.

[2] Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition, 2023. URL http://arxiv.org/abs/2304.05088.

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-NMS – Improving Object Detection With One Line of Code, 2017. URL http://arxiv.org/abs/1704.04503.

[4] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, 2018. URL http://arxiv.org/abs/1705.07750.

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. doi: 10.1109/TPAMI.2020.2991965. URL https://ieeexplore.ieee.org/document/9084270/.

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. IEEE, 2015. doi: 10.1109/CVPR.2015.7298698. URL http://ieeexplore.ieee.org/document/7298698/.

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection, 2018. URL http://arxiv.org/abs/1708.02002.

[8] Francisco Ordóñez and Daniel Roggen. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors*, 2016. doi: 10.3390/s16010115. URL http://www.mdpi.com/1424-8220/16/1/115.

[9] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. Da Silva. A Survey on Performance Metrics for Object-Detection Algorithms. In *2020 International Conference on Systems, Signals and*

Image Processing (IWSSIP). IEEE, 2020. doi: 10.1109/IWSSIP48289.2020.9145130. URL https://ieeexplore.ieee.org/document/9145130/.

[10] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Forster, Gerhard Troster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and Jose Del R. Millan. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. IEEE, 2010. doi: 10.1109/INSS.2010.5573462. URL http://ieeexplore.ieee.org/document/5573462/.

[11] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. TriDet: Temporal Action Detection with Relative Boundary Modeling, 2023. URL http://arxiv.org/abs/2303.07347.

[12] A. Roshan Zamir G. Toderici I. Laptev M. Shah Y.-G. Jiang, J. Liu and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes, 2014.

[13] Chenlin Zhang, Jianxin Wu, and Yin Li. ActionFormer: Localizing Moments of Actions with Transformers, 2022. URL http://arxiv.org/abs/2202.07925.

[14] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization, 2019. URL http://arxiv.org/abs/1712.09374.

[15] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, 2019. URL http://arxiv.org/abs/1911.08287.

A. Appendix

A.1. Datasets, Data and their abbrevations

About data of WEAR Dataset: The WEAR dataset has 'raw' and 'processed' folders. The folder 'raw' contains 'inertial' data from sensors and raw egocentric 'camera' data for 18 subjects. The sensor has a frequency of 50Hz and the egocentric videos were recorded at 60 fps. The folder 'processed' has Inflated 3D (I3D)[4] features, inertial features and combined features. Each of the above feature folders contains 120 frames and 60 Strides, 60 frames and 30 Strides, and 30 frames and 15 Strides. As the camera data has 60 frames per second, 120 frames and 60 Strides translate to a 2-second window and 50% overlap between each window, 60 frames and 30 Strides translate to a 1-second window and 50% overlap between each window, 30 frames and 15 Strides translate to a 0.5-second window and 50% overlap between each window. In brief, each I3D feature has context over the size of the window and the feature length of 2048 (two tensors with 1024-d features: for RGB and flow streams). In the context of inertial features, 2-second, 1-second, and 0.5-second windows have 100, 50, and 25 timestamps respectively and each timestamp has 12-axis sensor data which were concatenated to a feature with feature lengths of 1200, 600, and 300 respectively. The combined features are combined features of I3D and inertial features through concatenation (see Figure 9 for visual representation). Refer to Table 16 to see the folder structure.

Data Type (Root Folder)	**Processing Type**	**Frames and Stride**
Raw	Inertial (Sensor Readings)	NA
	Camera (Egocentric Videos)	NA
Processed	Inertial (Concatenated features)	30 Frames and 15 Stride
		60 Frames and 30 Stride
		120 Frames and 60 Stride
	Camera (I3D features)	30 Frames and 15 Stride
		60 Frames and 30 Stride
		120 Frames and 60 Stride
	Combined (Inertial + I3D features)	30 Frames and 15 Stride
		60 Frames and 30 Stride
		120 Frames and 60 Stride

Table 16.: WEAR Dataset

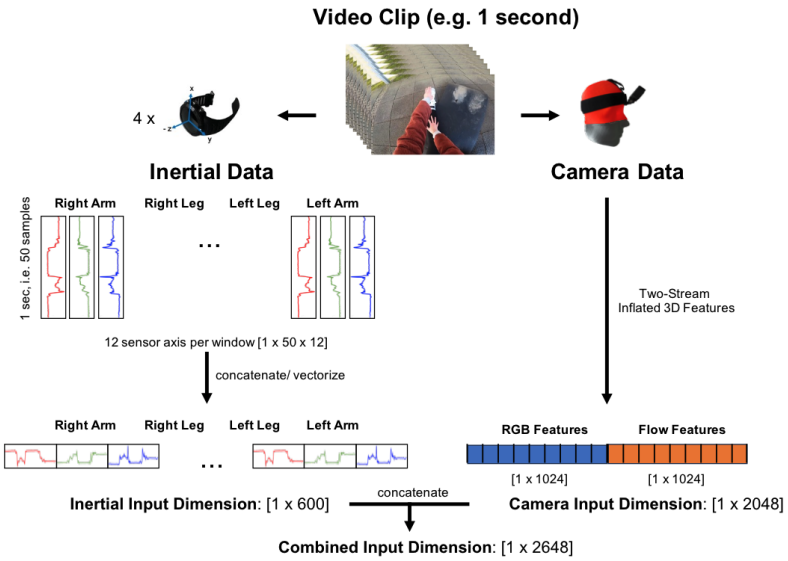

Figure 9.: Visualization of inertial data, camera data and Inertial feature embedding, Combined features [2]

In addition the raw inertial data is Normalized to obtain WEAR Inertial Normalized(**WIN**) using the methodology shown in sec A.3. These normalized data are also concatenated to obtain WEAR Normalized Inertial Features (WEAR Inertial Normalized Data Concatenated)**WNIF**. Refer to section 4.1 for details on obtaining pretrained inertial features. The WEAR inertial data and WEAR Inertial Normalized data are passed to the inertial network to obtain the LSTM and convolutional features. Further, the I3D features are concatenated to the inertial features to form combined features. Table 17 shows the different combinations of features and their abbreviations for the WEAR dataset.

Data Type	**Description**	**Abbreviation**
Raw Inertial	WEAR Inertial	**WI**
	WEAR Inertial Normalized	**WIN**
Inertial Features	WEAR Inertial Features (WEAR Inertial Data Concatenated)	**WIF**
	WEAR Normalized Inertial Features (WEAR Inertial Normalized Data Concatenated)	**WNIF**
	LSTM Features from **WI**	**WLF**
	LSTM Features from **WIN**	**WNLF**
	Convolutional Features from **WI**	**WCF**
Camera Features	WEAR I3D features	**WI3D**
Inertial + Camera Features (Combined Features)	**WIF** concatenated with **WI3D**	**WIF+I3D**
	WNIF concatenated with **WI3D**	**WNIF+I3D**
	WLF concatenated with **WI3D**	**WLF+I3D**
	WNLF concatenated with **WI3D**	**WNLF+I3D**
	WCF concatenated with **WI3D**	**WCF+I3D**

Table 17.: WEAR Data Description and their Abbreviations

About data of Opportunity ADL: The Opportunity dataset has 4 subjects and it has the drill and daily activities runs (ADL). The Opportunity ADL dataset is recorded at a frequency of 30 Hz and comprises data from 4 subjects, each with 5 ADL runs. To facilitate analysis, the 5 ADL runs for each subject are consolidated into a single ADL run. The data has 116 columns with timestamps as the first column, the axis of 3D accelerometers from columns 2 to 37, Inertial measurement units(IMUs) from columns 38 to 114, locomotion in column 115 and gestures in column 116. Accelerometers and IMUs have different scales and different units. Notably, the dataset contains numerous missing values, and addressing this, the Opportunity ADL (**OA**) data is derived by filling the missing values with zeros in the raw data.

The Opportunity ADL Normalized (**OAN**) data is generated by employing a methodology that involves filling in missing values and normalizing the data, as outlined in section 4.3. Similar to WEAR data **OA** and **OAN** datasets concatenate the data in the window resulting in Opportunity ADL Inertial Features (**OAIF**) and Opportunity ADL Normalized Inertial Features (**OANIF**) respectively. The OA and OAN datasets are subsequently input into an Inertial network, leading to the extraction of LSTM Features from OA (OALF) and LSTM Features from OAN (OANLF). For a detailed overview of the combinations of features and their abbreviations in the Opportunity ADL dataset, please refer to Table 18.

Data Type	**Description**	**Abbreviation**
Raw Inertial	Opportunity ADL	**OA**
	Opportunity ADL Normalized	**OAN**
Inertial Features	Opportunity ADL Inertial Features (Opportunity ADL Data Concatenated)	**OAIF**
	Opportunity ADL Normalized Inertial Features (Opportunity ADL Data Normalized and Concatenated)	**OANIF**
	LSTM Features from **OA**	**OALF**
	LSTM Features from **OAN**	**OANLF**

Table 18.: Opportunity ADL Data and their Abbreviations

A.2. Algorithm for HAR Data Pre-processing

The Algorithm shows the pseudocode for the methodology followed for the preprocessing. Algorithm 1 shows the procedure for filling in missing values and then obtaining the statistics of the dataset (mean, standard deviation, median) for further processing. Algorithm 2 shows the procedure to obtain the Lower and Upper bounds of all columns in the data. Algorithm 3 shows the procedure to preprocess each subject data using the obtained Lower and Upper bounds.

Algorithm 1 Proposed HAR Data Pre-Processing Algorithm (Part 1)

Input: Subjects data
Output: Normalized Subjects data

procedure FOR FILLING MISSING VALUES AND OBTAINING STATISTICS DICTIONARY(*Subject*$_1$, *Subject*$_2$, ..., *Subject*$_n$)
    Combined Data ← Combine the data from all n subjects.
    List of Activity Labelled Data ← Create an empty list for datasets divided based on activities.
    for all *Activity* ∈ *Activity labels* **do**
        Activity Labeled Data ← Divide Combined Data based on Activity to Activity Labeled Data.
        Append ← Add the Activity Labeled Data to the list.
    end for
    Statistics Dictionary ← Create an empty dictionary.
    if Dataset has Missing values **then**
        for all *Activity Labeled Data* ∈ *List of Activity Labelled Data* **do**
            for all *Column(Sensor Axis)* ∈ *Activity Labeled Data* **do**
                median ← Calculate the median ignoring missing values in column
                Statistics Dictionary (Activity, Column, Median) = *median*
            end for
        end for
        for all *Row, Column* ∈ *Combined Data* **do**
            if Data is Missing **then**
                Fill ← Fill using Statistics Dictionary(Activity in current Row, current Column, Median)
            end if
        end for
    end if
    for all *Activity Labeled Data* ∈ *List of Activity Labelled Data* **do**
        for all *Column(Sensor Axis)* ∈ *Activity Labeled Data* **do**
            mean ← Calculate the mean
            Statistics Dictionary (Activity, Column, Mean) = *mean*
            std ← Calculate the standard deviation
            Statistics Dictionary (Activity, Column, Std) = *std*
        end for
    end for
end procedure

Algorithm 2 Proposed HAR Data Pre-Processing Algorithm (Part 2)

procedure For Obtaining Lower Bound and Upper Bound Dictionaries of each column(*Statistics Dictionary*)
    Upper Bound Dictionary ← Create an empty dictionary.
    Lower Bound Dictionary ← Create an empty dictionary.
    for all *Column(Sensor Axis)* ∈ *Combined Data* **do**
        Column List z score +3 ← Create an empty list for appending each activity z scores.
        Column List z score -3 ← Create an empty list for appending each activity z scores.
        for all *Activity* ∈ *Activity labels* **do**
            Z score for ± *3* ← Calculate from Statistics Dictionary (Activity, Column, Mean) and Statistics Dictionary (Activity, Column, Std)
            Append ← Add z score for +3 to Column List z score +3
            Append ← Add z score for -3 to Column List z score -3
        end for
        Upper Bound Dictionary(column) = max(Column List z score +3)
        Lower Bound Dictionary(column) = min(Column List z score -3)
    end for
end procedure

Algorithm 3 Proposed HAR Data Pre-Processing Algorithm (Part 3)

procedure For Normalization of each Subject Data($Subject_1$, $Subject_2$, ..., $Subject_n$, *Statistics Dictionary, Upper Bound Dictionary, Lower Bound Dictionary*)
    for all *subject* ∈ *n subjetcs* **do**
        if Dataset has Missing values **then**
            if 20 consecutive values are missing **then**
                Fill by Interpolation
            else
                Fill ← Statistics Dictionary(Activity in current Row, Column, Median)
            end if
        end if
        for all *Column* ∈ *subject* **do**
            if *Lower Bound Dictionary(Column) <= Data(Column) <= Upper Bound Dictionary(Column)* **then**
                Data(Column) = Data(Column)
            else if *Data(Column) < Lower Bound Dictionary(Column)* **then**
                Data(Column) = Lower Bound Dictionary(Column)
            else if *Data(Column) > Upper Bound Dictionary(Column)* **then**
                Data(Column) = Upper Bound Dictionary(Column)
            end if
        end for
        for all *Column* ∈ *subject* **do**
            Column minimum ← Find minimum value in the column
            Column maximum ← Find maximum value in the column
            min-max normalization ← using Equation 3
        end for
    end for
end procedure

A.3. Data Pre-processing on WEAR Dataset

This section shows the preprocessing on the WEAR dataset to obtain the WEAR inertial Normalized data (**WIN**). The WEAR data consists of 18 subjects, each with 4 accelerometer sensors. Each sensor has 3 axes, resulting in 12 sensor axes. During recording, the values are limited to ±8 and there are no missing values.

In the accelerometer readings, the z-axis always has acceleration due to gravity, which is reflected on a different scale than the x and y axes. To give equal importance to all axes of data, normalizing the data is necessary. Figure 10 displays the data statistics of all axes (ordered by x, y, z). We can observe that the maximum readings of all z-axes are smaller compared to the x and y-axes.

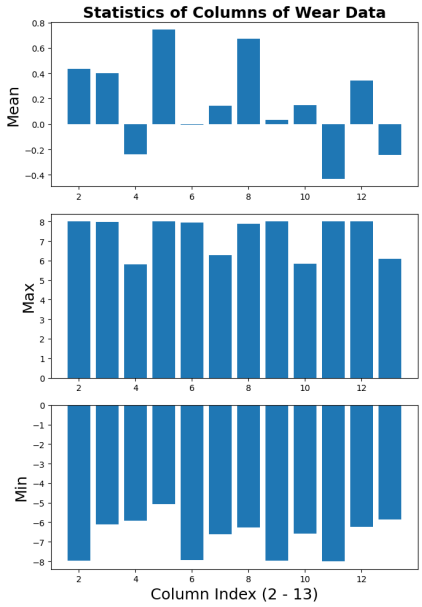

Figure 10.: WEAR data Statistics (mean, min, max)

As shown in the table 19 the WEAR data is also imbalanced, and the Null data is 20.17 % of the data.

Outliers Removal and Normalization Figure 11 shows the box plots with and without outliers for all the activities for column 3 of WEAR data of all 18 subjects. We can see there are outliers in each activity.

Figure 12 shows the z-score 3 sigma plot for all the activities for column 3 of WEAR data of all 18 subjects. We can see activities like jogging, jogging(rotating arms), jogging (skipping), jogging (side steps), situps, burpees, and jogging (butt-kicks) are below the z score of -3 of column data. Activities like jogging, jogging(rotating arms), jogging (skipping), jogging (side steps), situps

Table 19.: Data Distribution over Activities

Activity	Label	Percentage (%)
Null	0	20.17
Jogging	1	7.81
Jogging (Rotating Arms)	2	6.55
Jogging (Skipping)	3	6.20
Jogging (Sidesteps)	4	5.62
Jogging (Butt-Kicks)	5	5.35
Stretching (Triceps)	6	4.85
Stretching (Lunging)	7	4.70
Stretching (Shoulders)	8	4.35
Stretching (Hamstrings)	9	4.09
Stretching (Lumbar Rotation)	10	4.06
Push-ups	11	3.85
Push-ups (Complex)	12	3.79
Sit-ups	13	3.65
Sit-ups (Complex)	14	3.65
Burpees	15	3.65
Lunges	16	3.16
Lunges (Complex)	17	3.15
Bench-dips	18	2.73

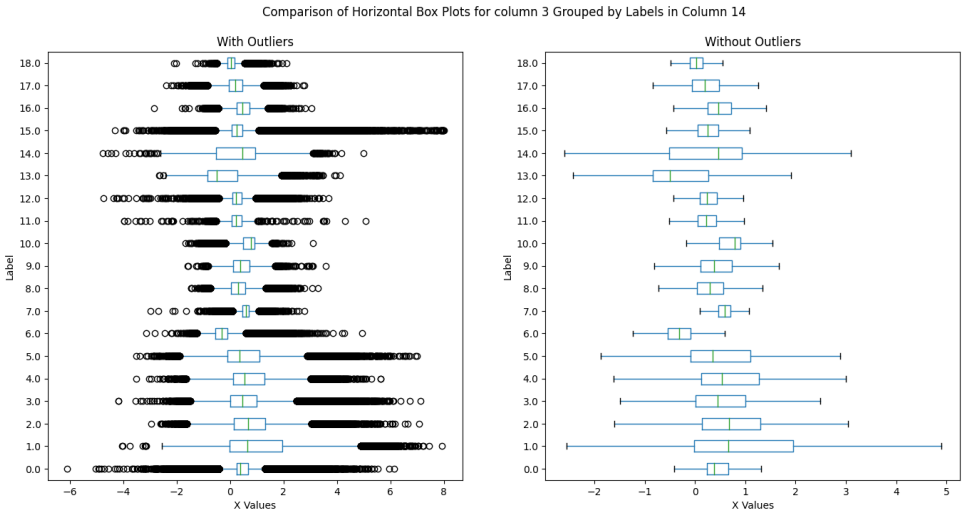

Figure 11.: WEAR data column 3 Box Plots

(complex), and jogging (butt-kicks) are above the z score of +3 of column data.

Table 20, shows the Lower and Upper Bound of z-scores (see figure 12 for column 3) and Box plots (see figure 11 for column 3) for the columns in WEAR data. As we can see both of them have different Upper and Lower Bound values, most of the Minimum Lower Bound and Maximum Upper Bound are Z scores, so considered Lower and Upper bounds of z-scores for the algorithm 1.

The left plot of Figure 13, shows the distribution of Colum 3 data, and the middle plot shows the clipped data based on the above minimum and maximum z-scores of all activities, the right plot is min-max normalization (see Equation 3) of the clipped data.

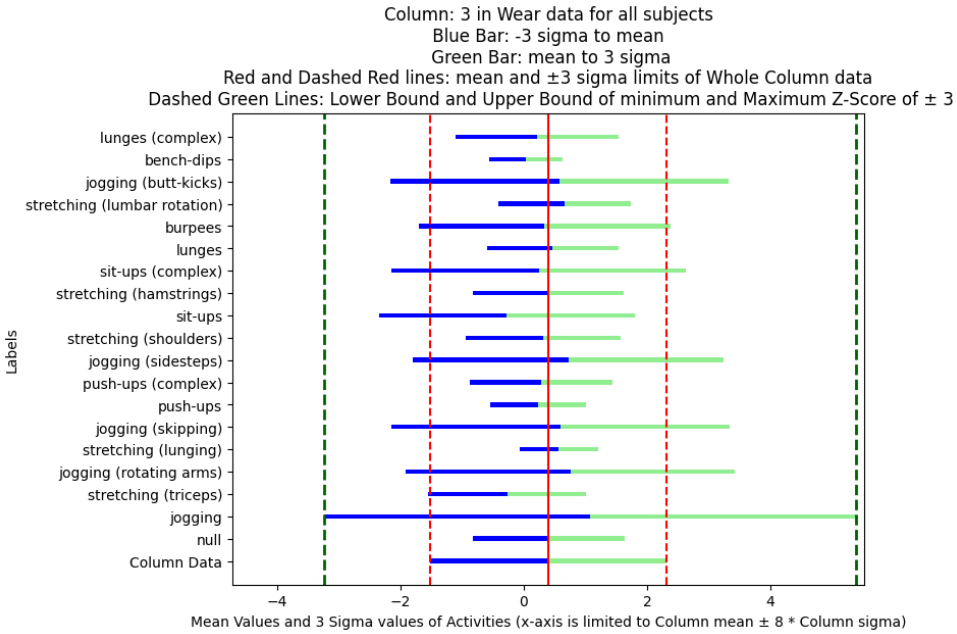

Figure 12.: WEAR data column 3: mean and 3 Sigma

Table 20.: Z-Scores and Box Plot Extremes for Outlier Detection (z: z-score, B: Box Plot) (min Lower Bound: Minimum of both Lower Bounds, max Upper Bound: Maximum of both Lower Bounds)

Columns	Lower Bound Z-Scores	Lower Bound Box plot	min Lower Bound	Upper Bound Z-Scores	Upper Bound Box plot	max Upper Bound
2	-3.26	-2.68	Z	5.36	4.34	Z
3	-3.24	-2.97	Z	5.38	4.89	Z
4	-2.2	-2.29	B	1.95	1.44	Z
5	-2.86	-2.49	Z	6.06	4.98	Z
6	-4.9	-3.12	Z	4.74	3.14	Z
7	-3.56	-3.63	B	3.39	3.64	B
8	-3.36	-2.44	Z	6.15	4.89	Z
9	-4.53	-2.79	Z	4.94	3.06	Z
10	-3.49	-3.52	B	3.39	3.55	B
11	-5.49	-4.47	Z	3.38	2.79	Z
12	-3.39	-3.09	Z	5.52	5.01	Z
13	-2.2	-2.12	Z	1.96	1.46	Z

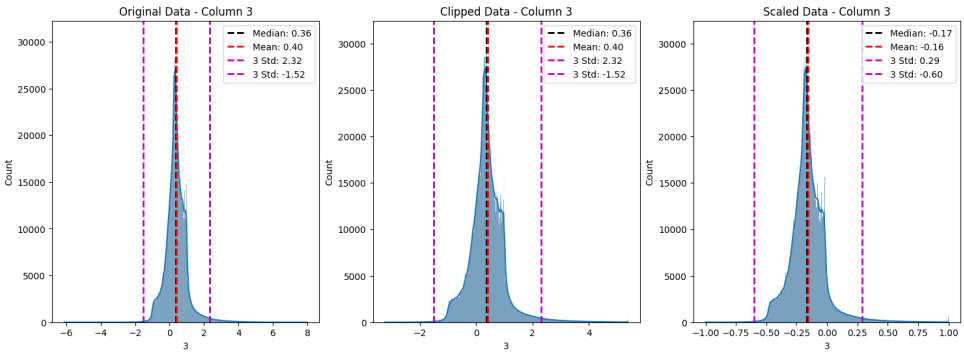

Figure 13.: WEAR data column 3: Original vs Clipped vs Scaled Data

A.4. Graphs, Confusion matrix of WEAR dataset

Figure 14 shows the Confusion Matrices for the ActionFormer model with **WIF** data, **WLF**, and **WCF** data. The True Positives of actions are improved with LSTM and Convolutional features. Refer to Table 19 for Action Labels.

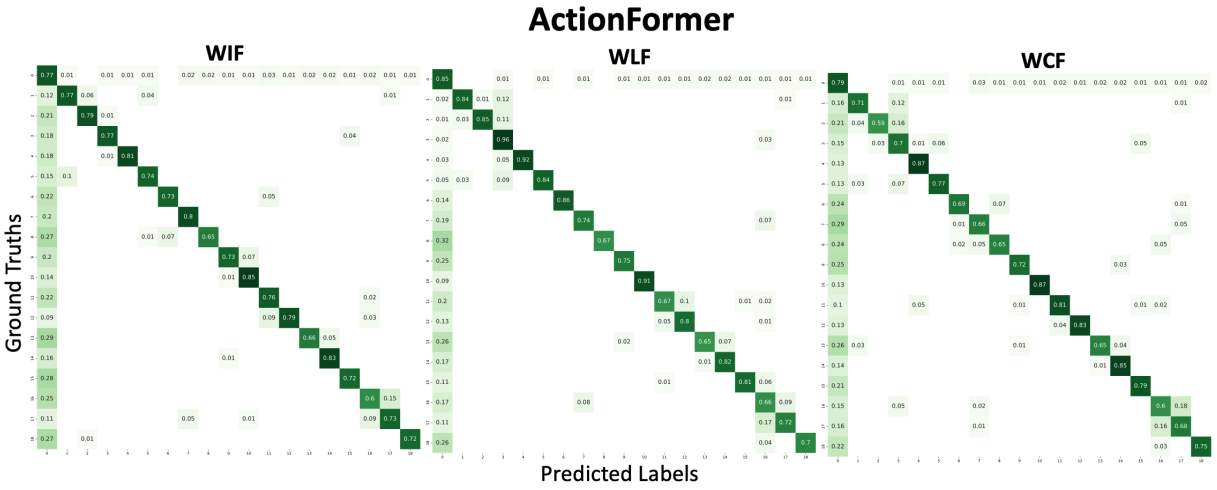

Figure 14.: Confusion matrices of ActionFormer, with **WIF** data, **WLF**, and **WCF** data, Refer to table 17 for abbreviations for the WEAR data

Figure 15 shows the Confusion Matrices for the TriDet model with **WIF** data, **WLF**, and **WCF** data. The True Positives of actions are improved with LSTM and Convolutional features. Refer to Table 19 for Action Labels.

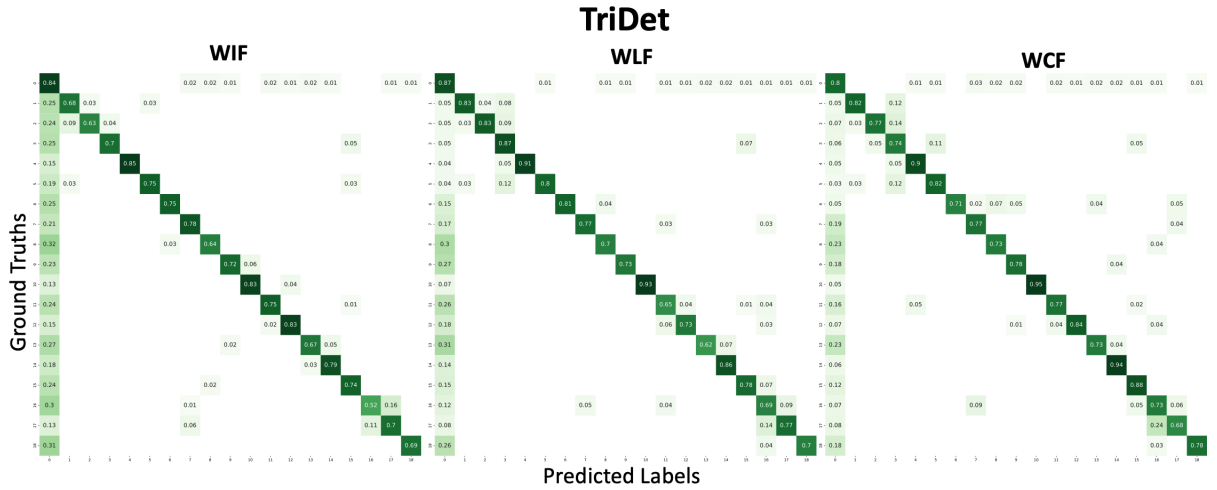

Figure 15.: Confusion matrices of TriDet, with **WIF** data, **WLF**, and **WCF** data

Figure 16 shows the graphs of train loss, validation loss, F1 score and mAP for 3 splits shown in table 1 for Tridet model with **WIF** data, **WLF**, and **WCF** data. Examining the training loss graphs reveals that the LSTM features exhibit faster convergence. Despite this, the validation loss for Convolutional features is lower. Notably, **WLF**, and **WCF** data achieve higher F1 and mAP more rapidly compared to the **WIF** data.

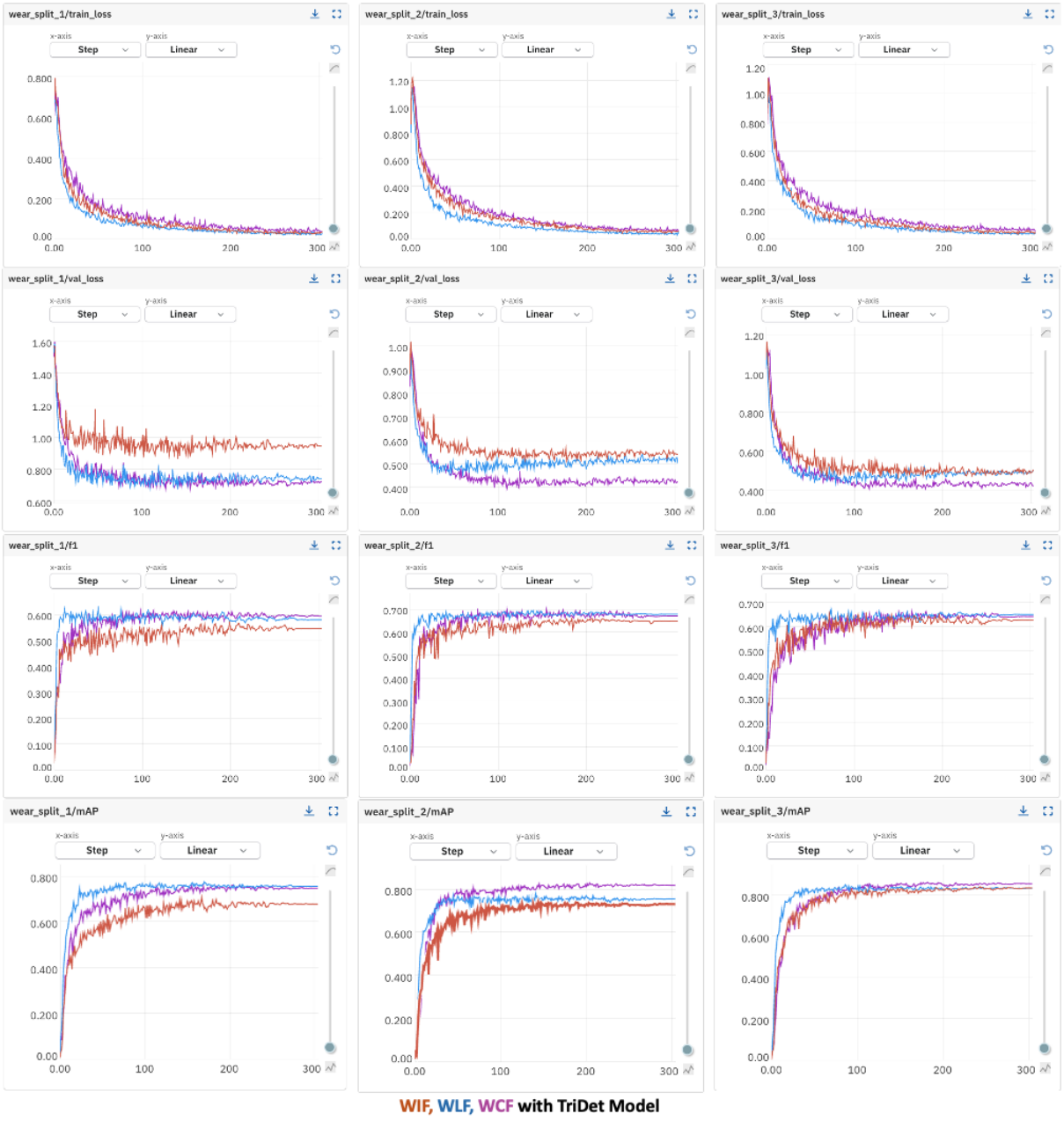

Figure 16.: Train Loss, Validation Loss, F1 Score and mAP graphs of TriDet, with **WIF** data, **WLF**, and **WCF** data, Refer to table 17 for abbreviations for the WEAR data

A.5. Graphs, Confusion matrix of Opportunity ADL dataset

Figure 17 shows the graphs of train loss, validation loss, F1 score and mAP for 3 splits shown in table 2 for Tridet model with **OAIF**, **OANF**, **OALF** and **OANLF**.

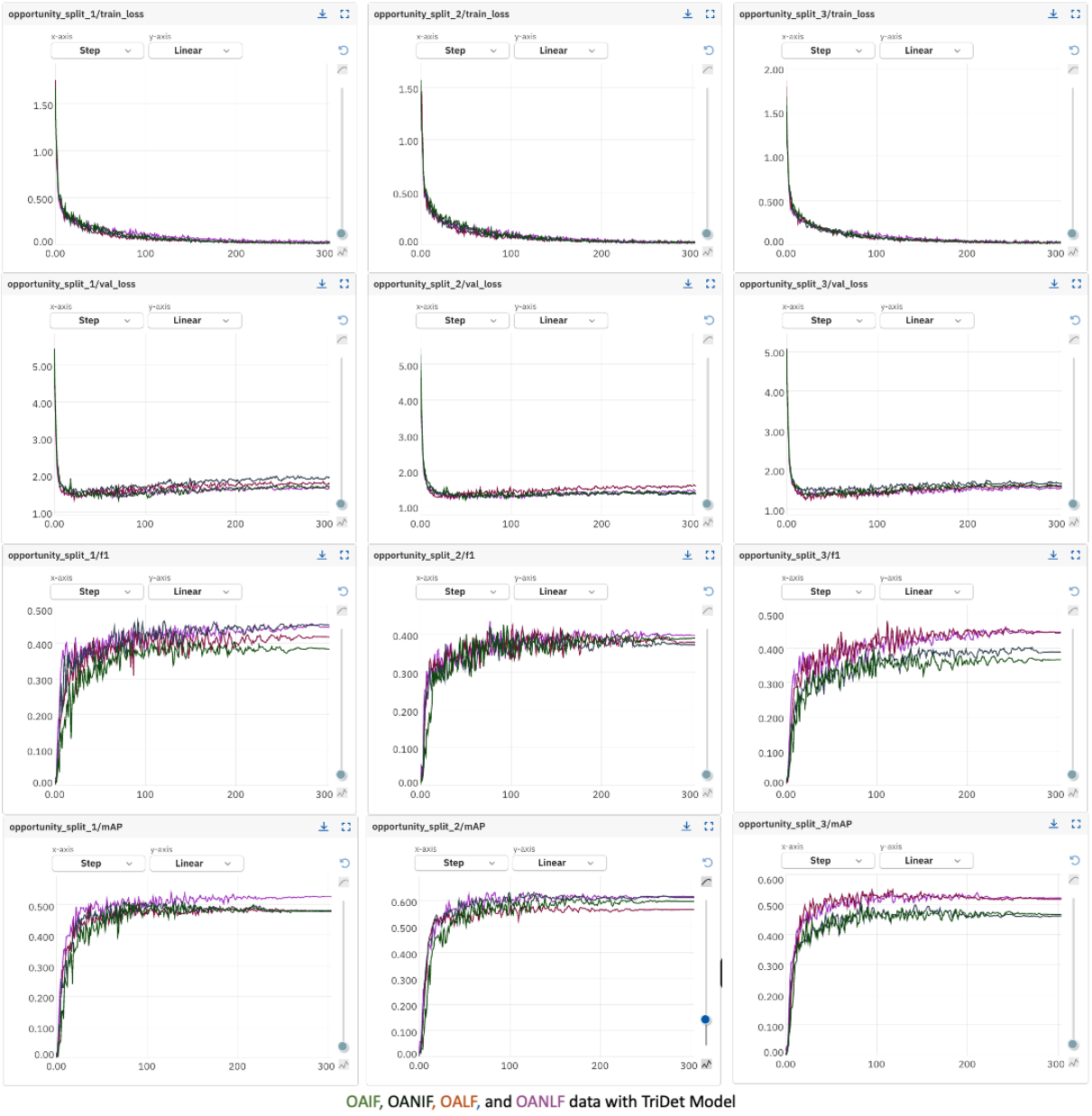

Figure 17.: Train Loss, Validation Loss, F1 Score and mAP graphs of TriDet model for Opportunity ADL dataset, Refer to table 18 for abbreviations for the Opportunity ADL Data

Figure 18 shows the Confusion Matrices for the Tridet model with**OAIF**, **OANF**, **OALF** and **OANLF**. Refer to Table 4 for Action Labels.

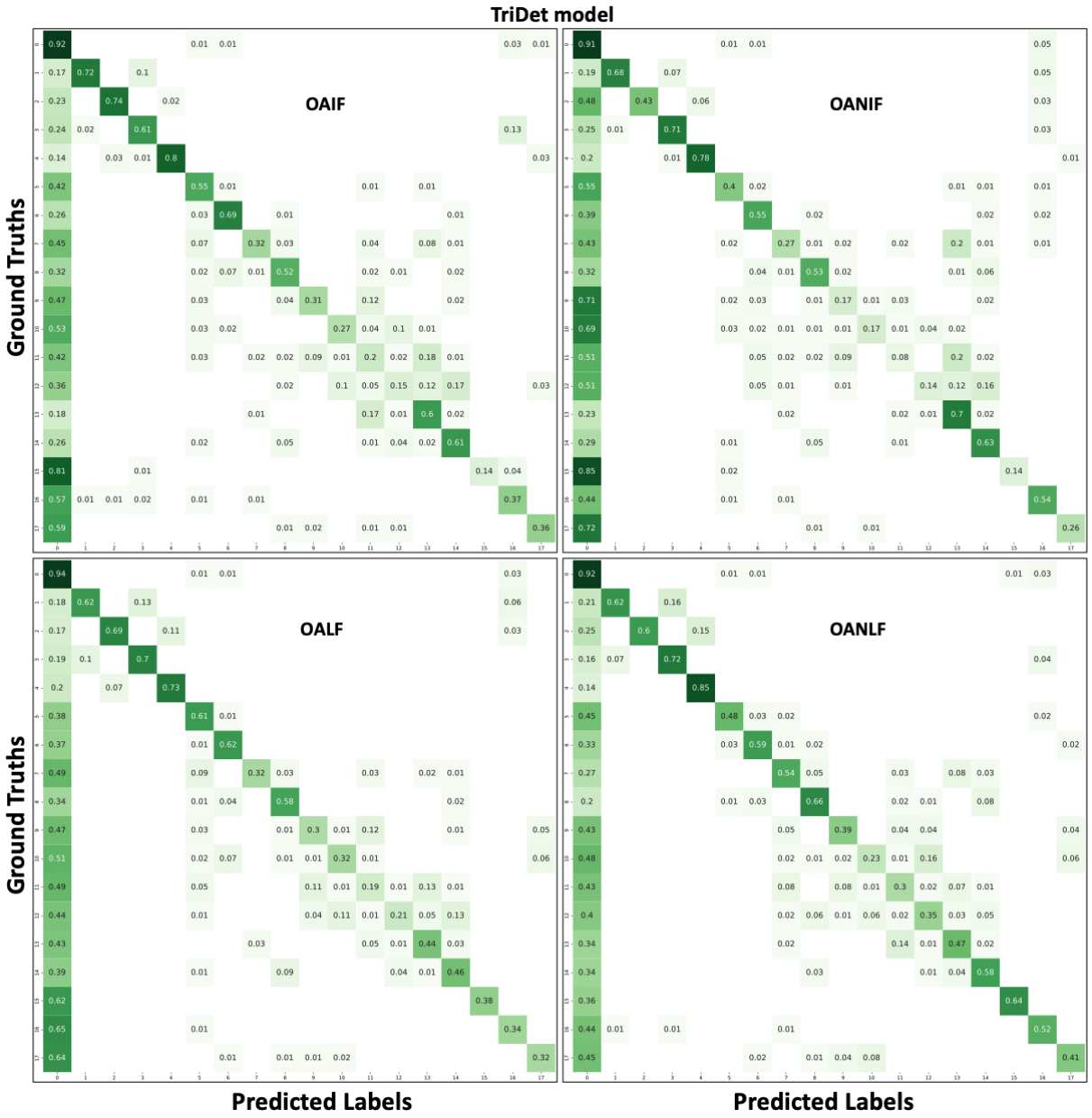

Figure 18.: Confusion Matrices of TriDet model for Opportunity ADL dataset, Refer to table 18 for abbreviations for the Opportunity ADL Data