Temporal Dynamics Modelling for People Counting in Point Clouds

An Extension on PointNet and MARS through LSTM Integration

Marina Escribano Esteban
Supervisors: Marco Zuñiga Zamalloa, Girish Vaidya
EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
22/06/2024

Name of the student: Marina Escribano Esteban
Final project course: CSE3000 Research Project
Thesis committee: Marco Zuñiga Zamalloa, Girish Vaidya, Michael Weinmann

Abstract

The People Counting Problem requires calculating the number of people in a region of interest. This is needed in crowd-monitoring scenarios but has become increasingly problematic when relying on video cameras, as they raise privacy concerns. Instead, we propose using a mmWave radar to detect people by creating point clouds from their radar signal reflections. This approach, however, can pose challenges when people walk closely together because their individual point clouds overlap and are seen as a single, larger cloud. It is difficult to count how many individuals this large point cloud holds, which can lead to miscounting the people in the scene. One approach to address this issue is leveraging the time dimension in people walking sequences, which can be done with Long Short-Term Memory (LSTM) models. Given this, we investigate how two state-of-the-art models, PointNet and MARS, perform for people counting from point clouds when extended through LSTMs. The results show how both PointNet and MARS improve performance when extended by LSTMs. Particularly, despite having over double the parameters, MARS+LSTM outperforms PointNet+LSTM in terms of accuracy and computational efficiency. MARS+LSTM can effectively capture small changes in the local structure of point clouds between frames, which PointNet loses due to max pooling. This highlights the importance of selecting a model architecture, like the CNN in MARS, that aligns with the data characteristics to maximise performance.

Keywords

LSTM, MARS, millimetre wave (mmWave), People Counting, Point Cloud, PointNet

1 Introduction

Precise people counting is crucial in various domains, including high-traffic zone optimisation, formulating strategic blueprints and conducting forensic investigations [1]. This challenge, referred to as *The People Counting Problem*, revolves around deciphering the patterns in the density of people within designated Regions of Interest (RoI) [2]. Yet, traditional methods that rely heavily on surveillance cameras face several issues. First, video cameras raise significant ethical concerns about privacy, legal compliance and the use of personally identifiable information (PII). Second, cameras require specific environmental conditions and proper lighting to capture scenes correctly. Additionally, maintenance of video systems is expensive.

To tackle these issues, we propose using a millimetre wave (mmWave) radar [3]. This technology creates point clouds to represent individuals, ensuring anonymity by not identifying specific people. An illustration of the cloud of points generated by an mmWave radar when a group of three people walk within its range can be seen in Figure 1. The issue we can observe from this figure is that, when people walk closely together, their point clouds can overlap and be mistakenly identified as a single, larger cloud rather than three separate clusters.

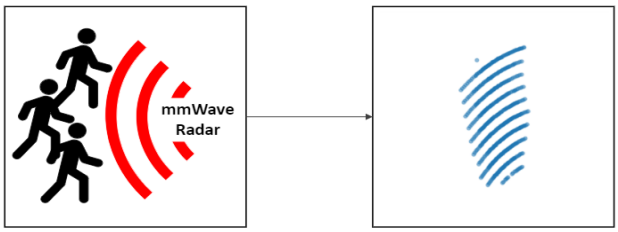

Figure 1: *mmWave radar capturing three people moving (left) and the resulting point cloud that would be generated from this sequence (right).*

To address the challenge of distinguishing and counting individuals within dense point cloud data, this research will employ two state-of-the-art models that can be adapted for people counting from point clouds: PointNet and MARS. In 2016, PointNet was presented as a deep neural network, which takes raw point cloud data as input and aims to classify and segment these 3D point sets into their true object class [4]. MARS, which stands for mmWave-based Assistive Rehabilitation System for Smart Healthcare, is a rehabilitation system that utilises mmWave sensors to reconstruct human poses in 3D by estimating the locations of 19 key joints [5]. However, PointNet's and MARS's capabilities to analyse temporal data from mmWave radar sequences remain constrained. Previous work has utilised Long Short-Term Memory (LSMT) models (derived from Recurrent Neural Networks) to implement sequence modelling, where body motion and appearance are captured over time [6], [7]. Nevertheless, this approach has not been applied to dynamic people counting using the proposed architectures.

This research aims to enhance the PointNet and MARS architectures by incorporating LSTM models to improve the counting of people from point clouds. We will assess the performance of PointNet and MARS for people counting, as opposed to object classification and posture reconstruction, respectively, and then evaluate how an LSTM extension can improve both the accuracy and processing time of the models. This study seeks to answer the question: **"How do PointNet and MARS perform when extended by an LSTM to count the people in a point cloud?"**

The paper's structure is tailored to address this question. Section 2 provides background information and related work, making it accessible to readers with varying levels of technological literacy. The methodology is covered in Section 3, Section 4 describes the

experimental setup and the results are presented and discussed in Section 5. Section 6 details how responsible research is adopted in the study, and Section 7 provides the conclusion and future work that could be investigated as an extension of this study.

2 Background

This section provides a detailed background on the problem through a technical description of the PointNet, MARS and LSTM architectures. It also covers related work done on these models and their utilities.

2.1 PointNet

PointNet is a deep neural network that directly consumes raw point clouds without the need to convert them to 3D voxel grids or image collections, effectively processing them in their irregular format [4]. PointNet can be used in applications such as object classification, part segmentation and scene semantic parsing. Its classification network can perform sixteen-class classification on point clouds.

PointNet's core strength revolves around its ability to handle the inherent disorder in raw point clouds, making it invariant to permutations of the input points. To achieve this robustness, two key features are introduced in its architecture. The first is the use of mini transformer networks (T-Nets), which align the input data into a standardised canonical space [8]. This alignment ensures that the model's outputs remain consistent despite variations in translation, scale, rotation, and other spatial transformations. The second and most crucial component is max pooling. Max pooling reduces the spatial dimensions of features by selecting the maximum value within each small window [9]. This process allows points to be in any order while keeping the output invariant to this randomness.

Figure 2 illustrates PointNet's structure [4]. The input to the classification network is an $n \times m$ data structure, where n is the number of points and m is the number of channels (or features). This input passes through the T-Nets and is processed through max pooling. The global feature vector obtained retains the most essential features while discarding less relevant ones. This vector is then processed by multi-layer perceptrons (MLPs), which extract local features without the complexities of convolutions to output the class probabilities.

2.2 mmWave-based Assistive Rehabilitation System (MARS)

MARS is a rehabilitation system that tracks patient at-home movements through a mmWave radar [5]. It utilises a deep neural network to estimate the positions of 19 joints on the human body from a point cloud input. The architectural foundation of MARS centres around a convolutional neural network (CNN) designed to process 5D point cloud data containing the spatial coordinates and movement dynamics of patients' joints.

The input layer of the CNN receives a stacked 5-channel feature map. Due to random data ordering posing a challenge to CNN design, MARS contains a preprocessing phase to ensure consistent and structured inputs. This involves the dataset being sorted by ascending coordinate values. The input then passes through two consecutive convolutional layers with 16 and 32 channels, respectively, that allow the network to extract spatial hierarchies from the data and help generate a flattened vector. The final fully connected layers learn non-linear combinations of the high-level features represented in the flattened vector and generate an output – 57 neurons, representing the 3D coordinates for the 19 joints.

2.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to effectively handle sequential data and capture dependencies [7]. In scenarios where data is split into sequences, LSTM networks have an exceptional ability for long-term modelling. Unlike standard RNNs, LSTM cells use a gating mechanism to control the flow of information through the network, as shown in Figure 3 [10]. The input gate $i^{(t)}$ and the forget gate $f^{(t)}$ determine what information from the input $x^{(t)}$ and previous hidden state $h^{(t-1)}$ should be added to the cell state $c^{(t)}$. The output gate $o^{(t)}$ controls the hidden state $h^{(t)}$ which is used in the current time step's output and as input in the next cell [6]. This selective retention ensures that only the most relevant information impacts the network's output, optimising both memory and processing power.

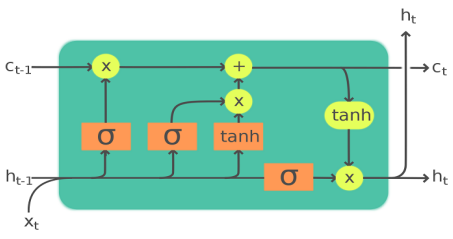

Figure 3: *This diagram is a visualisation of the internal architecture of an LSTM cell, showing how it structures the three gates (σ) — forget, input, and output [10].*

2.4 Related Work

Recent advancements in 3D point cloud processing have significantly influenced the object and gesture

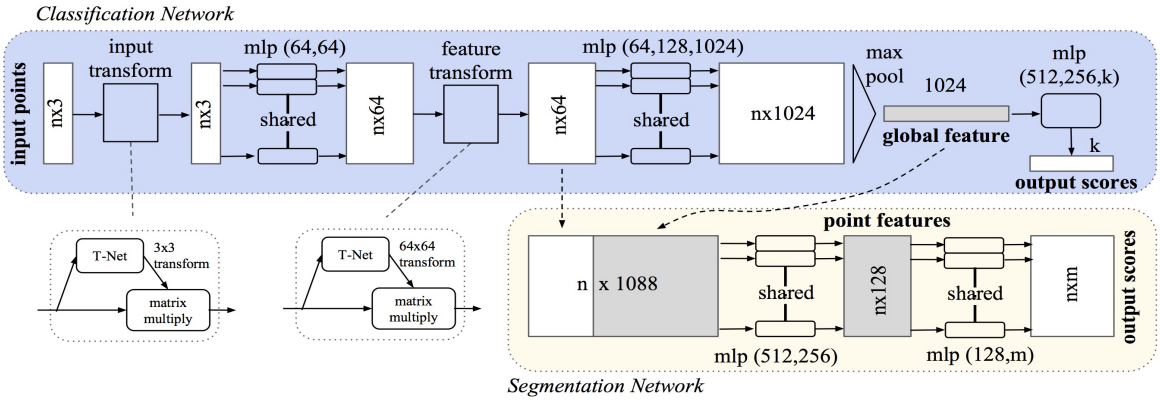

Figure 2: *This diagram illustrates the structure of PointNet [4]. The classification network transforms input points via two T-Nets (performing 3×3 and 64×64 transformations, respectively). These points pass through multiple layers of shared MLPs (Multi-Layer Perceptrons) before reaching a max pooling layer that aggregates global features. This is fed into another series of MLPs to produce final output scores for class probabilities. The segmentation network extends the classification pipeline by incorporating additional MLP layers that append per-point features to the global feature, necessary for detailed segmentation tasks.*

recognition domain, as well as people counting methodologies [11]. The integration of convolutional and recurrent neural networks shows promising results in harnessing spatial and temporal dimensions of point cloud data. Despite these advances, a notable gap remains in the literature: no existing studies have merged PointNet with temporal models for people counting, nor have they fully integrated the MARS architecture with such models. This paper seeks to bridge this gap. Two studies combining the architectures Point-CNN with Bi-LSTM and PointNet with LSTM, respectively, have demonstrated enhanced target and gesture recognition in diverse settings [12], [13]. This solid base can be adapted for people counting rather than gesture recognition. Moreover, studies demonstrating the effectiveness of CNN-LSTM networks in estimating human skeletons from radar data align closely with our proposed approach in data modality and network combination [14]. Similarly, other research efforts have leveraged LSTM-based frameworks for gesture recognition and anomaly detection, underscoring the robustness and versatility of LSTM methodologies in handling complex spatial-temporal data [15], [16].

3 Methodology

In this section, we describe the data utilised to run the experiments and the architecture of the two LSTM-extended models created in this study. Additionally, we cover the performance metrics used to evaluate the models.

3.1 Dataset

The dataset contains frames captured by the IWR6843ISK radar sensor from Texas Instruments (TI) [17]. This dataset consists of point cloud data, as seen in Figure 4, obtained from radar and video recordings of five volunteers walking individually and in various group sizes and formations. These frames are labelled into classes, namely 1, 2, 3, 4, 5 people and bikes, yielding 19,273 sequences of people and 829 sequences of bicycles.

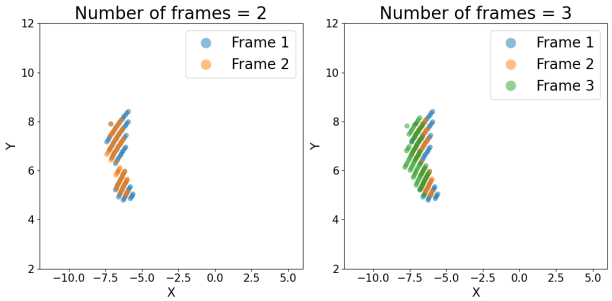

Figure 4: *Point cloud visualisation of consecutive frames. This depicts the movement of three people walking together for two frames (left) and three frames (right).*

3.2 Deep Learning Models

We adopt the deep learning models proposed in Section 2 for this paper's investigation on people counting. Moreover, we extend both models by an LSTM to leverage the time domain in sequences.

PointNet. The first deep learning model we create for the study is a modification of PointNet [4] implemented as a classification model. Our PointNet implementation classifies sequences of frames ranging from one to five people or bikes. In our application, the shape and size of a cluster relative to the distance from the radar is important information on the count of people. For this reason, we opt to remove the input and the feature transformations, as they change the size and distribution of point clouds.

This choice additionally reduces the model size and increases computational efficiency. Furthermore, we modify the configuration of the MLP to be (64, 128, 512), halving the global vector's size from 1024 to 512. Finally, we move the final shared MLP to follow the LSTM. We reduce its size to (512, 128, k), where k=6 in our six-class classification model, as opposed to PointNet's original sixteen-class classification. Our PointNet model architecture can be found in Figure 5.

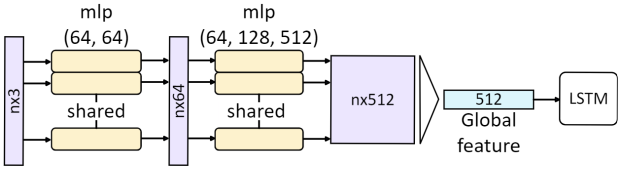

Figure 5: *Diagram of our modified PointNet architecture connected to an LSTM. The input is processed and transformed into a global feature vector. This vector is passed to an LSTM for sequential processing.*

MARS. The second deep learning model implemented is a modification of MARS [5] intended for people counting rather than pose reconstruction. Initially, we preserved the original architecture of the MARS model, including the flattening layer where data is transformed into a vector of 2048 features, rather than 512 features as done in PointNet. MARS's efficiency in processing time allows us to introduce a larger feature dimension to the LSTM. This decision is based on the principle that higher dimensional data contains more information, which can enhance the LSTM's ability to learn and the model's performance. As structured in our PointNet architecture, the LSTM is placed before the fully connected layers. These layers are composed of a shared MLP of size (512, 128, 6), where the final layer is of size six for the classification task. We replace the final regression function from MARS with the Logarithmic Softmax function used in classification tasks. Figure 6 shows our modified MARS architecture.

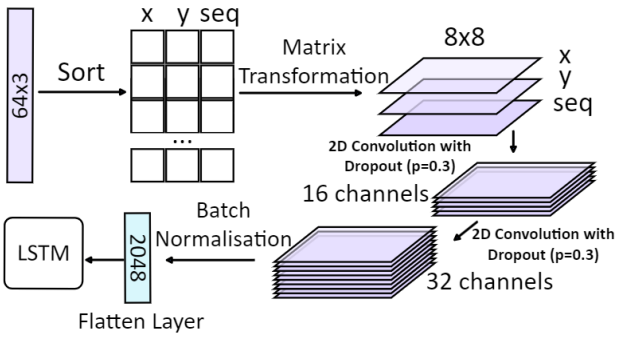

Figure 6: *Diagram of our modified MARS architecture connected to an LSTM. Input points are preprocessed, passed through the CNN and fed as a flattened vector into the LSTM.*

LSTM. The LSTM model takes inputs with features of 512 dimensions from PointNet and 2048 from MARS through one hidden layer of 512 neurons. The LSTM has a depth of one and a length corresponding to the number of frames. We process each frame consecutively with the output of one frame feeding into the next LSTM cell through the gate mechanism. The integrated LSTM architecture is seen in Figure 7.

The last output from the LSTM is fed into the **Feed Forward layer**, as shown in Figure 7. This layer begins with a ReLU activation function (Rectified Linear Unit) which introduces non-linearity to the network and allows the model to learn more complex data representations [18]. We then apply Batch Normalisation to minimise the internal covariate shift [19], and introduce a Fully Connected layer that reduces the 512 features from the LSTM down to 128 and introduces Dropout at a rate of 0.3. Dropout is explained in detail in Section 4.3 as a technique to prevent overfitting. This Feed Forward layer is repeated, with the second Fully Connected layer transforming the 128 features into outputs corresponding to the number of classes. In PointNet this was originally sixteen, but we perform a six-class classification instead.

3.3 Performance Metrics

We use a confusion matrix to evaluate the performance of the multiclass classification models, where the 'True Label' is on the y-axis, across from the 'Predicted Label' on the x-axis [20]. To evaluate performance quantitatively, we employ two numerical metrics on the results. The first metric utilised is Accuracy, which measures how many samples were correctly classified into their class from the total number of samples captured by the mmWave radar [21]. Mathematically, it can be expressed as:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

To account for the imbalance between datasets, we calculate a weighted average of individual class accuracies.

Additionally, the F1 score is used to evaluate the model's predictive power by looking at how well it performs for each class separately, as opposed to looking at performance as a whole, as Accuracy does. This score is the harmonic mean of Precision and Recall, where Precision represents the model's ability to identify instances of a specific class correctly and Recall represents the model's ability to identify all relevant cases of a specific class. The F1 score provides a single value that balances both metrics and is calculated using the formula:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

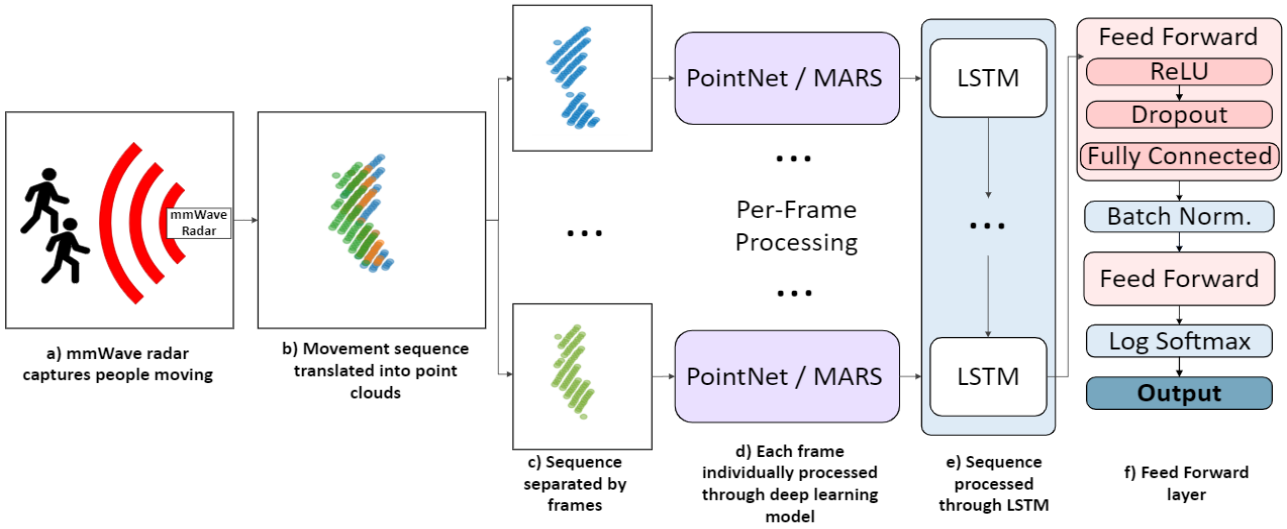

Figure 7: *This diagram illustrates the complete flow of data through the models and the feed-forward architecture. a) & b) mmWave radar captures movement and converts it into point clouds. c) Movement sequences are separated into individual frames. d) The deep learning models PointNet or MARS process each frame. e) The frames are fed into an LSTM network to capture temporal dependencies. f) The LSTM output is passed through a feed-forward network to produce the final output.*

This score ranges from 0 to 1, where 1 indicates perfect Precision and Recall, and 0 indicates the poorest performance.

4 Experimental Setup

This section will discuss the study's experimental and parameter settings and how Dropout is introduced as an overfitting preventive technique.

4.1 Data Preprocessing

After filtering the initial dataset to keep sequences with five frames, we obtain 19,346 sequences. It is important to note that each class comprises ∼4000 samples, but bikes are less represented with slightly over 800 samples. This dataset is shuffled and divided into a split of 60% for training, 20% for validation, and 20% for testing.

Before training the models, we preprocess the data to ensure it is clean and consistent. First, the clusters are standardised to equal sizes of 64 points per cloud. If the initial point cloud is larger than this target input size, we trim it by removing points with the lowest Signal-to-Noise values, which measure the strength of the radar signal relative to the background noise. This process ensures that the remaining points contain the most accurate information from the people in the scene and the minimal noise impact. If the point cloud is smaller than the expected size, we pad it through Agglomerative Clustering [22] in the case of PointNet. Alggomerative Clustering is a widespread technique to upsample data by introducing centroids according to the already existing points [23]. In the case of MARS, we pad it with zeros to be consistent with the original paper [5].

Moreover, we add random noise to the data in the PointNet model to improve its robustness and enhance generalisation [24]. For the MARS model, we sort the points in a consistent manner to the original MARS implementation, which facilitates the model's ability to process and learn from the spatial relationships in the data more effectively.

We refrain from implementing normalisation, as when points are scaled to a normalised space, important information on the coordinates and distribution of points within a point cloud is lost. Similarly, applying a random rotation would also distort the inherent spatial relationships and orientations of the points, potentially degrading the model's ability to interpret the structure of the data accurately. Therefore, we do not implement random rotation as a preprocessing step.

4.2 Hyperparameters

We introduce two important hyperparameters in the experiments that define what properties of the dataset are considered. The following are:

- ***Persistence***. This parameter represents the number of frames per sequence. Recordings of people walking can be of different lengths, which determine the number of frames used for each sequence. This parameter has a range from one to five, representing how many frames a sequence can have. The latter holds more information to be processed by the LSTM, however, it can result in longer run times due to increased information provided.

- ***Features***. This parameter represents what features of the data points should be considered when processing. This ranges from two to four features, where two features consider the x-coordinate and the y-coordinate, three additionally consider the SNR (Signal-to-Noise Ratio) value and four also consider the velocity.

We conduct training on an NVidia Tesla V100S GPU provided by the DelftBlue Supercomputer[1]. Running the PointNet and MARS models for 40 and 100 epochs, respectively, revealed the accuracy curves for the training and validation sets. PointNet is not tested for more epochs because of its high time complexity. It can be observed that the accuracy saturates at around 30 epochs in both cases (see Appendix A and Appendix B), leading us to train the models for this duration.

For performance comparison, we run both models without the LSTM extension on 30 epochs. These models retain the same architecture as described in Section 3.2, but the data bypass the LSTM and instead are directly processed by the Feed Forward layer (see Figure 7). To train these base models, the input is shaped as: (total number of points from all frames, number of features), with the number of points determined by persistence, as shown in Table 1. The experiments are conducted on sequences with persistence of five and points with four features, using a batch size of 64.

Persistence	Number of points
1	64
2	128
3	256
4	512
5	1024

Table 1: *Input size definitions based on persistence (number of frames per sequence).*

The input to the LSTM-extended models is shaped: (persistence, number of points per frame, number of features), where persistence is five and each frame holds 64 points after being standardised. Moreover, we process data with four features in batches of 64.

The loss used in training is the Negative Log-Likelihood Loss ('torch.nn.NLLLoss' in PyTorch), which is suitable for classification problems with C classes (C = 6 in these experiments) [25].

Furthermore, the Adam optimiser is employed to optimise the loss with a learning rate of 0.001 [26]. We chose the Adam optimiser for its ability to adaptively adjust weights to learn parameters individually. This improves convergence efficiency in these time-consuming training models.

[1] https://doc.dhpc.tudelft.nl/delftblue/

4.3 Dropout

Dropout layers are implemented to reduce the density of a neural network by randomly deactivating neurons during training. This technique helps mitigate overfitting by effectively averaging the network's weights [27]. We introduce these layers with a Dropout rate of 0.3 before the Fully Connected layers in the Feed Forward networks and after each convolutional layer in MARS. As shown in Appendix C and Appendix D, the models do not overfit when adjusting the Dropout probability to 0.3, hence we do not increase the rate further.

5 Results

This section showcases the results obtained from running the experiments as described in Section 4, and analyses them to obtain a reflection on the work.

5.1 Base Models

The base PointNet model performs to an accuracy of 62.5% when run on the discussed settings. The confusion matrix with specific class probabilities can be found in Appendix E. This model contains approximately 140,000 parameters and takes slightly under 17 hours to train (see Table 2).

The MARS base model can classify frames into the true number of people or bikes with an accuracy of 60.1%. This accuracy is comparable to PointNet's as the 2.4% difference can be accounted for in the randomness when training the models. For further details on the specific probability distribution of MARS, refer to Appendix F. The training runtime is around 28 minutes on this model of approximately 4,200,000 parameters (Table 2). Despite having forty times the parameters of PointNet, MARS has a thirty-two-times shorter training period. We can conclude this vast contrast is due to the models' different data handling and processing techniques. First, MARS reshapes input data into a 2D format, allowing it to process information in larger batches, but PointNet processes data sequentially. This sequential nature of PointNet limits its ability to use parallel processing. Second, each model's padding operation greatly affects its runtime. This is discussed in detail in Section 5.4.

5.2 PointNet+LSTM

The performance of the PointNet+LSTM model shows reasonable effectiveness in classifying objects, with an overall accuracy of 80% and an F1 score of 0.797, which indicates a good balance between Precision and Recall. The increase in Accuracy of 17.5% from the base PointNet model shows performance improvement when extending the architecture through

an LSTM. We can see the areas of strength in the confusion matrix, shown in Figure 8. These include higher accuracy in identifying between people classes but also reveal that the probability of misclassification is shifted to adjacent classes, particularly between the '3 person' and '4 person' classes. Figure 9 shows how point clouds for three and four people are similar in dimensions, reasoning the model's low ability to distinguish between them. This suggests there is potential for improvement in distinguishing between similar classes.

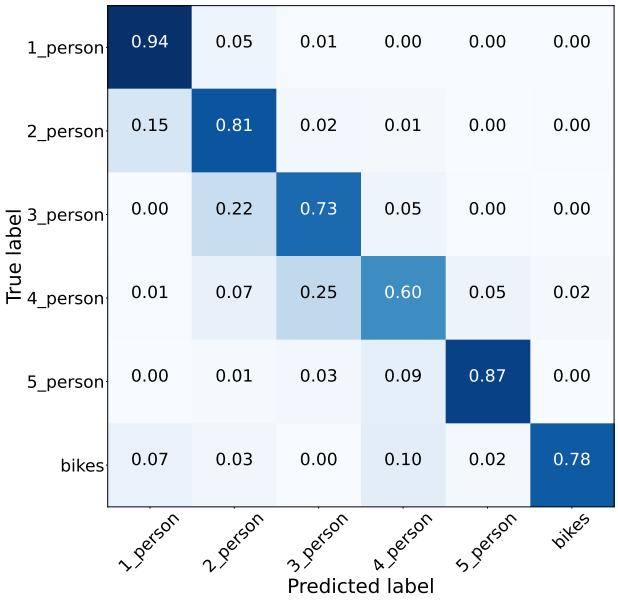

Figure 8: *Confusion Matrix of the PointNet+LSTM model on the test dataset.*

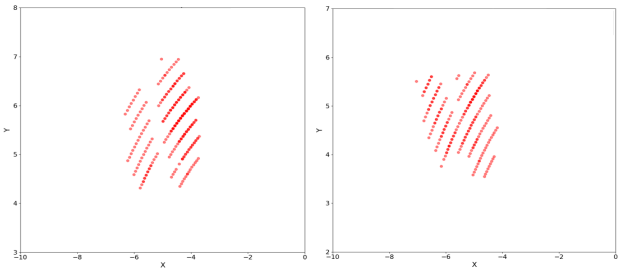

Figure 9: *Point clouds of three (left) and four (right) people walking closely together. It can be hard for the model to distinguish the exact count of people between three or four people.*

As shown in Table 2, PointNet+LSTM has sixteen times the parameters of the base PointNet but interestingly takes three hours less to train. We identified this arose from the use of Agglomerative Clustering when padding point clouds, which is an operation of time complexity $O(n^3)$. In PointNet, we pad clusters to 1024 points (Table 1), whereas in PointNet+LSTM we pad to 320 points, due to having 5 frames of 64 points each. Although this difference in padding sizes appears small, the $O(n^3)$ operation can lead to highly increased run times, as proven by our results.

Model	No. Parameters	Training Time	Accuracy
PointNet	142,790	16h	62.5%
MARS	4,200,758	25-30 mins	60.1%
PointNet +LSTM	2,243,974	13h	80%
MARS +LSTM	5,318,966	55-60 mins	89.1%

Table 2: *A summary of each model's total number of parameters, training time and accuracy.*

5.3 MARS+LSTM

When running MARS+LSTM on the settings described in Section 4, the model performs to an accuracy of 89.1%, showing an increase of 29% from the MARS base model, and of 9.1% from the accuracy of PointNet+LSTM. The model has an F1 score of 0.890, indicating a high effectiveness at correctly identifying and classifying relevant instances. The distribution of classification accuracy over classes can be seen on Figure 10, which demonstrates excellent recognition of individual classes, particularly for '1 person' at 97% accuracy.

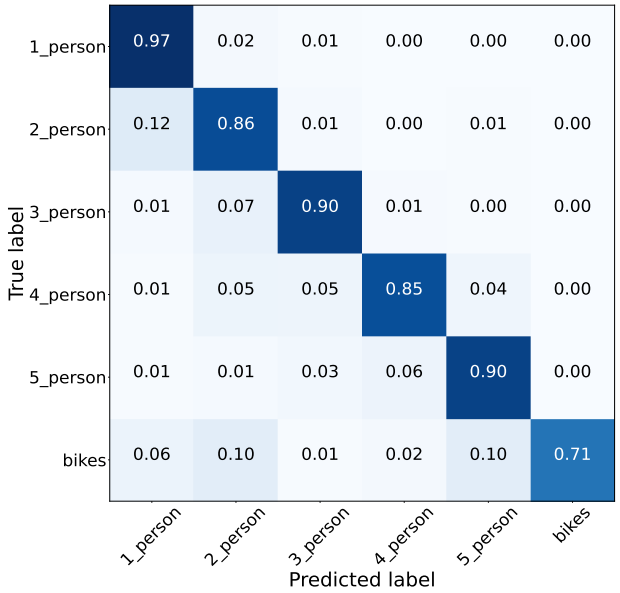

Figure 10: *Confusion Matrix of the MARS+LSTM model on the test data set.*

The model appears particularly robust in distinguishing between different person classes, showing a clear improvement in reducing adjacent misclassifications compared to the PointNet+LSTM model. Finally, the extended MARS+LSTM model has a run time of 1 hour to train – thirteen times less than

PointNet+LSTM.

5.4 Discussion

As presented in Table 2, the LSTM-extended models perform notably better than the base models, indicating that leveraging temporal data from people walking sequences through an LSTM improves the results of our proposed models. PointNet benefits from the LSTM's ability to learn long-term dependencies, enhancing its sequential data processing capabilities, while MARS leverages its CNN architecture to capture spatial hierarchies and local structures before feeding them into the LSTM for temporal analysis. This ability in the models to effectively integrate spatial and temporal information from point clouds achieves improved performance.

To understand why the PointNet+LSTM model performs worse than the MARS+LSTM model in both accuracy and training time, we must consider how each model processes data and the operations it performs.

Regarding runtime, both MARS models take at least twelve hours less than both PointNet models. As previously analysed, this difference stems from PointNet's use of Agglomerative Clustering for padding, which introduces a runtime complexity of $O(n^3)$. This operation increases complexity exponentially with more data being input. This increase in time is particularly pronounced with the large dataset being used in the experiments (Section 4). In contrast, MARS uses zero padding, with a runtime complexity of $O(n)$. When employing zero padding rather than Agglomerative Clustering in PointNet+LSTM, it takes slightly over an hour to run, a duration comparable to the hour that MARS+LSTM takes. However, zero padding was not used in PointNet because the models did not achieve accuracies higher than approximately 20%, due to the distortion of spatial structure introduced in the point clouds by zero-valued points, which PointNet struggles to learn from. Conversely, running MARS+LSTM with Agglomerative Clustering in place of zero padding results in a significantly longer runtime of seventeen hours, supporting the argument that more computationally intensive methods can lead to improved performance at the cost of increased processing time.

Additionally, MARS+LSTM achieves a modestly higher accuracy than PointNet+LSTM (9.1% higher). Since the LSTM architectures in the models process frames separately, they feed on the small changes in the local structure of point clouds between frames. This local information is lost in PointNet when applying max pooling on the data, which aggregates features and discards spatial hierarchies, making it more challenging for LSTMs to learn from the data. In contrast, the CNN architecture in MARS captures these discrepancies in local structure between point clouds more effectively, resulting in improved performance when extended by an LSTM. This difference underscores the importance of selecting the model architecture that aligns with the data characteristics and processing requirements to maximise performance.

6 Responsible Research

In this section, we discuss the ethical aspects of the research and the measures taken to ensure reproducibility and transparency.

6.1 Reproducibility

Intending to create reproducible work and contribute to the field of open research, we developed a transparent methodology and provided full visibility of the experiments' results. Furthermore, the source code of the base models used is publicly available[2], and any modifications are discussed in detail throughout the paper. This research adheres to the FAIR (Findable, Accessible, Interoperable, and Reusable) principles through detailed documentation to facilitate other researchers to understand and reproduce this work.

6.2 Ethical Considerations

The data obtained for this research is handled carefully to maintain participants' privacy. By processing point clouds rather than images of individuals, we ensure that no personally identifiable information (PII) is utilised. This approach aligns with ethical standards and privacy regulations, guaranteeing that participant anonymity is preserved.

Finally, this research adheres to the Netherlands Code of Conduct for Research Integrity, as it falls under the scope of scholarly work conducted at TU Delft and is governed by this code [28]. This entails that the principles (Honesty, Scrupulousness, Transparency, Independence and Responsibility) are followed, all results have been reported and responsible methodologies have been followed.

7 Conclusions and Future Work

This study effectively addresses the issue of counting people in motion from point cloud data. We extended the PointNet [4] and MARS [5] structures with Long Short-Term Memory (LSTM) models to test our hypotheses on whether accounting for temporal data would improve the performance of these models

[2] https://towardsdatascience.com/deep-learning-on-point-clouds-implementing-pointnet-in-google-colab-1fd65cd3a263 and https://github.com/SizheAn/MARS

when counting people from point clouds. The adjusted PointNet and MARS frameworks demonstrate enhancements in recognising the time-based patterns of groups walking together. The PointNet+LSTM outperforms the original PointNet model by 17.5% and reaches an accuracy of 80%. Moreover, the MARS+LSTM model achieves an accuracy of 89.1%, showing a significant increase of 16.7% from the base MARS model and of 9.1% from PointNet+LSTM.

These findings highlight the effectiveness of integrating temporal data processing using LSTMs into point cloud analysis, fully answering the original study question: "How do PointNet and MARS perform when extended by an LSTM to count the people in a point cloud?"

We should explore further work to improve the performance of the models proposed in this study. Firstly, the data set could be enhanced to be balanced across classes and expanded to include data samples of different environmental conditions and sensor positions. This could increase PointNet and MARS capabilities to generalise across various scenarios. Furthermore, more complex sequence modelling architectures such as Transformer networks should be implemented as an extension to PointNet and MARS, as they might capture spatial-temporal relationships more effectively than LSTMs. Finally, this research was limited by the extensive runtimes of the models. For future work, we should investigate the real-time processing capabilities of the proposed models, as the current runtimes are too long for them to sustain their performance in dynamic environments where instant data processing is needed.

In conclusion, there is further to be investigated in the field of point cloud data processing and how to leverage the temporal domain with other sequence modelling architectures. However, LSTMs show promising results when combined with state-of-the-art models, PointNet and MARS, to count the number of people from point clouds.

Bibliography

[1] A. Tomar, S. Kumar and B. Pant, 'Crowd Analysis in Video Surveillance: A Review', in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, Mar. 2022, pp. 162–168. DOI: 10.1109/DASA54658.2022.9765008. [Online]. Available: https://ieeexplore.ieee.org/document/9765008 (visited on 22nd Jun. 2024).

[2] L. Ren, A. G. Yarovoy and F. Fioranelli, 'Grouped People Counting Using mm-Wave FMCW MIMO Radar', *IEEE Internet of Things Journal*, vol. 10, no. 22, pp. 20107–20119, Nov. 2023, Conference Name: IEEE Internet of Things Journal, ISSN: 2327-4662. DOI: 10.1109/JIOT.2023.3282797. [Online]. Available: https://ieeexplore.ieee.org/document/10143995 (visited on 22nd Jun. 2024).

[3] C. Iovescu, 'The fundamentals of millimeter wave sensors', 2017. [Online]. Available: https://www.semanticscholar.org/paper/The-fundamentals-of-millimeter-wave-sensors-Iovescu/a5b2191ff17a96c82538d345ce61cbb4800d428e (visited on 22nd Jun. 2024).

[4] C. R. Qi, H. Su, K. Mo and L. J. Guibas, *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*, arXiv:1612.00593 [cs], Apr. 2017. DOI: 10.48550/arXiv.1612.00593. [Online]. Available: http://arxiv.org/abs/1612.00593 (visited on 24th Apr. 2024).

[5] S. An and U. Y. Ogras, 'MARS: mmWave-based Assistive Rehabilitation System for Smart Healthcare', *ACM Transactions on Embedded Computing Systems*, vol. 20, no. 5s, 72:1–72:22, Sep. 2021, ISSN: 1539-9087. DOI: 10.1145/3477003. [Online]. Available: https://doi.org/10.1145/3477003 (visited on 22nd Jun. 2024).

[6] J. Donahue, L. A. Hendricks, M. Rohrbach *et al.*, *Long-term Recurrent Convolutional Networks for Visual Recognition and Description*, arXiv:1411.4389 [cs], May 2016. DOI: 10.48550/arXiv.1411.4389. [Online]. Available: http://arxiv.org/abs/1411.4389 (visited on 22nd Jun. 2024).

[7] S. Hochreiter and J. Schmidhuber, 'Long Short-Term Memory', *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735 (visited on 22nd Jun. 2024).

[8] M. Jaderberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu, *Spatial Transformer Networks*, arXiv:1506.02025 [cs], Feb. 2016. DOI: 10.48550/arXiv.1506.02025. [Online]. Available: http://arxiv.org/abs/1506.02025 (visited on 22nd Jun. 2024).

[9] L. Zhao and Z. Zhang, 'A improved pooling method for convolutional neural networks', en, *Scientific Reports*, vol. 14, no. 1, p. 1589, Jan. 2024, Publisher: Nature Publishing Group, ISSN: 2045-2322. DOI: 10.1038/s41598-024-51258-6. [Online]. Available: https://www.nature.com/articles/s41598-024-51258-6 (visited on 22nd Jun. 2024).

[10] J. Howard and S. Gugger, *Deep Learning for Coders with Fastai and PyTorch: AI Applications Without a PhD*, en. O'Reilly Media, Incorporated, 2020, Google-Books-ID: xd6LxgEACAAJ, ISBN: 978-1-4920-4552-6.

[11] G. Vaidya and M. Zuniga, *A Deep-Learning Approach to Estimate People Count with mmWave Point Clouds*, Place: Hong Kong, May 2024. [Online]. Available: https://conferences.computer.org/iotDI/2024/program.html.

[12] P. Cheng, Q. Zhu and W. Wang, '3D point cloud target recognition based on the Bi-LSTM and PointCNN network', IEEE, Nov. 2022. DOI: 10.1109/cisp-bmei56279.2022.9980199. [Online]. Available: http://dx.doi.org/10.1109/CISP-BMEI56279.2022.9980199.

[13] N. Kern, T. Grebner and C. Waldschmidt, 'PointNet+LSTM for Target List-Based Gesture Recognition With Incoherent Radar Networks', *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, pp. 5675–5686, Dec. 2022. DOI: 10.1109/TAES.2022.3179248.

[14] M. M. Rahman, D. Martelli and S. Gurbuz, *Radar-Based Human Skeleton Estimation with CNN-LSTM Network Trained with Limited Data*. Oct. 2023, Pages: 4. DOI: 10.1109/BHI58575.2023.10313438.

[15] Y. Min, Y. Zhang, X. Chai and X. Chen, *An Efficient PointLSTM for Point Clouds Based Gesture Recognition*. Jun. 2020, Pages: 5769. DOI: 10.1109/CVPR42600.2020.00580.

[16] J. Ning, L. Chen, C. Zhou and D. Liu, 'Multi-object Spatial–Temporal Anomaly Detection Using an LSTM-Based Framework', *Neural Processing Letters*, vol. 53, Mar. 2021. DOI: 10.1007/s11063-021-10456-3.

[17] *IWR6843ISK Evaluation board | TI.com*. [Online]. Available: https://www.ti.com/tool/IWR6843ISK (visited on 22nd Jun. 2024).

[18] *Rectifier (neural networks)*, en, Page Version ID: 1221547062, Apr. 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Rectifier_(neural_networks)&oldid=1221547062 (visited on 22nd Jun. 2024).

[19] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv:1502.03167 [cs], Mar. 2015. DOI: 10.48550/arXiv.1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167 (visited on 22nd Jun. 2024).

[20] S. V. Stehman, 'Selecting and interpreting measures of thematic classification accuracy', *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, Oct. 1997, ISSN: 0034-4257. DOI: 10.1016/S0034-4257(97)00083-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425797000837 (visited on 22nd Jun. 2024).

[21] *Accuracy, precision, and recall in multi-class classification*, en. [Online]. Available: https://www.evidentlyai.com/classification-metrics/multi-class-metrics (visited on 22nd Jun. 2024).

[22] *Agglomerative Clustering*, en. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html (visited on 22nd Jun. 2024).

[23] S. Palipana, D. Salami, L. Leiva and S. Sigg, 'Pantomime: Mid-Air Gesture Recognition with Sparse Millimeter-Wave Radar Point Clouds', English, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, 2021, ISSN: 2474-9567. DOI: 10.1145/3448110.

[24] M. Zhou, T. Liu, Y. Li, D. Lin, E. Zhou and T. Zhao, 'Toward Understanding the Importance of Noise in Training Neural Networks', en, in *Proceedings of the 36th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, May 2019, pp. 7594–7602. [Online]. Available: https://proceedings.mlr.press/v97/zhou19d.html (visited on 22nd Jun. 2024).

[25] *NLLLoss — PyTorch 2.3 documentation*. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.NLLLoss.html (visited on 22nd Jun. 2024).

[26] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017. DOI: 10.48550/arXiv.1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 22nd Jun. 2024).

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting', *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html (visited on 22nd Jun. 2024).

[28] The Netherlands Organisation for Scientific Research (NWO), *The Netherlands Code of Conduct for Research Integrity*, 2018. [Online]. Available: https://www.nwo.nl/sites/nwo/files/documents/Netherlands%2BCode%2Bof%2BConduct%2Bfor%2BResearch%2BIntegrity_2018_UK.pdf.

Appendix

A PointNet+LSTM Accuracy

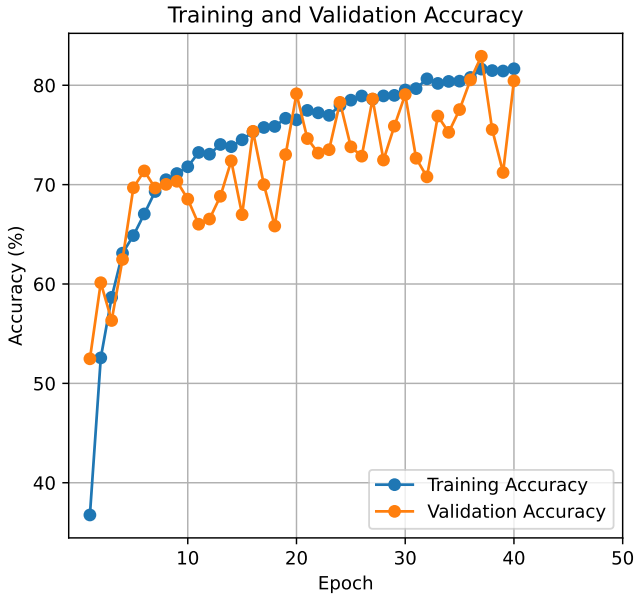

Figure 11: *PointNet+LSTM model accuracy diagram over 40 epochs for the training and validation data.*

B MARS+LSTM Accuracy

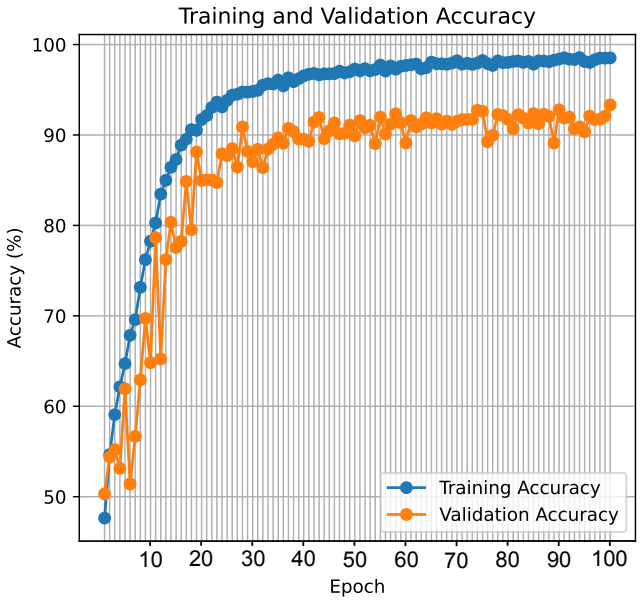

Figure 12: *MARS+LSTM model accuracy diagram over 100 epochs for the training and validation data.*

C PointNet+LSTM Loss

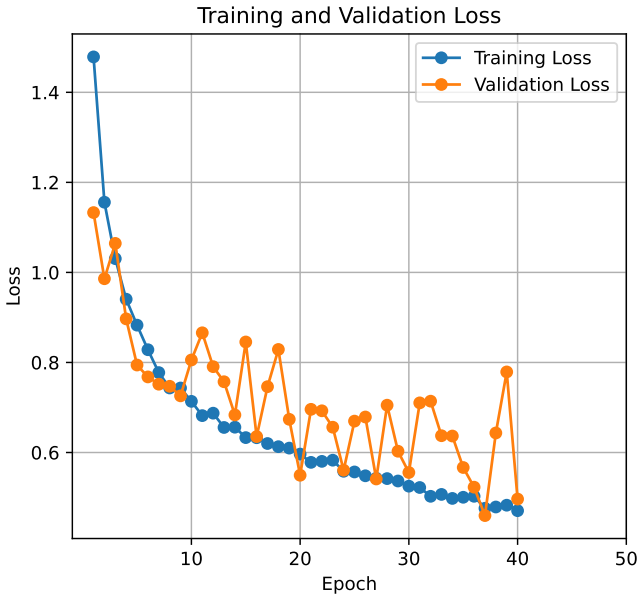

Figure 13: *PointNet+LSTM model loss diagram over 40 epochs for the training and validation data.*

D MARS+LSTM Loss

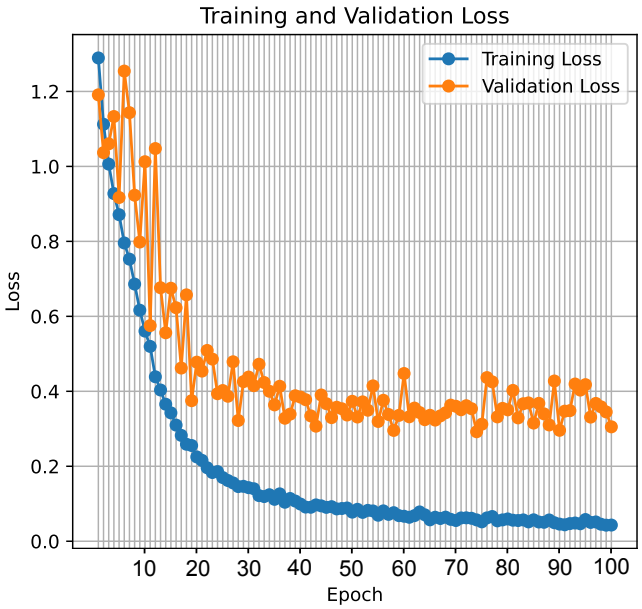

Figure 14: *MARS+LSTM model loss diagram over 100 epochs for the training and validation data.*

E PointNet Confusion Matrix

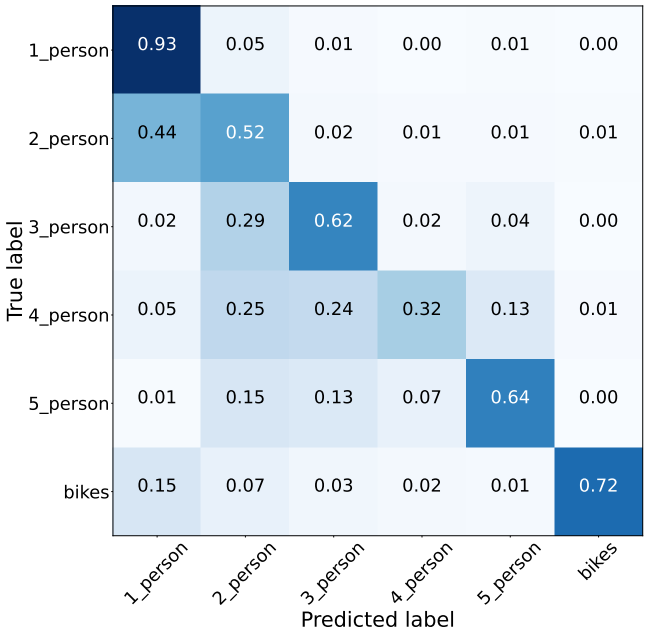

Figure 15: *Confusion Matrix of the PointNet model on the test data set.*

F MARS Confusion Matrix

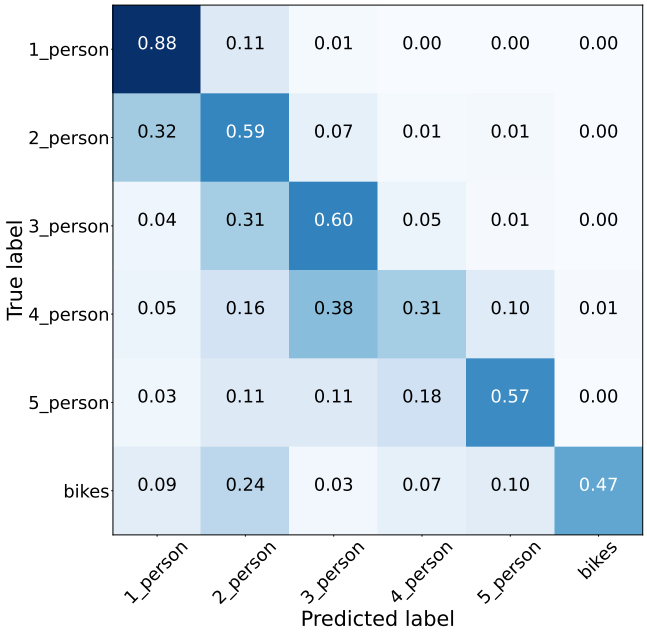

Figure 16: *Confusion Matrix of the MARS model on the test data set.*