

Article

Human-Aware Collaborative Robots in the Wild: Coping with Uncertainty in Activity Recognition

Beril Yalçinkaya ^{1,2,*} , Micael S. Couceiro ¹ , Salviano Pinto Soares ^{2,3,4}  and Antonio Valente ^{2,5} 

- ¹ Ingeniarius, Ltd., R. Nossa Sra. Conceição 146, 4445-147 Alfena, Portugal; micael@ingeniarius.pt
- ² Engineering Department, School of Sciences and Technology, University of Trás-os-Montes and Alto Douro (UTAD), Quinta de Prados, 5000-801 Vila Real, Portugal; salblues@utad.pt (S.P.S.); avalente@utad.pt (A.V.)
- ³ Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, 3810-193 Aveiro, Portugal
- ⁴ Intelligent Systems Associate Laboratory (LASI), University of Aveiro, 3810-193 Aveiro, Portugal
- ⁵ INESC TEC, Campus da Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
- * Correspondence: beril@ingeniarius.pt

Abstract: This study presents a novel approach to cope with the human behaviour uncertainty during Human-Robot Collaboration (HRC) in dynamic and unstructured environments, such as agriculture, forestry, and construction. These challenging tasks, which often require excessive time, labour and are hazardous for humans, provide ample room for improvement through collaboration with robots. However, the integration of humans in-the-loop raises open challenges due to the uncertainty that comes with the ambiguous nature of human behaviour. Such uncertainty makes it difficult to represent high-level human behaviour based on low-level sensory input data. The proposed Fuzzy State-Long Short-Term Memory (FS-LSTM) approach addresses this challenge by fuzzifying ambiguous sensory data and developing a combined activity recognition and sequence modelling system using state machines and the LSTM deep learning method. The evaluation process compares the traditional LSTM approach with raw sensory data inputs, a Fuzzy-LSTM approach with fuzzified inputs, and the proposed FS-LSTM approach. The results show that the use of fuzzified inputs significantly improves accuracy compared to traditional LSTM, and, while the fuzzy state machine approach provides similar results than the fuzzy one, it offers the added benefits of ensuring feasible transitions between activities with improved computational efficiency.

Keywords: human activity recognition and modelling; deep learning; human-robot collaboration; fuzzy logic; finite state machine; long short-term memory



Citation: Yalçinkaya, B.; Couceiro, M.S.; Soares, S.P.; Valente, A. Human-Aware Collaborative Robots in the Wild: Coping with Uncertainty in Activity Recognition. *Sensors* **2023**, *23*, 3388. <https://doi.org/10.3390/s23073388>

Academic Editor: José María Martínez-Otzeta

Received: 3 March 2023

Revised: 15 March 2023

Accepted: 20 March 2023

Published: 23 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Importance of Human Activity Recognition for Human-Robot Collaboration

For years, robots and humans have been separated in different workspaces, whether it be industrial or field applications. The reason for this separation is primarily for safety. Even though robots have been designed for specific tasks, in most cases, they are not aware of the environment and surrounding dynamic agents. As a result, these robots are often placed in cages or in a completely separate environment from human operators [1]. This separation has resulted in issues, such as low adaptability in different environments, costly setup, and limited flexibility, which do not align with the ideals of Industry 4.0, which demands fast production and efficiency.

To address these demands, Human-Robot Collaboration (HRC) has become a major trend in robotics in recent years. The goal is to improve efficiency and productivity by combining the benefits of humans' critical thinking and empathy, with robots' physical robustness in demanding and often dangerous conditions [2]. The idea is for humans and robots to work together towards a common goal. Research has demonstrated that

interaction and collaboration between humans and robots are crucial factors in achieving ergonomic systems and enhancing the quality and efficiency of the production process [3].

Collaborative robots, also known as “co-bots”, have become increasingly prevalent in industrial settings in recent years. However, they have also been utilized in a variety of other domains. For example, in the healthcare field, researchers are developing robotic walkers [4], wheelchairs [5], and elderly care robots [6]. Collaborative robots have the potential to assist humans in heavy and dangerous tasks as well, such as construction and search-and-rescue [7]. They can also be used in a range of industries and even in smart home applications. These robots can come in various forms, such as manipulators [8] and fully humanoid robots [2].

However, incorporating humans into the process presents many challenges, primarily due to the unpredictable nature of human behaviour. This can lead to difficulties with robots adaptability and robustness in changing and uncertain situations and environments. In HRC systems, robots are expected to understand human activities and intentions and, at times, even predict future human behaviour in order to efficiently achieve the shared goal. This can be a difficult task due to the inherent uncertainty of human behaviour.

Significant research has been dedicated to understanding human behaviour patterns through Human Activity Recognition (HAR), which involves analysing various sensor data to identify and detect simple and complex human activities. HAR has been applied not only to domains related to human daily life, such as healthcare, smart home applications, and elderly assistance [9], but also in robotics solutions where HRC is foreseen, being critical for the robot to have awareness of human actions. Traditional machine learning methods, such as Bayesian networks [10], random forest [11] and support vector machines [12] have been used to understand human behaviours. In addition to understanding human behaviour, some researchers have focused on predicting the most likely sequence of human actions. Probabilistic methods, such as Hidden Markov Models (HMM) [13] have been proposed to understand and predict human activities. Finite State Machines (FSM) have also been used as a tool to model dynamic changes over time and, when combined with fuzzy logic, to even handle uncertainty from sensor data through the use of linguistic variables [14]. Recently, deep learning has emerged as a new trend, as it has the ability to learn and identify complex patterns among large datasets. The major difference between deep learning and the previously described approaches is that it offers multiple hidden layers that are capable of feature extraction and transformation, thus significantly reducing the workload of human designers and developers. As a result, deep learning has been used in various other domains as well, such as image classification [15], speech recognition [16] and so on, and several deep learning algorithms, such as convolutional neural networks (CNNs) [17] and recurrent neural networks (RNNs) [18], have been key to improve the accuracy and robustness of HAR systems.

While these methods have shown promise, dealing with human uncertainty remains a challenge. One of the main difficulties is the high variability of human behaviour across different contexts, as well as the noise in the sensor data, which makes it difficult to generalize from training data. This uncertainty problem has a negative impact on trust and safety, which are critical measurements for any HRC system [19]: if the robot is unable to understand or anticipate human intention, this may lead it to make wrong decisions and even cause accidents and injuries, which will affect both acceptability and trustworthiness. Several authors point out to a panoply of solutions to eliminate uncertainty by extracting more information and rapid processing. However, there is no clear plan established for a constrained computing system, as robots and other facilitators of HRC (e.g., wearable technologies) often have. Given the complexity and addressed challenges associated with uncertainty in human behaviours, further research is still required to fully understand and address this problem.

1.2. Research Question and Objectives

This paper proposes a HAR framework capable of coping with uncertainty in human behaviours, resulting in positive improvements in trust and safety for HRC tasks. To this end, this paper presents three key incremental developments:

- i Enhance Long Short-Term Memory (LSTM) networks by incorporating fuzzy logic to model human uncertainty (Fuzzy-LSTM), building upon the work of [20]: the goal is to improve the performance of LSTM networks by incorporating fuzzy logic to model human uncertainty. In this method, features are extracted from sensor data, which may be uncertain due to the ambiguity of human behaviour or noise in the sensors. These features are then fuzzified using Tilt and Motion linguistic variables. This fuzzification step allows the model to handle uncertain data, making it more robust. The fuzzified features are then used as input to the LSTM network during training. The goal of this approach is to improve the accuracy of the LSTM network in handling uncertain sensor data.
- ii Further extend Fuzzy-LSTM representing the sequence of activities through finite-state machines (FSM), thus leading to the Fuzzy State LSTM (FS-LSTM): the goal of this method is to enhance the predictability of human activity sequences by combining the strengths of FSM and fuzzy logic in an LSTM-based model. In this approach, an LSTM network is trained for each state within the FSM. The output of the LSTM network is then used to determine the possible transitions between states.
- iii Estimate human uncertainty by aggregating predicted scores of the LSTM into a crisp output through defuzzification: this proposed method aims to estimate the uncertainty of the LSTM classifier's predictions by converting the classification scores into a crisp value through defuzzification. The classification scores are first converted into a fuzzy set to represent the degree of uncertainty in the predictions. Then, the fuzzy set is transformed into a crisp value to indicate the certainty of the classifier's predictions. This process allows for quantifying the uncertainty of the predictions, which is not only used within the FS-LSTM method to accept or reject transitions between states, but can also be useful in our future work in HRC, where certainty is important.

In addition to these three main contributions, a benchmark is presented to further investigate the impact of the proposed architecture, which compares the traditional LSTM and the incrementally developed novel architecture.

1.3. Organization of the Article

This article is structured as follows: In Section 2, a comprehensive review of relevant literature is provided. Section 3 outlines the use case and data collection through the developed simulator and a preliminary experimental study on generating synthetic data. The proposed method, including feature fuzzification and FSM learning with LSTM, is described in Section 4. At last, the results from the experimental studies are discussed in Section 5, followed by a description of future work and conclusions in Section 6.

2. Literature Review

HAR has gained significant attention for its ability to detect and identify human activities from sensor data [21]. The importance of HAR lies in its ability to handle the uncertainty that arises from the variability in human behaviour and ambiguity in activities. In robotics, understanding human behaviour and adapting accordingly is crucial for natural and safe interaction and collaboration with humans [22]. This literature review will examine the uncertainty problem in HAR and the methodologies used to address it, as well as the challenges and open research questions in the field.

Human uncertainty poses a significant challenge in HRC from various perspectives. One major aspect is the complex sequential decision-making required in dynamic environments during collaborative tasks, as discussed by Osman in her study [23] on complex dynamic control tasks. These tasks often require multiple decisions that have to accommo-

date many elements of the system to achieve a desired goal, which implies that there is a high degree of uncertainty introduced by humans regarding how they will behave in these changing conditions and environments. This makes it a difficult task for robots to predict and adapt. In addition to the dynamic nature of the environment and the complexity of collaborative tasks, the variability of human physical and cognitive abilities also contributes to their uncertainty. Human factors, such as fatigue, learning ability, and attentiveness can significantly impact a worker's efficiency and accuracy and even cause errors or safety issues in HRC systems [24]. This is aligned with the work of Vuckovic et al. [25], that highlighted the importance of human subjectiveness in creating uncertainty in human behaviours. According to the authors, individuals judge a stimulus and adapt their decisions accordingly to their judgments. This implies that human subjectivity has an important role in introducing uncertainty in human behaviours, as it leads them to perceive and react to situations differently based on their own experiences. Another important aspect in which uncertainty plays a role is in building trust between humans and robots in collaborative tasks. Trust is a crucial element in HRC, as it allows humans to rely on robots to safely perform tasks together. According to Law and Scheutz [26], understanding human needs and intentions, and effectively responding to them, is key to building trust.

Other researchers have emphasized the significant impact of human uncertainty on proactive planning for HRC. According to Kwon et al. [27], proactive planning involves a robot's ability to adapt to a dynamic environment by handling uncertainty. The authors note that the nature of the dynamic environment is not only affected by the robot's actions but also by human activities, which have complex temporal relationships. The uncertainty in these activities must be considered during planning as they are not easily predictable due to the robot's limited observation of the environment and the humans. Therefore, understanding and addressing uncertainty in collaborative tasks is essential for efficient planning in HRC.

Based on the literature reviewed, it is well understood that uncertainty introduced by humans poses a significant challenge in Human-Robot Collaboration (HRC). Therefore, a significant amount of research has been conducted in this field with the aim of mitigating the negative effects of uncertainty. These solutions mostly focus on the efficient and effective inference of human behaviour as a means of addressing uncertainty within the HRC context. One approach is the use of multimodal systems that combine different sensor types, such as video cameras, wearables, and even ambient sensors, such as infrared motion detectors. Video cameras are popular for HRC tasks, but they raise privacy concerns [28]. On the other hand, wearable sensors, such as inertial measurement units (IMUs), are widely used to cope with privacy and security concerns, but they also come with many challenges, such as limited representativeness of similar activities. Despite these challenges, wearable sensors are the most commonly used set of sensors in human activity monitoring. In [29], the authors proposed to use wearable sensors, such as accelerometers and gyroscopes worn at different positions on the human body, to capture activity data that are sampled at regular intervals to be used in HAR. Another study has been designing appropriate methodologies, such as utilizing data from individual accelerometers at the waist, which can identify basic daily activities, such as running, walking and lying down [30]. These works reported acceptable accuracy results for basic daily activities. However, they could not show good accuracy for more complex activities, such as transitions, e.g., standing up or sitting down. As said before, a way of improving such results would generally imply using a larger combination of sensors, although attaching many sensors to the human body is unfeasible and inconvenient for people's daily activities.

HAR is often treated as a pattern recognition problem, and many works have initially adopted machine learning techniques to recognize activities. Support Vector Machine (SVM) [31] and Hidden Markov Model (HMM) [32] classifiers are among the most commonly used methods for activity recognition. For example, Azim et al. [33] used an SVM classifier with trajectory features for activity classification and achieved an overall accuracy of 94.90% for the KTH online database and 95.36% for the Weizmann dataset

(<http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html> (accessed on 2 February 2023)). Kellokumpu et al. [34] used HMM and affine invariant descriptors, achieving an overall accuracy of 83.00%. While these works rely on offline data, Yamato et al. [35] used real-time sequential images and mesh features along with HMM, achieving a 90% accuracy. However, these traditional machine learning methods often rely on carefully designed and heuristic feature extraction methods, such as time-frequency transformation, statistical approaches, and symbolic representation. They lack a universal or systematic approach for effectively distinguishing human activities, and they are prone to overfitting and may perform poorly on unseen data [36].

To overcome these drawbacks, ensemble classifiers have been proposed, which involve training multiple models and combining their predictions to make a final decision. The aim of ensemble classifiers is to improve the performance of the model by combining the strengths of multiple models and mitigating their weaknesses [37]. Random forest is a popular ensemble classifier that is computationally efficient and commonly used in various domains, such as text and image classification. Random forest works by training multiple decision trees and combining their predictions through a voting procedure. This method is effective in addressing overfitting issues and has been shown to enhance accuracy by combining the outcome of each different classifier [38].

Both traditional machine learning and ensemble classifiers methods for feature extraction in HAR heavily rely on human experience and domain knowledge. However, these may not be effective for more general environments and may result in a lower chance of building an efficient recognition system. Additionally, the features learned by these methods are shallow, such as statistical information, and can only be used for low-level activity identification, such as walking or running, making it hard to detect high-level or context-aware activities, such as cooking. In contrast, in real-life scenarios, activity data comes in a stream and requires robust online learning from static data, which is a limitation of many of these traditional methods [39]. Deep learning methods, on the other hand, have been successful in learning complex activities due to their ability to learn features directly from the raw data hierarchically by performing nonlinear transformations. The layer-by-layer structure of deep models allows learning from simple to abstract features. Advances in computer resources have made it possible to use deep models to learn features from complex data from single or multimodal sensory systems. It is worth highlighting that deep neural networks can be detached and flexibly composed into a unified network, allowing for the integration of various deep learning techniques, such as deep transfer learning, deep active learning, and deep attention mechanism. This enables the integration of various effective solutions that can improve the performance of the recognition system [36].

Popular deep learning techniques include deep neural networks (DNN), convolutional neural networks (CNN), recurrent neural networks (RNN), and long short-term memory (LSTM) networks [28]. DNN are a type of Artificial Neural Network (ANN) that are characterized by a larger number of hidden layers. In contrast to traditional ANN, which often have only a few hidden layers, DNN can learn from large datasets more effectively. Hammerla et al. [40] adopted a five-hidden-layer DNN to perform automatic feature learning and classification. Vepakomma et al. [41] fed extracted hand-engineered features obtained from the sensors into a DNN model. CNNs are a type of neural network that exploit three key concepts: sparse interactions, parameter sharing, and equivariant representations. CNN have presented successful results in HAR application by utilizing local dependency, which refers to the nearby signals in a time-series that are most likely correlated. CNN also have shown the ability to handle variations in pace or frequency [39]. Several studies, such as [42,43] have employed one-dimensional (1D) on the individual univariate time-series signals for temporal feature extraction. Conventional 1D CNN have a fixed kernel size, which limits their ability to discover signal fluctuations over different temporal ranges. To address this, Lee et al. [17] combined multiple CNN structures of different kernel sizes to obtain the temporal features from different time scales. Nevertheless, this approach would demand more computational resources as well. Various deep learning methods have been

applied to temporal information including RNN. While traditional RNN cells suffer from vanishing gradient problems, LSTM, as a specific type of RNN, overcomes this issue. A sliding window is generally used to divide the raw data into individual pieces, which are then used to feed LSTM. In a typical LSTM-based temporal feature extraction, it is essential to carefully tune the hyper-parameters, such as the length and moving step of the sliding window. Some researchers adopted The Bidirectional LSTM (Bi-LSTM) structure for extracting temporal dynamics from both forward and backward directions in HAR [44]. On the other hand, Guan and Plötz have combined multiple LSTM networks in an ensemble approach and obtained superior results [45].

Another trend in HAR is combining different deep learning approaches by developing hybrid models to exploit their different aspects. For instance, Ordóñez and Roggen have combined CNN and LSTM for both local and global temporal feature extraction [46]. The idea is to exploit CNN's ability to capture the spatial relationship, while LSTM can extract the temporal relationship. According to the reported results, CNN combined with LSTM outperforms CNN combined with dense layers. Differently, in [47], the authors presented a hybrid model for HAR which first identifies the abstract activity by using random forest to classify it as static and moving. For static activities the authors have used SVM, while for moving activities they have adopted 1D CNN. Even though the overall accuracy of the system was 97.71%, their system was evaluated over a dataset and has not been tested in real environments and/or in runtime.

Despite these models having shown significant accuracy in HAR, the uncertainty of the activities remains a challenge due to several reasons, such as noise in sensors and human factors. Several studies adopted different methodologies to investigate the degree of certainty, or uncertainty, of a given performed activity. One of the methods adopted was a dynamic Bayesian mixture model (DBMM), which is a type of ensemble probabilistic model that combines the likelihood of multiple classifiers into a single form by attaching different weights to each classifier. DBMM uses an uncertainty measure, such as the posterior probability, as a confidence level, which is updated during the online classification [48]. Therefore, the classifier with the highest confidence level is the outcome of the classification process. In [49], the authors presented an architecture that recognises seven different actions performed by athletes using a single-channel electromyography (EMG) combined with positional data obtained by benchmarking ANN, LSTM and DBMM. According to the results, ANN and LSTM models were not the most reliable choice to identify these actions due to the low number of trials in the dataset. On the other hand, DBMM led to better results, with 96.47% accuracy and 80.54% F1-score. Similarly, in [50], human daily activities were recognized by using DBMM. The authors proposed a set of spatio-temporal features, including geometrical, energy-based and domain frequency features to represent the different daily activities which were then fed into DBMM. The overall classification performance for DBMM and LSTM, in terms of precision and recall, was 86.63% and 85.01%, respectively.

Other studies have explored fuzzy-based architectures in HAR, which allows for the incorporation of uncertainty in the decision-making process. While traditional probabilistic models represent the likelihood of an event using crisp values, fuzzy-based models use fuzzy membership values to represent the degree of partial truth by providing semantic expressiveness through the use of linguistic variables to handle uncertain data. Karthigasri and Sornam [20] fuzzified the input features to be used in a fuzzy FSM (FFSM), which is a methodology used to model dynamic sequences of events. The reported results of the approach outperformed decision trees, K-nearest neighbors, SVM, Gaussian naïve Bayes and quadratic discriminant analysis. Mohamed et al. [14] proposed a HAR architecture using data obtained from low-level sensory devices by enhancing FFSM with deep learning methods, namely LSTM and CNN. While both models have shown high scores of accuracy, the CNN-FFSM model showed more robust and reliable performance when applied to a larger dataset, while LSTM-FFSM outperformed CNN-FFSM for simple scenarios with a short period of a dataset. Despite the paper presenting promising results for

HAR, the methodology is presented in a high-level manner, lacking relevant technical and scientific details, which makes it impossible for the reader to understand and fully assess its reproducibility.

In conclusion, the literature reviewed in this study highlights the importance for robots to understand human activities and cope with uncertainty in HRC applications. To this end, a variety of studies have been conducted in this field to understand human behaviour by exploring HAR architectures. However, it is clear that there is still a need for further research in this area in order to not only measure human uncertainty during collaborative tasks with robots in runtime, as well as to use such knowledge to adapt accordingly.

3. Use Case and Data Collection

3.1. Use Case: The FEROX Project

FEROX <https://ferox.fbk.eu/> (accessed on 10 February 2023) is a project that aims to support workers collecting wild berries and mushrooms in wild and remote areas of Nordic countries by using robotic technologies. One of the key aspects of the project is its focus on HRC by deploying unmanned aerial vehicles (UAV) to monitor and assist groups of workers during field operations. This improves workers safety in remote environments, where access to help or assistance may be limited. The expected end results will be an increased worker trust in collaborating with robots, leading to larger number of berries harvested, higher quality berries for consumers, more efficient picking times, new level of worker safety in remote environments, and reduced worker exhaustion levels. Figure 1 depicts a view of the work field of the FEROX Project.

To achieve its aim, the FEROX project is exploring the use of wearable technology to infer the needs and states of the workers. One possible solution is to use a wearable device with integrated IMU (i.e., accelerometer, gyroscope and magnetometer) that can enable the identification of different activities, such as walking, running, sitting, collecting, and loading berries. Additionally, data from a global navigation satellite system (GNSS) (e.g., from the worker smartphone) can be used to infer activities performed over distance, such as driving a vehicle. It is foreseen that the combination of these two commonly adopted cheap devices would allow for more accurate and real-time monitoring of the workers' activities and needs, enabling the project to better support and assist them.



Figure 1. A view on the work field of the FEROX Project.

Figure 2 illustrates the conceptual overview of the architecture aimed to be implemented in the FEROX Project. As stated above, human workers are equipped with wearables and other technologies, which feed the herein proposed FS-LSTM architecture to assess their behaviors and the associated uncertainty for a high-level decision-making system. The system may integrate human physiological and kinematic data to identify human activities, such as (1) human locomotive activities, including idle and walking; (2) human work-related activities, such as berry picking; (3) potential detection of human injury, combining physiological data, such as heart rate, in the future; and (4) a multi-UAV system that provides assistance to the human workers based on the output of the high-level decision-making system, which takes into account the human state defined by FS-LSTM. The next phase of the study will focus on developing the high-level decision-making system to explore the areas, track human location, and assist with loading the collected berries to the collection point (see Section 6).

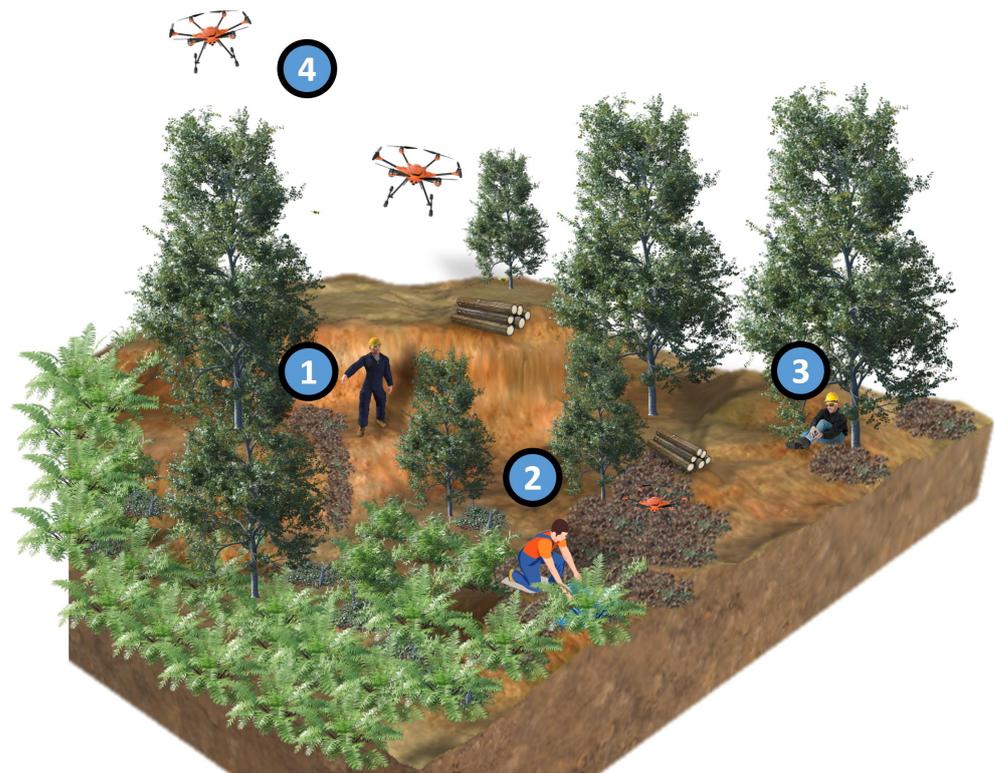


Figure 2. An overview of the use case scenario.

3.2. Data Collection: FEROX Simulator and Synthetic Data

In recent years, the performance of HAR systems has seen significant advancement due to the use of deep learning techniques. However, the acquisition and labelling of large datasets for training and evaluating these methods can be time-consuming and costly. To address these limitations, one solution is to use synthetic datasets that do not require manual labelling or expensive hardware for data capturing [51]. This approach has several advantages, such as producing labelled data without human input, being beneficial in fields where data acquisition is costly, such as field robotics [52].

As a preliminary study, we present a simulator that generates automatically labelled synthetic data by simulating a human character with a chest-worn virtual IMU and smartphone GNSS sensors. The first goal of this study is to develop a simulation environment that can produce synthetic human motion data to feed a HAR system capable of recognizing different locomotive actions. The focus of this research is on developing a system that can be trained using only synthetic labelled data, and then tested and evaluated with real data to justify its reliability for further studies.

3.2.1. FEROX Simulator Development and Virtual Sensor Modelling

We have developed the simulator using the Unity (<https://unity.com/> (accessed on 10 February 2023)) game engine with the ultimate goal of creating a game-like environment for HRC. To achieve this, we initially focused on setting up the forestry scenario by using the Unity Terrain High-Definition Render Pipeline (<https://assetstore.unity.com/packages/3d/environments/unity-terrain-hdrp-demo-scene-213198> (accessed on 10 February 2023)) and the avatar using the Mixamo library (<https://www.mixamo.com/> (accessed on 10 February 2023)), contemplating simple actions, such as idle, walking, running, sitting, falling down and getting up as shown in Figure 3. We also implemented work-related actions, such as collecting and loading berries, driving a vehicle, etc. To generate the animations, we used a keyframe-based method that models connected virtual human body joints in a sequence of frames.

To establish communication between the different agents in the simulation and the developed framework, we integrated the ROS TCP Connector (<https://github.com/Unity-Technologies/ROS-TCP-Connector> (accessed on 10 February 2023)) to set up a TCP connection between Unity and the widely popular Robot Operating System (ROS) framework [53]. This allows us to generate C# classes to serialize and deserialize ROS messages, specifically the synthetic data obtained by the virtual IMU and GNSS sensors. The GNSS coordinate data is published at a rate of 5 Hz to the ROS network as a sensor_msgs/NavSatFix standard message. The synthetic IMU data is published at 50 Hz to the ROS network as a sensor_msgs/Imu standard message that stores the data over. Additionally, we also publish the current activity (label) being performed at 50 Hz to the ROS network under the message type std_msgs/String. All these three types of messages include a timestamp, which ensures that the activity labels can be synchronized with a given GNSS and IMU data stream.

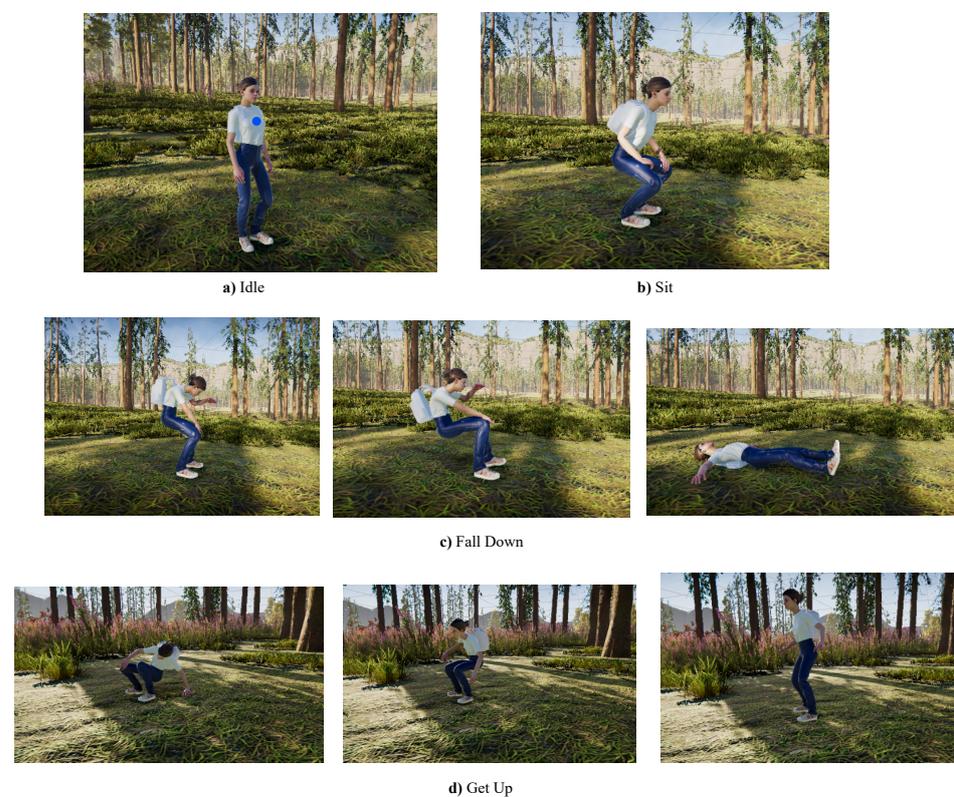


Figure 3. The avatar performs locomotive actions.

In this study, we implemented a virtual model of the RION AH200C IMU sensor (http://en.rion-tech.net/products_detail/productId=158.html, (accessed on 10 February 2023)) which integrates an accelerometer, a gyroscope and a magnetometer, thus combining them and providing readings of linear acceleration, angular velocity and orientation. It is possible to place a virtual sensor in any desired position, as long as it is attached to a human joint. In our case, the virtual IMU sensor was placed on the chest of the avatar, as it is shown with a blue mark in Figure 3. The linear acceleration was calculated by taking into account the discrete derivative of the velocity with respect to the time as shown in Equation (1):

$$a(t_n) = K \left(\frac{v(t_n) - v(t_{n-1})}{T_s} \right) \quad (1)$$

where a , K , v , and T stands for the linear acceleration, gain factor, velocity and time of the cycle, respectively. More particularly, we captured the position of the virtual IMU attached

to the chest joint of the avatar and calculated its second discrete derivative every 20 ms on its own frame. In order to do that, we made use of the parent-child concept of Unity, which relies on a hierarchical structure between transform frames (position and rotation), as the pose of the child changes accordingly to the pose of the parent. The position of the child is applied from the current position of the avatar's chest. The rotation of the parent is adopted from the current rotation of the avatar's chest, while the position of the parent is adopted from the avatar's chest position in the previous frame as illustrated in the Figure 4 in which the transparent human figure represents the virtual IMU position of the previous cycle. This concept allows us to obtain the position of the virtual IMU in its local frame.

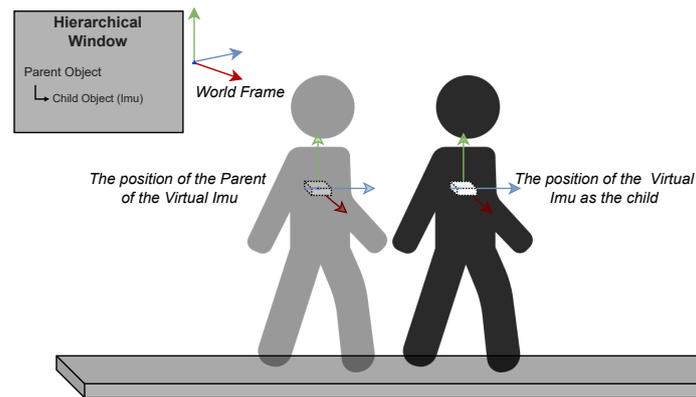


Figure 4. The parent-child relationship is adopted to obtain the position data in virtual IMU's local frame.

It is noteworthy that, due to the successive discrete derivative, the linear acceleration is greatly affected by noise, which is not observable in the data retrieved from the real IMU sensor. Therefore, we have applied a linear interpolation followed by smoothing the data using an exponential smoothing algorithm commonly employed in time-series data to remove high-frequency noises, as in Equation (2), where x_t is the data sequence, s_t is the output of the exponential smoothing algorithm, t is time and α is the smoothing factor:

$$\begin{aligned} s_t &= x_t, & t &= 0 \\ s_t &= \alpha x_t + (1 - \alpha)s_{t-1}, & t > 0 \text{ and } 0 < \alpha \leq 1 \end{aligned} \quad (2)$$

Additionally, and because real-world accelerometers are generally affected by gravitational acceleration, we have calculated the gravity vector in the local frame of the sensor by making use of the Unity physics engine.

In order to have the orientation information, we extracted the virtual sensor's rotation in quaternions. Quaternions provide a convenient mathematical notation for representing the orientation of objects in space, being represented with complex numbers in the following form as shown in Equation (3), where q_x , q_y , q_z are the vector units and q_w is the scalar unit. Then, similarly to acceleration, we applied smoothing algorithm to smooth the quaternion data.

$$q = q_x + q_y + q_z + q_w \quad (3)$$

At last, to obtain the angular velocity, we made use of the orientation described above, converting quaternions to Euler angles, and applying the related discrete derivative at every 20 ms, as represented in Equation (4), where ω , θ , and t represent angular velocity, rotation angle in radians, and time, respectively.

$$\omega = \frac{\Delta\theta}{\Delta t} \quad (4)$$

In line with the approach taken to model the virtual IMU, a similar methodology was adopted to model GNSS data. Specifically, a smartphone positioning system was utilized as a reference for modelling GNSS data. To simulate the GNSS data, the avatar's position in space was leveraged and converted into longitudinal and latitudinal coordinates using the GpsConverter package (<https://github.com/MichaelTaylor3D/UnityGPSConverter> (accessed on 10 February 2023)). It should be noted that the smartphone GNSS data had already undergone filtering, thus negating the need for additional data smoothing techniques [54].

While the data generated closely matches its real counterpart, real sensors are often affected by noise. Preliminary tests using the data generated from the aforementioned approach led to the overfitting of the models. The real IMU and GNSS sensors have noise characteristics due to some calibration errors or environmental noise that affects the sensor readings. Therefore, in order to make the synthetic data more realistic, a Gaussian Noise was injected on both sensors, more specifically affecting the longitude, latitude, linear acceleration and angular velocity variables. Noise was not added to quaternion data as the data provided by the real IMU sensor already comes from Extended Kalman Filter, which leads to a noiseless signal [55]. To add variability to the virtual IMU sensor data, the velocity and sequence of movements in the virtual avatar were adjusted in runtime. Different velocity levels can result in different patterns in the sensor data, while different sequences of activities can affect the overall variability of the data.

3.2.2. Data Preparation

To justify the reliability of the synthetic data, we conducted a preliminary study by using MatLab to deploy the sequence classifier for training and testing, benefiting from both Deep Learning Toolbox (<https://www.mathworks.com/products/deep-learning.html> (accessed on 10 February 2023)) (for sequence data classification) and ROS Toolbox (<https://www.mathworks.com/products/ros.html> (accessed on 10 February 2023)) (for seamless communication with the ROS master).

We have started by building our own dataset, containing the synthetic data obtained by the virtual IMU and GNSS sensors, as well as the real data obtained by the real RION AH200C IMU sensor and GNSS data of a smartphone. At this stage, the dataset included data from only four activities (Sit, Fall Down, Get Up and Idle) with automatic labelling being performed for the synthetic data as described in the previous section, and manual labeling being performed for the real data, which would be required to not only assess the feasibility of the virtual IMU and GNSS models, but also to validate and evaluate the classifier.

We have implemented a method for synchronizing the timestamps of IMU and GNSS sensors. For each timestamp of the label data, the closest GNSS and IMU timestamps are found. The GNSS data is associated with the IMU and label data over a short time of 10 timestamps. Then the GNSS route involving longitude (ϕ) and latitude (λ) has been converted to Cartesian x and y coordinates.

Taking into account the use of a single IMU and GNSS, be it virtual or real, we have considered a feature vector $s(t)$ that includes linear acceleration (a_x, a_y, a_z), angular velocity ($\omega_x, \omega_y, \omega_z$), quaternion (q_x, q_y, q_z, q_w), x and y being represented as follows:

$$s(t) = [a_x \ a_y \ a_z \ \omega_x \ \omega_y \ \omega_z \ q_x \ q_y \ q_z \ q_w \ x \ y] \quad (5)$$

To tackle this classification problem, we adopted a Long Short-Term Memory (LSTM) network, which is known to be state-of-the-art supervised method for sequence data classification. As previously stated, LSTM is an improved type of recursive neural network and, instead of having a single neural network layer, it has four interacting layers, namely, cell state layer, input gate layer, forget gate layer and output gate layer. This enables it with the ability to "remember" information for a certain period, enabling learning-term dependencies [49]. Further detailed information on LSTM structure can be found in Section 4.2.

3.2.3. Synthetic Data Validation

We conducted a preliminary experiment to investigate if the synthetic data is as adequate as the real data. For real-world experiments, we have used a smartphone and the RION AH200C IMU sensor in a chest-worn sensor setup. For both real-world and virtual experiments, we have recorded 192 activities as 48 samples of each action, for up to 3 s, at 5 Hz and 50 Hz, respectively for GNSS and IMU data. We also created the categorical array that holds the labels corresponding to these actions. The LSTM network was trained with the synthetic data and subsequently tested with the real data. The adaptive moment estimation optimizer was adopted, with a maximum epoch of 200.

The results of these experiments are presented in Figure 5 with the confusion matrix depicting the accuracy of the experiment including the performance of each activity. A result of 84.9% indicates that although the model performs with acceptable accuracy, several Sit and Get Up actions were incorrectly classified as Idle. While the initial findings indicate that the model trained using synthetic data can accurately classify the four specified activities when presented with real-world data, the upcoming sections will delve deeper into the evaluation of HAR using synthetic data across a wider range of activities. Within the context of the FEROX project, and to propose a more encompassing architecture, more complex activities will be included, such as forestry-work-related ones. Therefore, due to the simulator feasibility for generating data, the next sections encompass data collection from 13 different activities and, likewise, a novel approach for HAR under uncertainty.

<i>Fell</i>	25.0%	0.0%	0.0%	0.0%	100.0%
<i>Get Up</i>	0.0%	19.3%	0.0%	0.0%	100.0%
<i>Idle</i>	0.0%	5.7%	25.0%	9.4%	62.3%
<i>Sit</i>	0.0%	0.0%	0.0%	15.6%	100.0%
	100.0%	77.1%	100.0%	62.5%	84.9%
	<i>Fell</i>	<i>Get Up</i>	<i>Idle</i>	<i>Sit</i>	

Figure 5. The confusion matrix of the LSTM network.

4. Fuzzy State Long-Short Term Memory (FS-LSTM)

The proposed FS-LSTM framework for HAR under uncertainty is presented in Figure 6. The framework is comprised of five blocks, labeled A, B, C, D, and E. Block A is responsible for collecting human-related data through multimodal sensors, hereby assessed using a chest-worn IMU and a GNSS smartphone positioning system. Block B processes the IMU and GNSS data, including linear acceleration, angular velocity, orientation from the IMU sensor, as well as longitude and latitude from the GNSS, which are published at 50 Hz and 5 Hz, respectively and as previously described in Section 3. This data is then transformed into linguistic labels for Motion and Tilt through a fuzzification process in Block C, as further described in Section 4.1. These fuzzified Motion and Tilt features serve as inputs for both Block D and E. In Block D, the fuzzified feature set is used as input in LSTM

state machine learning, where multiple networks are trained for each state to be executed during runtime, including a recovery state called Lost. This process is further detailed in Section 4.2. In Block E, uncertainty is managed through defuzzification in a closed-loop. The fuzzified inputs are used in the classification network, which was established in the previous iteration, and the generated classification score is first fuzzified and then defuzzified into a crisp value to determine whether to progress to the next state or remain in the current one. The details of this process will be further explained in Section 4.3.

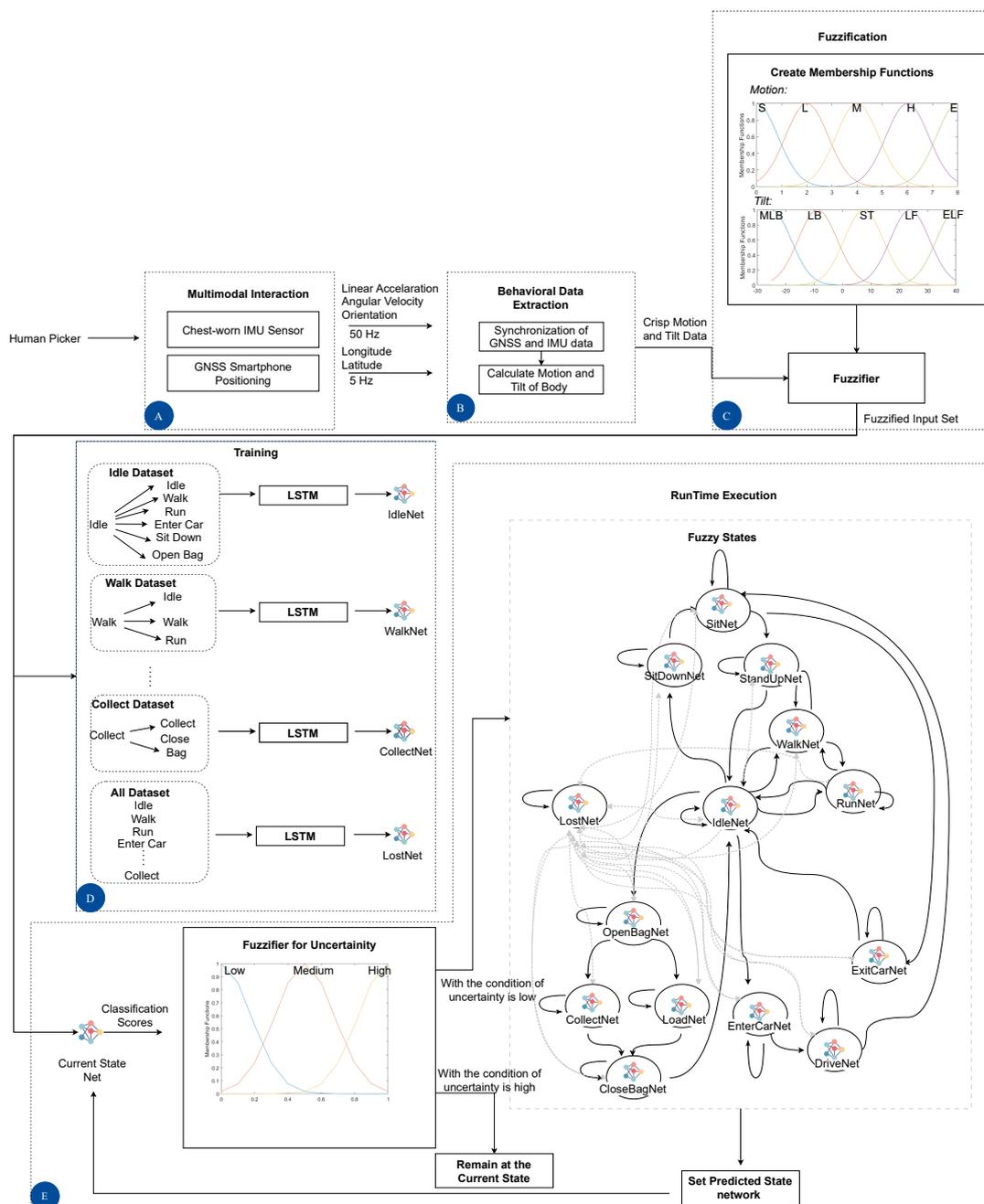


Figure 6. The diagram of the proposed architecture.

4.1. Fuzzification of Features

This section presents the design of the feature fuzzification process for converting IMU and GNSS data into linguistic variables that will serve as inputs in the proposed FS-LSTM model. This stage is marked as C block in Figure 6.

Fuzzy logic allows computer systems to mimic the human-like thinking and make decisions under uncertain and imprecise information. For instance, the subjective and ambitious statement “the food is good” is enough for a person to decide how much to tip. This way of handling uncertainty is important when the reliable exact information is not available.

The numeric data collected by the FEROX simulator, as described in Section 3, includes linear acceleration (a_x, a_y, a_z) , angular velocity $(\omega_x, \omega_y, \omega_z)$, and quaternion (q_x, q_y, q_z, q_w) , as well as the x and y coordinates. The process of fuzzification involves converting these crisp numerical data from sensors (either synthetic or real) into linguistic variables for use in the proposed FS-LSTM model. In particular, we have selected Motion and Tilt variables for fuzzification. We obtained velocity information as crisp data inputs for Motion and y-axis Euler angle (or pitch) for Tilt. To calculate the velocity, we benefit from the synchronized GNSS and IMU data (previously addressed in Section 3) to first generate a smooth 2D Cartesian position, x and y . The initial velocity is calculated as the derivative of the position:

$$\begin{aligned} V_{x_0}(t) &= \frac{x(t) - x(t-1)}{\delta t}, \\ V_{y_0}(t) &= \frac{y(t) - y(t-1)}{\delta t} \end{aligned} \quad (6)$$

At each time step, we then calculate the velocity along the x and y axis as:

$$\begin{aligned} V_x(t) &= V_{x_0}(t) + a_x(t) * t, \\ V_y(t) &= V_{y_0}(t) + a_y(t) * t \end{aligned} \quad (7)$$

where t is the elapsed time and a_x and a_y are the linear acceleration measurements along x and y axis, respectively, obtained after applying a rotation to the linear acceleration measurements provided by the IMU, so as to align the body frame with the world frame. Finally, V is calculated as the magnitude of the velocity vector:

$$V(t) = \sqrt{V_x(t)^2 + V_y(t)^2} \quad (8)$$

To quantify Tilt, the y-axis Euler angle θ (pitch), which represents the tilt of the chest forward or backward was calculated. The extracted quaternion values were converted to Euler angles in degrees following the principle presented in Equation (9):

$$\begin{aligned} \phi &= \text{atan2}(2(q_w q_x + q_y q_z), 1 - 2(q_x^2 + q_y^2)) \\ \theta &= \text{arcsin}(2(q_w q_y - q_z q_x)) \\ \psi &= \text{atan2}(2(q_w q_z + q_x q_y), 1 - 2(q_y^2 + q_z^2)) \end{aligned} \quad (9)$$

The calculated V and θ are inputs utilized in the fuzzification process. In particular, V is mapped into five linguistic labels for Motion, while pitch (θ) is translated into five linguistic labels for Tilt. The numerical data for both Motion and Tilt were mapped into five linguistic variables by Gaussian membership function using MatLab’s Fuzzy Logic Toolbox as follows:

$$S_U(t) = \begin{cases} \text{Motion} \rightarrow \{S, L, M, H, E\} \\ \text{Tilt} \rightarrow \{MLB, LB, ST, LF, ELF\} \end{cases} \quad (10)$$

where S, L, M, H and E represent the linguistic variables Stopped, Low, Medium, High, and Extreme for Motion, respectively, while MLB, LB, ST, LF and ELF represent Medium Lean Back, Lean Back, Straight, Lean Front, and Extremely Lean Front, for Tilt, respectively. $S_U(t)$ represents the feature vector which then is used as input for the FS-LSTM method.

In Figure 7, the five linguistic variables for Motion and Tilt are displayed, alongside an activity sequence in which the human is first idle, then walks, then becomes idle again, and finally opens the bag and collects some berries.

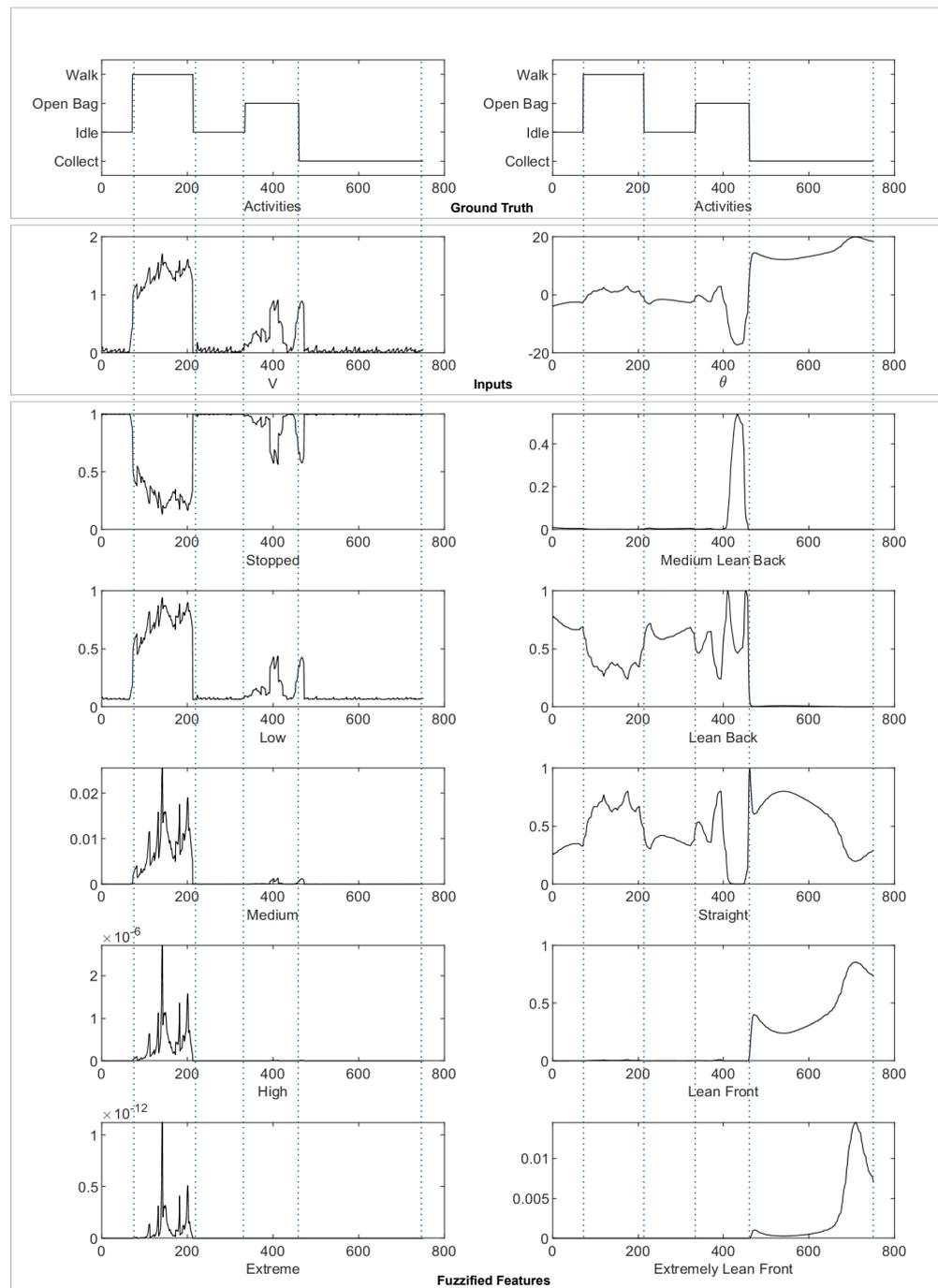


Figure 7. Motion and Tilt plotting along an activity sequence.

The activity sequence in this short period is represented as: Idle, Walk, Idle, Open Bag, and Collect, which are shown in the figure in the ground truth section. Additionally, the figure shows the precise numerical values of the Velocity (V) and pitch (θ) as the inputs for Motion and Tilt, respectively. It can be seen from the figure that the chosen linguistic variables effectively depict the activities. For instance, one can easily recognize when the human is walking by noticing that the Motion value falls between Low and Medium. The Open Bag and Collect activities also showcase distinct characteristics, such as leaning back while opening the bag and leaning forward while collecting berries. Thus, these activities and their fuzzified representations are clearly illustrated and may be used as features to the HAR architecture.

4.2. State Machine Learning

This section presents the modelling of human activity by exploring the transitions between different activity states. These states may involve specific activity sequences or individual activities. As an example, the use case discussed in Section 3 is depicted through a state diagram in Block E of Figure 6. This particular use case models the sequential activities of a human picker, including locomotive movements (such as Idle, Walk, Run), berry collection and loading, and vehicle driving. The flow is represented by 14 states (with the recovery state Lost) and includes states that occur in a sequential manner, such as sitting and driving a car, and transitions that occur between these states, such as Sit Down to Sit, Sit to Stand Up or Enter Car to Drive. There are also states that lead to multiple possibilities, such as the Idle state, which can transition to activities such as Open Bag, Walk, Run, Enter Car, or Sit Down.

The state transitions in a given process are modelled individually through the use of LSTM networks. As previously mentioned, LSTM is capable of extracting hidden patterns from long-term sequential data by handling gradient exploding or vanishing gradients problems [56]. In more detail, the internal structure of an LSTM network consists of multiple gates, including input gate i_t , forget gate f_t , and output gate o_t , that control the flow of information towards the final output. The input gate updates the information, while the forget gate processes information from both the input gate i_t and the previous state C_{t-1} , selectively removing information from the current state C_t when necessary. The output gate forwards the final output to the next LSTM unit and retains the output value for subsequent sequence predictions. The recurrent unit, on the other hand, estimates the state of the previous cell C_{t-1} and the current input x_t using a \tanh activation function. The value of h_t can then be calculated as the scalar product of the output gate o_t and the \tanh of the C_t . The ultimate output is obtained by passing h_t to a softmax classifier [57].

Each LSTM network within the proposed HAR architecture is designed to receive sequential data as input (fuzzy features addressed in the previous section) and generate output predictions for only the feasible transition states. The objective of this approach is to ensure that there are no impossible transitions between states. For example, in a use case where the activities include Sit Down and Stand Up, or Collect and Load, it is possible that similar characteristics may result in an incorrect transition from one activity to the other. However, these transitions should not occur according to the expert-designed state flow, hereby represented as an FSM. The aim of this modelling approach is to guarantee that the predicted next state will be one of the possible states, providing important information for decision-making in an HRC system. Moreover, the number of possible classes significantly affects the size of the model, and a larger number of possible classes leads to a more complex model structure. Such a complex model demands additional computational resources, which would result in a longer runtime execution [58]. For instance, a Sit Down LSTM network with a reduced number of possible classes, such as Sit Down (remains in the same state) and Sit (moves to only the next possible state), is more efficient when compared to an LSTM network trained with all 13 possible activities, particularly when many of the outputs are unlikely to be the actual state.

The implementation of the LSTM layer in MatLab was carried out using the Deep Learning Toolbox. For the purpose of predicting the subsequent activity sequence, a sequence-to-sequence classification approach was employed. Each LSTM network corresponding to a specific state was trained using the same sequential input data. The input vector $S_U(t)$ comprises ten features, including the fuzzified values for Motion and Tilt, as described in Section 4.1 and depicted in Equation (11):

$$S_U(t) = [S \ L \ M \ H \ E \ MLB \ LB \ ST \ LF \ ELF] \quad (11)$$

where,

$$\begin{aligned}
 S &= \text{Stopped} & MLB &= \text{Medium Lean Back} \\
 L &= \text{Low} & LB &= \text{Lean Back} \\
 M &= \text{Medium} & ST &= \text{Straight} \\
 H &= \text{High} & LF &= \text{Lean Front} \\
 E &= \text{Extreme} & ELF &= \text{Extreme Lean Front}
 \end{aligned}
 \tag{12}$$

It is noteworthy, however, that each state-based LSTM network output sets differ, with each output set corresponding to a specific state, including the potential transitions, and including the possibility of remaining in the same state, as it is shown in Figure 8. Any non-feasible state names are labelled as Lost. The Lost network, which uses the same sequential input data and an output dataset of all states, serves as a recovery mechanism, being only activated when such class is output by the previous LSTM network.

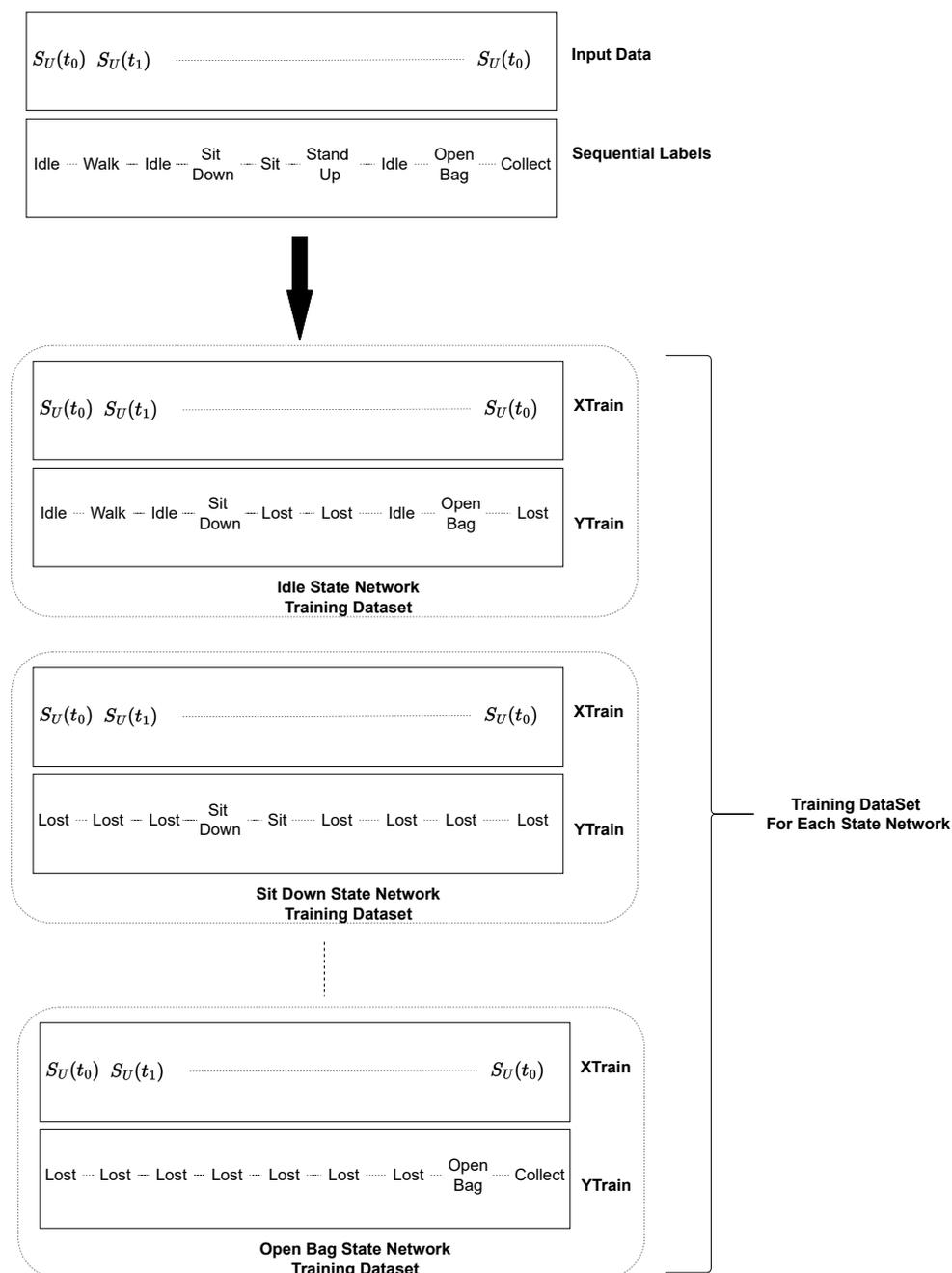


Figure 8. LSTM training datasets.

The data generated in this study have been made publicly available (https://gitlab.ingeniarius.pt/ingeniarius_public/ferox/hrc-ferox.git (accessed on 22 February 2023)). The dataset includes the following features, as shown in Equation (13): t_s is the timestamp, $s(t)$ is the feature vector containing the raw sensor data shown in Equation (5), $s_U(t)$ is the feature vector containing the fuzzified sensor data shown in Equation (11), and G is the ground truth of activities presented as a string array where each element represents the activity label at a specific timestamp.

$$[t_s \quad s(t) \quad s_U(t) \quad G] \quad (13)$$

$$G = ["Idle", "Idle", "Sit", \dots, "Close Bag"] \quad (14)$$

4.3. Coping with the Uncertainty through Defuzzification

The trained network models described in the preceding section are employed in a closed-loop architecture to predict the subsequent state, as depicted in Block E of Figure 6. The fuzzified outputs generated by Block C serve as inputs to the trained network model associated with the current state, which is referred to as the Lost network in the first iteration and is updated in subsequent iterations in accordance with the flow of the architecture.

The proposed architecture makes use of network model classification scores as posterior probabilities based on the fuzzified input set. These probabilities are calculated based on Bayes' Theorem:

$$\hat{P}(B | A) = \frac{P(A | B)P(B)}{\sum_{j=1}^R P(A | j)P(j)} \quad (15)$$

where $\hat{P}(B | A)$ is the posterior probability that an observation A of given class B , $P(A | B)$ is the conditional probability of A given class B , $P(B)$ is the prior probability for class B and R is the number of classes in the response variable [59]. The classification scores are represented as an z -by- R matrix, where z is the number of observations in the data and R is the number of unique classes. The matrix indicates the probability of each observation belonging to a specific class, with the predicted class being determined by the class with the highest score.

FS-LSTM falls on the assumption that state network models may struggle to make confident predictions if the highest score is low or if the scores are similar across classes, leading to a certain level of uncertainty. Yet, it uses such level of uncertainty to still generate a prediction based on the class with the highest score. It is therefore important to consider the level of confidence in the prediction and interpret the scores before making any decisions based on the model's output. The uncertainty in the scores is evaluated using the fuzzy logic system.

Hence, similar as carried out for Motion and Tilt fuzzy variables, classification scores are used for the fuzzification to produce fuzzy linguistic labels for Uncertainty as Low, Medium, and High. The triangular fuzzifier is used to determine the degree of membership for each value. Unlike Motion and Tilt fuzzy variables, however, inference is followed by adopting a set of rules designed for each state, which are then later utilized to assess the uncertainty in the classification scores generated by the current state network. An example of fuzzy rules is presented for the Close Bag state. The uncertainty is assessed in the classification scores generated for the possible transition outputs of the Idle, Close Bag, and Lost recovery state by implementing these rules:

1. IF Idle is Low and Close Bag is Low and Lost is Low THEN Uncertainty is High
2. IF Idle is Medium and Close Bag is Medium and Lost is Medium THEN Uncertainty is High
3. IF Idle is High and Close Bag is High and Lost is High THEN Uncertainty is High
4. IF Idle is High and Close Bag is Low and Lost is Low THEN Uncertainty is Low
5. IF Idle is Low and Close Bag is High and Lost is Low THEN Uncertainty is Low

6. IF Idle is Low and Close Bag is Low and Lost is High THEN Uncertainty is Low
7. IF Idle is High and Close Bag is not Low and Lost is not Low THEN Uncertainty is High
8. IF Idle is not Low and Close Bag is High and Lost is not Low THEN Uncertainty is High
9. IF Idle is not Low and Close Bag is not Low and Lost is High THEN Uncertainty is High

In the fuzzy inference, the rules are applied to the fuzzified inputs to calculate the degree of fulfilment for each rule through aggregation. The following step is defuzzification, which transforms the fuzzy outputs into crisp outputs by using the fuzzy sets and their corresponding membership degrees. The result of the aggregation is converted into a crisp output value through the centroid method. The output of this system expresses the uncertainty as a crisp value between 0 and 1. This crisp value coming from the fuzzy logic system is then used to assess the confidence level of each network model before determining the next state. This is performed by comparing it to an experimentally-defined threshold. If the crisp value of uncertainty generated through defuzzification is lower than the threshold, this implies the model is confident in the prediction and the classification is carried out based on the highest score, which is identified as the next state. This state could be a different state or remain unchanged. The network model that corresponds to the predicted state is then selected for use in the next iteration. If the level of uncertainty exceeds the established threshold, the system remains in the same state. In this case, the next iteration performs the classification utilizing the current network model. This iterative process continues in accordance with the closed-loop architecture until the model reaches a sufficient level of confidence in its prediction.

Figure 9 shows a small section of the timeline of activity recognition handling the uncertainty. Between 62 and 65 samples, the model experiences low confidence in its predictions as the uncertainty is above the threshold and remains in the same state for subsequent selections. At sample 184, the model erroneously classifies the data as Walk while the ground truth remains Run. While it may seem like a premature classification, it could be a coincidence since transitions from Run to Walk are possible. However, at sample 185, it is observed that the Walk state network model lacks certainty in the classification, causing it to remain in the Walk state. The uncertainty level drops below the threshold once the ground truth and the predicted classes align at sample 186.

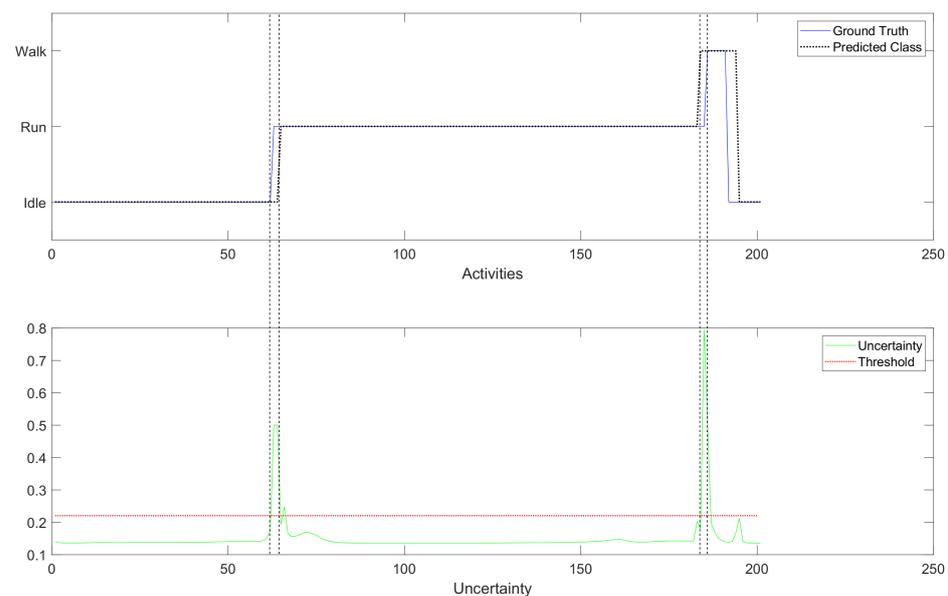


Figure 9. Accessing uncertainty in the activity recognition.

5. Results and Discussion

5.1. Benchmark of Activity Recognition Performance

In this section, we present the results of our experimental study designed to evaluate the effectiveness of the proposed approach. We used a training dataset consisting of 13 different activities (Idle, Walk, Run, Sit Down, Sit, Stand Up, Exit Car, Drive, Enter Car, Open Bag, Collect, Load, Close Bag) as depicted in Figure 6 block E. The training dataset was collected over a period of 17.85 min and includes 539 activities, with a total of 53,553 samples of virtual IMU data and 5355 samples of virtual GNSS data. The testing dataset consists of 168 activities and comprises 21,188 samples of virtual IMU data and 2118 samples of virtual GNSS data, spanning a total duration of 7.06 min. In this section, the number of hidden layers was set to 64 for all LSTM networks considered.

We have benchmarked three different methodologies:

(a) Traditional LSTM: a LSTM model that was trained using raw sensor input, similar to what has been presented in Section 3, though with all 13 states instead of four.

(b) Fuzzy LSTM: a LSTM model that was trained with the fuzzified features described in Section 4.1 and outputs all 13 states.

(c) FS-LSTM: the multiple LSTM models that were trained for each state, using fuzzified features, and each model only outputs the feasible states, as per presented in Figure 6 and described in the previous Section 4.

For the Traditional LSTM model, the input feature vector $s(t)$ is the one previously presented in Equation (5), consisting of linear acceleration values (a_x, a_y, a_z) , angular velocity values $(\omega_x, \omega_y, \omega_z)$, quaternion values (q_x, q_y, q_z, q_w) , and the x and y Cartesian coordinates. The input vector for both the Fuzzy LSTM and FS-LSTM models, $S_U(t)$, includes the five linguistic labels previously described in Section 4.1 for each of the extracted Motion and Tilt variables (Equation (11)).

The results of the three methodologies are presented in Figure 10 as confusion matrices, which depict the outcomes of (i) Traditional LSTM; (ii) Fuzzy-LSTM; and (iii) FS-LSTM. The rows represent the target classes, while the columns represent the output classes. Superior classification accuracy results are identified in bold. The results indicate that utilizing solely raw sensor data leads to a significantly low accuracy of 23.2%. Conversely, by utilizing fuzzified inputs, the Fuzzy-LSTM approach markedly enhances the accuracy to 93.2%. This demonstrates how a fuzzy logic system handles data ambiguity and achieves correct classification, where the Traditional LSTM method frequently falls short. The proposed FS-LSTM methodology achieved an accuracy of 90.9%. This result was obtained by treating samples classified as Lost as unchanged. This approach ensures that the system waits until it recovers from the Lost state, which runs the same model as the Fuzzy-LSTM approach (with 13 state outputs), before transmitting the predicted output to a higher-level decision-making system to ensure the correctness of the transmitted output.

Although the overall accuracy does not show a significant difference from the Fuzzy-LSTM, being even slightly inferior in terms of accuracy, the FS-LSTM prevents transitions that should not occur from happening. This might result in slightly superior performance on Sit Down-Sit-Stand Up transitional states when compared to Fuzzy-LSTM. This not only avoids passing incorrect information to the higher-level management system but also improves the probability of predicting the next state. Figure 11 illustrates this prevention of wrong transitions more clearly for one of the many sequences generated by the aforementioned models. In the figure, a sequence of activities is given with their ground truth in the blue dotted line and predicted outputs with the straight black line obtained through Traditional LSTM (top), Fuzzy-LSTM (middle), and FS-LSTM (bottom). Once again, it is shown that Traditional LSTM is unable to classify activities correctly, alternating between Walk and Idle in this sequence. For the Fuzzy-LSTM, around sample 150, the predicted class is Enter Car, while the ground truth class is Idle. In contrast, in the same sample in FS-LSTM, the classification of the current network model (Idle state network) is either correct, or the uncertainty is high, and the system remains unchanged in Idle. This prevents the system from making infeasible transitions to Enter Car, which would break the sequence

of predicted activities. Fuzzy-LSTM, however, makes infeasible transitions multiple times, such as in sample 665, where it transitions from Idle to Close Bag, which should never happen. FS-LSTM handles such infeasible transitions through uncertainty defuzzification and the state machine approach generally well. As it is described earlier, FS-LSTM handles misclassifications via recovery of the Lost state. In the confusion matrix Figure 10, the Lost classified outputs are not shown as they were treated as unchanged states. However, in practice, when the network is unable to classify a sample in any of the possible transitions, the decision-making moves to the Lost state. For instance, in the sample around 956 marked with a red circle, the system is in the Lost state. This occurred because the Idle network predicted Enter Car instead of Sit Down, which was feasible but wrong. The Enter Car state network model was unable to classify the samples in any feasible transitions, and the system went to the Lost state before recovering to the Sit Down state.

In comparison, while Fuzzy-LSTM mostly performs poorly in the presented time window, it still achieves overall accurate classification, as demonstrated by the corresponding confusion matrix. However, it is important to note that Fuzzy-LSTM should be regarded as a sequence of activity flow prediction rather than individual sample prediction, underscoring the superiority of FS-LSTM. Furthermore, FS-LSTM is expected to achieve the same level of accuracy, or even higher, under constrained resources, since the multiple LSTM networks it encompasses are expected to not require the same number of hidden layers given the reduced number of outputs foreseen by each. This is further explored in the next section.

Close Bag	i) 0.2%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 100%				
	ii) 2.9%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.2%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 92.2%
	iii) 2.9%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 94.1%
Collect	i) 0.0%	i) 0.2%	i) 0.1%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 61.5%
	ii) 0.1%	ii) 9.4%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.2%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 95.9%
	iii) 0.1%	iii) 9.4%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 98.7%
Drive	i) 0.0%	i) 0.1%	i) 0.7%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.1%	i) 74.6%
	ii) 0.0%	ii) 0.0%	ii) 7.2%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.5%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.3%	ii) 88.4%	
	iii) 0.0%	iii) 0.0%	iii) 7.1%	iii) 0.0%	iii) 0.2%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.5%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.3%	iii) 86.9%	
Enter Car	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.4%	i) 0.1%	i) 0.4%	i) 0.1%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.2%	i) 0.1%	i) 0.2%	i) 0.2%	i) 27.6%	
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 4.8%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.2%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.1%	ii) 91.8%	
	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 4.2%	iii) 0.0%	iii) 0.2%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 92.0%	
Exit Car	i) 0.3%	i) 0.0%	i) 0.4%	i) 0.0%	i) 0.0%	i) 1.8%	i) 0.6%	i) 0.1%	i) 0.0%	i) 0.0%	i) 0.1%	i) 0.8%	i) 1.2%	i) 1.2%	i) 0.0%	
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 6.5%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 98.6%	
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 6.4%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 0.1%	iii) 97.1%	
Idle	i) 1.7%	i) 5.0%	i) 0.7%	i) 4.1%	i) 3.2%	i) 18.1%	i) 4.3%	i) 1.8%	i) 2.9%	i) 1.7%	i) 1.5%	i) 0.0%	i) 5.3%	i) 5.3%	i) 36.1%	
	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.5%	ii) 0.0%	ii) 23.9%	ii) 0.4%	ii) 0.1%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.6%	ii) 0.6%	ii) 93.5%	
	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.7%	iii) 0.0%	iii) 24.0%	iii) 0.9%	iii) 0.1%	iii) 0.0%	iii) 0.2%	iii) 0.0%	iii) 0.0%	iii) 0.7%	iii) 0.7%	iii) 89.8%	
Open Bag	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 1.4%	
	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.7%	ii) 8.5%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.1%	ii) 0.1%	ii) 89.5%	
	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 0.2%	iii) 0.0%	iii) 0.0%	iii) 7.9%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 94.6%	
Run	i) 0.6%	i) 0.9%	i) 2.4%	i) 0.1%	i) 1.2%	i) 2.4%	i) 2.1%	i) 1.4%	i) 0.4%	i) 0.1%	i) 0.5%	i) 2.5%	i) 4.3%	i) 7.5%		
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 4.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.4%	ii) 90.8%		
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 2.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.2%	iii) 86.7%		
Sit	i) 0.2%	i) 2.3%	i) 0.1%	i) 0.2%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.1%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.1%	i) 0.0%		
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 3.4%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 97.6%		
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 4.0%	iii) 0.0%	iii) 0.2%	iii) 0.0%	iii) 0.0%	iii) 93.5%		
Sit Down	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.2%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.2%	i) 0.1%	i) 0.0%	i) 0.1%	i) 0.0%	i) 0.2%	i) 0.0%		
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 95.1%		
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 1.8%	iii) 0.0%	iii) 0.0%	iii) 96.7%		
Stand Up	i) 0.1%	i) 0.4%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.1%	i) 0.0%		
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.6%	ii) 0.0%	ii) 2.9%	ii) 0.0%	ii) 0.0%	ii) 80.2%		
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.2%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 3.0%	iii) 0.0%	iii) 0.0%	iii) 95.1%		
Load	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.5%	i) 0.0%		
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 6.3%	ii) 0.0%	ii) 99.0%		
	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 6.2%	iii) 0.0%	iii) 98.7%						
Walk	i) 0.2%	i) 0.5%	i) 2.8%	i) 0.3%	i) 2.2%	i) 2.4%	i) 1.8%	i) 0.9%	i) 1.2%	i) 0.3%	i) 0.8%	i) 2.0%	i) 1.7%	i) 9.8%		
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.3%	ii) 0.0%	ii) 0.3%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 11.6%	ii) 94.7%		
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.2%	iii) 0.0%	iii) 0.4%	iii) 0.0%	iii) 2.5%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 11.9%	iii) 79.3%		
Lost	i) 5.3%	i) 2.4%	i) 10.2%	i) 7.4%	i) 0.0%	i) 72.0%	i) 0.1%	i) 30.7%	i) 0.0%	i) 0.0%	i) 0.0%	i) 7.2%	i) 12.7%	i) 23.2%		
	ii) 92.6%	ii) 98.9%	ii) 99.4%	ii) 89.3%	ii) 99.1%	ii) 94.7%	ii) 95.2%	ii) 89.6%	ii) 74.7%	ii) 82.5%	ii) 87.7%	ii) 99.5%	ii) 88.1%	ii) 93.2%		
	iii) 92.6%	iii) 98.7%	iii) 98.4%	iii) 79.5%	iii) 97.3%	iii) 95.2%	iii) 88.6%	iii) 42.5%	iii) 87.7%	iii) 84.7%	iii) 92.6%	iii) 99.0%	iii) 90.2%	iii) 90.9%		
	Close Bag	Collect	Drive	Enter Car	Exit Car	Idle	Open Bag	Run	Sit	Sit Down	Stand Up	Load	Walk			

Figure 10. The confusion matrix of Traditional LSTM, Fuzzy-LSTM and FS-LSTM. Superior classification accuracy results are identified in bold.

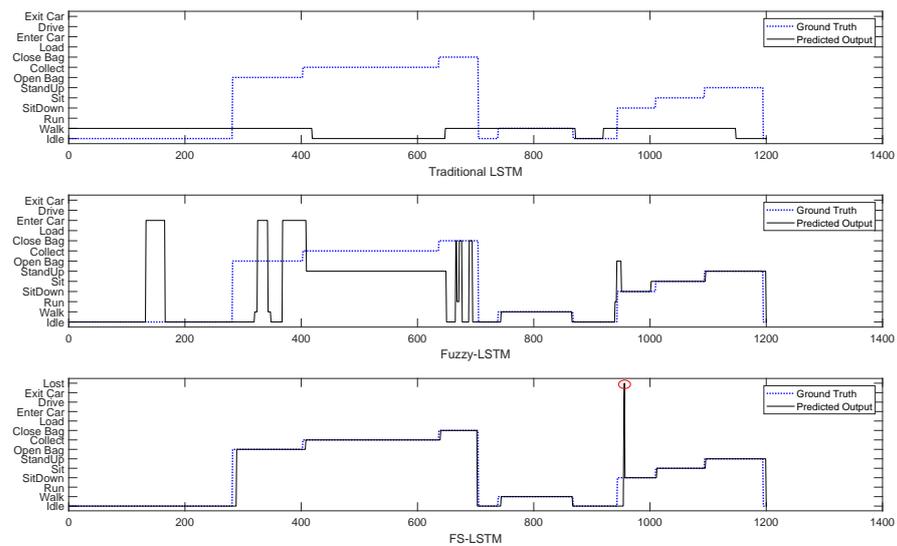


Figure 11. The benchmark of three predicted output sequences via Traditional LSTM, Fuzzy-LSTM and FS-LSTM. The Lost state is marked with the red circle.

5.2. Benchmark of Efficiency

In this section, we evaluate the classification performance of the proposed FS-LSTM compared to Fuzzy-LSTM from the perspective of GPU resource efficiency. In the previous section, both models were trained with 64 hidden layers, and while Fuzzy-LSTM classified 13 states, FS-LSTM only used this complex network in the Lost state. To better understand the impact of this difference on computer resources, we monitored GPU utilization and power consumption during the classification runtime process for a period of 25 minutes. Our experiments were conducted using an NVIDIA GeForce GTX 1050. The benchmark chart for both models is presented in Figures 12 and 13.

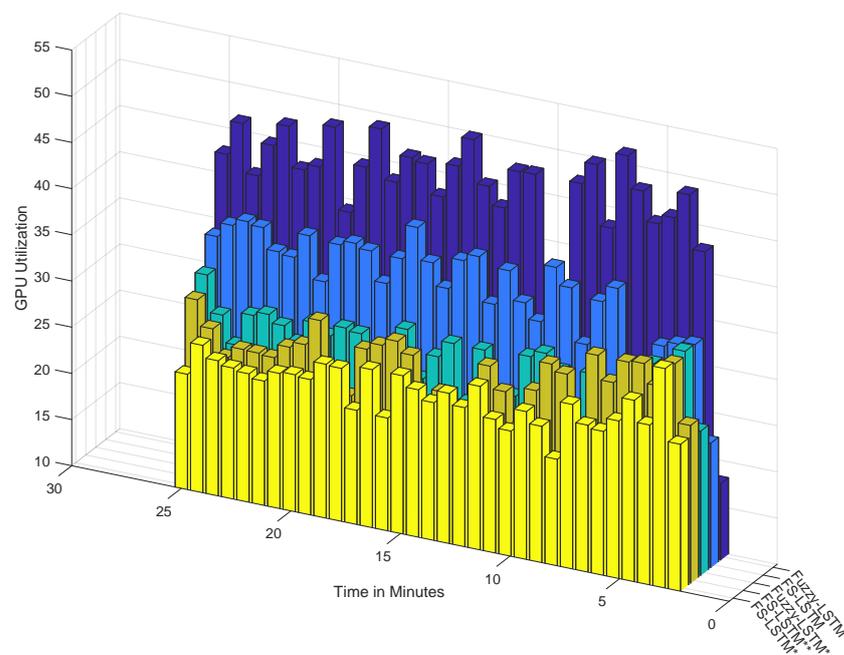


Figure 12. The benchmark of GPU utilization during testing. Fuzzy-LSTM and FS-LSTM networks trained with 32 layers (denoted with *), and a hybrid version of FS-LSTM (denoted with **).

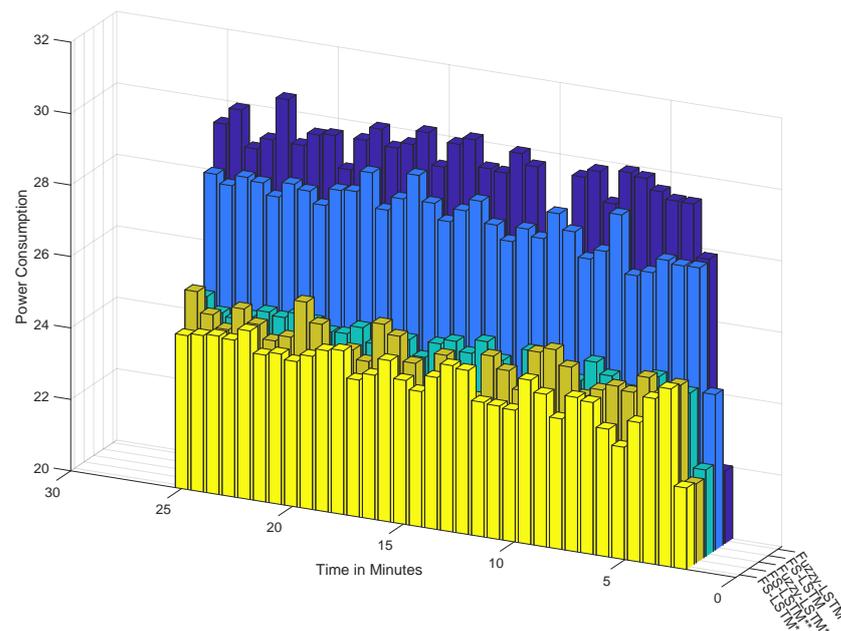


Figure 13. The benchmark of power consumption during testing. Fuzzy-LSTM and FS-LSTM networks trained with 32 layers (denoted with *), and a hybrid version of FS-LSTM (denoted with **).

As shown in the last two rows of columns from Figure 12, despite both methods performing with high accuracy, Fuzzy-LSTM (represented by the dark blue bars) relies more heavily on GPU resources than the FS-LSTM approach (represented by the light blue bars). This is a critical consideration in any long-term outdoor application where computer resources are constrained. While in the last two rows of columns from Figure 13 the power consumption did not show any significant difference between these methods over a short-term test, high GPU load is a critical consideration in any long-term outdoor application where computer resources are constrained. A high GPU load indicates that the GPU is being heavily utilized to complete the classification task, which can cause the GPU to generate more heat and consume more power, ultimately affecting the overall performance and energy efficiency of the system over an extended period.

A question may arise as to whether reducing the number of hidden layers in both models can reduce its complexity, then leading to better efficiency, while still maintaining the desired accuracy. To answer this, we extended the benchmark study to five different combinations: both Fuzzy-LSTM and FS-LSTM networks trained with 32 layers (denoted with *), and a hybrid version of FS-LSTM (denoted with **) trained with 32 layers, except for the Lost network, which was trained with 64 layers. As shown in Figures 12 and 13, reducing the number of layers significantly decreases GPU utilization and power consumption. Table 1 further compares the mean and standard deviations of GPU utilization and power consumption, demonstrating that a smaller model size with fewer output and hidden layers leads to significantly higher efficiency. However, reducing the number of hidden layers sacrifices accuracy, as shown in the confusion matrix in Figure 14.

Close Bag	i) 1.3%	j) 0.0%	j) 0.1%	i) 0.0%	i) 0.0%	j) 0.1%	i) 0.0%	i) 0.0%	j) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 89.2%
	ii) 1.6%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 92.6%
	iii) 2.9%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.3%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 90.3%
Collect	i) 0.1%	i) 4.5%	i) 0.0%	i) 0.2%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 94.2%
	ii) 0.1%	ii) 5.7%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 98.2%
	iii) 0.1%	iii) 9.2%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 98.3%
Drive	i) 0.0%	i) 0.0%	i) 6.9%	i) 0.0%	i) 0.2%	i) 0.3%	i) 0.0%	i) 0.2%	i) 0.5%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.5%	i) 80.1%
	ii) 0.0%	ii) 0.0%	ii) 6.8%	ii) 0.0%	ii) 0.3%	ii) 0.3%	ii) 0.0%	ii) 0.2%	ii) 0.5%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.3%	ii) 81.6%
	iii) 0.0%	iii) 0.0%	iii) 7.2%	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.5%	iii) 0.0%	iii) 81.4%				
Enter Car	i) 0.3%	i) 4.5%	i) 0.0%	i) 2.7%	i) 0.0%	i) 0.1%	i) 0.3%	i) 0.0%	i) 0.4%	i) 0.2%	i) 0.3%	i) 0.8%	i) 0.1%	i) 0.1%	i) 27.7%
	ii) 0.6%	ii) 0.8%	ii) 0.4%	ii) 3.7%	ii) 0.0%	ii) 0.5%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 61.0%
	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 4.2%	iii) 0.0%	iii) 0.4%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 88.1%
Exit Car	i) 0.0%	i) 0.1%	i) 0.0%	i) 0.0%	i) 6.3%	i) 0.1%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.1%	i) 0.1%	i) 95.2%
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 6.3%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 95.8%
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 6.4%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 96.7%
Idle	i) 0.9%	i) 0.0%	i) 0.0%	i) 0.9%	i) 0.0%	i) 23.2%	i) 4.0%	i) 0.1%	i) 0.0%	i) 0.1%	i) 0.1%	i) 0.0%	i) 0.0%	i) 1.5%	i) 75.0%
	ii) 0.2%	ii) 0.2%	ii) 0.0%	ii) 1.1%	ii) 0.0%	ii) 22.6%	ii) 3.3%	ii) 0.2%	ii) 0.4%	ii) 1.2%	ii) 0.5%	ii) 0.6%	ii) 2.0%	ii) 69.6%	
	iii) 0.1%	iii) 0.2%	iii) 0.0%	iii) 1.1%	iii) 0.0%	iii) 23.0%	iii) 3.3%	iii) 0.3%	iii) 0.0%	iii) 1.2%	iii) 0.0%	iii) 0.6%	iii) 2.0%	iii) 72.2%	
Open Bag	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.3%	i) 0.0%	i) 1.0%	i) 3.8%	i) 0.0%	i) 0.0%	i) 0.5%	i) 0.0%	i) 0.2%	i) 0.3%	i) 0.3%	i) 62.6%
	ii) 0.3%	ii) 1.8%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.8%	ii) 5.6%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 59.0%
	iii) 0.0%	iii) 0.8%	iii) 5.5%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 84.7%				
Run	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.1%	i) 0.0%	i) 0.1%	i) 0.0%	i) 4.1%	i) 0.0%	i) 0.0%	i) 0.3%	i) 0.0%	i) 0.5%	i) 0.5%	i) 80.6%
	ii) 0.0%	ii) 1.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.1%	ii) 0.0%	ii) 2.7%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.3%	ii) 64.5%
	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 2.7%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.3%	iii) 88.7%				
Sit	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.6%	i) 0.0%	i) 0.1%	i) 0.0%	i) 0.0%	i) 3.7%	i) 0.7%	i) 0.1%	i) 0.0%	i) 0.0%	i) 0.0%	i) 69.4%
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 2.6%	ii) 0.3%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 86.9%					
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.4%	iii) 0.0%	iii) 95.5%									
Sit Down	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.2%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.5%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 65.3%
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.8%	ii) 0.6%	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 38.8%					
	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.1%	iii) 3.7%	iii) 0.9%	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 17.8%
Stand Up	i) 0.0%	i) 0.2%	i) 0.2%	i) 0.3%	i) 0.0%	i) 0.0%	i) 0.3%	i) 0.0%	i) 0.0%	i) 0.0%	i) 2.4%	i) 0.0%	i) 0.1%	i) 0.1%	i) 67.0%
	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.5%	ii) 0.0%	ii) 0.3%	ii) 0.0%	ii) 0.0%	ii) 0.2%	ii) 0.0%	ii) 2.5%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 73.0%
	iii) 0.0%	iii) 0.2%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 3.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 95.3%				
Load	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 5.3%	i) 0.0%	i) 0.0%	i) 99.1%					
	ii) 0.1%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 4.8%	ii) 0.0%	ii) 0.0%	ii) 98.9%				
	iii) 0.1%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 5.6%	iii) 0.0%	iii) 0.0%	iii) 98.3%				
Walk	i) 0.5%	i) 0.1%	i) 0.0%	i) 0.1%	i) 0.0%	i) 0.4%	i) 0.2%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 0.0%	i) 10.1%	i) 85.1%
	ii) 0.2%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.4%	ii) 0.0%	ii) 1.5%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 0.0%	ii) 10.4%	ii) 83.3%	
	iii) 0.0%	iii) 0.4%	iii) 0.1%	iii) 1.5%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 0.0%	iii) 10.8%	iii) 84.6%					
	i) 41.9%	i) 47.1%	i) 95.2%	i) 50.4%	i) 96.7%	i) 91.9%	i) 43.0%	i) 87.9%	i) 79.9%	i) 24.9%	i) 71.9%	i) 83.8%	i) 76.2%	i) 74.7%	
	ii) 51.9%	ii) 59.6%	ii) 94.3%	ii) 70.0%	ii) 96.2%	ii) 89.5%	ii) 62.7%	ii) 57.6%	ii) 57.6%	ii) 28.1%	ii) 77.9%	ii) 75.8%	ii) 79.0%	ii) 76.0%	
	iii) 90.8%	iii) 96.7%	iii) 98.8%	iii) 78.0%	iii) 97.8%	iii) 91.2%	iii) 61.5%	iii) 58.7%	iii) 7.9%	iii) 40.7%	iii) 94.2%	iii) 89.4%	iii) 81.7%	iii) 81.7%	
	Close Bag	Collect	Drive	Enter Car	Exit Car	Idle	Open Bag	Run	Sit	Sit Down	Stand Up	Load	Walk		

Figure 14. The confusion matrix of Fuzzy-LSTM* (i), FS-LSTM* (ii) and FS-LSTM** (iii). Superior classification accuracy results are identified in bold.

These conclusions are not new, though what can be seen is that Fuzzy-LSTM* (i) and FS-LSTM* (ii), show a decrease in accuracy of 74.7% and 76.0%, with FS-LSTM slightly dethroning Fuzzy-LSTM under a lower number of resources (32 hidden layers instead of 64 hidden layers), while still requiring less GPU utilization. Furthermore, while the FS-LSTM 64 hidden layer Lost network (with 32 hidden layers for all other states), FS-LSTM** (iii), presents a similar GPU utilization and power consumption than the Fuzzy-LSTM with a 32 hidden layer (Fuzzy-LSTM*), its accuracy rises to 81.7% (versus the 74.7% of Fuzzy-LSTM*). While this is not an outstanding result, it can be a useful compromise between model size, performance, and energy efficiency in certain applications. One such application is covered by the FEROX Project, where the system is expected to run on smaller portable devices, such as smartphones and wearables. In such scenarios, an efficient utilization of computing resources becomes crucial, and the FS-LSTM model may offer a viable solution without significantly reducing the human activity recognition accuracy.

Table 1. The mean ± standard deviation of GPU utilization and power consumption. Fuzzy-LSTM and FS-LSTM networks trained with 32 layers (denoted with *), and a hybrid version of FS-LSTM (denoted with **).

Mean ± SD	Fuzzy-LSTM	FS-LSTM	Fuzzy-LSTM *	FS-LSTM *	FS-LSTM **
GPU Utilization	44.55 ± 9.85	35.21 ± 9.57	28.05 ± 9.56	25.58 ± 9.46	27.54 ± 10.16
Power Consumption	29.25 ± 2.04	28.14 ± 1.91	24.66 ± 0.89	24.25 ± 0.89	24.70 ± 1.34

5.3. Discussion

As described in Section 1.2, this paper presented three key incremental developments, which have been successfully achieved and summarized as it follows:

- i Enhance Long Short-Term Memory (LSTM) networks by incorporating fuzzy logic to model human uncertainty (Fuzzy-LSTM): the objective of this research was to enhance the accuracy of LSTM networks by incorporating fuzzy logic to model human uncertainty. Even though the preliminary results, as shown in Figure 5, depicted an accuracy of 84.9% for recognizing four activities, such accuracy dropped to 23.3% when trying to recognize 13 different activities. By utilizing fuzzified Motion and Tilt features, the Fuzzy-LSTM model was capable to effectively handle uncertain data. The initial results of the study showed that Fuzzy-LSTM improved the accuracy of activity recognition by a significant margin, achieving 93.2% accuracy compared to the traditional LSTM model using raw sensor data.
- ii Extend the Fuzzy-LSTM approach by incorporating finite-state machines (FSM) to model activity sequences, resulting in the Fuzzy State LSTM (FS-LSTM) model: the primary objective of this research was to improve the predictability of human activity sequences by identifying possible transitions between states. While Fuzzy-LSTM achieved 93.2% accuracy under unconstrained GPU resources, approximately 3% more than FS-LSTM, it often resulted in infeasible transitions. However, when using constrained resources such as embedded systems or limited GPU resources, the FS-LSTM showed superior performance compared to Fuzzy-LSTM. As shown in Figure 14, FS-LSTM has a trade-off between accuracy and computational resources, but it offers significant benefits for long-term real-time outdoor applications.
- iii Develop a defuzzification-based method to estimate human uncertainty by aggregating predicted scores of the LSTM model: this proposed approach aimed to estimate the uncertainty associated with the LSTM classifier's predictions through defuzzification. By waiting until the prediction became certain, the system could achieve an accuracy of 90.0%. This development is crucial to prevent the system from making wrong transitions between states before the model becomes certain, thereby improving the overall performance of the system. Although this study did not show the direct impact of this issue, it could significantly affect the high-level decision-making process for robots, where the system needs to consider the current human state. Any infeasible transitions could compromise the performance of the system, causing trust and safety issues.

This study presents a few drawbacks and limitations that should be taken into consideration. Firstly, the results were obtained using only synthetic data. Although the preliminary results (Figure 5) show that models trained with such data are transferable to real-world data, this may still not accurately reflect real-world scenarios or domains without further modification and customization. Therefore, caution should be exercised when applying the proposed approach to other contexts. Furthermore, while synthetic data provides advantages such as low-cost data generation and easy labelling, it may not fully capture the complexity and variability of real-world data. Thus, a further study is needed to ensure the generation of data with the same characteristics as real data when using a different type of sensor. Additionally, in a real-world setup, the sensor placement and other aspects of the experimental setup may need to be adjusted to account for different environmental conditions and potential interferences.

6. Conclusions and Future Work

This study presents an architecture for human activity recognition and modelling intended for use in human-robot collaboration in field applications. The approach employs multiple LSTM networks, each trained to recognize feasible states within an FSM architecture. The FSM architecture is further enhanced with fuzzy logic to determine the uncertainty level of the classification made by the LSTM, thereby preventing unfeasible

activity transitions in high-level decision-making systems. The proposed approach is compared to a traditional LSTM model trained on raw sensory data and a Fuzzy-LSTM model that used fuzzified sensory data as inputs to train a single LSTM network.

The proposed approach achieves high accuracy, with a rate of 90.9%, while efficiently utilizing computer resources. The system's performance is evaluated using synthetic data generated from a berry collection use case developed in a simulator. Future work will involve assessing the system's performance using real-world data within the FEROX project, as well as optimizing the developed work to operate on a small platform such as a smartphone.

As a continuation of this work, the next step will involve developing a high-level decision-making system that utilizes the human state predicted by the proposed FS-LSTM, as well as its associated uncertainty, to make informed decisions for each agent in a multi-robot and multi-human system. The decision-making system will be developed based on explicit and implicit relationships between agents, building upon the presented study's understanding of human activity.

Author Contributions: Conceptualization, B.Y. and M.S.C.; methodology, B.Y. and M.S.C.; software, B.Y.; validation, B.Y.; investigation, B.Y. and M.S.C.; resources, S.P.S. and A.V.; writing—original draft preparation, B.Y.; writing—review and editing, M.S.C., S.P.S. and A.V.; visualization, B.Y.; supervision, M.S.C., S.P.S. and A.V.; project administration, M.S.C.; funding acquisition, S.P.S. and A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partly funded by the EU FEROX project (<https://ferox.fbk.eu/> (accessed on 10 February 2023)). FEROX has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101070440. Views and opinions expressed are however those of the authors only and the European Commission is not responsible for any use that may be made of the information it contains. This work was also partially funded by FCT—Fundação para a Ciência e a Tecnologia (FCT) I.P., through national funds, within the scope of the UIDB/00127/2020 project (IEETA/UA, <http://www.ieeta.pt/> (accessed on 25 February 2023)). This work was also supported by Ingeniarius Ltd with provided materials and co-funded by UTAD.

Data Availability Statement: The data generated in this study are openly available here: https://gitlab.ingeniarius.pt/ingeniarius_public/ferox/hrc-ferox.git (accessed on 22 February 2023)).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HRC	Human Robot Collaboration
HAR	Human Activity Recognition
FS-LSTM	Fuzzy State-Long Short-Term Memory
Bi-LSTM	Bidirectional Long Short-Term Memory
SVM	Support Vector Machines
HMM	Hidden Markov Models
FSM	Finite State Machines
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
DNN	Deep Neural Networks
ANN	Artificial Neural Networks
DBMM	Dynamic Bayesian Mixture Model
IMU	Inertial Measurement Units
ROS	Robot Operating System
GPU	Graphics Processing Unit

References

1. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* **2018**, *55*, 248–266. [CrossRef]
2. Ajoudani, A.; Zanchettin, A.M.; Ivaldi, S.; Albu-Schäffer, A.; Kosuge, K.; Khatib, O. Progress and prospects of the human–robot collaboration. *Auton. Robot.* **2018**, *42*, 957–975. [CrossRef]
3. Chueshev, A.; Melekhova, O.; Meshcheryakov, R. Cloud Robotic Platform on Basis of Fog Computing Approach. In *Interactive Collaborative Robotics, Proceedings of the Interactive Collaborative Robotics, Leipzig, Germany, 18–22 September 2018*; Ronzhin, A., Rigoll, G., Meshcheryakov, R., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 34–43.
4. Rodriguez-Losada, D.; Matia, F.; Jimenez, A.; Galan, R.; Lacey, G. Implementing Map Based Navigation in Guido, the Robotic SmartWalker. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005*; pp. 3390–3395. [CrossRef]
5. Jia, P.; Hu, H. Head gesture based control of an intelligent wheelchair. In *Proceedings of the 11th Annual Conference of the Chinese Automation and Computing Society in the UK [CACSSUK05], Sheffield, UK, 10 September 2005*; pp. 85–90.
6. Montemerlo, M.; Pineau, J.; Roy, N.; Thrun, S.; Verma, V. Experiences with a mobile robotic guide for the elderly. *AAAI/IAAI* **2002**, *2002*, 587–592.
7. Bauer, A.; Wollherr, D.; Buss, M. Human–robot collaboration: A survey. *Int. J. Humanoid Robot.* **2008**, *5*, 47–66. [CrossRef]
8. Albu-Schäffer, A.; Haddadin, S.; Ott, C.; Stemmer, A.; Wimböck, T.; Hirzinger, G. The DLR lightweight robot: Design and control concepts for robots in human environments. *Ind. Robot. Int. J.* **2007**, *34*, 376–385. [CrossRef]
9. Nweke, H.F.; Teh, Y.W.; Mujtaba, G.; Al-garadi, M.A. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Inf. Fusion* **2019**, *46*, 147–170. [CrossRef]
10. Xiao, Q.; Song, R. Action recognition based on hierarchical dynamic Bayesian network. *Multimed. Tools Appl.* **2018**, *77*, 6955–6968. [CrossRef]
11. Hu, C.; Chen, Y.; Hu, L.; Peng, X. A novel random forests based class incremental learning method for activity recognition. *Pattern Recognit.* **2018**, *78*, 277–290. [CrossRef]
12. Abidine, B.; Fergani, L.; Fergani, B.; Oussalah, M. The joint use of sequence features combination and modified weighted SVM for improving daily activity recognition. *Pattern Anal. Appl.* **2018**, *21*, 119–138. [CrossRef]
13. Ronao, C.A.; Cho, S.B. Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models. In *Proceedings of the 2014 10th International Conference on Natural Computation (ICNC), Xiamen, China, 19–21 August 2014*; IEEE: New York, NY, USA, 2014; pp. 681–686.
14. Mohamed, G.; Lotfi, A.; Pourabdollah, A. Enhanced fuzzy finite state machine for human activity modelling and recognition. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 6077–6091. [CrossRef]
15. Tan, T.H.; Gochoo, M.; Huang, S.C.; Liu, Y.H.; Liu, S.H.; Huang, Y.F. Multi-resident activity recognition in a smart home using RGB activity image and DCNN. *IEEE Sens. J.* **2018**, *18*, 9718–9727. [CrossRef]
16. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
17. Lee, S.M.; Yoon, S.M.; Cho, H. Human activity recognition from accelerometer data using Convolutional Neural Network. In *Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (Bigcomp), Jeju, Republic of Korea, 13–16 February 2017*; IEEE: New York, NY, USA, 2017; pp. 131–134.
18. Inoue, M.; Inoue, S.; Nishida, T. Deep recurrent neural network for mobile human activity recognition with high throughput. *Artif. Life Robot.* **2018**, *23*, 173–185. [CrossRef]
19. Devitt, S. Trustworthiness of autonomous systems. *Foundations of Trusted Autonomy (Studies in Systems, Decision and Control, Volume 117)*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 161–184.
20. Karthigasari, R.; Sornam, M. Evolutionary Model and Fuzzy Finite State Machine for Human Activity Recognition. Available online: <http://www.ijcnes.com/documents/%20V8-11-P7.pdf> (accessed on 1 March 2023) .
21. Kong, Y.; Fu, Y. Human action recognition and prediction: A survey. *Int. J. Comput. Vis.* **2022**, *130*, 1366–1401. [CrossRef]
22. Kostavelis, I.; Vasileiadis, M.; Skartados, E.; Kargakos, A.; Giakoumis, D.; Bouganis, C.S.; Tzovaras, D. Understanding of human behavior with a robotic agent through daily activity analysis. *Int. J. Soc. Robot.* **2019**, *11*, 437–462. [CrossRef]
23. Osman, M. Controlling uncertainty: A review of human behavior in complex dynamic environments. *Psychol. Bull.* **2010**, *136*, 65. [CrossRef]
24. Golan, M.; Cohen, Y.; Singer, G. A framework for operator–workstation interaction in Industry 4.0. *Int. J. Prod. Res.* **2020**, *58*, 2421–2432. [CrossRef]
25. Vuckovic, A.; Kwantes, P.J.; Neal, A. Adaptive decision making in a dynamic environment: A test of a sequential sampling model of relative judgment. *J. Exp. Psychol. Appl.* **2013**, *19*, 266. [CrossRef]
26. Law, T.; Scheutz, M. *Trust: Recent Concepts and Evaluations in Human-Robot Interaction*; Academic Press: New York, NY, USA, 2021; pp. 27–57. [CrossRef]
27. Kwon, W.Y.; Suh, I.H. Planning of proactive behaviors for human–robot cooperative tasks under uncertainty. *Knowl.-Based Syst.* **2014**, *72*, 81–95. [CrossRef]
28. Ramasamy Ramamurthy, S.; Roy, N. Recent trends in machine learning for human activity recognition—A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1254. [CrossRef]

29. Dua, N.; Singh, S.N.; Semwal, V.B. Multi-input CNN-GRU based human activity recognition using wearable sensors. *Computing* **2021**, *103*, 1461–1478. [[CrossRef](#)]
30. Narayanan, M.R.; Scalzi, M.E.; Redmond, S.J.; Lord, S.R.; Celler, B.G.; Lovell, N.H. A wearable triaxial accelerometry system for longitudinal assessment of falls risk. In Proceedings of the 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vancouver, BC, Canada, 20–25 August 2008; pp. 2840–2843. [[CrossRef](#)]
31. Anguita, D.; Ghio, A.; Oneto, L.; Parra, X.; Reyes-Ortiz, J.L. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Proceedings of the International Workshop on Ambient Assisted Living*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 216–223.
32. Kolekar, M.H.; Dash, D.P. Hidden markov model based human activity recognition using shape and optical flow based features. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), Singapore, 22–25 November 2016; IEEE: New York, NY, USA, 2016; pp. 393–397.
33. Abdul-Azim, H.A.; Hemayed, E.E. Human action recognition using trajectory-based representation. *Egypt. Inform. J.* **2015**, *16*, 187–198. [[CrossRef](#)]
34. Kellokumpu, V.; Pietikäinen, M.; Heikkilä, J. Human activity recognition using sequences of postures. In Proceedings of the MVA, Tsukuba Science City, Japan, 16–18 May 2005; pp. 570–573.
35. Yamato, J.; Ohya, J.; Ishii, K. Recognizing human action in time-sequential images using hidden Markov model. In Proceedings of the Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, IL, USA, 15–18 June 1992; pp. 379–385. [[CrossRef](#)]
36. Chen, K.; Zhang, D.; Yao, L.; Guo, B.; Yu, Z.; Liu, Y. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *Acm Comput. Surv.* **2021**, *54*, 77. [[CrossRef](#)]
37. Parmar, A.; Katariya, R.; Patel, V. A review on random forest: An ensemble classifier. In *International Conference on Intelligent Data Communication Technologies and Internet of Things*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 758–763.
38. Song, Q.; Liu, X.; Yang, L. The random forest classifier applied in droplet fingerprint recognition. In Proceedings of the 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 15–17 August 2015; IEEE: New York, NY, USA, 2015; pp. 722–726.
39. Wan, S.; Qi, L.; Xu, X.; Tong, C.; Gu, Z. Deep learning models for real-time human activity recognition with smartphones. *Mob. Netw. Appl.* **2020**, *25*, 743–755. [[CrossRef](#)]
40. Hammerla, N.Y.; Halloran, S.; Plötz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv* **2016**, arXiv:1604.08880.
41. Vepakomma, P.; De, D.; Das, S.K.; Bhansali, S. A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In Proceedings of the 2015 IEEE 12th International conference on wearable and implantable body sensor networks (BSN), Cambridge, UK, 9–12 June 2015; IEEE: New York, NY, USA, 2015; pp. 1–6.
42. Bai, L.; Yao, L.; Wang, X.; Kanhere, S.P.S.; Xiao, Y. Prototype similarity learning for activity recognition. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Zhangjiajie, China, 15–17 August 2015; Springer: Berlin/Heidelberg, Germany, 2020; pp. 649–661.
43. Duffner, S.; Berlemont, S.; Lefebvre, G.; Garcia, A. 3D gesture classification with convolutional neural networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: New York, NY, USA, 2014; pp. 5432–5436.
44. Ishimaru, S.; Hoshika, K.; Kunze, K.; Kise, K.; Dengel, A. Towards reading trackers in the wild: Detecting reading activities by EOG glasses and deep neural networks. In *UbiComp '17: Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 704–711.
45. Guan, Y.; Plötz, T. Ensembles of deep lstm learners for activity recognition using wearables. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2017**, *1*, 11. [[CrossRef](#)]
46. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115. [[CrossRef](#)]
47. Hossain Shuvo, M.M.; Ahmed, N.; Nouduri, K.; Palaniappan, K. A Hybrid Approach for Human Activity Recognition with Support Vector Machine and 1D Convolutional Neural Network. In Proceedings of the 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 13–15 October 2020; pp. 1–5. [[CrossRef](#)]
48. Faria, D.R.; Premebida, C.; Nunes, U. A probabilistic approach for human everyday activities recognition using body motion from RGB-D images. In Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, UK, 25–29 August 2014; IEEE: New York, NY, USA, 2014; pp. 732–737.
49. Nunes Rodrigues, A.C.; Santos Pereira, A.; Sousa Mendes, R.M.; Araújo, A.G.; Santos Couceiro, M.; Figueiredo, A.J. Using artificial intelligence for pattern recognition in a sports context. *Sensors* **2020**, *20*, 3040. [[CrossRef](#)]
50. Vital, J.P.; Faria, D.R.; Dias, G.; Couceiro, M.S.; Coutinho, F.; Ferreira, N.M. Combining discriminative spatiotemporal features for daily life activity recognition using wearable motion sensing suit. *Pattern Anal. Appl.* **2017**, *20*, 1179–1194. [[CrossRef](#)]
51. Martínez-González, P.; Oprea, S.; García-García, A.; Jover-Alvarez, A.; Orts-Escolano, S.; García-Rodríguez, J. Unrealrox: An extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Real.* **2020**, *24*, 271–288. [[CrossRef](#)]

52. Puig, X.; Ra, K.; Boben, M.; Li, J.; Wang, T.; Fidler, S.; Torralba, A. VirtualHome: Simulating Household Activities via Programs. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
53. Quigley, M.; Gerkey, B.; Conley, K.; Faust, J.; Foote, T.; Leibs, J.; Berger, E.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
54. Zangenehjad, F.; Gao, Y. GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. *Satell. Navig.* **2021**, *2*, 24. [[CrossRef](#)] [[PubMed](#)]
55. Kim, A.; Golnaraghi, M. A quaternion-based orientation estimation algorithm using an inertial measurement unit. In Proceedings of the PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556), Monterey, CA, USA, 26–29 April 2004; pp. 268–272. [[CrossRef](#)]
56. Haq, I.U.; Ullah, A.; Khan, S.U.; Khan, N.; Lee, M.Y.; Rho, S.; Baik, S.W. Sequential learning-based energy consumption prediction model for residential and commercial sectors. *Mathematics* **2021**, *9*, 605. [[CrossRef](#)]
57. Khan, I.U.; Afzal, S.; Lee, J.W. Human activity recognition via hybrid deep learning based model. *Sensors* **2022**, *22*, 323. [[CrossRef](#)] [[PubMed](#)]
58. Han, S.; Kang, J.; Mao, H.; Hu, Y.; Li, X.; Li, Y.; Xie, D.; Luo, H.; Yao, S.; Wang, Y.; et al. Ese: Efficient speech recognition engine with sparse lstm on fpga. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Washington, DC, USA, 14–18 August 2017; pp. 75–84.
59. Berrar, D. Bayes' theorem and naive Bayes classifier. *Encycl. Bioinform. Comput. Biol. ABC Bioinform.* **2018**, *403*, 412.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.