

Article

IoT and Deep Learning-Based Farmer Safety System

Yudhi Adhitya ^{1,*} , Grathya Sri Mulyani ¹, Mario Köppen ¹  and Jenq-Shiou Leu ² ¹ Department of Computer Science and Systems Engineering (CSSE), Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka-shi 820-8502, Fukuoka, Japan² Department of Electronic and Computer Engineering (ECE), National Taiwan University of Science and Technology, Taipei City 106, Taiwan

* Correspondence: adhitya.yudhi898@mail.kyutech.jp

Abstract: Farming is a fundamental factor driving economic development in most regions of the world. As in agricultural activity, labor has always been hazardous and can result in injury or even death. This perception encourages farmers to use proper tools, receive training, and work in a safe environment. With the wearable device as an Internet of Things (IoT) subsystem, the device can read sensor data as well as compute and send information. We investigated the validation and simulation dataset to determine whether accidents occurred with farmers by applying the Hierarchical Temporal Memory (HTM) classifier with each dataset input from the quaternion feature that represents 3D rotation. The performance metrics analysis showed a significant 88.00% accuracy, precision of 0.99, recall of 0.04, F_Score of 0.09, average Mean Square Error (MSE) of 5.10, Mean Absolute Error (MAE) of 0.19, and a Root Mean Squared Error (RMSE) of 1.51 for the validation dataset, 54.00% accuracy, precision of 0.97, recall of 0.50, F_Score of 0.66, MSE = 0.06, MAE = 3.24, and = 1.51 for the Farming-Pack motion capture (mocap) dataset. The computational framework with wearable device technology connected to ubiquitous systems, as well as statistical results, demonstrate that our proposed method is feasible and effective in solving the problem's constraints in a time series dataset that is acceptable and usable in a real rural farming environment for optimal solutions.

Keywords: time series dataset; quaternion; hierarchical temporal memory; cascade classifier; probability prediction; farming activity monitoring; smart farming



Citation: Adhitya, Y.; Mulyani, G.S.; Köppen, M.; Leu, J.-S. IoT and Deep Learning-Based Farmer Safety System. *Sensors* **2023**, *23*, 2951. <https://doi.org/10.3390/s23062951>

Academic Editor: Rongxing Lu, Sajjad Dadkhah, Jianting Ning and Beibei Li

Received: 14 February 2023

Revised: 2 March 2023

Accepted: 3 March 2023

Published: 8 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

Within the past two decades, the Internet of Things (IoT) concept has evolved. IoT-connected devices can exchange information, commands, and decisions independently through intelligent networks within their ecosystem. The rapid development of IoT is opening up exceptional opportunities for various implementation sectors, including agriculture, healthcare, and manufacturing [1]. Implementing IoT infrastructure is expected to improve efficiency, reduce costs, and save energy, as well as prevent loss and accidents and open up new business opportunities.

Agriculture sectors frequently disregarded business cases for IoT implementation as it requires adapting to well-known farming activity [2–4]. Nevertheless, due to the remoteness of farming operations, IoT implementation proposes the possibility of transforming farmers' practices. Even though this concept has yet to gain widespread acceptance, it is one of the IoT-based applications that must be acknowledged. Smart farming will become a significant application field in agriculturally-based countries [5–7]. Smart farming involves applying data and information technology in a complex farming system to be maximized. Data, information, and communication technologies are implemented to utilize machinery, equipment, and installation of sensors used in agricultural production systems. By leveraging the above-mentioned technologies, IoT [8] and cloud computing can strengthen the effectiveness and efficiency even further, one with innumerable sophisticated sensors, including the implementation of artificial intelligence into agriculture [9–12].

Work involving the management of livestock, tractors, and heavy agricultural equipment, exposure to toxic chemicals, or laboring in open fields under the sun's heat or in a closed chamber, all have significant risks [13–15]. Employment in agriculture requires farm workers to work at elevations to finish their jobs. Falls from these elevations are the most expected agricultural casualties. Falling from a certain altitude can cause fractures, brain damage, and additional severe injuries. Falls, as the leading causes of unintentional injuries worldwide, with 37.3 million cases of falls each year, require medical attention [16].

Wearable technology appliances that use accelerometers and gyroscopes datasets have developed ecosystems and consumers [12,17–20]. Advanced technology gadgets that are linked to specific devices have become ubiquitous in our daily lives, ushering in a new trend in the mobile industry. These devices enable data generation, enabling the collection of massive quantities of data and information. Three angles can represent the 3D rotation [21,22] in studies using the Euler angle, which decomposes the rotation within three separate angles. Although using Euler angles is a natural step in the process of representing 3D rotations [23,24], it can cause difficulties in performing calculations, such as rounding errors in the conversion results, which can lead to distortion. More importantly, the Euler angle representation needs to be more linear. Farmers are recognizing the benefits of advancements in wearable technology, which have promising applications in healthcare products, monitoring, and personal health. Wearable technology can also be utilized in agriculture [25–27] and is expected to prepare preventative measurements for diverse categories of casualties in agriculture.

Deep learning can take advantage of IoT technology implementation. IoT sensors that can provide the input data of farmers' positions and heart rates can be analyzed utilizing intelligent algorithms, combining data collected with previous data and training models to identify patterns and make predictions about the safety conditions of farmers in working environments, as depicted in Figure 1.

To overcome this problem, we proposed a quaternion to represent 3D rotation [28] as an input feature. While most of the current literature studies use three-dimensional data from accelerometer data and gyroscope data for activity prediction, we exploited quaternion as four-dimensional data (as a feed input for the Hierarchical Temporal Memory (HTM) classifier). We evaluated the approaches of large-scale egocentric datasets and farming pack mocap datasets; our proposed method is practical for computational efficiency and results. As a result, the primary goal of this study was to evaluate extracted quaternion features for hierarchical temporal memory input for accurate activity prediction.

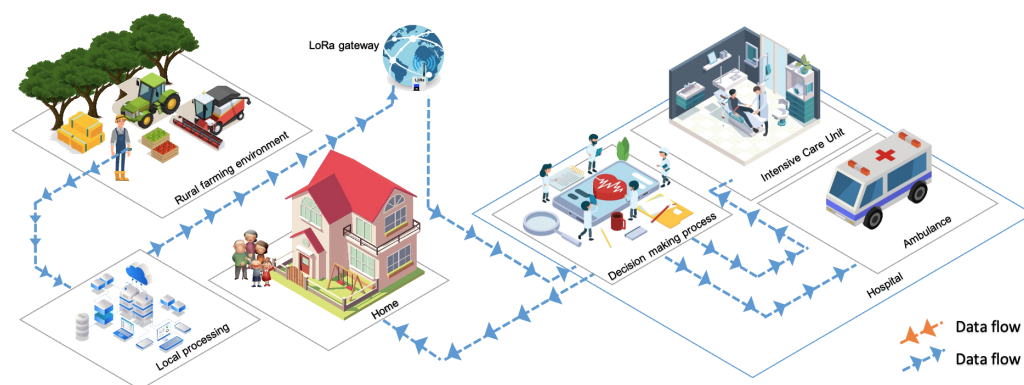


Figure 1. The Proposed farmer safety system in a smart farming environment.

The general framework for farmer safety systems in the smart farming environment is proposed in Figure 1. The farmer's use of wearable sensor devices as an IoT subsystem has the potential to improve data readability in today's medical systems, which can provide substantial sensor tracking of multiple physiological functions within the body, as well as cost-effective and high-efficiency services in a wide range of fields. These wearable devices generally use the battery as their power source and implement various power

management strategies to extend battery life [29]. Most wearable devices can connect to the internet via Wi-Fi, and some even have their own cellular services. Accelerometer data and gyroscope data are collected on the device for local processing and transferred to the cloud by using the long-range (LoRa) network for decisions. This type of network has a distance range that can cover rural areas from the farm itself. The output of this computational process is then sent to the paramedics at the hospital. With these details, the caregiver can determine the proper action for preventing and following medical treatment measures [30–32] on what actions to take next; for example, sending an ambulance to the farmer, sending information to the farmer's family about what happened to the farmer, and applying medical procedures at the hospital.

The goal is to exploit quaternion as four-dimensional data as the feed input for the classifier and compare the proposed algorithm to the other algorithm based on their strengths and vulnerabilities. In accordance with the findings from this study, the proposed framework significantly outperformed the others. The contributions of the proposed work are as follows:

1. We utilized a quaternion as an input feature to represent 3D rotation.
2. We evaluated the approaches on both large-scale egocentric datasets and farming pack mocap datasets, and demonstrate that our proposed method is practical regarding computational efficiency and result.
3. Using the proposed algorithm, we determined the optimal solutions for the proposed system.
4. With the same train test data, the proposed algorithm, HTM classifier, and (k-Nearest Neighbor) kNN classifier were compared.
5. The performances of the two algorithms were compared using different performance metrics.

The rest of this paper is structured as follows: as this study is an IoT and deep learning-based farmer safety system, the related research on the farming activity monitoring method and the smart farming application include work on human motion classification and hierarchical classification algorithms, as introduced in Section 2. Section 3 presents the problem definition and clarifies the challenges behind the hierarchical temporal classification problem and its specificity. In Section 4, the material is proposed, i.e., in Section 4.1. Section 4.2, the proposed method and our proposed framework are formulated. The experimental results are presented in Section 5. Section 6 presents the discussion, and several studies that were compared to the existing ones are reported with numerical computations. Section 7 discusses future research directions and draws several conclusions.

2. Related Works

Health is a crucial priority [33], despite being the top revenue earner for numerous countries. Each country is exploring innovative practices to deliver quality healthcare [34,35]. The use of technology, particularly the internet, is essential to manage solutions effectively and efficiently. One such technology is machine learning algorithms combined with wearable healthcare devices [36]. Data and medical records can be collected to form a model that could be earlier by applying a machine learning algorithm [37,38].

Xiao Guo et al. [39] researched human motion prediction. The applied Skeleton Network (SkelNet) for local structure representation on different body components was used as input. The Skeleton Temporal Network (Skel-TNet) and Recurrent Neural Network (RNN) were used to apply the learning spatial and temporal dependencies. The final result was with the Residual Recurrent Neural Network (RRNN) and Convolutional Sequence to Sequence (CSS). Regardless, the analysis primarily focused on predicting short-term human motion.

The study by Judith Bütepage et al. [40] focuses on the use of the Deep Learning (DL) framework architecture to analyze human motion capture data. The study comprehends a generic representation and generalizes the motion capture data from generative models of 3D human motion. These feature representations of human motion are extracted via an encoding–decoding network that quantifies learned features from the current history,

representing various output layers for action classification. Three proposed methods, i.e., the Symmetric Temporal Encoder (S-TE), Convolutional Temporal Encoder (C-TE), and Hierarchical Temporal Encoder (H-TE) were compared with data point feature classification and Principal Component Analysis (PCA). The study covered long-term predictions, which are more accurate with sliding windows. However, it needs further investigation concerning the parameter number and each network structure.

Pham The Hai et al. [41] developed a Human Action Recognition System (HARS) in both indoor and outdoor environments. Their approach involves observation sequences to train the Hidden Markov Model (HMM) and differentiate between seven activities. They extracted preferred features from a separate sequence of a human skeletal joint mapping, which involved Baum–Welch and the forward–backward algorithm to discover the optimal parameter of the individual HMM model. An automatic database update is needed to execute the system based on infrared light and time-of-flight technology from Kinect V2.

In different applications and domains, linear combination approaches across multiple levels of the hierarchy are suggested by Evangelos Spiliotis et al. [42]. Their strategy enables a more general non-linear combination of the reference prediction, which combines the accuracy and coherence of continued-to-optimize post-sample evidence-based forecasting, combined in a straightforward and appropriate model, without the need for comprehensive information for each series and level of the reconciled forecast, and using accuracy and bias as an evaluation method. The study was evaluated with Mean Absolute Scaled Error (MASE), Root Mean Squared Scaled Error (RMSSE), and Absolute Mean Scaled Error (AMSE). Correspondingly, the study enclosed several alternative paths (by focusing on cross-sectional hierarchical structures cases), did not generalize optimization objectives to each aggregation level, and did not maintain forecast uncertainty or computational complexity.

Alaa Sagheer et al. [43] developed the Deep Long Short-Term Memory (DLSTM) model in autoencoder. They applied transfer learning to mitigate the need to update each Long Short-Term Memory (LSTM) cell's weight vector in hierarchical architectures and involve transfer learning methods to reduce the time-consuming and substantial quantity of data from various dimensions. The learned features are redistributed to the higher levels of the hierarchy for concurrent training to obtain more detailed conclusions and produce better forecasting models. The DLSTM was applied to publicly available datasets for Brazilian electrical power generation and Australian domestic tourism traveler nights. However, the study evaluated two criteria: forecasting accuracy and producing a coherent forecast but did not ensure forecast coherency at all hierarchy levels on a cross-temporal framework. The author also focused on the tourism and energy domains as examples.

Eduardo P. Costa et al. [44] is a machine learning technique used to address hierarchical classification problems in molecular biology related to proteins (organized hierarchically). They experimented with four hierarchical classification variant models (i.e., flat classification based on leaves, flat classification for all levels, top-down classification, and big-bang models) by using the G-protein-Coupled Receptor (GPCR) and enzyme protein families to predict protein functions by utilizing signature-generated protein sequences. The system requires a considerable level of detail for performance evaluation at the deepest classification level for each assessment, which is a system-automated procedure.

Yuliang Xiu et al. [45] developed a pose-tracking system for multi-person articulated poses by building associations between cross-frame and form poses. The system uses pose flows and non-maximum suppression to reduce redundant pose flows and re-link temporal disjoint. Nevertheless, the authors analyzed the proposed pose tracker's short-term action recognition and scene understanding.

Hao-Shu Fang et al. [46] identified a subtle error in localization and recognition in the single-person pose estimator. By proposing a regional multi-person pose estimation framework, inaccurate human bounding boxes can be reduced. It has three-segment approaches and can deal with incorrect bounding boxes and redundant detections. Their research focused on human bounding box poses rather than the human detector and had

difficulty distinguishing between overlapped people and similar objects when using human detection, resulting in undetected human poses.

Wojke et al. [47] proposed an approach to multi-object tracking. Their method integrates the appearance of simple online and real-time tracking. They applied offline pre-training and online application techniques to establish measurements to track associations using nearest neighbor queries in visual appearance spaces, which could track objects through more prolonged periods of occlusion. The study does not observe track jumping, which frequently occurs between false alarms.

Martinez et al. [48] proposed a method for predicting 3D positions from a 2D joint location. This system employs a simple deep feed-forward network to demonstrate the large portion of errors in a modern deep 3D pose estimation based on a visual analysis similar to a multi-layer perceptron. The approach needs more direct connections to visual evidence, which would also result in lower performance.

Hierarchical temporal memory has been linked to a Long Short-Term Memory (LSTM) network and cascade classifier [49–51]. There are principal distinctions, which is why LSTM ‘gating’ is not proper because of the neuron model complexity of the HTM model, enforcing straightforward Hebbian learning rules locally and unsupervised, without utilizing back-propagation. Procedures in HTM are based on the activation and connectivity between neurons, utilizing binary units and weights.

Several HTM use cases in the research studies [52–69] and practices that have already been commercially tested include server anomaly detection using Grok [70], stock volume anomalies [71], rogue human behavior [72], natural language prediction [73], and geospatial tracking [74].

3. Problem Definition

This section clarifies some of the fundamental challenges behind hierarchical classification, including the motivations for evaluating significance or merit. It also explains the formal definition of classification, temporal classification, and hierarchical temporal classification to better comprehend this study. Furthermore, these detailed aspects make up the complete structure of an HTM network (as shown in Figure 2).

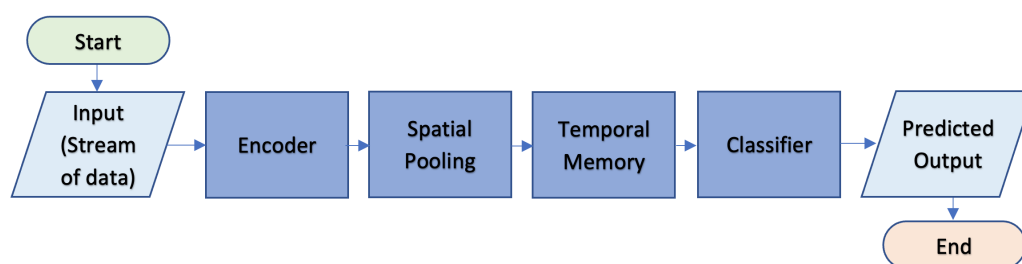


Figure 2. Hierarchical Temporal Memory (HTM) algorithm flowchart.

3.1. Classification

Classification challenges have already traditionally been considered as core components of deep learning, which would include supervised and semi-supervised learning problems.

A fundamental classification problem can be defined as producing an estimated label y from the K -dimensional of the input vector x , where:

$$x \in X \subseteq \mathbf{R}^K \quad (1)$$

For most of the common machine learning algorithms, the specific input variables must be of real value with the following:

$$y \in Y = \{C_1, C_2, \dots, C_q\} \quad (2)$$

This task employs the following classification rule or function:

$$g : X \rightarrow Y \quad (3)$$

to predict new pattern labels [75,76].

3.2. Temporal Classification

For temporal classification terminology [77–79], there are several terms defined: channel, stream, and frame.

The channel formally can be defined as a function c , which maps from a set of time-stamps T to a set of values, V , denoted as:

$$c : T \rightarrow V \quad (4)$$

where T can be the set of times at which the value of channel c is defined, assume that:

$$T = [0, 1, \dots, t_{max}] \quad (5)$$

The stream involves all of the values of the channels at once (collectively). A stream as a channel s sequence, for the same domain in each channel, is expressed as follows:

$$s = [c_1, c_2, \dots, c_n] \quad s.t. \quad \text{domain}(c_1) = \text{domain}(c_2) = \dots = \text{domain}(c_n) \quad (6)$$

where the number of channels in stream s can be noted as n .

The frame can be referred to as the values of all channels at a specific given point of time; formally, a frame can be defined as a function on an s stream and a t time. For a given stream, $s = [c_1, c_2, \dots, c_n]$, function fr is denoted as:

$$fr : \text{domain}(c_1) \rightarrow \text{range}(c_1) \times \text{range}(c_2) \times \dots \times \text{range}(c_n) \quad (7)$$

$$fr(s, t) = [c_1(t), c_2(t), \dots, c_n(t)] \quad (8)$$

For the temporal classification, let S be a training example set from a fixed distribution $D_{x \times z}$, input space $X = (\mathbf{R}^m)^*$ as a set of all sequences in m dimensional real-valued vectors, and the target space $z = l^*$ involves all sequences over the alphabet L of labels.

Each example in s consists of a pair of sequences (x, z) . The target sequence $y = (Y_1, Y_2, \dots, Y_u)$ is, at most, as long as the input sequence of $x = (X_1, x_2, \dots, x_T)$, so $U \leq T$. The aim is to use s to train a temporal classifier $g : X \rightarrow Y$ [80].

3.3. Hierarchical Temporal Classification

As an example of the biologically inspired theories of machine intelligence, the Hierarchical Temporal Memory (HTM) theory was developed in 2008 by Jeff Hawkins [81,82]. The HTM Cortical Learning Algorithm (CLA), or simply, the HTM learning algorithm, is a type of HTM learning algorithm.

In [83,84], the authors look at the neocortex theory and its principles. HTM simulates the architecture and processes of the neocortex in the biological components of the brain. Comprehensive presentation of information about the main components of CLA, including the encoder, Spatial Pooler (SP) algorithm, and Temporal Memory (TM), which are classification and prediction algorithms, are listed below.

The hierarchical temporal classification task can be simplified as classifying a stream of data into a class hierarchy.

Stream classification can be formalized as learning instances from a stream s appearing incrementally as sequences of examples labeled as $\{x^t, y^t\}$ for $t = 1, 2, \dots, t$, where x is a vector of attribute values and y is a class label $y \in \{K_1, \dots, K_l\}$. The result in example x^t is classified by classifier C , which predicts its class label [85,86].

A hierarchical classification task can be formulated as input $X : s$, where s can be denoted as the input stream. The category tree is denoted as T with the hierarchical layer as L . Categories as classes to be predicted are denoted by Y .

4. Materials and Methods

4.1. Materials

When conducting this research, the proposed framework for prediction probability was evaluated by considering two datasets. The dataset includes both a live validation experiment dataset and a farmer mocap dataset for practical evaluability assessment. The dataset analyzed is described as follows.

4.1.1. Validation Dataset

Using accelerometer and gyroscope datasets obtained from [87], a large-scale egocentric vision dataset was created. The dataset consists of 20 daily activities with the following details: it is egocentric, non-scripted, uses a native environment, has 11.5 M frames, with 432 sequences, 39,596 action segments, 149 action classes, 454,255 object bounding boxes, 323 object classes, created by 32 participants, with 32 environments [88].

We utilized this dataset to calibrate and validate the gyroscope and accelerometer data, which were then implemented into quaternion calculations. The dataset representations are visualized in Figure 3.

Figure 3a shows the plot record for the accelerometer dataset. The blue color indicates the accelerometer coordinate for the X-axis. The green color indicates the accelerometer coordinate for the Z-axis, and the orange color indicates the accelerometer coordinate for Y-axis.

Figure 3b shows the plot record for the gyroscope dataset; the blue color indicates the gyroscope coordinate for the X-axis, the green color indicates the gyroscope coordinate for the Z-axis, and the orange color indicates the gyroscope coordinate for the Y-axis.

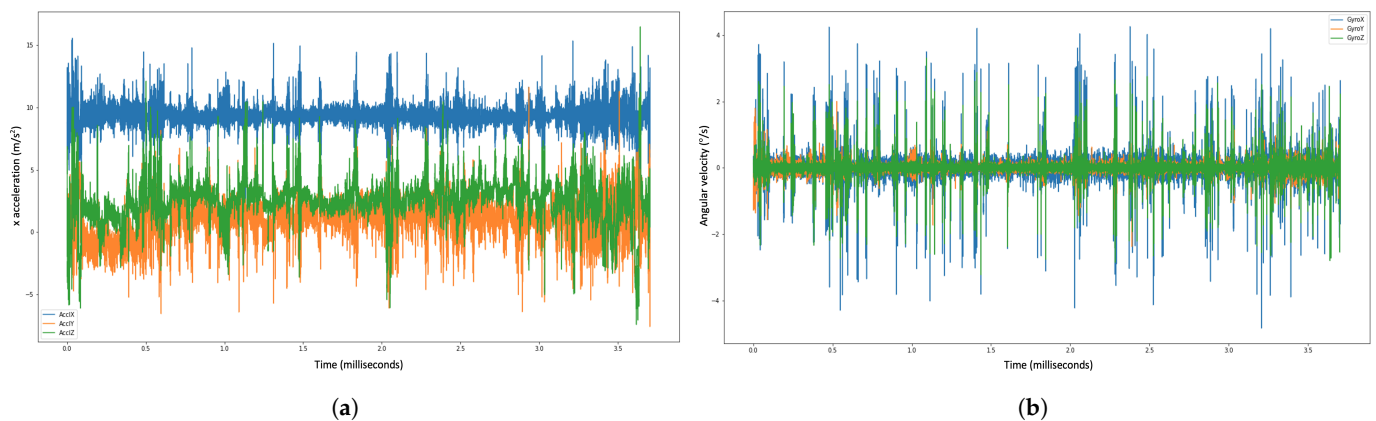


Figure 3. Example of the accelerometer and gyroscope data visualization for the validation dataset. *AccX*, *AccY*, and *AccZ* denote the x -, y -, and z -axis of the 3D accelerometer, respectively. Correspondingly, *GyroX*, *GyroY*, and *GyroZ* denote the x -, y -, and z -axis of the 3D-gyroscope sensor, respectively. (a) Accelerometer dataset. (b) Gyroscope dataset.

4.1.2. Farming-Pack Mocap Dataset

This dataset [89] is the Farming-Pack mocap animation, as shown in Figure 4. It contains in total 24 mocap datasets with related farming activity source for Figure 4, e.g., working with a wheelbarrow (5 mocap datasets), picking fruit (3 mocap dataset), milking a cow (1 mocap dataset), watering a plant (1 mocap dataset), holding activity (4 mocap datasets), planting (1 mocap dataset), planting a tree (1 mocap dataset), pull out (2 mocap datasets), digging and plant seeds (1 mocap dataset), kneeling (1 mocap dataset), and working with a box (4 mocap datasets). Based on [90–95] for the sensor attachment, we evaluated the extracted rotation dataset for the rotation position.



Figure 4. Farming-Pack mocap dataset [89]. (a) Wheelbarrow walk. (b) Wheelbarrow dump. (c) Picking fruit. (d) Milking cow. (e) Watering Plant. (f) Planting. (g) Pull plant. (h) Box walk. (i) Plant seeds. (j) Dig.

4.2. Method

Classification is one of the tasks performed by machine learning. It is defined as the process of making predictions about the class or category of observed values or relevant data provided. In this investigation, information was collected from wearable devices that could read motion data and afterward perform a classification process depicted in Figure 5. Another tracking sensor developed by Sony, “mocopi [96]” uses a small light sensor that allows anyone to easily conduct full-body tracking in 3D. Vive [97] created another tracking sensor for full-body tracking.

Then, as a comparison, additional classification was performed to see how the outcome performances were compared to the mocap dataset.

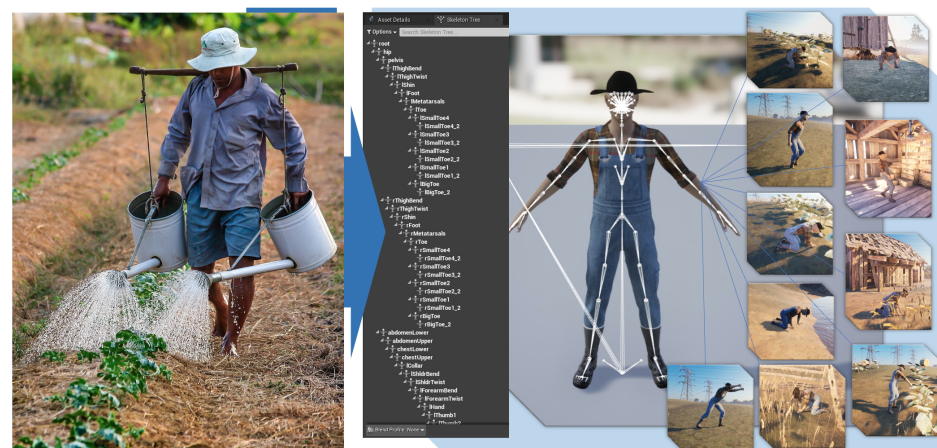


Figure 5. How to train the classifier.

Regarding local processing in a wearable device, we proposed several stages of the classification process, as shown in Figure 6.

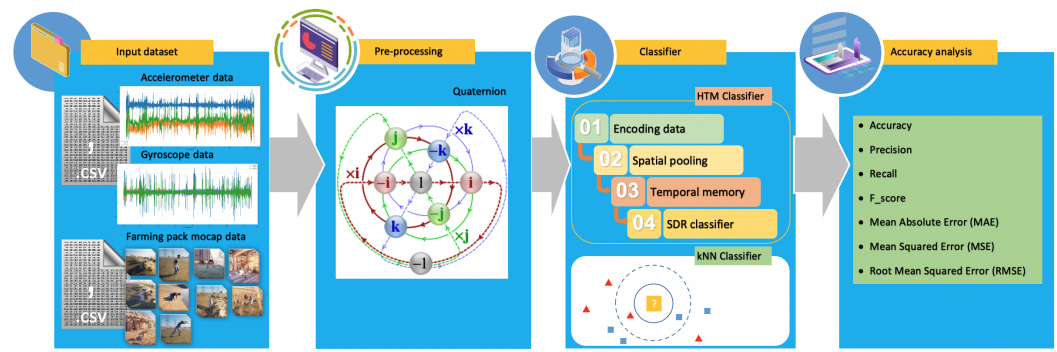


Figure 6. Proposed framework for prediction probability.

4.2.1. Feature Extraction

This section differentiates the visible components of the accelerometer and gyroscope to simplify the computation of input data interpretation. Two types of data received, i.e., the accelerometer and gyroscope dataset, were previously pre-processed by extracting quaternion [21,98–100] data from each dataset. Quaternion is a comprehensive way of describing a complex number; it was discovered by William Rowan Hamilton in the mid-19th century in 1843 [101]. The quaternion is a parameter representation of spatial rotation in three dimensions, extended to four dimensions. The representation of the quaternion calculation is described in Figure 7 [102], representing a wearable device in a smartwatch to read the accelerometer and gyroscope data.

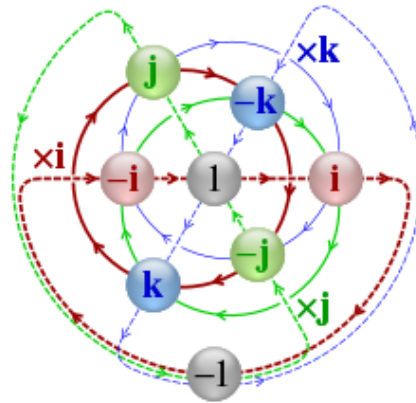


Figure 7. Quaternion visualization between axes (X, Y, Z) on a smartwatch-embedded gyroscope sensor and accelerometer sensor.

The quaternion formula is denoted as follows [103]:

$$q = q_w + q_x i + q_y j + q_z k \quad (9)$$

$$i^2 = j^2 = k^2 = ijk = -1, ij = -ji = k, ki = -ik = j, jk = -kj = i \quad (10)$$

It is applied to x, y, z as follows, with θ being the rotation angle with vector $e = (e_x, e_y, e_z)$.

$$q = \begin{pmatrix} q_w \\ q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} \cos \theta / 2 \\ e_x \sin \theta / 2 \\ e_y \sin \theta / 2 \\ e_z \sin \theta / 2 \end{pmatrix} \quad (11)$$

where w is denoted as the angle of rotation, and x, y, z are vectors representing the axes of rotation.

Quaternions are good choices for representing internal object rotations due to their efficiency in interpolation and single-orientation representation. However, the user interface presentation of quaternions is less intuitive compared to Euler angles, which are more

familiar, intuitive, and predictable. Euler angles suffer from singularity and code-level issues, such as sequential rotation order storage, performance, and permutation support.

The matrix rotation contains nine elements with three rotational degrees of freedom, making this rotation matrix redundant. This matrix can illustrate the orientation of the reference frame, which is a more compact and intuitive way to define an orientation. Furthermore, the most convenient performance for reliable interpolation is represented by a quaternion.

4.2.2. Encoding Data

Using a sparse pattern for input data on Hierarchical Temporal Memory (HTM), called Sparse Distributed Representation (SDR) [104], we used two types of encoders, i.e., *CategoryEncoder* and *RandomDistributedScalarEncoder*. *CategoryEncoder* is used to convert activity category values, as follows: ['P04', 'P35', 'P30', 'P25', 'P26', 'P12', 'P23', 'P28', 'P22', 'P36', 'P03', 'P06', 'P33', 'P09', 'P11', 'P07', 'P01', 'P37', 'P02', 'P27'] for the first datasets (EPIC-KITCHEN-100 dataset). "P##" (e.g., "P01") is a participant or activity for each person. For the numbering of category data, we do not sort the data, but still use the category based on the dataset from [87,88]. For the Farming-Pack mocap dataset, *CategoryEncoder* is as follows: [M01, M02, M03, M04, M05, M06, M07, M08, M09, M10, M11], as shown in Table 1.

Table 1. *CategoryEncoder* for the Farming-Pack mocap dataset.

<i>CategoryEncoder</i>	<i>Activity</i>
M01	Working with a wheelbarrow
M02	picking fruit
M03	milking cow
M04	watering plant
M05	holding activity
M06	planting a plant
M07	planting a tree
M08	pull-out a plant
M09	digging and planting seeds
M10	kneeling
M11	working with a box

Moreover, for other data, for example, quaternions, we used the encoder, namely *RandomDistributedScalarEncoder* to encode numeric values in the form of floating-point values into an array of bits (in this case, sparse patterns).

4.2.3. Spatial Pooling

The second step for the classifier is spatial pooling. In this step, we run the spatial pooler, which handles the column in a region and the input bit relation. The spatial pooler process returns a list of *activeColumns* columns.

4.2.4. Temporal Memory

The next step for the classifier is sequence memory, temporal pooler, or temporal memory. In this step, we perform one step of the temporal memory algorithm for learning purposes, based on the computation of temporal memory to obtain the next time step prediction by identifying patterns that transition over time and recognizing spatial patterns across temporal sequences.

4.2.5. Sparse Distributed Representation (SDR) Classifier

The classification algorithm used for the classification and prediction tasks in this experiment uses the SDR classifier. SDR is a data structure whose elements are binary, 1, or 0, representing the brain activity and, in the context of HTM, is a biologically-realistic model of neurons. The classifier receives input from the temporal memory algorithm's SDR output, also known as the activation pattern. Furthermore, additional encoder information

describes the target's actual input. The SDR classifier performs this classification task similar to implementing a single-layered feedforward neural network.

There are three detailed stages in the SDR classification model: initialization, inference, and learning. All classes are initialized with zero values to equalize the probability values before learning in the initialization stage. In the next stage of inference, they calculate the probability distribution by applying the activation function at the activation level by performing calculations in the form of class prediction probabilities for each received input pattern. The final step of this classification is learning, which proportionally adjusts the weight to the gradient, calculates the error value for each output unit, and adds it to the appropriate element weight matrix.

4.2.6. Performance Metrics Evaluation

The quaternion dataset as input data in this paper is a continuous value. In order to achieve reliability and validate the prediction of our suggested methodology, we applied related performance metrics. We used the following metrics to assess our prediction error rates and model the performance metrics: accuracy, precision, recall, *F*-score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

In classification problems, accuracy is used to calculate the percentage of correct predictions made by a model. In machine learning, the accuracy score is an evaluation metric that compares the number of correct predictions made by a model to the total number of predictions made. We calculate it by dividing the total number of predictions by the number of correct predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (12)$$

Precision is calculated as the percentage of correct predictions for a specific class out of all predictions made for that class.

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

In the case of binary classification, where we have an imbalanced classification problem, recall is calculated using the following equation:

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

where True Positives (*TP*) are positive classes that are correctly predicted to be positive. False Positives (*FP*) are negative classes that are incorrectly predicted to be positive. True Negatives (*TN*) are negative classes that are predicted to be negative. False Negatives (*FN*) are positive classes that are incorrectly predicted to be negative.

The *F*-score (also referred to as the *F1* score or *F*-measure) is a performance metric for machine learning models. It is a single score that combines precision and recall.

$$F - score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (15)$$

MAE represents the difference between the initial input data and predicted results extracted by averaging the absolute difference over the dataset.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}| \quad (16)$$

MSE represents the difference between the initial input data and predicted results extracted by squaring the average difference over the dataset, which can evaluate the degree of change in the data.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2 \quad (17)$$

The error rate multiplied by the square root of the *MSE*, or the average size of the error, is denoted by *RMSE*. It is the square root of the average squared difference between the predicted and observed values.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2} \quad (18)$$

where y_i is the observed value, $\hat{y}$ is the predicted value of y_i , $\bar{y}$ is the average of the observed value of y_i , and N is the number of samples.

5. Results

Data from the validation dataset and the Farming-Pack mocap dataset are time-series-based because they are sequential (temporal) or continuous data.

The first step of our framework is feature extraction. We extracted the accelerometer and gyroscope data from the validation dataset and convert it to quaternions based on these values. We also applied the same process to the Farming-Pack mocap dataset, where we extracted the position of the bones from the included (.bvh) file. The (.bvh) file is a text file that contains motion data. We converted the resulting positions to quaternions.

The quaternion results from the two datasets were then processed into the classifier in the following stages: data encoding, spatial pooling, temporal memory, and then classified using the Hierarchical Temporal Memory (HTM) classifier, and k-Nearest Neighbor (kNN) classifier.

At first, the HTM classifier did not learn the data patterns, resulting in poor performance. When learning runs, the classifier learns data patterns and makes reasonably accurate predictions. After learning for a while, the classifier can adapt to new patterns and make better predictions.

After running the experiment and achieving the results for specific data, we evaluated performance evaluation metrics in machine learning and statistics for this experiment. The evaluation performances of the HTM classifier with accuracy, precision, recall, F_Score, MSE, MAE, and RMSE are shown in Table 2. The rate of the performance metric is proportional to the percentage of the correct classification patterns applied to the dataset. We focused on solving hierarchical classification problems rather than specific algorithm problems due to the limited options for performance comparisons. We used the kNN algorithm in correlation with our validation and Farming-Pack mocap datasets.

The HTM can predict the data value in the next step based on previously stored patterns based on the structure of the HTM classifier system. When new data arrives and is processed, HTM compares it to the predicted value and calculates the difference.

Considering HTM as a continuous learning theory, there is no training, validation, or test set against the input data [105]. Data are learned and predicted continuously across all of the datasets.

Table 2. Performance metrics evaluation for the validation dataset and Farming-Pack mocap dataset.

Metrics	HTM ¹		kNN ²	
	Validation Dataset	Farming-Pack Mocap Dataset	Validation Dataset	Farming-Pack Mocap Dataset
Accuracy (%)	88.00	54.00	13.00	40.00
Precision	0.99	0.97	0.14	0.16
Recall	0.04	0.50	0.13	0.40
F_Score	0.09	0.66	0.12	0.23
MSE ³ (%)	5.10	0.06	30.00	25.00
MAE ⁴ (%)	0.19	3.24	13.00	3.39
RMSE ⁵ (%)	0.38	1.51	5.47	5.00

¹ Hierarchical Temporal Memory (HTM); ² k-Nearest Neighbor (kNN); ³ Mean Squared Error (MSE); ⁴ Mean Absolute Error (MAE); ⁵ Root Mean Squared Error (RMSE)

6. Discussion

From our experiment, HTM can achieve 88.00% accuracy, precision of 0.99, recall of 0.04, and a F_Score of 0.09 for the validation dataset; it can achieve 54.00% accuracy, precision of 0.97, recall of 0.50, and a F_Score of 0.66 for the Farming-Pack mocap dataset.

Moreover, for the lower value of the average, we achieve an MSE = 5.10, MAE = 0.19, and RMSE = 0.38 for the validation dataset, and MSE = 0.06, MAE = 3.24, and RMSE = 1.51 for the Farming-Pack mocap dataset, which implies the higher performance of the prediction model.

When compared with other methods proposed in Tables 3 and 4, our method achieves the following results: on the Human3.6M dataset, the RRNN method achieves an average of 0.97 and 0.23 for standard deviation, while CSS results in an average of 0.77 and 0.21 for standard deviation. The SkelNeT method achieves an average of 0.76 and 0.21 for standard deviation and our Skel-TNet method achieves an average of 0.73 and 0.21 for standard deviation. With the CMU mocap dataset, the CSS method results in an average of 0.61 and standard deviation of 0.18, the SkelNet method achieves an average of 0.60, and a standard deviation of 0.21, and our Skel-TNet method results in an average of 0.55 and standard deviation of 0.21.

Table 3. The evaluation results of our proposed method compared with those of similar methods applied to the Mocap databases.

By	Dataset	Method	Results
[39]	Human3.6M	RRNN	Average/standard deviation 0.97/0.23
		CSS	0.77/0.21
		SkelNeT	0.76/0.21
		Skel-TNet	0.73/0.21
	CMU mocap dataset	CSS	Average/standard deviation 0.61/0.18
		SkelNeT	0.60/0.21
		Skel-TNet	0.55/0.21
[40]	CMU mocap dataset	Data Point	Classification rate 0.76
		PCA	0.73
		DSAE	0.72
		S-TE	0.78
		C-TE	0.78
		H-TE	0.77
[41]	Training Database	HMM	Accuracy Stand = 88.67 Walk = 86.20 Run = 82.70 Jump = 76.30 Fall = 72.03 Lie = 86.23 Sit = 92.70

The Carnegie Mellon University (CMU) mocap dataset was also evaluated by [40], who achieved classification rate results using various methods, i.e., data point = 0.76, PCA = 0.73, Deep Sparse Autoencoder (DSAE) = 0.72, S-TE = 0.78, C-TE = 0.78, and H-TE = 0.77. Another study by [41] used the HMM method with a training database and achieved accuracy results for activities such as stand = 88.67, walk = 86.20, run = 82.70, jump = 76.30, fall = 72.03, lie = 86.23, and sit = 92.70.

In the time-series dataset, [42] applied the xGBoost method on a tourism dataset and achieved MASE = 0.92, RMSSE = 1.16, and AMSE = 0.49. They also applied the random forest method and achieved MASE = 0.45, RMSSE = 0.67, and AMSE = 0.30. [43] applied the DLSTM method to the Brazilian electrical power generation dataset and achieved RMSE levels of 0 = 0.60, 1 = 1.02, and 2 = 2.91. They also applied the method to the Australian visitor nights or domestic tourism dataset and achieved RMSE levels of 0 = 2.27, 1 = 5.26, 2 = 6.34, and 3 = 8.07.

Table 4. Comparison evaluation results with a similar method applied on the time series and enzyme dataset.

By	Dataset	Method	Results
[42]	Tourism	xGBoost	Metrics
			MASE = 0.92
	Sales	Random Forest	RMSSE = 1.16
			AMSE = 0.49
[43]	Brazilian electrical power generation	DLSTM	Metrics
			MASE = 0.45
			RMSSE = 0.67
			AMSE = 0.30
	Australian visitor nights of domestic tourism	DLSTM	Metrics: RMSE
			level 0 = 0.60
			level 1 = 1.02
			level 2 = 2.91
[44]	Tourism	Flat (Leaves)	Metrics: accuracy
			61.33
			Flat (all levels)
			87.80
	Enzyme protein families	Top-Down	87.80
			Big-Bang
			91.13
			Metrics: accuracy
		Flat (leaves)	82.73
			Flat (all levels)
			89.78
			Top-Down
			89.78
			Big-Bang
			96.36

The study by [44] used hierarchical classification on two datasets. For the tourism dataset, they achieved the following accuracies: flat (leaves) = 61.33, flat (all levels) = 87.80, top-down = 87.80, and big-bang = 91.13. For the enzyme protein families dataset, they achieved the following accuracies: flat (leaves) = 62.73, flat (all levels) = 89.78, top-down = 89.78, and big-bang = 96.36.

Overall, this research represents the most dependable implementation with the optimum results. Wearable devices are used as IoT subsystem technology to collect motion sensor data, which are then processed locally and transferred to the cloud via the long-range (LoRa) network for decision-making.

Our statistical results confirm that our proposed method is both feasible and effective in addressing the constraints of time series datasets, and it is suitable for implementation in real rural farming environments for optimal solutions.

7. Conclusions

With the demand for farming activity monitoring in rural farming environments, this paper presents a classification prediction method for hierarchical temporal memory using the quaternion feature for farming safety activity monitoring. The obtained results support the proposed method's ability to classify multiple activity classes. With 88.00% accuracy, precision of 0.99, recall of 0.04, F_Score of 0.09, MSE = 5.10, MAE = 0.19, and RMSE = 0.38 for the validation dataset, and 54.00% accuracy, precision of 0.97, recall of 0.50, F_Score of 0.66, MSE = 0.06, MAE = 3.24, and RMSE = 1.51 for the Farming-Pack mocap dataset, which implies the higher performance of a prediction model. Nevertheless, it would be exciting to look into the possibility of combining, extending, and evaluating other datasets and machine learning methods, which could be an interesting research direction. Examining computational complexity will be another future direction to maximize the proposed method further.

Author Contributions: Conceptualization, Y.A., M.K., and J.-S.L.; data curation, G.S.M.; formal analysis, Y.A.; investigation, Y.A. and G.S.M.; methodology, Y.A. and M.K.; project administration,

Y.A.; resources, G.S.M.; supervision, M.K. and J.-S.L.; validation, M.K. and J.-S.L.; visualization, Y.A.; writing—original draft, Y.A.; writing—review and editing, M.K. and J.-S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by a collaborative research project between the Kyushu Institute of Technology (Kyutech) and the National Taiwan University of Science and Technology (Taiwan-Tech).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors gratefully acknowledge the support from the Kyushu Institute of Technology—National Taiwan University of Science and Technology Joint Research Program, under grant Kyutech-NTUST-111-04.

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

References

1. Martin, G.; Martin-Clouaire, R.; Duru, M.; Farming system design to feed the changing world: A review. *Agron. Sustain. Dev.* **2013**, *33*, 131–149. [CrossRef]
2. United Nations Department of Economic and Social Affairs. Growing at a Slower Pace, World Population is Expected to Reach 9.7 Billion in 2050 and Could Peak at Nearly 11 Billion Around 2100. Available online: <https://www.un.org/development/desa/en/news/population/world-population-prospects-2019.html> (accessed on 1 June 2021).
3. Alexandratos, N.; Bruinsma, J. *World Agriculture towards 2030/2050: The 2012 Revision*; ESA Working Paper No.12-03.; Agricultural Development Economics Division, Food and Agriculture Organization of the United Nations: Rome, Italy, 2012.
4. Food.; of the United Nations, A.O. *How to Feed the World 2050: Proceedings of a Technical Meeting of Experts, Rome, Italy, 24–26 June 2009*; Food and Agriculture Organization of the United Nations (FAO): Rome, Italy, 2009.
5. Achim Walter, Robert Finger, R.H.; Buchmann, N. Smart farming is key to developing sustainable agriculture. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 6148–6150. [CrossRef]
6. Mohamed, E.S.; Belal, A.A.; Abd-Elmabod, S.K.; El-Shirbeny, M.A.; Gad, A.; Zahran, M.B. Smart farming for improving agricultural management. *Egypt. J. Remote. Sens. Space Sci.* **2021**, *24*, 971–981. [CrossRef]
7. Rose, D.C.; Chilvers, J. Agriculture 4.0: Broadening Responsible Innovation in an Era of Smart Farming. *Front. Sustain. Food Syst.* **2018**, *2*, 87. [CrossRef]
8. Kumar, D.; Shen, K.; Case, B.; Garg, D.; Alperovich, G.; Kuznetsov, D.; Kuznetsov, D.; Gupta, R.; Durumeric, Z. All Things Considered: An Analysis of IoT Devices on Home Networks. In Proceedings of the 28th USENIX Conference on Security Symposium, Santa Clara, CA, USA, 14–16 August 2019; USENIX Association: Berkeley, CA, USA, 2019; pp. 1169–1185.
9. Swaroop, K.N.; Chandu, K.; Gorreputu, R.; Deb, S. A health monitoring system for vital signs using IoT. *Internet Things* **2019**, *5*, 116–129. [CrossRef]
10. Saminathan, S.; Geetha, K. A Survey on Health Care Monitoring System Using IoT. *Int. J. Pure Appl. Math.* **2017**, *117*, 249–254.
11. Saranya, M.; Preethi, R.; adn Dr.S.Veena, M.R. A Survey on Health Monitoring System by using IOT. *Int. J. Res. Appl. Sci. Eng. Technol.* **2018**, *6*, 778–782. [CrossRef]
12. Fukatsu, T.; Nanseki, T. *Farm Operation Monitoring System with Wearable Sensor Devices Including RFID*; IntechOpen: London, UK, 2011.
13. Health and Safety Executive. *Fatal Injuries in Agriculture, Forestry and Fishing in Great Britain (1 April 2021 to 31 March 2022)*; Health and Safety Executive: London, UK, 2022.
14. Thibaud, M.; Chi, H.; Zhou, W.; Piramuthu, S. Internet of Things (IoT) in high-risk Environment, Health and Safety (EHS) industries: A comprehensive review. *Decis. Support Syst.* **2018**, *108*, 79–95. [CrossRef]
15. Bergen, G.; Chen, L.H.; Warner, M.; Fingerhut, L.A. *Injury in the United States: 2007 Chartbook (March 2008)*; National Center for Health Statistics: Hyattsville, MD, USA, 2008.
16. World Health Organization. Falls. Available online: <https://www.who.int/news-room/fact-sheets/detail/falls> (accessed on 1 June 2021).
17. Fukatsu, T.; Nanseki, T. Monitoring system for farming operations with wearable devices utilized sensor networks. *Sensors* **2009**, *9*, 6171–6184. [CrossRef] [PubMed]
18. Delahoz, Y.S.; Labrador, M.A. Survey on Fall Detection and Fall Prevention Using Wearable and External Sensors. *Sensors* **2014**, *14*, 19806–19842. [CrossRef] [PubMed]
19. Rungnapakan, T.; Chintakovid, T.; Wuttidittachotti, P. Fall Detection Using Accelerometer, Gyroscope & Impact Force Calculation on Android Smartphones. In Proceedings of the CHIU XiD '18: Proceedings of the 4th International Conference on Human-Computer Interaction and User Experience in, Indonesia, Yogyakarta, Indonesia, 23–29 March 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 49–53.
20. Lim, D.; Park, C.; Kim, N.H.; Kim, S.H.; Yu, Y.S. Fall-Detection Algorithm Using 3-Axis Acceleration: Combination with Simple Threshold and Hidden Markov Model. *J. Appl. Math.* **2014**, *2014*, 896030. [CrossRef]

21. EuclideanSpace. EuclideanSpace—Maths-Quaternion. Available online: <https://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/index.htm> (accessed on 1 June 2021).
22. Yannick Millot, P.P.M. Active and passive rotations with Euler angles in NMR. *Concepts Magn. Reson* **2022**, *40A*, 215–252. [CrossRef]
23. Allgeuer, P.; Behnke, S. Fused Angles and the Deficiencies of Euler Angles. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 5109–5116.
24. Janota, A.; Šimák, V.; Nemec, D.; Hrbček, J. Improving the precision and speed of Euler angles computation from low-cost rotation sensor data. *Sensors* **2015**, *15*, 7016–7039. [CrossRef]
25. Tastan, M. IoT Based Wearable Smart Health Monitoring System. *Celal Bayar Univ. Fen Bilim. Derg.* **2018**, *14*, 343–350.
26. Wcislik, M.; Pozoga, M.; Smerdzynski, P. Wireless Health Monitoring System. *IFAC-PapersOnLine* **2015**, *48*, 312–317. .. [CrossRef]
27. Yacchirema, D.; de Puga, J.S.; Palau, C.; Esteve, M. Fall detection system for elderly people using IoT and Big Data. *Procedia Comput. Sci.* **2018**, *130*, 603–610.
28. Challis, J.H. Quaternions as a solution to determining the angular kinematics of human movement. *BMC Biomed. Eng.* **2020**, *2*, 5. [CrossRef]
29. Rong, G.; Zheng, Y.; Sawan, M. Energy Solutions for Wearable Sensors: A Review. *Sensors* **2021**, *21*, 3806. [CrossRef]
30. Wu, F.; Wu, T.; Yuce, M.R. An Internet-of-Things (IoT) Network System for Connected Safety and Health Monitoring Applications. *Sensors* **2019**, *19*, 21. [CrossRef] [PubMed]
31. Samad Barri Khojasteh, José R Villar, C.C.V.M.G.E.d.I.C. Improving Fall Detection Using an On-Wrist Wearable Accelerometer. *Sensors* **2018**, *18*, 1350. [CrossRef] [PubMed]
32. Wu, F.; Zhao, H.; Zhao, Y.; Zhong, H. Development of a Wearable-Sensor-Based Fall Detection System. *Int. J. Telemed. Appl.* **2015**, *2015*, e576364. [CrossRef] [PubMed]
33. United Nations. *The Sustainable Development Goals Report 2022*; United Nations: New York, NY, USA, 2022.
34. Banerjee, A.; Chakraborty, C.; Kumar, A.; Biswas, D. Chapter 5—Emerging trends in IoT and big data analytics for biomedical and health care technologies. *Handbook of Data Science Approaches for Biomedical Engineering*; Academic Press: Cambridge, MA, USA, 2020; pp. 121–152. [CrossRef]
35. Pawar, M.V.; Pawar, P.; Pawar, A.M. Chapter 2—HealthWare telemedicine technology (HWT) evolution map for healthcare. *Wearable Telemedicine Technology for the Healthcare Industry*; Academic Press: Cambridge, MA, USA, 2022; pp. 17–32. [CrossRef]
36. Ranganathan Chandrasekaran, V.K.; Moustakas, E. Patterns of Use and Key Predictors for the Use of Wearable Health Care Devices by US Adults: Insights from a National Survey. *J. Med. Internet Res.* **2020**, *22*, e22443. [CrossRef]
37. Rezayi, S. Chapter 5—Controlling vital signs of patients in emergencies by wearable smart sensors. *Wearable Telemedicine Technology for the Healthcare Industry*; Academic Press: Cambridge, MA, USA, 2022; pp. 71–86. [CrossRef]
38. Perez-Pozuelo, I.; Spathis, D.; Clifton, E.A.; Mascolo, C. Chapter 3—Wearables, smartphones, and artificial intelligence for digital phenotyping and health. *Digital Health*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 33–54. . [CrossRef]
39. Guo, X.; Choi, J. Human Motion Prediction via Learning Local Structure Representations and Temporal Dependencies. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 2580–2587. [CrossRef]
40. Bütetage, J.; Black, M.J.; Kragic, D.; Kjellström, H. Deep Representation Learning for Human Motion Prediction and Classification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1591–1599.
41. Hai, P.T.; Kha, H.H. An efficient star skeleton extraction for human action recognition using hidden Markov models. In Proceedings of the 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), Novotel, Ha Long, Vietnam, 27–29 July 2016; pp. 351–356. [CrossRef]
42. Spiliotis, E.; Abolghasemi, M.; Hyndman, R.J.; Petropoulos, F.; Assimakopoulos, V. Hierarchical forecast reconciliation with machine learning. *Appl. Soft Comput.* **2021**, *112*, 107756. [CrossRef]
43. Sagheer, A.; Hamdoun, H.; Youness, H. Deep LSTM-Based Transfer Learning Approach for Coherent Forecasts in Hierarchical Time Series. *Sensors* **2021**, *21*, 4379. [CrossRef] [PubMed]
44. Costa, E.P.; Lorena, A.C.; Carvalho, A.C.P.L.F.; Freitas, A.A.; Holden, N. Comparing Several Approaches for Hierarchical Classification of Proteins with Decision Trees. In *Advances in Bioinformatics and Computational Biology: Second Brazilian Symposium on Bioinformatics, Proceedings of the Advances in Bioinformatics and Computational Biology, Angra dos Reis, Brazil, 29–31 August 2007*; Sagot, M.F., Walter, M.E.M.T., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 126–137.
45. Xiu, Y.; Li, J.; Wang, H.; Fang, Y.; Lu, C. Pose Flow: Efficient Online Pose Tracking. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3–6, 2018*; BMVA Press: London, UK, 2018; p. 53.
46. Fang, H.; Xie, S.; Tai, Y.; Lu, C. RMPE: Regional Multi-person Pose Estimation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE Computer Society: Los Alamitos, CA, USA, 2017; pp. 2353–2362. [CrossRef]
47. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
48. Martinez, J.; Hossain, R.; Romero, J.; Little, J.J. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE Computer Society: Los Alamitos, CA, USA, 2017; pp. 2659–2668. [CrossRef]

49. Cascade Classification—Opencv 2.4.13.7 Documentation. Available online: https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html?highlight=cascadeclassifier#cascadeclassifier (accessed on 1 June 2021).
50. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. 1.
51. Obukhov, A. Chapter 33—Haar Classifiers for Object Detection with CUDA. In *GPU Computing Gems Emerald Edition*; Hwu, W.-m.W., Ed.; Applications of GPU Computing Series; Morgan Kaufmann: Boston, MA, USA, 2011; pp. 517–544. [\[CrossRef\]](#)
52. Monakhov, V.; Thambawita, V.; Halvorsen, P.; Riegler, M.A. GridHTM: Grid-Based Hierarchical Temporal Memory for Anomaly Detection in Videos. *Sensors* **2023**, *23*, 2087. [\[CrossRef\]](#) [\[PubMed\]](#)
53. Luo, J.; Tjahjadi, T. Gait Recognition and Understanding Based on Hierarchical Temporal Memory Using 3D Gait Semantic Folding. *Sensors* **2020**, *20*, 1646. [\[CrossRef\]](#) [\[PubMed\]](#)
54. Zhang, K.; Zhao, F.; Luo, S.; Xin, Y.; Zhu, H.; Chen, Y. Online Intrusion Scenario Discovery and Prediction Based on Hierarchical Temporal Memory (HTM). *Appl. Sci.* **2020**, *10*, 2596. [\[CrossRef\]](#)
55. Nguyen, T.V.; Pham, K.V.; Min, K.S. Hybrid Circuit of Memristor and Complementary Metal-Oxide-Semiconductor for Defect-Tolerant Spatial Pooling with Boost-Factor Adjustment. *Materials* **2019**, *12*, 2122. [\[CrossRef\]](#)
56. Nguyen, T.V.; Pham, K.V.; Min, K.S. Memristor-CMOS Hybrid Circuit for Temporal-Pooling of Sensory and Hippocampal Responses of Cortical Neurons. *Materials* **2019**, *12*, 875. [\[CrossRef\]](#) [\[PubMed\]](#)
57. Perea-Moreno, A.J.; Aguilera-Ureña, M.J.; Meroño-De Larriva, J.E.; Manzano-Agugliaro, F. Assessment of the Potential of UAV Video Image Analysis for Planning Irrigation Needs of Golf Courses. *Water* **2016**, *8*, 584. [\[CrossRef\]](#)
58. Ding, N.; Gao, H.; Bu, H.; Ma, H.; Si, H. Multivariate-Time-Series-Driven Real-time Anomaly Detection Based on Bayesian Network. *Sensors* **2018**, *18*, 3367. [\[CrossRef\]](#) [\[PubMed\]](#)
59. Van-Horenbeke, F.A.; Peer, A. NILRNN: A Neocortex-Inspired Locally Recurrent Neural Network for Unsupervised Feature Learning in Sequential Data. *Cogn. Comput.* **2023**.
60. Dzhevlikian, E.; Latyshev, A.; Kuderov, P.; Panov, A.I. Hierarchical intrinsically motivated agent planning behavior with dreaming in grid environments. *Brain Inform.* **2022**, *9*, 1–28. [\[CrossRef\]](#) [\[PubMed\]](#)
61. Dobric, D.; Pech, A.; Ghita, B.; Wennekers, T. On the Importance of the Newborn Stage When Learning Patterns with the Spatial Pooler. *SN Comput. Sci.* **2022**, *3*, 179. [\[CrossRef\]](#)
62. Chakraborty, P.; Bhunia, S. BINGO: Brain-inspired learning memory. *Neural Comput. Appl.* **2022**, *34*, 3223–3247. [\[CrossRef\]](#)
63. Ding, C.; Zhao, J.; Sun, S. Concept Drift Adaptation for Time Series Anomaly Detection via Transformer. *Neural Process. Lett.* **2022**.
64. Sin Yong Teng, Vítězslav Máša, M.T.M.V.H.L.L.P.S. Waste-to-energy forecasting and real-time optimization: An anomaly-aware approach. *Renew. Energy* **2022**, *181*, 142–155. [\[CrossRef\]](#)
65. Ivan Rodkin, Petr Kuderov, A.I.P. Stability and Similarity Detection for the Biologically Inspired Temporal Pooler Algorithms. *Procedia Comput. Sci.* **2022**, *213*, 570–579. [\[CrossRef\]](#)
66. Melnykova, N.; Kulievych, R.; Vycluk, Y.; Melnykova, K.; Melnykov, V. Anomalies Detecting in Medical Metrics Using Machine Learning Tools. *Procedia Comput. Sci.* **2022**, *198*, 718–723. [\[CrossRef\]](#)
67. Rodríguez-Flores, T.C.; Palomo-Briones, G.A.; Robles, F.; Ramos, F. Proposal for a computational model of incentive memory. *Cogn. Syst. Res.* **2023**, *77*, 153–173. [\[CrossRef\]](#)
68. George, A.M.; Dey, S.; Banerjee, D.; Mukherjee, A.; Suri, M. Online time-series forecasting using spiking reservoir. *Neurocomputing* **2023**, *518*, 82–94. [\[CrossRef\]](#)
69. Thill, M.; Konen, W.; Wang, H.; Bäck, T. Temporal convolutional autoencoder for unsupervised anomaly detection in time series. *Appl. Soft Comput.* **2021**, *112*, 107751. [\[CrossRef\]](#)
70. Numenta. Numenta Releases Grok for IT Analytics on AWS. Available online: <https://numenta.com/press/2014/03/25/numenta-releases-grok-for-it-analytics-on-aws/> (accessed on 1 June 2021).
71. Numenta. Detect Anomalies in Publicly Traded Stocks Using Trading and Twitter Data. Available online: <https://numenta.com/assets/pdf/apps/htmforstocks.pdf> (accessed on 1 June 2021).
72. Numenta. Rogue Behavior Detection. Available online: <https://numenta.com/assets/pdf/whitepapers/Rogue%20Behavior%20Detection%20White%20Paper.pdf> (accessed on 1 June 2021).
73. Numenta. The Path to Machine Intelligence. Available online: <https://numenta.com/assets/pdf/whitepapers/Numenta%20-%20Path%20to%20Machine%20Intelligence%20White%20Paper.pdf> (accessed on 1 June 2021).
74. Numenta. Geospatial Tracking. Available online: <https://numenta.com/assets/pdf/whitepapers/Geospatial%20Tracking%20White%20Paper.pdf> (accessed on 1 June 2021).
75. Bifet, A.; Gavalda, R.; Holmes, G.; Pfahringer, B. *Machine Learning for Data Streams with Practical Examples in MOA*; MIT Press: Cambridge, MA, USA, 2018. Available online: <https://moa.cms.waikato.ac.nz/book/> (accessed on 1 June 2021).
76. Pérez-Ortiz, M.; Jiménez-Fernández, S.; Gutiérrez, P.A.; Alexandre, E.; Hervás-Martínez, C.; Salcedo-Sanz, S. A Review of Classification Problems and Algorithms in Renewable Energy Applications. *Energies* **2016**, *9*, 607. [\[CrossRef\]](#)
77. Cotofrei, P.; Stoffel, K. Classification Rules + Time = Temporal Rules. In *Proceedings of the Computational Science, ICCS 2002, Amsterdam, The Netherlands, 21–24 April 2002*; Sloot, P.M.A., Hoekstra, A.G., Tan, C.J.K., Dongarra, J.J., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 572–581.
78. Kadous, M.W. Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series. Ph.D. Thesis, School of Computer Science and Engineering, The University of New South Wales, Kensington, Australia, 2002.

79. Kadous, M.W.; Sammut, C. Classification of Multivariate Time Series and Structured Data Using Constructive Induction. *Mach. Learn.* **2005**, *58*, 179–216. [\[CrossRef\]](#)
80. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; Association for Computing Machinery: New York, NY, USA, 2006; pp. 369–376. [\[CrossRef\]](#)
81. Hawkins, J.; Blakeslee, S. *On Intelligence*; Times Books; Henry Holt and Company: New York, NY, USA, 2008.
82. Hawkins, J.; Lewis, M.; Klukas, M.; Purdy, S.; Ahmad, S. A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex. *Front. Neural Circuits* **2019**, *12*, 121. [\[CrossRef\]](#) [\[PubMed\]](#)
83. Hawkins, J.; Ahmad, S.; Cui, Y. A Theory of How Columns in the Neocortex Enable Learning the Structure of the World. *Front. Neural Circuits* **2017**, *11*, 81. [\[CrossRef\]](#) [\[PubMed\]](#)
84. Lewis, M.; Purdy, S.; Ahmad, S.; Hawkins, J. Locations in the Neocortex: A Theory of Sensorimotor Object Recognition Using Cortical Grid Cells. *Front. Neural Circuits* **2019**, *13*, 22. [\[CrossRef\]](#) [\[PubMed\]](#)
85. Stefanowski, J.; Brzezinski, D. Stream Classification. In *Encyclopedia of Machine Learning and Data Mining*; Sammut, C.; Webb, G.I., Eds.; Springer: Boston, MA, USA, 2016; pp. 1–9. [\[CrossRef\]](#)
86. Cerri, R.; Barros, R.C.; de Carvalho, A.C. Hierarchical multi-label classification using local neural networks. *J. Comput. Syst. Sci.* **2014**, *80*, 39–56. [\[CrossRef\]](#)
87. Damen, D.; Doughty, H.; Farinella, G.M.; Fidler, S.; Furnari, A.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; et al. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
88. Damen, D.; Doughty, H.; Farinella, G.; Fidler, S.; Furnari, A.; Kazakos, E.; Moltisanti, D.; Munro, J.; Perrett, T.; Price, W.; et al. The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4125–4141. [\[CrossRef\]](#)
89. Adobe Mixamo-Farming Pack. Available online: <https://www.mixamo.com/#/?page=1&query=farming+pack&type=Motion%2CMotionPack> (accessed on 12 August 2022).
90. Abdulla, U.A.; Taylor, K.; Barlow, M.; Naqshbandi, K.Z. Measuring Walking and Running Cadence Using Magnetometers. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, VIC, Australia, 16–18 July 2013; pp. 1458–1462. [\[CrossRef\]](#)
91. Gjoreski, H.; Lustrek, M.; Gams, M. Accelerometer Placement for Posture Recognition and Fall Detection. In Proceedings of the 2011 Seventh International Conference on Intelligent Environments, Nottingham, UK, 25–28 July 2011; pp. 47–54. [\[CrossRef\]](#)
92. Cleland, I.; Kikhia, B.; Nugent, C.; Boytsov, A.; Hallberg, J.; Synnes, K.; McClean, S.; Finlay, D. Optimal placement of accelerometers for the detection of everyday activities. *Sensors* **2013**, *13*, 9183–9200. [\[CrossRef\]](#) [\[PubMed\]](#)
93. Pannurat, N.; Thiemjarus, S.; Nantajeewarawat, E.; Anantavasilp, I. Analysis of Optimal Sensor Positions for Activity Classification and Application on a Different Data Collection Scenario. *Sensors* **2019**, *17*, 774. [\[CrossRef\]](#) [\[PubMed\]](#)
94. Nguyen, N.D.; Bui, D.T.; Truong, P.H.; Jeong, G.M. Position-Based Feature Selection for Body Sensors regarding Daily Living Activity Recognition. *J. Sensors* **2018**, *2018*, 9762098. [\[CrossRef\]](#)
95. Arvidsson, D.; Fridolfsson, J.; Börjesson, M. Measurement of physical activity in clinical practice using accelerometers. *J. Intern. Med.* **2019**, *286*, 137–153. [\[CrossRef\]](#) [\[PubMed\]](#)
96. Sony. Mocopi. Available online: <https://www.sony.jp/mocopi/> (accessed on 2 March 2023).
97. HTC Corporation. Vive Tracker. Available online: <https://www.vive.com/jp/accessory/vive-tracker/> (accessed on 2 March 2023).
98. Unity Documentation. Unity Documentation: Quaternion 2021. Available online: <https://docs.unity3d.com/ScriptReference/Quaternion-w.html> (accessed on 1 June 2021).
99. OpenGL. OpenGL-Tutorial: Tutorial 17: Rotation. Available online: <https://www.opengl-tutorial.org/intermediate-tutorials/tutorial-17-quaternions/> (accessed on 1 June 2021).
100. OpenGL. Wiki SecondLife: Quaternion. Available online: <https://wiki.secondlife.com/wiki/Quaternion> (accessed on 1 June 2021).
101. Jia, Y.-B. Quaternions and Rotations. Available online: <https://web.cs.iastate.edu/cs577/handouts/quaternion.pdf> (accessed on 1 June 2021).
102. Wikipedia. Quaternion. Available online: <https://en.wikipedia.org/wiki/Quaternion> (accessed on 1 June 2021).
103. Kou, K.I.; Xia, Y.H. Linear Quaternion Differential Equations: Basic Theory and Fundamental Results. *Stud. Appl. Math.* **2018**, *141*, 3–45. [\[CrossRef\]](#)
104. Wu, J.; Zeng, W.; Yan, F. Hierarchical Temporal Memory method for time-series-based anomaly detection. *Neurocomputing* **2018**, *273*, 535–546. [\[CrossRef\]](#)
105. Sousa, R.; Lima, T.; Abelha, A.; Machado, J. Hierarchical Temporal Memory Theory Approach to Stock Market Time Series Forecasting. *Electronics* **2021**, *10*, 1630. [\[CrossRef\]](#)

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.