Discover Internet of Things

Discover

Research

"Chatty Devices" and edge-based activity classification

Mike Lakoju[1] · Amir Javed[2] · Omer Rana[2] · Pete Burnap[2] · Samuelson T. Atiba[2] · Soumaya Cherkaoui[3]

Received: 16 November 2020 / Accepted: 24 December 2020

Abstract
With increasing automation of manufacturing processes (focusing on technologies such as robotics and human-robot interaction), there is a realisation that the manufacturing process and the artefacts/products it produces can be better connected post-production. Built on this requirement, a "chatty" factory involves creating products which are able to send data back to the manufacturing/production environment as they are used, whilst still ensuring user privacy. The intended use of a product during design phase may different significantly from actual usage. Understanding how this data can be used to support continuous product refinement, and how the manufacturing process can be dynamically adapted based on the availability of this data provides a number of opportunities. We describe how data collected on product use can be used to: (i) classify product use; (ii) associate a label with product use using unsupervised learning—making use of edge-based analytics; (iii) transmission of this data to a cloud environment where labels can be compared across different products of the same type. Federated learning strategies are used on edge devices to ensure that any data captured from a product can be analysed locally (ensuring data privacy).

Keywords Product use activity classification · IoT · Chatty factories · Federated learning · Edge computing

1 Introduction

Increasing availability and reduction in size of sensors and actuators has enabled integration of these devices within a number of physical objects. The use of sensors within built environments has existed for a number of years, enabling control systems (e.g. Building Energy Management Systems) to process and undertake control actions based on data obtained from such sensors. The term "Industry 4.0" has drawn significant attention over recent years, providing an ecosystem of sensing, actuation and analytics to support product development and process automation. A key theme is the transformation of a production environment from manual (but machine supported) design and manufacturing to digital design and manufacturing [1]—often also the basis for significant recent interest in developing "digital twins". Within this paradigm, interconnected computer systems, intelligent devices and smart materials communicate with one another while interacting with the environment with little to no human involvement [2]. The manufacturing industry, in particular, has been significantly impacted by the use of such sensing and actuation technologies [3]. In the typical manufacturing system, taking a product to market requires multiple discrete and highly specialised activities across research, design and manufacturing disciplines.

The *Chatty Factories* vision for the manufacturing factory of the future is to take these traditionally discrete activities and collapse them into one seamless process that is capable of continuous real-time product refinement at an industrial scale. The Industrial Internet of Things (IIoT) offers the opportunity to draw new insights through real-time data harvested

✉ Mike Lakoju, mLakoju@cardiffmet.ac.uk | [1]Cardiff School of Technologies, Cardiff Metropolitan University, Western Avenue, Cardiff CF5 2YB, UK. [2]School of Computer Science & Informatics, Cardiff University, 5 The Parade, Cardiff CF24 3AA, UK. [3]Interlab Lab., Dept. of Electical & Comp. Engineering, Université de Sherbrooke, Sherbrooke, Quebec J1K 2R1, Canada.

Published online: 24 February 2021

from sensors embedded in products. "Chatty Factories" seeks to leverage product use data from "Chatty" (smart devices that are able to communicate their usage back to the factory) devices and equip design engineers and manufacturers within the Industry 4.0 paradigm with the ability to understand how the products are being used *in the wild*, providing the basis for new forms of creative design and leading to tailor-built manufacturing processes.

A key challenge, as outlined in this work, is to ensure that product use data is collected whilst still preserving user privacy. Placing IoT-enabled data driven systems at the core of design and manufacturing processes provides a number of opportunities for refining product design and aligning this more closely with its actual (vs. estimated/ expected) use. We focus on collecting data from IoT-enabled sensors embedded in products during real-time use by consumers, to understand how that data might be immediately transferred into usable information to inform design. We also consider what characteristics of the manufacturing environment might optimise the response to such data in a privacy-preserving manner. However, achieving continuous product refinement in response to real-time data on product use requires: data driven systems that provide an auditable, secure flow of information between all operations inside and outside the factory. By implication, this raises concerns around privacy, security and scale of data. Recent approaches to achieve this include Federated Learning, which can be used to train a globally shared model by exploiting a large amount of user-generated data samples on sensorized/chatty devices while preventing data leakage [4]. A *surrogate* model is developed on each device, which can then be combined at a cloud data centre. Product use activity classification is carried out on data at each device at the network edge, using federated learning approaches. This is achieved through the use of an Autoencoder at the edge to identify unknown product use activities, and then uses unsupervised machine learning methods to identify repeated patterns/ anomalous activities, to distinguishing between noise and actual product use. Furthermore, we use entangled ethnography to create labels for these newly identified activities. The new use activity labels created are pushed to the cloud which aids federated learning and model correction from a collective of chatty devices while maintaining appropriate privacy and trust. This paper contributes to the broader literature on edge computing and federated learning by:

- Proposing a novel system that identifies *product-use* activities at the edge, thereby ensuring user privacy while providing for model correction;
- Demonstrating the system's capability to detect unknown *product-use* activities on chatty devices at the edge;
- Recognising previously "seen" activities in the presence of noise;
- Rigorously testing the novel system against two independent datasets (a chatty dataset collected by us and a publicly available dataset on human activity recognition).

2 The chatty factories vision

The Chatty Factories project explores the transformative potential of placing IoT-enabled data driven systems at the core of design and manufacturing processes. The aims of the research project can be summarised as [5]: (i) enabling new forms of agile engineering product development via "chatty" products that continue to relay their experiences from consumers back to designers and product engineers through the mediation provided by embedded sensors, IoT and data driven design tools; (ii) enabling the manufacturing environment to dynamically respond to changes in physical configuration and ethical re-skilling of production elements such as robots and humans by utilising theory from exopedagogy and interpretable data analytics; (iii) addressing the challenges of data volume, privacy and cybersecurity to develop an access-controlled manufacturing ecosystem in which product use data will be collated, analysed and disseminated across the factory floor, merging Operational Technology (OT) on the factory floor with Information Technology (IT) in the business. Figure 1 illustrates the key vision of Chatty Factories. With the "Chatty concept used to describe how sensors retrofitted on products in "the wild" can be used to capture their use. The chatty process flow is initiated at the Product Data from the wild phase as illustrated in Fig. 1. Acquired product use labels are subsequently transferred for data annotation and pattern detection at the Data Annotation" phase. As denoted by the Product Use Models" phase, the successful labelling of newly identified product use activities is followed by the creation of product use models through supervised machine learning algorithms assisted by designer led rules. After various analyses, new forms of data-driven design suggestions are put forward to the design team at the "New Forms of Design" phase. Insights gathered at this stage can be used to support enhancements to existing products and/ or the creation of a totally new product. As illustrated by the "Rapid Product Manufacture" phase, this design and manufacturing process is rapidly sped up by the interaction between the *design* digital twin and the *manufacturing* digital twin. The entire process runs constantly and iteratively. This

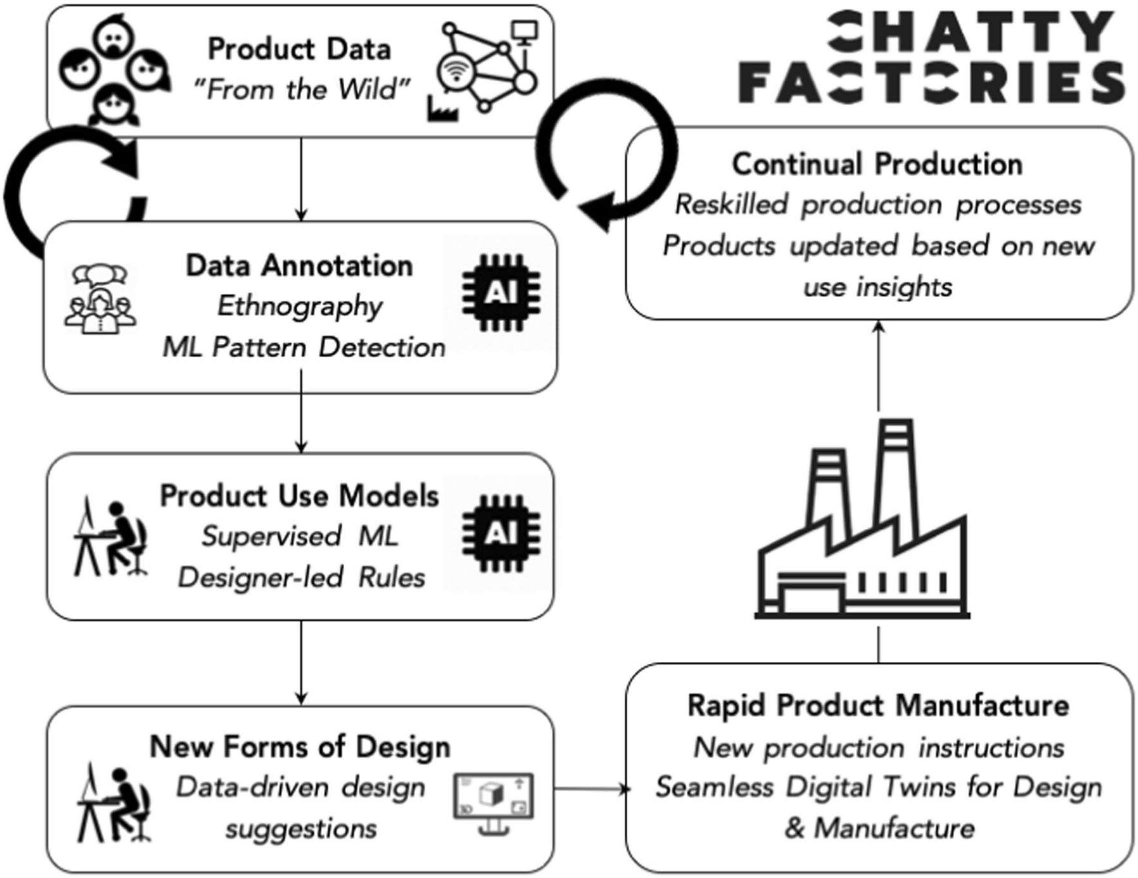

CHATTY FACTORIES
Product Data
"From the Wild"
Data Annotation
Ethnography
ML Pattern Detection
AI
Product Use Models
Supervised ML
Designer-led Rules
AI
New Forms of Design
Data-driven design suggestions
Rapid Product Manufacture
New production instructions
Seamless Digital Twins for Design & Manufacture
Continual Production
Reskilled production processes
Products updated based on new use insights

Fig. 1 The chatty factories vision

is highlighted by the "Continual Production" phase in Fig. 1. The approach proposed in this work automatically produces new activity labels (based on user behaviour) on Edge devices while protecting the privacy of the user. Furthermore, the proposed system can be adapted by analysing the *social chatter* between devices.

3 Related work

This section describes a number of related concepts reported in literature, such as classification of device use activities by human users and federated learning on user owned devices.

3.1 Activity classification

While research on product use activity recognition is still in its infancy, there has been substantial research dedicated to activity recognition and classification especially in human activity classification, smart homes, activities of daily living (ADL) and medical domains. Advancements in sensor technology have fuelled the growing interest in context-aware systems for different domain-specific applications [6–8]. This provides a better understanding of sensing and reasoning to establish a context for people's actions. Researchers have particularly considered how data from sensors can be used to investigate fall and activities of daily living (ADL) [9–11]. General research on activity recognition with sensor technology can be categorised into three general themes: wearable sensors, vision-based detection and environmental sensors [11–14].

Trabelsi et al. [15] investigated the use of an unsupervised learning method for human activity recognition, and they focused on using three accelerometers which were attached to the right thigh, left ankle and chest. Furthermore, they grounded their work on the Hidden Markov Model Regression. One drawback to their approach is that it assumes that the number of "k" activities is already known. Some Authors like Banos et al. [16] investigated the significance of signal segmentation. Their experiments utilised nine inertial sensors on different parts of the body while varying the sliding window sizes between 0.25 and 7s. By employing supervised learning models, they suggested that the most accurate results are gotten for little windows (0.25–0.5 s). Their study has nothing to do with understand product use activities. Gao, Bourke, and Nelson [17] made a comparison between single-sensor wearable systems and multi-sensor systems through five recognisers. Their focus was to identify six activities such as standing, sitting, climbing up and down, and level walking. They suggested that the single-sensor systems did not outdo the multi-sensor systems on the recognition accuracy regardless of a much higher sampling rate, more complex features, and more focused classifier.

There is also interest in vision-based detection sensors for identifying patterns using image processing techniques. Ambient sensor-based systems have the capability of tracking movements, and can be used for analysis of gait and trajectory of user [18, 19] movements. The system uses sensor readings of pressure, sound, vibrations of the floor and infrared imaging to determine and characterise movements. These approaches are still constrained by the perimeter (radius of the area) of sensor coverage.

Wearable sensors (continuously reducing in cost) are a hybrid of different types of environment sensors. Wearable devices are commonly used in sports, health applications, entertainment etc. [20, 21]. The availability of this category of sensors has led to additional uses in activity recognition along with efficient processing techniques [22, 23]. Sensors such as an accelerometer, a gyroscope, a magnetometer, a camera etc are now not only wearable but can also be retrofitted into high-value products, thus enabling an ecosystem of sophisticated systems which can aid the analysis of any information captured by them, without a restriction of being indoors or outdoors.

Li et al. [24] carried out experiments on fall detection using a sample thresholding method between a gyroscope and an accelerometer. Lee and Carlisle [25] proposed a two-thresholding method to identify movements and the different simulated falls using a smartphone and accelerometers. Human behaviour research has gained significant attention due to its relevance in pervasive and mobile computing, health, ambient assistive living, surveillance-based security and context-aware computing [7]. For example, Tabia et al. [26] identified human activities such as running and jogging by using Nave Bayes and Support Vector Machine (SVM) classifiers. Jayasinghe et al. [27] conducted experiments using a combination of clothing mounted sensors and wearable sensors for movement analysis and activity classification with a focus on running, walking, sitting and riding in a bus. By calculating correlation coefficients for each sensor pair, they suggest that even though the two data streams have some notable differences, results indicate high classification accuracy.

Moussa et al. [28] approached human activity classification from a spatial features-based method. They utilised the Scale Invariant Feature Transform (SIFT) algorithm to identify key points of interest. Furthermore, they created a Bag of Words using the K-means algorithm, which aided the assignment of each descriptor generated from the key points of interest to the nearest visual word. They also calculated the frequency histogram of the visual words. Lastly, an SVM classifier was used to determine the human activity class.

By using a wearable Shimmer device, Kerdjidj et al. [11] proposed a system for fall detection and detection of activities of daily living (ADL). Their experiment employed 17 subjects performing a set of movements. They focused on three distinct systems: the first detects the presence or absence of a fall; the second detects static or dynamic movements including a fall, and the last recognizes the fall and six other ADL activities. In order to reduce the size of transmitted data and also energy consumption, they applied a compressive sensing (CS) method, reporting an accuracy of 99.9% from their experiment. Similarly, Hui and Zhongmin [29] proposed the use of a compressed sensing method to identify activities in which a combination of acceleration data and phone placement information are used. Their experiment focused on three placements of mobile phones: trouser pocket, handbag and the hand.

3.2 Federated learning

Federated Learning (FL) is a unique implementation of a distributed machine learning approach that enables training on decentralised data residing on devices like mobile phones, tablets or any chatty device [30, 31]. FL is one instance of *pushing code to the data*, rather than data being pushed to the code which therefore supports user privacy, ownership and locality of data. A more detailed background of Federated Learning has been provided by McMahan and Ramage [32]. Likewise, Konecny et al. [33] and McMahan et al. [34] have elaborated on its theory. Federated Learning infrastructure implementation allows for either asynchronous or synchronous training algorithms. Even though literature reveals substantial successful work on deep learning using asynchronous training, e.g Dean et al. [35], more recent studies have tilted towards synchronous large batch training even in the data centers [36, 37]. McMahan et al. [30] describe a Federated Averaging algorithm which also follows a similar approach. Other studies include differential privacy by McMahan et al. [34] and Secure Aggregation by Bonawitz et al. [38] also use FL to enhance privacy. They operate on the notion of synchronization on a fixed set of devices, which allows the server-side of the learning algorithm to only consume a simple aggregate of the updates from multiple users.

4 Systems design and implementation

In this section we describe how devices on the edge equipped with a baseline autoencoder model can be used to identify known product use activities. Using a threshold-based detection mechanism, if the results of the activity cannot be classified into a known class, an unsupervised machine learning model is automatically triggered to cluster such an activity. If the outcome of the unsupervised model identifies a particular cluster which persists over time, we posit that the cluster is a new activity and not just random noise that the autoencoder is unfamiliar with. At this point, an ethnographer creates new labels for this activity cluster—enabling these clusters to to be aggregated based in similarity using federated learning mechanisms on a cloud platform. Cloud aggregation also enables model correction, enabling this new activity detection to be deployed on the edge. This flow of work is illustrated in Fig. 2.

4.1 Using federated learning

The use of federated learning is particularly suited to “Social Chatter" for two reasons. First, the data generated by a device is directly related to how the device is being used. This essentially leads to non-independent and identically distributed (non-iid) and unbalanced distributions. Second, the chatty device’s data can also happen to be privacy sensitive. In a general federated learning approach, a global model is sent to several devices. This model is then trained at each device using local data. Information about the locally trained models (e.g. gradients) are then sent from the devices back to the central server (e.g. at the cloud) where they are aggregated to form a combined enhanced global model which is sent back to the devices for subsequent rounds of training. For the edge-based activity classification, the baseline autoencoder model is used to identify known and unknown activities and an unsupervised machine learning model is used to cluster unknown activities.

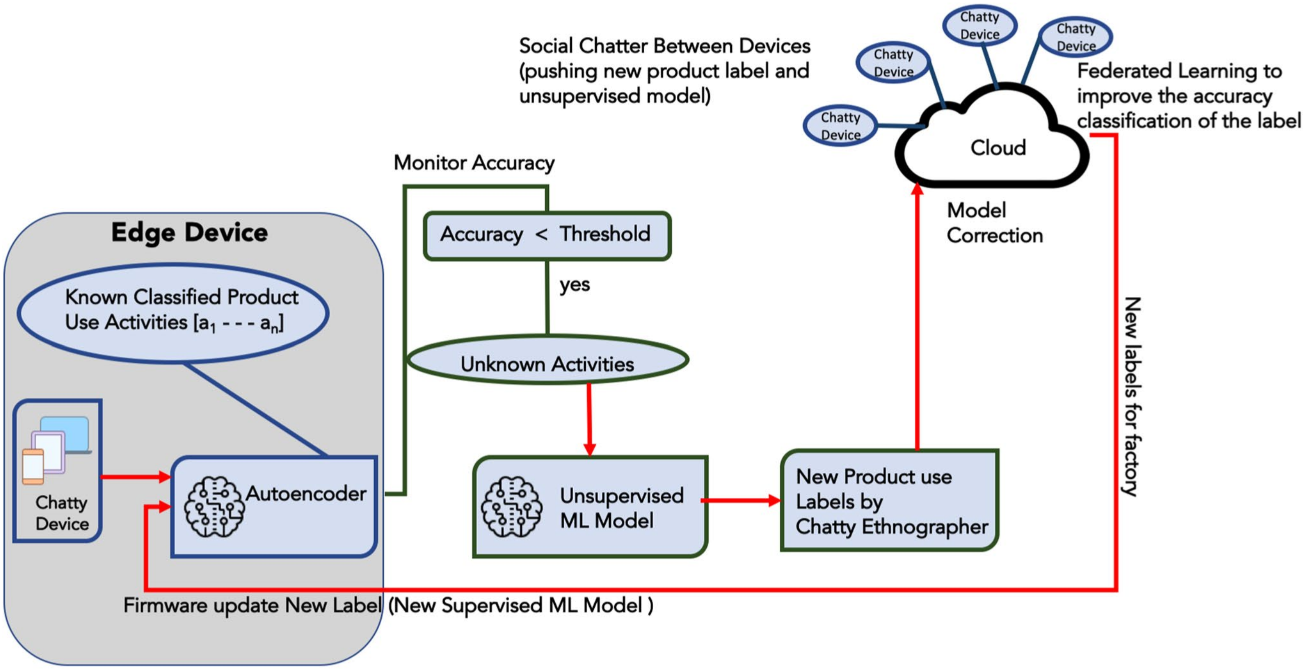

Social Chatter Between Devices (pushing new product label and unsupervised model)
Chatty Device
Chatty Device
Chatty Device
Chatty Device
Federated Learning to improve the accuracy classification of the label
Cloud
Monitor Accuracy
Accuracy < Threshold
yes
Unknown Activities
Model Correction
New labels for factory
Edge Device
Known Classified Product Use Activities [a1 - - - an]
Chatty Device
Autoencoder
Unsupervised ML Model
New Product use Labels by Chatty Ethnographer
Firmware update New Label (New Supervised ML Model)

Fig. 2 Flow of work

4.2 Date exchange protocol and frequency

In each communication round, the server residing at the cloud sends up-to-date global autoencoder to a set of devices. The devices run the autoencoder during a training period with the unlabeled data. When the autoencoder detects activities with an accuracy below a threshold, this triggers the use of an unsupervised machine learning method to cluster the new activity. At the end of the communication round, the devices send the gradients of the resulting model to the server. The server uses the Federated Averaging Algorithm (FedAvg) to calculate the resulting model which will be sent in the next round. If the unsupervised model identifies a cluster which continues over time a new label is generated. The server also updates the global classifier using the new label and sends the resulting classifier to the selected devices with the global autoencoder.

The number of scheduled devices is limited by the available bandwidth that can be allocated to clients at each round. Also, a communication round cannot be considered valid unless a minimum number of updates is collected from devices. The frequency of the communication and aggregation can be reduced with additional local new activity learning rounds. For an optimal learning round duration, bandwidth resource use should be adapted based on the number of local iterations of unknown activities at each device, and the number of global learning rounds.

5 Data collection and annotation

In this study, two datasets were used: one a publicly available Human Activity Recognition (HAR) dataset and a second one which was captured from devices in the Chatty Factories project.

5.1 Experimental set up

A 6th generation Apple iPad was selected as the “Chatty device" for the experiment. Both iOS and Android devices are equipped with a variety of sensors, providing a number of potential mechanisms to collect data for research. iOS devices typically utilise a wide range of inbuilt sensors eg. proximity sensors, accelerometers, gyroscope, etc. [39]. Using this “Chatty device" a dataset was created while performing common product use activities, such as: walking, sitting, standing with the device, dropping, picking up the device, placing the device stationary on a surface and a vibrating surface. The data was collected from the following four in-built sensors.

- Acceleration: 3-axes acceleration data.
- Orientation: Azimuth, Pitch, and Roll.
- Angular velocity: 3-axes gyroscopes for rotational motion.
- Magnetic Field: 3-axes magnetic field.

5.2 Data acquisition

5.2.1 Chatty Dataset

The MATLAB Mobile application [40] was used to stream data from in-built sensors of Chatty-device. A sampling rate of 100Hz was selected for logging the data. The product use Data was streamed into a MathWorks Cloud for analysis. Two subjects (a male and female) were selected to carry out each activity for a continuous period of two minutes each. A total of four minutes of sensor data readings were collected for every activity, producing a total of 148,778 data points. The steps highlighted below were followed:

1. The MATLAB Mobile was installed and configured on the Chatty device through the Apple App store.
2. A registered user account was used and the Chatty device was connected to MathWorks Cloud.
3. Using the MATLAB Mobile interface, the sensors were turned on.
4. A sampling rate of 100 Hz was selected and the log button was used to initiate recording of the sensor data readings.

5. After the expected duration, the logged results were saved to the cloud and given file names which corresponded to the activities carried out.

5.3 Human activity recognition dataset

The publicly available Human Activity Recognition (HAR) data set [41] covers acceleration, GPS, gyroscope, light, magnetic field and sound level data of activities. Where the activities that were performed by fifteen participants (age 31.912.4, height 173.16.9, weight 74.113.8, eight males and seven females) were, climbing stairs down and up, jumping, lying, standing, sitting, running/jogging, and walking. For each activity, a log file was generated that simultaneously recorded the acceleration of the body based on chest, forearm, head, shin, thigh, upper arm, and waist positions during each activity. Each activity was performed for a period of 10 min except for jumping due to physical exertion (1.7 min).

5.4 Data pre-processing

All generated product use data were stored locally on the device and automatically uploaded to the Mathworks cloud. The files were then downloaded and imported into MATLAB desktop application and processed. Each sensor reading was timestamped, pre-processed and then labelled for each activity. A master labelled data log file was generated that contained all sensor readings. A summary of the activities from the datasets used in this work is given in Table 1.

The Chatty dataset was used to train and test the model. The model was further tested against an unseen publicly available dataset. For training of the model, out of the six activities defined in Chatty dataset, four were used. While two activities, namely: Standing and “Drop and Pick up" were used to evaluate the new activity detecting capability of the system.

5.5 Autoencoders

Unlabelled data is captured at the edge to identify known product use activities and flag any unknown activity. To accomplish this problem we propose the use of the Autoencoder method for new activity detection. Autoencoder is a deep learning architecture that has the capability to identify outliers in a dataset [42–44]. Thus it does not need labelled datasets which contain both known and unknown product use activities for training purposes. Considering the data will be collected in real time, it will not be free from noise, a stacked denoising autoencoder was chosen to detect product use activities. A stacked denoising autoencoder, shown in Fig. 3, is a neural network that learns a reconstruction function that produces a denoised vector from a corrupted, noisy input [45]. It is a feed-forward non-recurrent net with the input layer connected to the output layer through several hidden layers which contract from the input to the latent code at the bottleneck layer and then expand to the output layer [46].

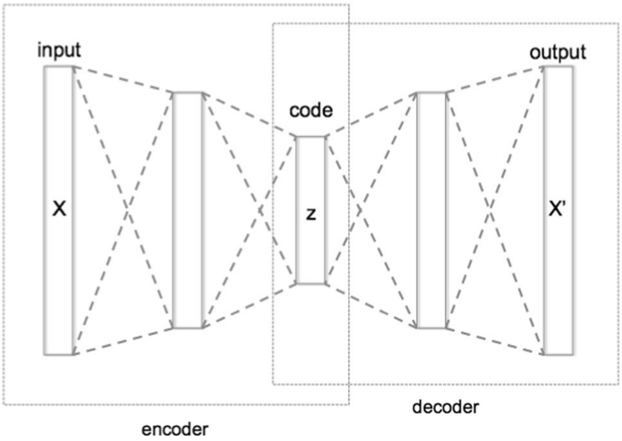
The basic autoencoder learns to find a parameter vector θ which minimizes a reconstruction function of the form

$$J_{AE}(\theta) = \sum_t L(x^{(t)}, g_\theta(f_\theta(x^{(t)}))) \quad (1)$$

Table 1 Summary of datasets

Activities	Chatty	Public HAR
Known	99001	61606
Walking	25267	30734
Sitting	24563	30871
Vibration	24558	–
Stationary on table	24613	–
Unknown	49776	26264
Standing	25214	–
Drop and Pickup	24562	–
Jumping	–	4325
Climbing down stairs	–	21939

Fig. 3 Stacked autoencoder architecture



where $x^{(t)}$ represents a training sample of the known product use activities from the Chatty dataset. However, the stacked denoising autoencoder adds a corruption term $\mathbb{E}_{q(\tilde{x}|x^{(t)})}[\cdot]$ such that the reconstruction function becomes

$$J_{AE}(\theta) = \sum_{t} \mathbb{E}_{q(\tilde{x}|x^{(t)})}[L(x^{(t)}, g_\theta(f_\theta(x^{(t)})))] \tag{2}$$

where the notation $\mathbb{E}_{q(\tilde{x}|x^{(t)})}[\cdot] = \sum_{x^{(t)}} q(\tilde{x}|x^{(t)}) \times [\cdot]$.

The corruption term so introduced reduces the sensitivity of the network to noise in the input and thus increases the ability of the network to generalize to previously unseen input data.

Experiment settings and Hyperparameters: a stacked denoising Autoencoder was implemented in Python 3.8.5 with Keras 2.4.0 and TensorFlow 2.3.0-rc0 with GPU support. The experiment was carried out on a Macbook Pro with six (6) core Intel Core i7 processor running at 2.6GHz, 16GB RAM and a Radeon Pro 555X GPU. Table 2 summarizes the Autoencoder hyperparameter configuration. Linear activation was used on the output and hidden layers to preserve correspondence between the input and output domains since no normalization was used on the input data. For optimization, the Adam optimizer was used since it achieves faster convergence and better performance compared to RMSProp, Adadelta and vanilla SGD [47, 48]. The number of training epochs was fixed to 64 since the training loss flattens out considerably at about 55 epochs. To combat overfitting, an L1_L2 activity regularizer was used on the bottleneck layer to introduce sparsity on that layer. A 0.6 standard deviation Gaussian Noise layer was also used after the input layer to reduce sensitivity of the model to noise in the input data and enhance the models ability to generalize to previously unseen data samples. The noise layer was only active during training, it was inactive at inference time.

5.6 Unsupervised machine learning

While the focus of this paper is not to explore various unsupervised methods for clustering, it plays a crucial role in helping to distinguish between an actual product activity from random noise. If the clustering method repeatedly identifies a specific cluster of data points, then we infer that those data points belong to an actual product activity and not just random noise that may occur infrequently. To this end, it is critically important to determine an optimal number of clusters that can be fed to the clustering algorithm given that the product use activities are being generated during actual use, and as such is unknown.

Table 2 Autoencoder hyperparameters

Hyperparameters	Description	Values
N_l	Number of hidden layers	3
N_h	Number of units on each hidden layer	5, 3, 5
N_{epoch}	Number of training epochs	64
α	Learning rate	0.1

Table 3 Reconstruction loss distribution

	Train (Chatty)	Known activity (Chatty)	Unknown activity (Chatty)	Known activity (Public) sitting	Unknown activity (Public) climb down
Mean	0.129162	0.129993	0.462168	1.054878	1.222696
Std	0.173400	0.174832	0.511599	0.027329	0.385470
Min	0.012407	0.012829	0.018665	0.768975	0.331072
25%	0.028761	0.028780	0.068573	1.047389	0.975942
50%	0.032796	0.033920	0.385282	1.052245	1.123178
75%	0.151830	0.152089	0.773422	1.058461	1.359943
Max	1.363831	1.375441	11.554137	2.316129	3.493574

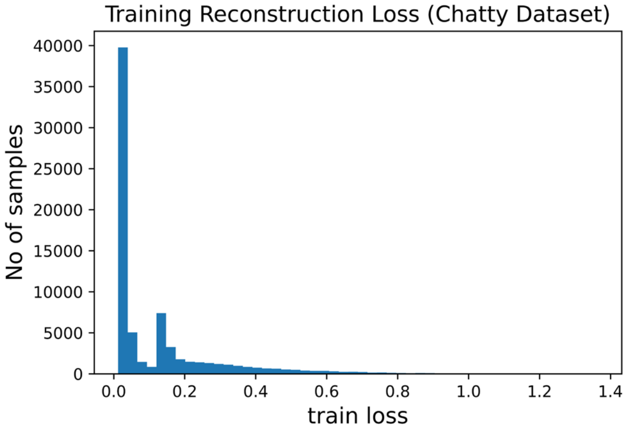


Fig. 4 Training reconstruction loss (Chatty Dataset)

Various methods can be used to find the optimal number of clusters. The elbow method is one such methods [49] and is selected to achieve this task in this study. In this algorithm, the centre of clusters is determined by using K-means clustering algorithm (to execute the task of clustering), and the *Within-Cluster-Sum-of-Squares* (WCSS) is estimated for each identified cluster. The WCSS is referred to as the sum of squares of the distances for every data point in all clusters to their corresponding centroids [50, 51].

To demonstrate the clustering of data points in this study, *K-means* algorithm was used. It is characterized by its distance-based technique which executes in an iterative manner.

Let $X = \{x_i\}, i = 1, 2, \dots, n.$ be the n objects to be clustered, $S = S_1, S_2, \dots, S_k$ is the set of clusters. Let μ be the mean of cluster S_i. The squared-error between μ_i and the objects in cluster S_i is defined as $WCSS(S_i) = \Sigma_{X_{j \in Si}} ||X_j - \mu_i||^2$. The K-means algorithm aims to minimize the sum of the squared error over all k clusters, that is $min(WCSS(S)) = argmin_s \sum_{i=1}^{k} \sum_X j \in Si||X_j - \mu_i||^2$, where WCSS denotes the sum of the squared error in the inner-cluster. A good point to note is that one drawback to the K-means algorithm is that it requires the specification of the number of ″k“ (clusters) beforehand and also and the inappropriateness for discovering non-convex clusters [52].

6 Results and analysis

Once the auto-encoder was configured and deployed, the Chatty dataset was divided into train and test datasets using a 75–25% split for the train and test datasets respectively. For the train dataset only known product user activities were included. However, the dataset that was used to test the model contained both known and unknown product use activities.

Though Autoencoder have a deep learning architectures they differ from neural network based on what they produce as an output. The target output of the autoencoder is not classes but the input itself, primarily to learn the representation of the data or to remember the pattern of known product use activities. During training the autoencoder only known product activities were used to build the model. The rationale behind this was that the model was able to recognise only known product activities and was capable to successfully able to reconstruct the error.

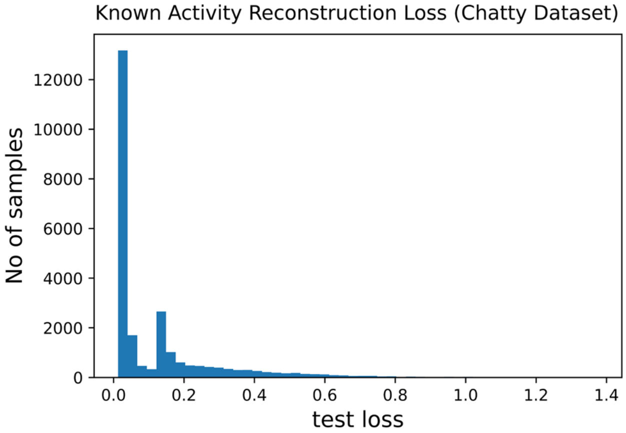


Fig. 5 Test loss known activity (Chatty Dataset)

While training the model on known product use activities the model was able to reconstruct the train and test set data with an accuracy of 99.35% and mean reconstruction error of 0.129 (see Table 3 and Figs. 4, 5). Similarly, the model was tested against known product use activities (walking and sitting) from a publicly available dataset and the performance was consistent with results obtained with the Chatty dataset. The model recorded a performance of 98.03% with a mean of 1.05 and max of 2.3 for the reconstruction error (see Table 3 and Fig. 6).

However, when the model was tested on unknown Chatty product use activities (Standing and Drop and Pickup) the accuracy dropped to 89.51% and the maximum reconstruction error increased by around 743%. This indicated that the model was unable to recognise the data, thus an increase in the reconstruction error was observed. Similarly, we checked the model against unknown product use activities (jumping and climbing down stairs) from a publicly available data set. Again we observed a drop in performance to 83.59% and a rise in reconstruction error (see Figs. 8, 9). In Fig. 7 we

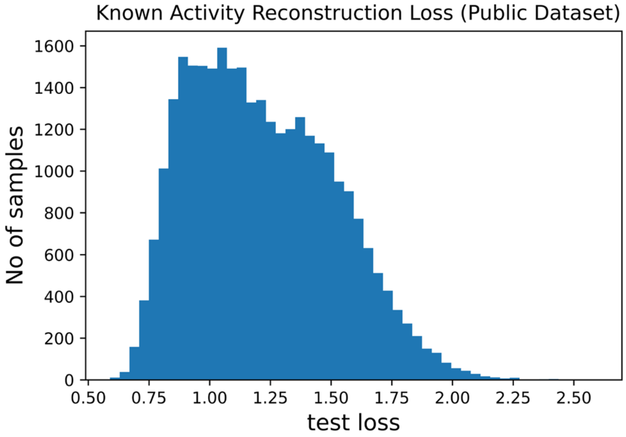


Fig. 6 Known activity reconstruction loss (Public Dataset)

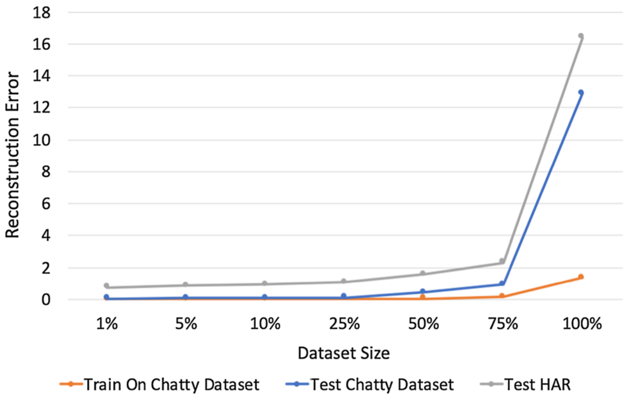


Fig. 7 Reconstruction error observed while training and testing the model

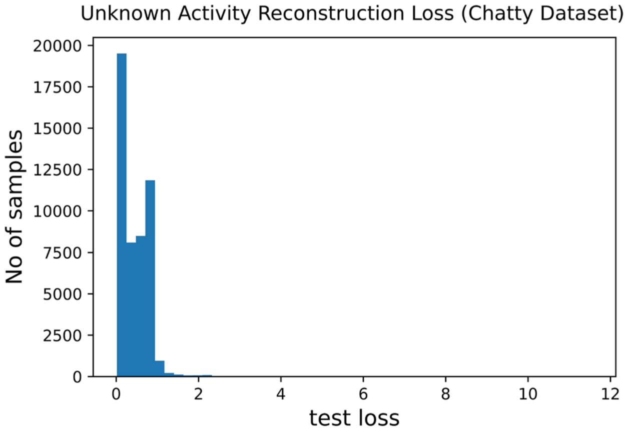


Fig. 8 Unknown activity reconstruction loss (Chatty Device)

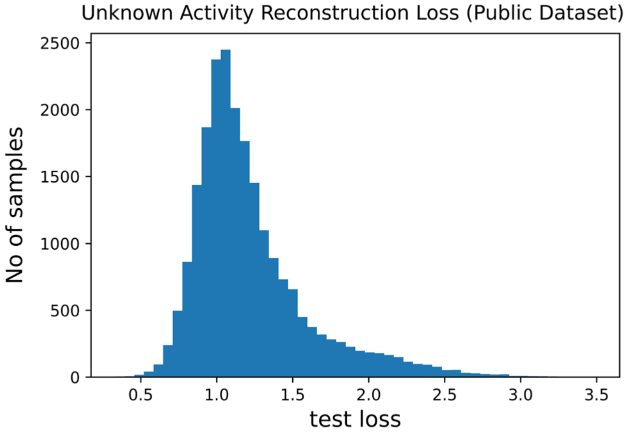


Fig. 9 Unknown activity reconstruction loss (Public Dataset)

Table 4 Anomaly detection model performance

Dataset	Accuracy (%)
Chatty train set (known)	99.35
Chatty test set (known)	99.35
Chatty test set (unknown)	89.51
Public test set (known)	98.03
Public test set (unknown)	83.59

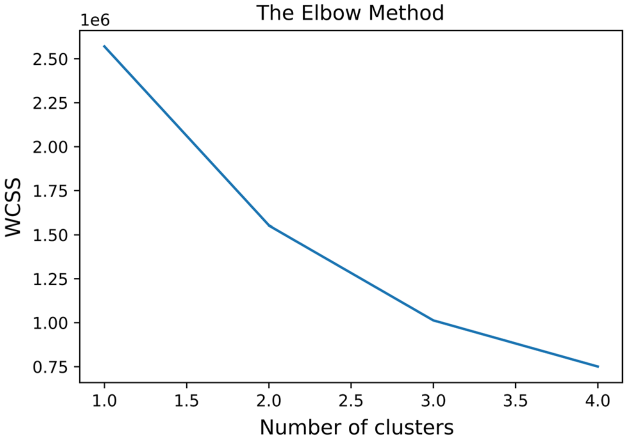
can see how the reconstruction error changes as the model encounters unknown product use activity resulting in drop of overall performance.

Table 4 presents the performance of the model on known and unknown product use activities from the chatty dataset and the publicly available HAR dataset. The results demonstrate that the model is able to generalize to previously unseen datasets of known activities with an accuracy of 98.06%. As expected, the accuracy drops significantly for unknown activities from the held out test data from the Chatty device and the publicly available HAR dataset.

In order to create a trigger for the model to flag unknown product use activities, a threshold value was to be decided. From the observed results, an accuracy of 93.7% was selected as the threshold for minimum acceptable accuracy for known activities. This threshold is 1.5 standard deviations lower than the minimum observed accuracy for known activities across the chatty datasets and the publicly available HAR dataset. The selected threshold allows the model to maximally discriminate between known and unknown activities while preserving the ability of the model to generalize to previously unseen datasets.

The data that had been categorised as unknown product use activity, was further processed to check if it was actually an activity data or it was just noise. In order to check that the data was passed through unsupervised machine learning model to cluster similar data points together. While building the unsupervised clustering model, Elbow

Fig. 10 Elbow method to select number of Clusters



method was used to determine the number of clusters and subsequently, the corresponding graph is plotted with respect to the cluster numbers and WCSS (as reflected in Fig. 10).

The optimal number of clusters occurs when no further improvement is observed in the “clustering index" as additional clusters are added. Figure 10 demonstrates clusters for the unknown product usage activities discovered by the autoencoder. From Fig. 10, we can infer that the optimal number of clusters is 2.

Figure 11 presents empirical results using K-means clustering algorithm on unknown activity data. Two clear clusters can be observed in the visualisation, as already seen from the WCSS results. Figure 12 illustrates a line plot of the triaxial accelerometer signals for the unknown activities. Therefore the data flagged as unknown product use data corresponds to user activity and not to noise.

Fig. 11 Acceleration 3D plot: clustering of unknown product use activity

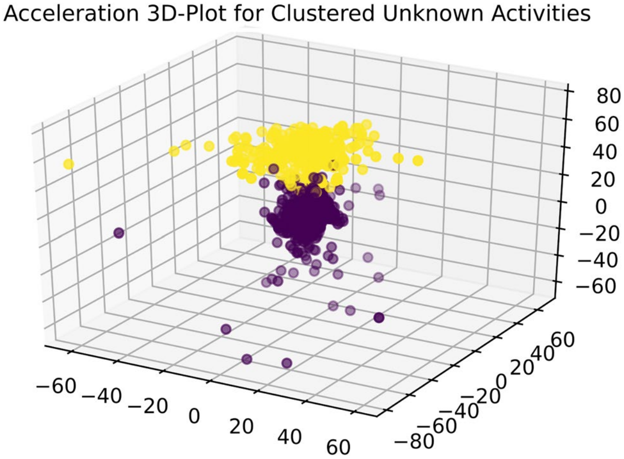
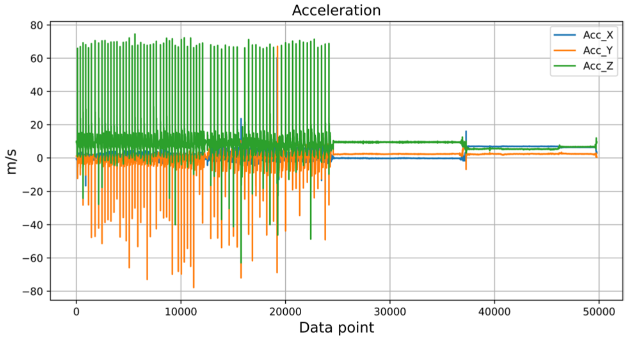


Fig. 12 Acceleration line plot: for unknown product use activity



7 Discussion and conclusion

A system that can distinguish between known and unknown product use activities, while providing model correction using federated learning is described. The proposed system is designed to enable user data to be processed at the edge to construct a model, and any correction needed to the model to be undertaken on a cloud environment. This ensures that no user related data has to be transferred beyond proximity to the device. To distinguish between known and unknown product use activities a stacked denoising autoencoder was chosen because of its ability to reduce sensitivity to noise in the input dataset and thus increase the ability to generalise to previously unseen input dataset. The autoencoder was trained on dataset collected from devices in the Chatty Factories project. It was tested on a part of unseen data collected from *chatty devices* (i.e. devices that have in-built sensors and where such data can be used to associate particular use of the device from this data) and from a publicly available dataset, the Human Activity Recognition (HAR) dataset.

The model was successful in identifying known product use activities with an accuracy of 99.35% while training the model, and an accuracy of 99.35% (Chatty dataset) and 98% (HAR dataset) while testing the model. In order to identify unknown product use activities a performance threshold of 97.5% was defined. A drop in performance below this level would indicate that the model had encountered unknown product use activities. Based on this threshold the model was successful in identifying unknown product use activities in both Chatty and HAR datasets. This led to the creation of a new label representing unknown product use activities, which a product manufacturer could use to understand how the product was *actually* being used. The data could also be used to improve product design for a future version of the product. The methodology and approach outlined in this work can lead to a new generation of privacy-aware data-driven adaptive manufacturing process, which can be used in a number of different manufacturing environments.

Using a 6th gen. Apple iPad as a "chatty device" (with acceleration, orientation, angular velocity and magnetic field sensors) we demonstrate how product use activities can achieve a classification accuracy of 99.35%. A comparison is also undertaken with the Human Activity Recognition (HAR) data set, achieving an accuracy of 98%. Our approach demonstrates how semantic activity labels can be associated with product use, and subsequently used to improve product design.

Acknowledgements This research was partially funded by the UK Engineering and Physical Sciences Research Council (EPSRC) projects: EP/R021031/1: "Chatty Factories" and EP/R033439/1: "Privacy-aware Cloud Ecosystems".

Authors' contributions ML,AJ, OR and PB conceived the research design. ML collected, transformed and classified the data. SA developed the autoencoder. ML, AJ and SA tested and validated the proposed model. SC contributed towards architecture design, particularly towards Federated learning. All authors wrote, proofread and approved the final manuscript.

Data access statement Information on the data underpinning the results presented here, including how to access them, can be found in the Cardiff University data catalogue at 10.17035/d.2021.0125677664 and as a web link http://doi.org/10.17035/d.2021.0125677664.

Competing interests The authors declare that they have no conflict of interest.

References

1. Oztemel E. Gursev samet literature review of industry 4.0 and related technologies. 2020.
2. Ghobakhloo M. Industry 4.0, digitization, and opportunities for sustainability. 2020.
3. Muscio A, Ciffolilli A. What drives the capacity to integrate Industry 4.0 technologies? Evidence from European R&D projects. Econ Innov New Technol. 2020;29(2):169–83.
4. Qiong W, He K, Chen X. Personalized federated learning for intelligent IoT applications: a cloud-edge based Framework. 2020.

5. Burnap P, Branson D, Murray-Rust D, Preston J, Richards D, Burnett D, Edwards N, Firth R, Gorkovenko K, Khanesar MA, Lakoju M, Smith T, Thorp J Chatty factories: a vision for the future of product design and manufacture with IoT. In *Living in the Internet of Things (IoT 2019)*, pages 4 (6 pp.)–4 (6 pp.). Institution of Engineering and Technology, 2019.
6. Guidoux R, Duclos M, Fleury G, Lacomme P, Lamaudière N, Manenq H, Paris L, Ren L, Rousset S. A smartphone-driven methodology for estimating physical activities and energy expenditure in free living conditions. J Biomed Inf. 2014;52:271–8.
7. Uddin MZ. A wearable sensor-based activity prediction system to facilitate edge computing in smart healthcare system. J Parallel Distrib Comput. 2019;123:46–53.
8. Preece SJ, Goulermas JY, Kenney LPJ, Howard D, Meijer K, Crompton R. Activity identification using body-mounted sensors a review of classification techniques. Physiol Measur. 2009;30(4):R1–R33.
9. Casilari E, Santoyo-Ramón JA, Cano-García JM. Analysis of public datasets for wearable fall detection systems. Sensors. 2017;17(7):1513.
10. Micucci D, Mobilio M, Napoletano P, Tisato F. Falls as anomalies? An experimental evaluation using smartphone accelerometer data. J Ambient Intell Hum Comput. 2017;8(1):87–99.
11. Kerdjidj O, Ramzan N, Ghanem K, Amira A, Chouireb F. Fall detection and human activity classification using wearable sensors and compressed sensing. J Ambient Intell Hum Comput. 2020;11(1):349–61.
12. Igual R, Medrano C, Plaza I. Challenges, issues and trends in fall detection systems. 2013.
13. Mubashir M, Shao L, Seed L. A survey on fall detection: principles and approaches. Neurocomputing. 2013;100:144–52.
14. Alhimale L, Zedan H, Al-Bayatti A. The implementation of an intelligent and video-based fall detection system using a neural network. Appl Soft Comput J. 2014;18:59–69.
15. Trabelsi D, Mohammed S, Chamroukhi F, Oukhellou L, Amirat Y. An unsupervised approach for automatic activity recognition based on Hidden Markov Model regression. IEEE Trans Automat Sci Eng. 2013;10(3):829–35.
16. Banos O, Galvez JM, Damas M, Pomares H, Rojas I. Window size impact in human activity recognition. Sensors. 2014;14(4):6474–99.
17. Gao L, Bourke AK, Nelson J. Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems. Med Eng Phys. 2014;36(6):779–85.
18. Feng G, Mai J, Ban Z, Guo X, Wang G. Floor pressure imaging for fall detection with fiber-optic sensors. IEEE Pervas Comput. 2016;15(2):40–7.
19. Daher M, Diab A, Najjar MEBE, Khalil MA, Charpillet F. Elder tracking and fall detection system using smart tiles. IEEE Sensors J. 2017;17(2):469–79.
20. Abbasi QH, Rehman MU, Qaraqe K, Alomainy A. Advances in body-centric wireless communication: applications and state-of-the-art. Institution of Engineering and Technology, 1st edition, 2016.
21. Makhlouf A, Boudouane I, Saadia N, Cherif AR. Ambient assistance service for fall and heart problem detection. J Ambient Intellig Hum Comput. 2019;10(4):1527–46.
22. Ozcan K, Velipasalar S, Varshney PK. Autonomous fall detection with wearable cameras by using relative entropy distance measure. IEEE Trans Hum Mach Syst. 2017;47(1):31–9.
23. Gibson RM, Amira A, Ramzan N, Casaseca-De-La-Higuera P, Pervez Z. Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic. Appl Soft Comput J. 2016;39:94–103.
24. Qiang L, Stankovic John A, Hanson Mark A, Barth Adam T, John L, Gang Z. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In: Proceedings- 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks. BSN. 2009;2009(2009):138–43.
25. Raymond Y, Lee W, Carlisle AJ. Detection of falls using accelerometers and mobile phone technology. Age Ageing. 2011;40(6):690–6.
26. Tabia H, Gouiffes M, Lacassagne L. Motion histogram quantification for human action recognition. In: 21st International Conference on Pattern Recognition (ICPR 2012), number Icpr, 2012;2404–2407.
27. Jayasinghe U, Harwin WS, Hwang F. Comparing clothing-mounted sensors with wearable sensors for movement analysis and activity classification. Sensors. 2019;20(1):82.
28. Moussa MM, Hamayed E, Fayek MB, El Nemr HA. An enhanced method for human action recognition. J Adv Res. 2015;6(2):163–9.
29. Hui S, Zhongmin W. Compressed sensing method for human activity recognition using tri-axis accelerometer on mobile phone. J China Univ Posts Telecommun. 2017;24(2):3171–772.
30. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics, AISTATS 2017. 2016.
31. Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konečný J, Mazzocchi S, McMahan HB, Van Overveldt T, Petrou D, Ramage D, Roselander J. Towards federated learning at scale: system design. 2019.
32. McMahan B, Daniel R. Federated learning: collaborative machine learning without centralized training data; 2017.
33. Konečný J, McMahan HB, Ramage D, Richtárik P. Federated optimization: distributed machine learning for on-device intelligence. 2016.
34. McMahan H Brendan, Ramage Daniel, Talwar Kunal, Zhang Li. Learning Differentially Private Recurrent Language Models. In: 6th international conference on learning representations, ICLR 2018—conference track proceedings. 2018.
35. Dean J, Corrado GS, Monga R, Chen K, Devin M, Le QV, Mao MZ, Ranzato A, Senior A, Tucker P, Yang K, Ng AY. Large scale distributed deep networks. Technical report. 2012.
36. Smith SL, Kindermans PJ, Ying C, Le QV. Don't decay the learning rate, increase the batch size. In: International conference on learning representations (ICLR), 2018. 2018.
37. Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L, Kyrola A, Tulloch A, Jia Y, He K. Accurate, large minibatch SGD: training ImageNet in 1 hour. 2017.
38. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the ACM conference on computer and communications security, New York, New York, USA, oct 2017. Association for Computing Machinery, pages 1175–1191.
39. Sam C. The sensors that make the iPhone So Cool. 2019.
40. Makers of matlab and simulink.

41. Sztyler T, Stuckenschmidt H. On-body localization of wearable devices: An investigation of position-aware activity recognition. In: 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE Computer Society, 2016, pages 1–9. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7456521.
42. Pratomo BA, Burnap P, Theodorakopoulos G. Unsupervised approach for detecting low rate attacks on network traffic with autoencoder. In: 2018 international conference on cyber security and protection of digital services (Cyber Security)
43. Hao Y, Sheng Y, Wang J. Variant gated recurrent units with encoders to preprocess packets for payload-aware intrusion detection. IEEE Access. 2019;7:49985–98.
44. Schneider P, Böttinger K. High-performance unsupervised anomaly detection for cyber-physical system networks. In: Proceedings of the 2018 workshop on cyber-physical systems security and privacy, 2018;1–12.
45. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intellig. 2013;35(8):1798–828.
46. Autoencoder with recurrent neural networks for video forgery detection. *Electronic Imaging*, 2017;(7), 2017.
47. Ruder S. An overview of gradient descent optimization algorithms. CoRR. arxiv: abs/1609.04747, 2016.
48. Kingma Diederik P. Ba Jimmy. Adam: A method for stochastic optimization; 2017.
49. Kodinariya Trupti M, Makwana Prashant R. Review on determining number of cluster in k-means clustering. Int J. 2013;1(6):90–5.
50. David P. Strong consistency of k-means clustering. Ann Stat. 1981;56:135–40.
51. Chakrabarty N, Rana S, Chowdhury S, Maitra R. RBM based joke recommendation system and joke reader segmentation. In: International conference on pattern recognition and machine intelligence. Springer; 2019;229–239.
52. Likas Aristidis, Vlassis Nikos, Verbeek Jakob J. The global k-means clustering algorithm. Pattern Recogn. 2003;36(2):451–61.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.