Article

Leveraging Edge Computing ML Model Implementation and IoT Paradigm Towards Reliable Postoperative Rehabilitation Monitoring

Evanthia Faliagka [*] , Vasileios Skarmintzos , Christos Panagiotou , Vasileios Syrimpeis , Christos Antonopoulos , Nikolaos Voros

Posted Date: 21 June 2023

doi: 10.20944/preprints202306.1530.v1

Keywords: Edge computing; Internet of Things system; Knee rehabilitation; Machine Learning rehabilitation; Movement detection

Article

Leveraging Edge Computing ML Model Implementation and IoT Paradigm towards Reliable Postoperative Rehabilitation Monitoring

Evanthia Faliagka [1,*], Vasileios Skarmintzos [1], Christos Panagiotou [1,2], Vasileios Syrimpeis [1], Christos Antonopoulos [1] and Nikolaos Voros [1]

[1] Electrical & Computer Engineering Department, University of Peloponnese, M. Alexandrou 1, Patras, Greece; e.faliagka@esdalab.ece.uop.gr; ece2023smart-ic@go.uop.gr; ch.panagiotou@esdalab.ece.uop.gr; vsyrimpeis@gmail.com; ch.antonop@uop.gr; voros@uop.gr

[2] AVN Innovative Technology Solutions ltd, Limassol, Cyprus; chpanag@avntechgroup.com

* Correspondence: e.faliagka@esdalab.ece.uop.gr

Abstract: In this work, an IoT system with edge computing capability is proposed facilitating postoperative surveillance of patients who have undergone knee surgery. The main objective is to reliably identify whether a set of orthopedic rehabilitation exercises are executed correctly, which is critical since, it is often necessary to supervise patients during the rehabilitation period so as to avoid injuries or long recovery periods. The proposed system leverages Internet of Things (IoT) paradigm, in combination with Deep Learning and Edge Computing to classify the extend-flex movement of one's knee via embedded Machine Learning (ML) classification algorithms. The contribution of the proposed work is multilayered. Furthermore as an outcome of this work a dataset of labeled knee movements is freely available on https://www.kaggle.com/datasets/billskarm/knee-range-of-motion to the research community. It also provides real time movement detection with an accuracy reaching 100%, which is achieved with a ML model trained to fit a low cost, off the shelf, Bluetooth Low Energy platform. The proposed Edge Computing approach allows predictions to be performed on-device rather than solely relying on a cloud service. This yields critical benefits in terms of wireless bandwidth and power conservation, drastically enhancing device autonomy, while delivering reduced event detection latency. In particular, the "on device" implementation is able to yield a drastic 99,9% wireless data transfer reduction, a critical 39% prediction delay reduction and a valuable 17% increase of prediction event. Finally, enhanced privacy comprises another significant benefit from the implemented Edge Computing ML model as sensitive data can be processed on-site and only events or predictions are shared with the medical personnel.

Keywords: edge computing; Internet of Things system; knee rehabilitation; machine Learning rehabilitation; movement detection

1. Introduction

Over the last few years multiple reports have highlighted the drastic need for a shift of caregiving technologies towards remote and personalized service provision [1,2]. What can be easily extracted is that advancements in information and communications technology domains (ICT domains) such as embedded systems, communication technologies, and Artificial Intelligence/Machine Learning, place cyber-physical end-to-end platforms in the position of key enablers for future healthcare systems. A recent attention-grabbing survey highlights the potential of ICT based health care paradigm [3].

Patients nowadays increasingly require personalized and continuous interaction with their therapists combining more effective treatments with cost-efficiency. At the same time, it is commonly accepted that the typical health care structures (i.e. hospitals, clinics, public organizations etc.) are reaching critical capacity levels leading to low quality service provision, medical personal exhaustion with potential negative effect on patients' health and even life endangerment. However, it is of outmost importance that medical staff offer services of high value and successful treatments to patients in need.

The last decade IoT devices have emerged in health care systems worldwide. Their advent has revolutionized the way medical staff interacts with patients. This is because, on one hand, they allow a wide range of services to be provided remotely (i.e. directly at the patient's home environment) with critical benefits for all parties involved. On the other hand, state of the art IoT devices and accompanied platforms leverage edge computing allowing the detection of highly complex events/conditions by the device itself drastically expanding the reliability, scalability, lifetime, autonomy, and intelligence of the system. However, these new and rich data sources have started to overwhelm the health systems, putting forward new challenges to those who develop them. Consequently, digitalization of the systems has provided many opportunities to transform typical healthcare solutions into systems that continue to treat patients with less suffering and also meet the cost, quality and safety challenges.

The goal is to deliver health care services to patients and provide the ideal supervision in real-time. In this paper, the main objective is to develop a novel system leveraging IoT technologies, in combination with Deep Learning and Edge Computing to offer a comprehensive and effective solution. Deep Learning refers to the subset of algorithms of Machine Learning (ML), through which a system would imitate the way a human brain would function in order to learn. It is a set of programmable actions which use mathematical models to acquire a set of data, properly process the information and bring a final output, which is a solution to a problem that has been determined. Deep Learning can be accomplished through the organization of a Neural Network, where input data are forwarded to the so-called hidden layers, a set of processing nodes properly connected between them. Certain processes are performed to these units and a final outcome is presented by the output layer. Through repetitive processes and re-evaluation of weights (connections) between neurons (nodes) and with a suitable activation function (a process called back-propagation) the network succeeds to learn. Thus, any new given input can successfully be categorized to a class. Deep learning provides all the necessary tools to acquire, process and properly use the data that come from sensors and in that way build models that satisfy special applications for medical use. Edge Machine Learning is a technique where Smart Devices can process data using machine and deep learning algorithms, reducing reliance on Cloud networks while offering performance advantages.

Many efforts have been done to combine IMUs (electronic devices that measures and reports a body's movement and orientation, using a combination of accelerometers, gyroscopes, and magnetometers) and healthcare services. However, most of them [4, 5] exploit sensors and mainly accelerometers and are compared to traditional methods for evaluating the rehabilitation process as patient-reported outcome measures and performance-based outcome measures. Through the work in the paper, authors aim to address a gap, since few studies have dealt with the use of IMUs for remote healthcare [6] and even fewer with kinetics prediction using deep learning techniques with wearable sensors [5].

Only the last two years efforts are reported making use of Machine Learning-based embedded devices in the field of healthcare [7]. The existing studies use embedded devices to propose a hearing aid device [8], to monitor vital signs (body temperature, breathing pattern, and blood oxygen saturation) [9] and to recognize human activity [10]. In all these studies TinyML [11], a field that facilitates running machine learning at the embedded edge devices was used. This is the first effort to create a rehabilitation system for flex-extend classification using AI techniques and additionally enhancing the method using neural networks on computing devices to reduce the computational resources and increase privacy.

One of the areas where respective systems can have a profound positive effect is orthopedic rehabilitation. The successful outcome of such processes depends on the consistent, accurate and meticulous repetition of specific sets of exercises by the patient for possibly extended time periods (several weeks or months), the reliable, concise, and personalized monitoring and reporting to the medical staff and the timely feedback and encouragement back to the patient [12]. Driven by this perspective, this paper puts forward a respective IoT, edge computing capable system applying

doi:10.20944/preprints202306.1530.v1

respective technologies in postoperative surveillance of patients who have undergone knee surgery, to identify whether the rehabilitation exercises are executed correctly. It should be noted that knee arthroscopy is among the most common surgeries performed, with more than four million arthroscopic knee procedures to be performed worldwide each year, according to the American Orthopedic Society for Sports Medicine [13]. In addition, the most commonly performed joint replacement surgery is the total knee arthroplasty with over 600,000 surgeries occurring annually. By 2030, the number of knee replacements is projected to rise to over 3 million per year [14]. Traditional outpatient follow-up appointments provide limited opportunities for review, while approximately 20% of patients continue to suffer from postoperative pain and functional limitations, impacting recovery [15, 16].

Such references highlight the need for optimized rehabilitation strategies that focus on the early identification of patients with a suboptimal outcome through the assessment of early postoperative physical function [17, 18].

Taking into consideration all the above information, we aim to apply and evaluate deep learning techniques and algorithms in integrated systems in the field of orthopedic rehabilitation. In this paper we propose a novel approach where a machine learning model is created using the Edge Impulse platform [19] and an Arduino Nano 33 sense board [20]. Edge Impulse is an innovative developing platform for machine learning application on edge devices. It empowers developers to build, deploy, and scale embedded ML applications to create and optimize solutions with real-world data. It is fully compatible with the Arduino Nano 33 BLE Sense, an off-the-shelf, low cost, IoT embedded system with sensors to detect color, proximity, motion, temperature, humidity, audio and more.

A critical challenge addressed concerns the training of a model that can distinguish different degrees of angles of a moving knee that can be deployed locally at an embedded device. By making an extend-flex movement of one's knee, the model will be able to classify this movement in the given degrees, via embedded ML classification algorithms. The importance of this classification is apparent in orthopedic exercises that a therapist may ask a patient to execute. The final goal is to evaluate the edge computing capability and performance of executing the trained model on an Arduino Nano 33 board, detecting the targeted event and conveying the information wirelessly to a backend IoT platform where data is stored, forwarded and presented to the medical personnel. So, the patient only needs to use a wearable device and practice the "flex-extend" exercises given by the therapist, without the need to be transported to a doctor's office or a medical/rehabilitation center. The proposed system, by enabling a machine learning algorithm running on device, yields valuable reduction concerning event detection delays, as well as bandwidth and energy requirements and at the same time improve privacy. Furthermore, by distribution the workload between the device and the cloud infrastructure some layers of a neural network can be calculated on device and the outputs then be transferred through a gateway to cloud for further processing. So, a considerable amount of processing is transferred to these small devices increasing utilization. Machine learning techniques require a great number of resources to work properly. The goal here is to develop appropriate algorithms that are tailored for embedded platforms. What we assume here is that a machine learning model has been trained offline on a prebuild dataset. Usually, data gathered by the edge device would be sent back to a cloud platform. This platform is responsible to perform the training and finally distribute the model that has been built back to the edge device.

The rest of the paper is organized as follows. Section 2 outlines the knee rehabilitation problem presenting background information required to better understand the design approach. Section 3 describes the architecture of the implemented system and the information workflow. Section 4 presents the machine learning process including the training phase, the event detection, the edge computing implementation and the challenges of the proposed methods and the future work. Section 5 highlights the main conclusions of the paper.

2. Knee rehabilitation Background Information

The knee joint (Figure 1) is a modified hinge joint (ginglymus). The knee joint is not a very stable joint. Knee range of motion is conducted through the following movements: Flexion, Extension, Internal rotation, External rotation.

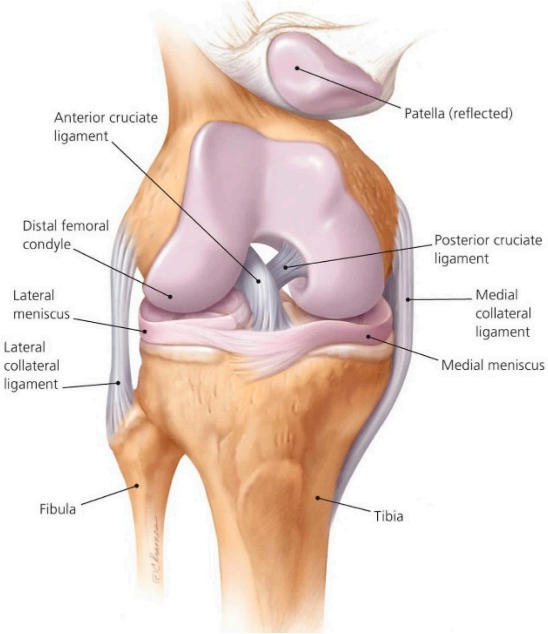

Patella (reflected)
Anterior cruciate ligament
Distal femoral condyle
Posterior cruciate ligament
Lateral meniscus
Medial collateral ligament
Medial meniscus
Lateral collateral ligament
Fibula
Tibia

Figure 1. The Knee Joint.

Flexion and extension are the main movements. They take place in the upper compartment of the joint, above the menisci. The flexion of the knee can be measured using a goniometer. Flexion over 120 ° is regarded as normal. Loss of flexion is common after local trauma, effusion and arthritic conditions.

Flexion and extension differ from the ordinary hinge movements in two ways: The transverse axis around which these movements take place is not fixed. During extension, the axis moves forwards and upwards, and in the reverse direction during flexion. The distance between the center and the articular surfaces of the femur changes dynamically with the decreasing curvature of the femoral condyles. These movements are invariably accompanied by rotations (conjunct rotation). Medial rotation of the femur occurs during the last 30 degrees of extension and lateral rotation of the femur occurs during the initial stages of flexion when the foot is on the ground. When the foot is off the ground, the tibia rotates instead of the femur, in the opposite direction. The description of flexion and extension showcases that the movements that this paper studies are more complex than simple back-and-forth movements. It makes clear the importance of machine learning as ML algorithms consider all the micro-movements that are not even visible.

The Knee Range of Motion (ROM) is an important indication of the ability of doing certain activities. Following a surgery on the knee or any distortion that may (accidentally) occur, therapists have to take into serious consideration the Range of Motion of the knee. In the table above, it is apparent that when someone needs to do any activity, the degrees of the bending knee play a significant role.

Range of Motion is a composite movement, made up by extension and flexion. The combination of the two movements, creates the ability to practice the movement of the knee. Extension means that the leg is straight (0 ° -completely straight leg) and flexion means a flexed leg (at around 130 °).

Extension of the knee is of great importance as it may lead to instabilities and (dangerous) falls and also, if legs cannot go completely straight then quadriceps muscles are continuously activated and are tired quickly.

Knee flexion is equally important, because it is this exact movement of the knee that lets any activity be executed. Bending the knee is crucial if one needs to stand up from a chair or safely climb stairs. The major clinical problem that patients, physiotherapists and medical doctors have to face after a knee surgery is the loss of the range of motion of the knee, due to immobilization, that very quickly results to joint stiffness, especially in the knee joint. Unfortunately, joint stiffness is directly connected to permanent and severe pain, even when the knee joint lacks a few degrees of motion. Moreover, when referring to knee extension, even a 5° loss of extension results to an obvious pathological phenomenon that is lameness. This result is considered extremely bad, since more than four million arthroscopic knee procedures are performed worldwide each year, according to the American Orthopaedic Society for Sports Medicine, and in their majority are conducted in young patients.

Therefore, it is of great clinical importance to provide a system that is able to detect, monitor for extended periods of time and estimate the postoperative knee range of motion restoration. Driven by this need, in this paper we have examined the knee movement of 20 °, 45 °, 60 °, 90 ° and 120 °. The aforementioned angles are chosen to be detected because the 20 ° is the normal anatomical position of the knee, the 45 ° is an important milestone in knee flexion, the 60 ° shows progression over the 45 ° milestone and as you can see in Table 1, is very close to achieve the activity of walking without a limp, the 90 ° is related to the activity of ascending-descending stairs, see Table 1, and the 120 ° is considered to be a knee flexion that will not cause any future pain even after a knee injury or surgery.

Table 1. Movements and required Range of Motion.

Activity	Required Knee Range of Motion
Walk without a Limp	**70°**
Safely climb stairs	**83°**
Safely descend stairs	**90°**
Get in and out of car	**100°**
Get up from chair	**105°**
Ride a Bike	**115°**
Garden	**117°**
Squat	**125°**

3. Proposed System Architecture

In Figure 2 the system's high-level architecture is presented indicating the main components and the main process steps. The training process involved 10 volunteered subjects that all of them gave their informed consent for inclusion before they participated in the study. The study was conducted in accordance with the Declaration of Helsinki, and the protocol was approved by the Local Ethics Committee.

All subjects had different characteristics such as age, weigh, height so as to have a representative sample. They all performed the same movements and the data collected were forwarded to a cloud backend platform [21]. After a first preprocessing all data are stored in a cloud database and sent to the edge impulse platform. The training procedure followed as it is described in Section 4.1 and the classification model was created. Following required iterations in order to achieve the most suitable model adjusting adequately the parameters, the optimal trained mode is transferred to the embedded device and it is ready to be used on the device. Each time the patient makes a movement it is classified to one of the predetermined classes and the prediction can be sent to the backend infrastructure or doctor's mobile using Bluetooth wireless interface.

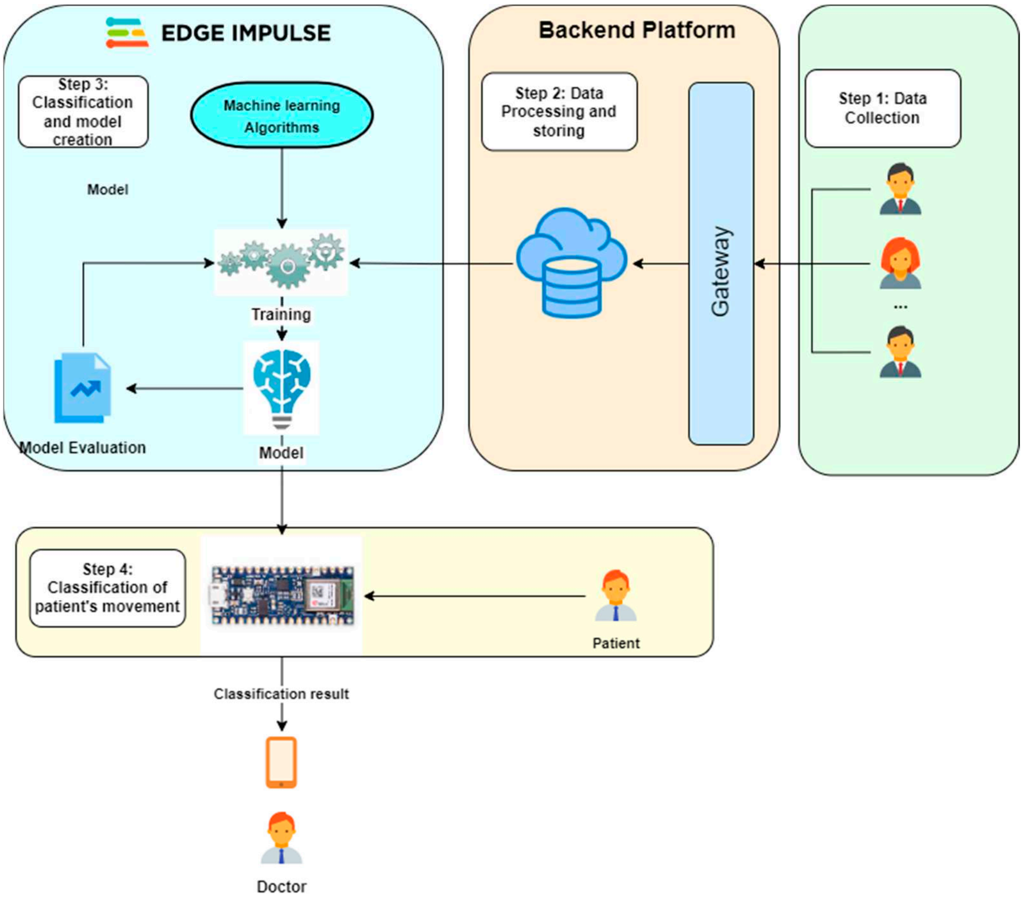

EDGE IMPULSE
Step 3: Classification and model creation
Machine learning Algorithms
Model
Training
Model Evaluation
Model
Backend Platform
Step 2: Data Processing and storing
Gateway
Step 1: Data Collection
...
Step 4: Classification of patient's movement
Patient
Classification result
Doctor

Figure 2. System's Architecture and main workflow.

3.1. Data aquition and experimentation setup

The purpose of this experiment is to create an embedded machine learning model that is able to distinguish the different positions an adult can bend his/her leg. The degrees of the angles that we want the model to recognize are 20 º, 45 º, 60 º, 90 º and 120 º.

To achieve this objective, as indicated in the system architecture section, an Arduino Nano 33 BLE sense Board was used to acquire the required training data from the onboard IMU sensor. This board is considered ideal for IoT solutions because it offers a variety of sensors (IMUs and wireless interfaces being the most relevant to our experiment) in small dimensions (18 x 45 mm) as it supports trained models by the Edge Impulse platform. Using the board's sensors, the movement of the leg is monitored, focusing on up- and-down movement of ones' leg, from the knee and below and respective angle this movement forms are recorded.

In order to objectively record the ground truth measurements of the angles of interest two instruments were used: i) a universal goniometer and ii) a telescopic RangeOfMotion knee brace with built-in goniometer. The Arduino Nano 33 was adjusted to each of these devices and three different setups are considered. In the first case the board was adjusted on the edge of the goniometer (Figure 3a). The goniometer was then placed on a steady surface. Then, while holding the one brace of the goniometer steady, we started moving the other brace (with the Arduino board attached) up-and-down, forming, in that way, the desired angle.

The second case followed the same procedure but this time the goniometer was placed on the knee of an adult person (Figure 3b) thus taking into consideration all the involuntary rotations and movements in the two other axes existing when a realistic leg movement is done. The person (starting from complete stretch of the leg – 0 degrees) started moving the leg up-and-down forming the desired angle, according to the indication of the goniometer.

The third case involved a telescopic RangeOfMotion knee brace (Figure 3c). This knee brace is a universal, easily adjustable and secure support to patients recovering from knee surgery, injuries or instabilities. The range of motion can be easily adjusted via the quick-clips and provide a simple 'pull and slide' motion. The mechanism of the device has degree indications so we can easily choose the range of motion of the leg in degrees. For example, by pulling and sliding the regulator to 45^{o}, we have a range of motion (from full extend) of 45^{o} degrees. After wearing this device, Arduino Nano was adjusted to the lowest part of the brace and we start experimenting with the different angles. The dataset created with the third method was the one used for the edge computing implementation and is freely available on https://www.kaggle.com/datasets/billskarm/knee-range-of-motion.

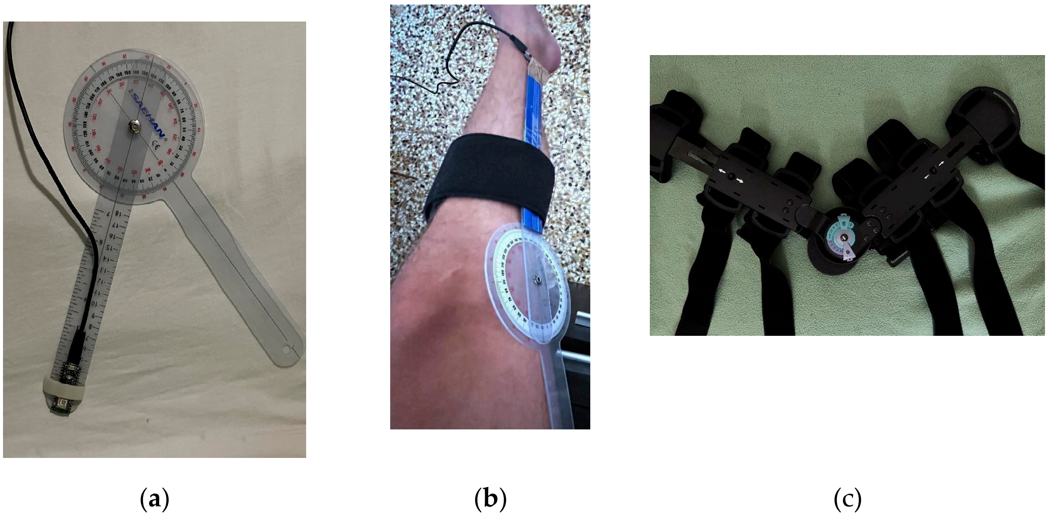


Figure 3. (a) Goniometer with Arduino attached**; (b)** Goniometer on knee; **(c)** Knee brace.

An image of how the raw data collected are presented in the platform of Edge Impulse is shown below (Figure 4). As the accelerometer on the development board has three axes three different lines will be graphed out, one for each axis (accX, accY, accZ). Accordingly, there are three lines for the gyroscope measurements (gyrX, gyrY, gyrZ) and three more for the magnetometer measurements (magX, magY, magZ).

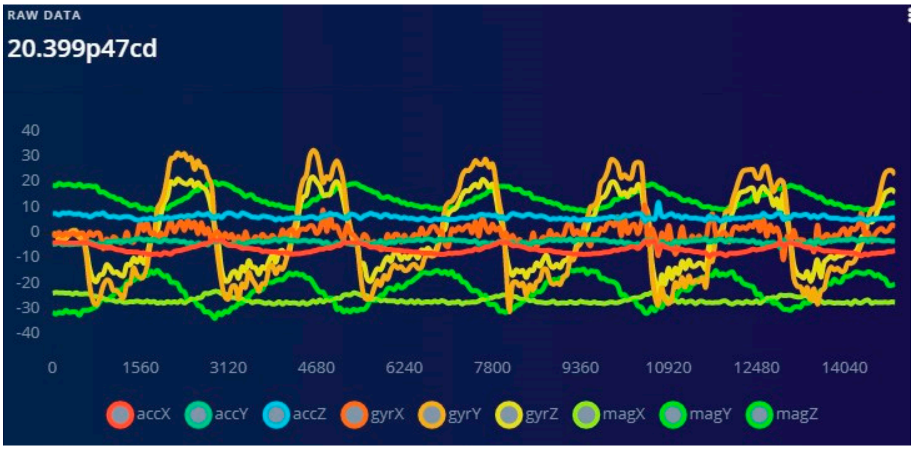


Figure 4. Samples of a Training Set.

Overall, it is quite obvious that the installation where the knee brace was used offers the most realistic environment for the experiment. Knee brace brings stability and efficiency. Arduino is held steadily on the brace and the position of it remains unchanged. The two installations of the experiment, which include the goniometer, give a more typical way to try and create a model. By simply recording the movement of a goniometer gives a falser prediction, without taking into consideration the mechanics of the movement of a human knee. By observing the anatomy of a human leg, we can understand that a goniometer cannot express the complexity of the extension and flexion movement. The knee brace on the other hand gives a more natural way of recording the flexion and extension as the device that is planned to be assembled, will be used by a patient that probably has a knee brace adjusted.

4. Machine Learning Process

In this section the machine learning process is described with all the required steps. To accurately and reliably detect the required event a machine learning model was created using the Edge Impulse platform. The main objective is to train a model that can distinguish different degrees of angles of a moving knee taking into consideration the different rotations existing in flex and extension movements as well as the fact that these are highly personalized movements as analyzed in Section 2. By making an extend-flex movement of one's knee, the model will be able to classify this movement in the given degrees. The importance of this classification is apparent in orthopedic exercises that a therapist may ask a patient to execute. The final goal, leveraging the edge computing paradigm, is to import and execute the trained model on an Arduino Nano 33 embedded platform to take advantage of the respective wireless interfaces and to send the predictions made by the model running on device. So, the patient only needs to have a wearable device and practice the "flex-extend" exercises without the need to be transported to a doctor's office or a medical/rehabilitation center.

The training phase included 3 cases to gather the most suitable training set: A goniometer with the embedded device attached (Figure 3a), a goniometer on the knee (Figure 3b) and a telescopic RangeOfMotion knee brace (Figure 3c). These 3 cases were selected in order to examine

After preparing the model, based on the subsequent evaluation process the optimal model was selected for the actual classification process. In the last subsection the edge machine learning process is presented focusing on the scenario where the selected model is transferred to the embedded device to perform the classification process based on the edge computing paradigm, thus offering increase of performance capability and resource conservation.

4.1. Machine Learning Model Training

For each installation, the steps that are already explained were followed. Firstly, the Training and the Test Set were formed. For each case, each volunteer performed leg extensions and flexion continuous movements for each angle and several samples were collected lasting 10-12 seconds each. The Training data set represents the angles 20, 45, 60, 90 and 120 °. In total, a training set consisting of 49 samples was created. These samples are almost equally distributed to the classes. Specifically, the 20° class has 9 samples, the 45° class has 11 samples, the 60° class has 10 samples, the 90° class has 11 samples and the 120° class has 8 samples.

The Test Set consists of the collected data that were captured and used to verify the model created. The Test Set consists of 13 samples, from different angles and has been gathered by the volunteers, who made some extra movements for the experiment. An 80/20 train/test split ratio for the data of every class (or label) in the dataset was used to make sure that the size of the training set is big enough and the test set represents most of the variance in the dataset.

On the Edge Impulse platform, the model was built as a pipeline using the NN Classifier and the proper parameters were selected in order to achieve the most accurate model. A neural network consists of layers of interconnected neurons and each connection has a weight. One such neuron in the input layer would be the height of the first peak of the X-axis e.g. AccX and one neuron in the output layer would be 20° (one the classes). When defining the neural network all these connections are initialized randomly, and thus the neural network will make random predictions. During training the network makes a prediction using the raw data, and then alters the weights depending on the outcome. After a lot of iterations, the neural network learns; and will eventually become much better at predicting new data. The number of training cycles shows the number of the iterations.

doi:10.20944/preprints202306.1530.v1

The parameters chosen are shown below and in the following table (Table 2) and a graph showing the neural network is shown in Figure 5. They were chosen to increase accuracy and simultaneously avoid overfitting:

- Window size is 5000ms and Window increase 50ms. The window size is the size of the raw features that is used for the training. The window increase is used to artificially create more features (and feed the learning block with more information)
- The number of training cycles was chosen to be 50 and the learning rate was 0.0005.

Table 2. Parameters and their values for the training process.

Number of training Cycles	50
Learning Rate	0.0005
Validation set size	20%
Sampling Frequency	62,5 (Hz)
Input layer	33 features
First Dense layer	20 neurons
Second Dense layer	10 neurons

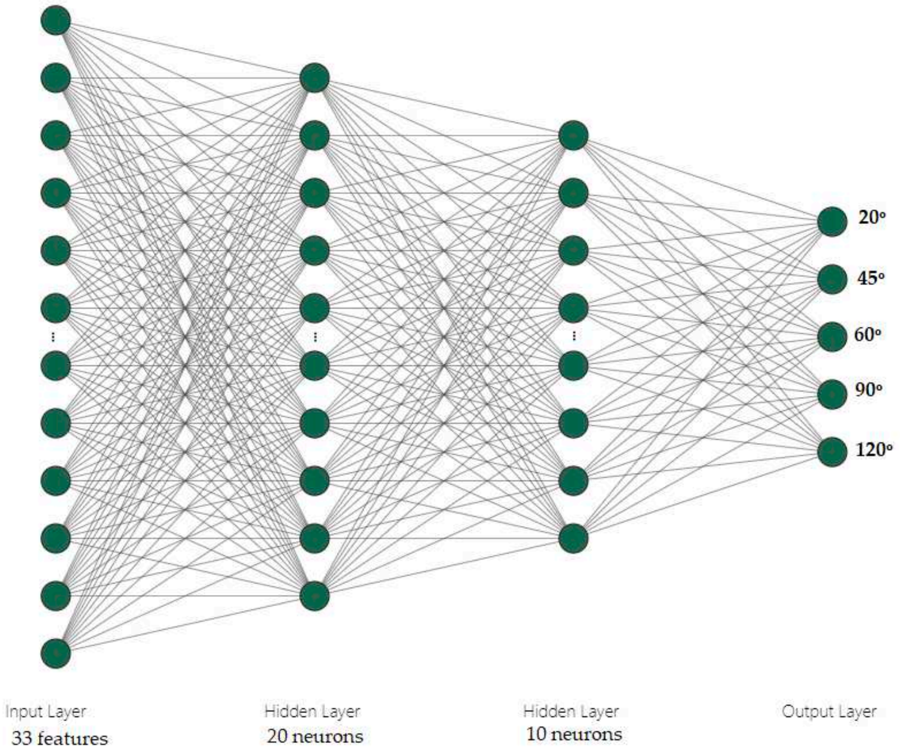


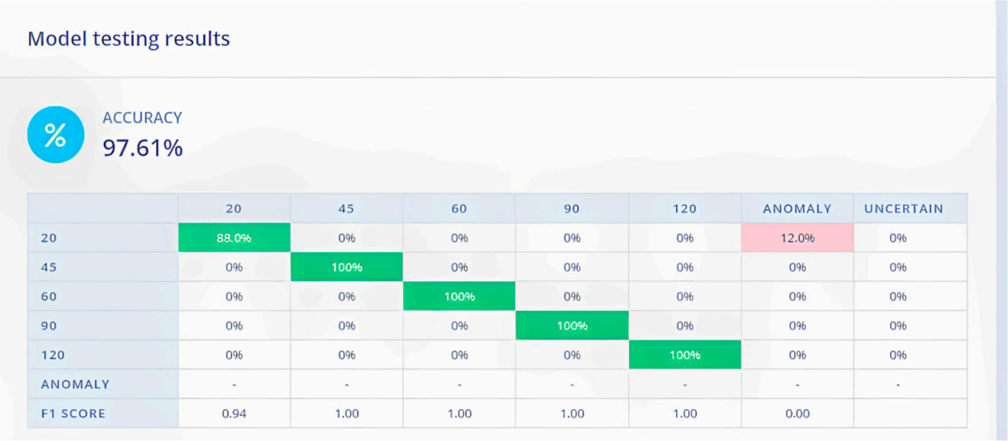
Figure 5. The neural network graph.

4.2. Machine Learning Model Creation

In Classification, an algorithm learns from a given dataset and classifies new data into a number of classes or groups. In this paper, the model aims to classify the angle of the patient's knee into one of the following classes: 20°, 45°, 60°, 90° and 120° using Neural Networks.

On the first setup, a goniometer was used, and the board was adjusted on its edge. The goniometer was placed on a steady surface. While holding the one brace steady, we started moving the other brace (with the Arduino board attached) up-and-down, forming in that way the desired angle. We tried to simulate the extension and flexion of a human knee, in the angles of interest.

In order to have a well-trained model, we ran 50 training cycles. As it is indicated in Figure 6, the accuracy of this model is very good (accuracy: 97.61%). The F1 score (a machine learning evaluation metric that combines the precision and recall scores of a model and measures a model's accuracy) is 1.00 except from the case of 20. In order to have a system that can deal with data it has never seen before (like a new gesture) edge impulse provides the Anomaly Detection block. This block was used in the system proposed so that the neural network can identify the odd values and not classify them into the predefined categories. In the confusion matrix, there is a small difficulty in discerning this case with the 12% of the predictions giving the wrong classification (Figure 6).



Model testing results

% ACCURACY 97.61%

	20	45	60	90	120	ANOMALY	UNCERTAIN
20	88.0%	0%	0%	0%	0%	12.0%	0%
45	0%	100%	0%	0%	0%	0%	0%
60	0%	0%	100%	0%	0%	0%	0%
90	0%	0%	0%	100%	0%	0%	0%
120	0%	0%	0%	0%	100%	0%	0%
ANOMALY	-	-	-	-	-	-	-
F1 SCORE	0.94	1.00	1.00	1.00	1.00	0.00	

Figure 6. Goniometer results after test set.

4.3. Machine Learning Event Detection-Classification

After having the model designed, trained and verified it was deployed back to the device so as to classify the Test Set. The Test Set was acquired with the same method that was used to create the Training set and presented in section 4.1.

By solely relying on the Test Set the model works very well. In the first case where the Arduino was attached to the goniometer it had an adequately high accuracy percentage of 88,45%. There is a small anomaly percentage on 20 degrees, meaning that the model cannot recognize some of the samples taken. It is worth noting that even at scenarios where the angles differ less than 30 degrees (45 and 60 degrees) the model's accuracy is very high and it performed successful classification results.

In the second case representing a more realistic setup without specialized equipment, the goniometer was placed on the knee of an adult person. The person (starting from complete extension of the leg) started moving the knee up-and-down, to the desired angle, according to the indication of the goniometer. The angles that were tested here are also 20°, 45 °, 60 °, 90 ° and 120 °. As this case involved realistic leg movements, the model trained achieved an overall accuracy of the model lower than the first case and equal to 78,54%. The 20-degrees scenario provided very good results reaching an accuracy of 100%. The 60 degrees scenario yields also good prediction percentage of 73,4%.

The 20 ° scenarios perform much better than the other angles- scenarios, but the overall accuracy and F1 score are lower than the ones in the other two cases (Table 3). So, although this setup is much easier to be applied, requires no specialized medical equipment and acceptable performance is offered for a range of angles, a notable deviation to the optimal accuracy is identified.

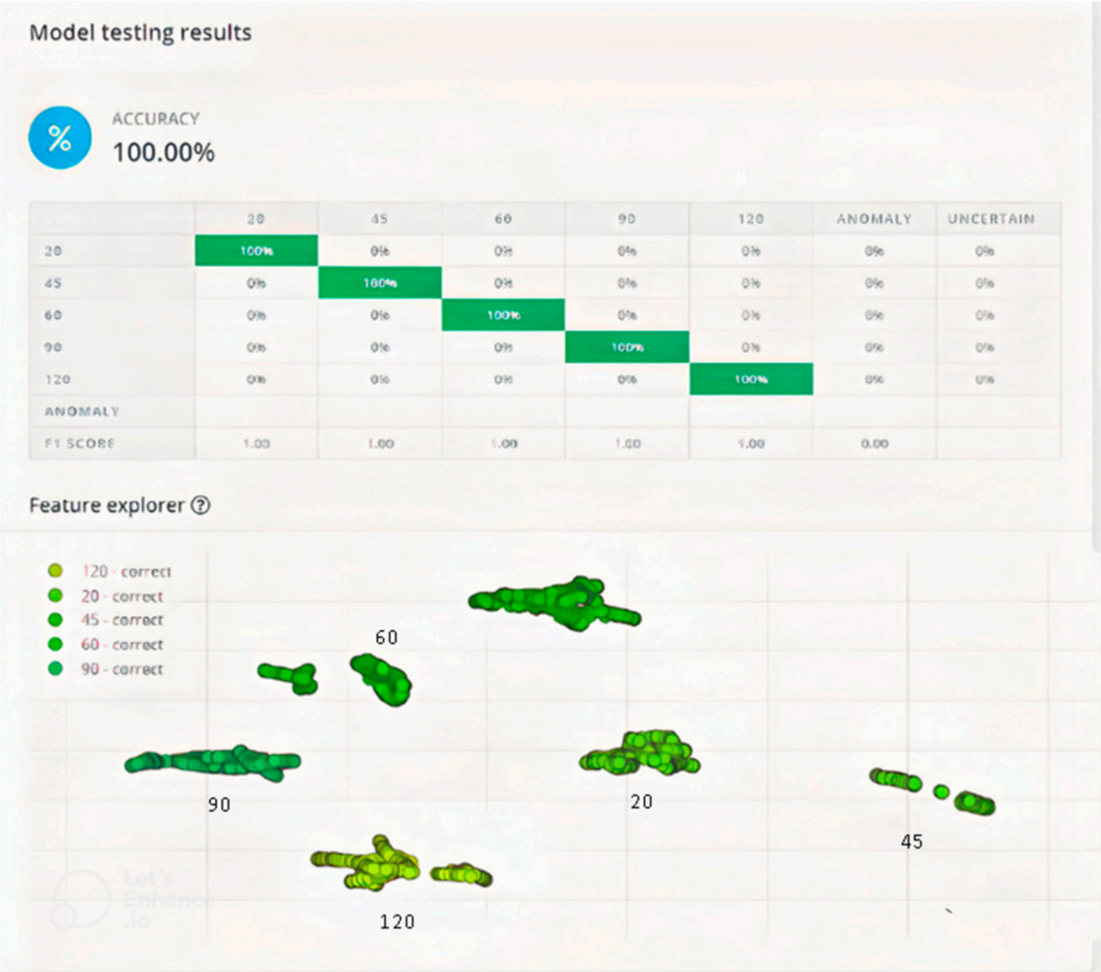
Table 3. Accuracy results of all cases.

		Accuracy						F1
		20	45	60	90	120	Overall	
Training	**Goniometer**	100%	100%	99,5%	94,9%	93,65 %	97,61%	0,98
	Goniometer on knee	100%	100%	95,7%	100%	100%	99,1%	0,99
	Knee brace	100%	100%	100%	100%	100%	100%	1.00

Classification	**Goniometer**	87,2%	76,6%	84,2%	99,8%	99,5%	88,45%	0,92
	Goniometer on knee	100%	69,7%	73,4%	76,4%	79,6%	78,54%	0,84
	Knee brace	100%	100%	100%	100%	100%	100%	1.00

In the third case a telescopic RangeOfMotion knee brace was used combining a realistic leg movement with a more controllable experiment due to the use of specialized equipment. Using the brace degree indications, we can easily choose the range of motion of the leg in angle degrees. Range of motion (ROM) means the extent or limit to which a part of the body can be moved around a joint or a fixed point; the totality of movement a joint is capable of doing. For example, by pulling and sliding the regulator to 45°, we have a range of motion (from full extend) of 45°. After adjusting the device, we properly stick Arduino Nano 33 BLE to the lowest part of the brace and we start experimenting with the different angles.

The angles tested here are also 20 °, 45 °, 60 °, 90 ° and 120 °. The main observation is that model created with considering the 3rd setup scenarios, offers excellent classification ability of the different angles we investigated. As depicted in Figure 7 the accuracy of this method is much better than the other methods, reaching up to 100% with very high F1_score (close to 1.00). Equally important the training of the specific model was much faster requiring only 19 training cycles compared to 50 in the previous setup scenarios. In the bottom of Figure 7 the graph with the correct predictions is presented where the predictions of the degrees (20 °, 45 °, 60 °, 90 °, 120°) are shown clearly clustered, which is another evidence of the well performing model created.



Model testing results

ACCURACY
100.00%

	20	45	60	90	120	ANOMALY	UNCERTAIN
20	100%	0%	0%	0%	0%	0%	0%
45	0%	100%	0%	0%	0%	0%	0%
60	0%	0%	100%	0%	0%	0%	0%
90	0%	0%	0%	100%	0%	0%	0%
120	0%	0%	0%	0%	100%	0%	0%
ANOMALY							
F1 SCORE	1.00	1.00	1.00	1.00	1.00	0.00	

Feature explorer ⓘ

Figure 7. Knee Brace Live Classification results.

For reasons of completeness, in Table 3 some samples of the test set are shown. For each sample the degrees predicted are shown, as well as the duration of the movement, the anomaly percentage and the accuracy.

Table 3. Classification samples of test set.

Prediction	Length	Anomaly	Accuracy
45	10 sec	-0,21	100%
60	5 sec	-0,16	100%
20	15 sec	-0,12	100%
120	15 sec	-0,34	100%
120	15 sec	-0,27	100%
90	15 sec	-0,25	100%
90	15 sec	-0,23	100%
60	15 sec	-0,26	100%
60	15 sec	-0,27	100%
45	15 sec	-0,20	100%
45	15 sec	-0,19	100%
20	15 sec	-0,25	100%
20	15 sec	-0,17	100%

Another worth noting result is that the first classification result of each movement lacks in accuracy. In the example explained here, the prediction is for 20 degrees and Neural Network algorithm was used as in all experiments mentioned. Figure 8 shows that in the first classification the model struggles to identify the movement as it predicts a movement of 20° with an accuracy of 54,6% and a movement of 45° with an accuracy of 45,3%. The model gives the correct classification, but the accuracy of the model is fairly low. After the first classification, and while the movement of the knee continues, the model makes the correct classification, while the accuracy percentages are improved.

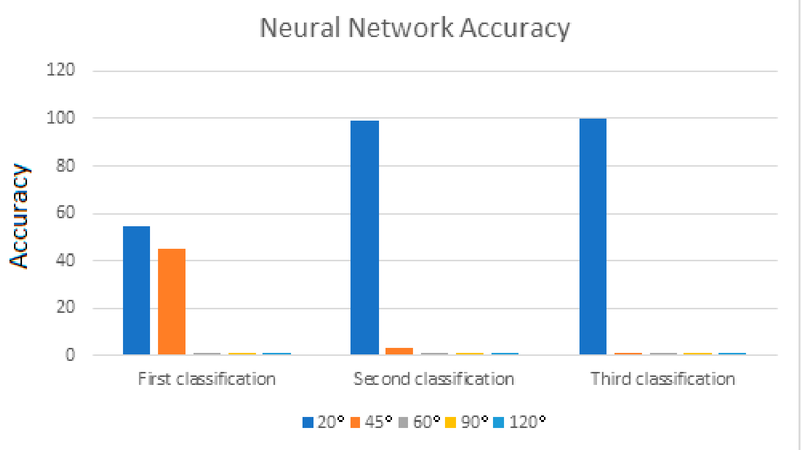


Figure 8. Classification accuracy in three first attempts.

4.4. Edge computing implementation and performance evaluation

A critical contribution of this paper is the implementation and porting of the training model to the Arduino Nano 33 BLE board to make the prediction on-device. Additionally, the code was modified in order to enable the Bluetooth Low Energy technology to send the classification result to the doctor's mobile. With BLE enabled, the decision of the algorithm can be sent to a remote station e.g. a mobile device and the angle prediction is executed on device to accurate evaluate respective performance of the edge computing implementation of proposed solution.

The objectives of this exercise are: 1) the feasibility of executing a demanding ML model on off-the-shelf, low cost, IoT embedded systems, 2) performance comparison with respect to cloud execution and 3) evaluation of the benefits yielded by such an edge computing paradigm.

In order to offer objective performance evaluation, the same users executed the same scenario in both cases (cloud and on device) and the accuracy results are quite similar (Table 4) indicating that the embedded platform is able to reliably execute the trained model.

Table 4. Accuracy results of all cases.

		Cloud	**On Device**
Classification	**Goniometer**	88,45%	87,4%
	Goniometer on knee	78,54%	80,03%
	Knee brace	100%	100%

The advantages of performing computations on the edge can be summarized below:

- One of the critical challenges when executing a ML at the cloud stem from the great volume to data that need to be acquired from the sensors and transferred wirelessly. This is especially pronounced when low bandwidth IoT communication technologies are utilized like the highly popular and widely integrated BLE communication protocol. In such cases the wireless network can easily fall into high congestion scenarios with negative effects on throughput, transmission delay and data transfer reliability metrics. This is where edge computing paradigm comes into play and can drastically reduce all these negative effects since instead of continuously streaming all the sensor raw data, the embedded platform can just send a single event detected, when it is detected. This can drastically reduce the required bandwidth effectively avoiding congestion situation as well as allowing for higher number of wireless sensors to share the same transmission medium. This case is clearly depicted in the proposed system case as follow: For each classification a batch of 60kbytes must be transferred to the cloud server. This exact transfer is achieved through a USB cable. Using the sensors of the embedded device, and before the execution of the algorithm, the sampling is completed and the data are used right away with no need for any sample transfer. In the cloud method the whole batch of measurements is transferred from the Arduino to the cloud, which is 60 kb, while in the on-device method, only the result of the prediction is transferred via BLE, which is 2-3 bytes. So, there is a bandwidth reduction of 99,9%. If consider that no event is identified (e.g. because the user is resting) then as it can easily be understood the bandwidth conservation can reach 100%.
- Another advantage promised by edge computing is reduced event detection latency in the sense that the need for data to be transferred to the cloud infrastructure to actually detect the event is effectively omitted. Such approach can significantly improve the real-time data processing and feedback provision. This is especially critical in cases where virtual coaching is offered where monitoring and feedback on specific exercises need to comply with strict time constrained demands. With the on-device method the process of sending data from the Arduino to the edge impulse platform that is on the cloud using a USB cable is bypassed. In the cloud method there is the extra step of uploading the sample data, which requires approximately 13 sec, while the 'on device' algorithm needs about 8 seconds. So, the on-device method achieves 39% time reduction in the prediction process.
- Furthermore, the fact that only detected events need to be wirelessly transmitted instead of continuously streaming all raw acquired data, allows the system to deactivate the wireless interface for extended periods of time contributing to power conservation. Considered that the wireless interfaces on such low power IoT devices comprises one of the most power-hungry

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 June 2023 doi:10.20944/preprints202306.1530.v1

components, it is easily extracted that such capability can drastically increase the lifetime and autonomy of respective devices being a possibly critical feature when commercialization is considered.

Interestingly enough, is was also indicated that the edge computing implementation was able to deliver higher number of predictions at the same time period of 60 seconds. In the "on device" case the model makes a prediction every 6 seconds (5 seconds for sampling and about 1 second to have a prediction). If we add up the 2 seconds of waiting time before sampling, every iteration of the model takes in total 8 seconds. In one minute (60 sec) we have 60/8 = 7,5 so basically 7 predictions / minute. In the other case we measured 6 predictions/minute. So, there is a 17% increase of event prediction rate when the "on device" method us used, which is explained due to the inevitable overhead of raw data transmission in the cloud case.

- Finally, a qualitative advantage concerns user privacy and safety of the data comprising an equally important advantage of edge computing especially in the health domain. Edge devices have the ability to discard information that need to be kept private, as only certain information and with special encoding.

4.5. Challenges for future work

Following a complete experimentation set of the indicated three different setups where the embedded system board was used to acquire the required data and also gathered the model testing results, in this section we discuss the main challenges we encountered.

In the first setup case where the Arduino Nano 33 is attached to the goniometer, we can see that the uncertainty is higher than the two other cases. This observation can be attributed to the fact that besides the desired 'up-down' movement, on the vertical axis, the goniometer also moves even slightly in the horizontal axis. The way the goniometer is assembled gives this flexibility. Specifically, the goniometer's moving brace is connected with the circular part of the goniometer with a small metallic bolt. These small vibrations of the goniometer in the horizontal axis makes the acquired training set samples more complex to distinguish as it is less stable than the Knee brace.

In general vibrations and position of Arduino play an important role on the results of the model. We can infer that even small changes on direction and position of Arduino or any vibrations could give different results or bring uncertainty to the models.

As for future work we plan to expand the training dataset including more people and more arthroses movements and address the challenges mentioned before. We will also compare the results of this study with another method of motion recognition using computer vision. Finally, an integrated platform is planned to be implemented that will offer a complete knee rehabilitation plan to the patients.

5. Conclusions

In this paper we have illustrated an edge computing capable system in detail, described the algorithms used and shown the basic scenario of a high impact health application's usage. The proposed platform is based on an integrated embedded IoT platform and cloud service that facilitates the automatic knee movement detection based on both classic machine learning techniques and on the edge classifications which is critical in the context of a postoperative rehabilitation monitoring. All the system components and processes are analyzed in detail validating the importance as well as feasibility of the solution proposed. A number of tests were performed for evaluating the developed techniques both qualitatively and quantitatively in order to test their stability and their accuracy which reached 100%. After concluding to the most accurate model, respective tests were undertaken to compare the on-device prediction and the classic machine learning process with great results as with the edge classification were transferred 99,9% less data, with a 39%-time reduction and a 17% increase of event prediction were achieved. Finally, the main challenges and future prospects of this work are discussed while the dataset of labeled knee movements is freely available on https://www.kaggle.com/datasets/billskarm/knee-range-of-motion to the research community comprising a critical contribution of the work conducted.

Acknowledgments: This work has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No 872614 -SMART4ALL: Selfsustained Cross-Border Customized Cyberphysical System Experiments for Capacity Building among European Stakeholder.

Ethical Statements: All 10 volunteered subjects gave their informed consent for inclusion before the participation in the study. The study is conducted in accordance with the Declaration of Helsinki, and the protocol is approved by the ESDA Ethics Committee.

References

1. Spekowius, G.; Wendler, T. (Eds.). Advances in healthcare technology: shaping the future of medical care; Springer Science & Business Media, 2006; Volume 6.
2. Czaja, S.; Beach, S.; Charness, N.; Schulz, R. Older adults and the adoption of healthcare technology: Opportunities and challenges. Technologies for active aging, 2013; pp. 27-46.
3. Sundaravadivel, P., Kougianos, E., Mohanty, S. P., & Ganapathiraju, M. K. Everything you wanted to know about smart health care: Evaluating the different technologies and components of the internet of things for better health. IEEE Consumer Electronics Magazine, 2017; Volume 7(1), pp. 18-28.
4. Poitras, I., Dupuis, F., Bielmann, M., Campeau-Lecours, A., Mercier, C., Bouyer, L. J., & Roy, J. S. Validity and reliability of wearable sensors for joint angle estimation: A systematic review. Sensors, 2019; Volume 19(7), pp. 1555.
5. Dejnabadi, H., Jolles, B. M., & Aminian, K. A new approach to accurate measurement of uniaxial joint angles based on a combination of accelerometers and gyroscopes. IEEE Transactions on Biomedical Engineering, 2005; Volume 52(8), pp. 1478-1484
6. Godfrey, A. C. R. M. D. O. G., Conway, R., Meagher, D., & ÓLaighin, G. Direct measurement of human movement by accelerometry. Medical engineering & physics, 2008; Volume 30(10), pp. 1364-1386.
7. Tsoukas, V., Boumpa, E., Giannakas, G., & Kakarountas, A. A review of machine learning and tinyml in healthcare. In 25th Pan-Hellenic Conference on Informatics 2021; pp. 69-73.
8. Fedorov, I., Stamenovic, M., Jensen, C., Yang, L. C., Mandell, A., Gan, Y., ... & Whatmough, P. N. TinyLSTMs: Efficient neural speech enhancement for hearing aids. 2020; arXiv preprint arXiv:2005.11138.
9. Fyntanidou, B., Zouka, M., Apostolopoulou, A., Bamidis, P. D., Billis, A., Mitsopoulos, K., ... & Fourlis, A.. IoT-based smart triage of Covid-19 suspicious cases in the Emergency Department. In 2020 IEEE Globecom Workshops, 2020; pp. 1-6
10. Gupta, S., Jain, S., Roy, B., & Deb, A. A TinyML Approach to Human Activity Recognition. In Journal of Physics: Conference Series, 2022 ; Vol. 2273, No. 1, p. 012025. IOP Publishing.
11. Warden, P., & Situnayake, D. Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers. O'Reilly Media, 2019.
12. Weber, M. D., & Woodall, W. R.. Knee rehabilitation. Physical Rehabilitation of the Injured Athlete; Expert Consult-Online and Print, 2012; 377.
13. American Orthopaedic Society for Sports Medicine: https://www.sportsmed.org (accessed on 29/03/2023)
14. Inacio, M., Paxton, E. W., Graves, S. E., Namba, R. S., & Nemes, S. Projected increase in total knee arthroplasty in the United States - an alternative projection model. Osteoarthritis and cartilage, 2017; 25(11), 1797–1803. https://doi.org/10.1016/j.joca.2017.07.022
15. Bell, K. M., Onyeukwu, C., Smith, C. N., Oh, A., Devito Dabbs, A., Piva, S. R., ... & McClincy, M. P. A portable system for remote rehabilitation following a total knee replacement: a pilot randomized controlled clinical study. Sensors, 2020; Volume 20(21), pp. 6118.
16. Burland, J. P., Outerleys, J. B., Lattermann, C., & Davis, I. S. Reliability of wearable sensors to assess impact metrics during sport-specific tasks. Journal of sports sciences, 2021; Volume 39(4), pp. 406-411.
17. Luna, I.E.; Kehlet, H.; Peterson, B.; Wede, H.R.; Hoevsgaard, S.J.; Aasvang, E.K. Early patient-reported outcomes versus objective function after total hip and knee arthroplasty. Bone Jt. J. 2017; Volume 99-B, pp. 1167–1175.
18. Yi, C., Jiang, F., Bhuiyan, M. Z. A., Yang, C., Gao, X., Guo, H., ... & Su, S. Smart healthcare-oriented online prediction of lower-limb kinematics and kinetics based on data-driven neural signal decoding. Future Generation Computer Systems, 2021; Volume 114, pp. 96-105.
19. Edge Impulse. Available online: https://www.edgeimpulse.com/ (accessed on 21/2/2023)
20. Arduino cc. Available online: https://docs.arduino.cc/hardware/nano-33-ble-sense (accessed on 21/2/2023)

21. Antonopoulos, C. P., Antonopoulos, K., Panagiotou, C., & Voros, N. S. Tackling Critical Challenges towards Efficient CyberPhysical Components & Services Interconnection: The ATLAS CPS Platform Approach. Journal of Signal Processing Systems, 2019; Volume 91(11-12), pp. 1273-1281.
22. Knee angle recognition dataset: https://www.kaggle.com/datasets/billskarm/knee-range-of-motion (accessed on 18/3/2023)

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.