Embedded neural network for real-time animal behavior classification

D. Gutierrez-Galan, Juan P. Dominguez-Morales, E. Cerezuela-Escudero, A. Rios-Navarro, R. Tapiador-Morales, M. Rivas-Perez, M. Dominguez-Morales, A. Jimenez-Fernandez, A. Linares-Barranco

Robotic and Technology of Computers Lab, Department of Architecture and Technology of Computers, University of Seville, Av. Reina Mercedes s/n, 41012Sevilla, Spain

Keywords:
Monitoring wildlife
Wireless sensor network
Neural network
Multilayer perceptron
Sensor fusion
Embedded device

A B S T R A C T

Recent biological studies have focused on understanding animal interactions and welfare. To help biologists to obtain animals' behavior information, resources like wireless sensor networks are needed. Moreover, large amounts of obtained data have to be processed off-line in order to classify different behaviors. There are recent research projects focused on designing monitoring systems capable of measuring some animals' parameters in order to recognize and monitor their gaits or behaviors. However, network unreliability and high power consumption have limited their applicability.

In this work, we present an animal behavior recognition, classification and monitoring system based on a wireless sensor network and a smart collar device, provided with inertial sensors and an embedded multi-layer perceptron-based feed-forward neural network, to classify the different gaits or behaviors based on the collected information. In similar works, classification mechanisms are implemented in a server (or base station). The main novelty of this work is the full implementation of a reconfigurable neural network embedded into the animal's collar, which allows a real-time behavior classification and enables its local storage in SD memory. Moreover, this approach reduces the amount of data transmitted to the base station (and its periodicity), achieving a significantly improving battery life. The system has been simulated and tested in a real scenario for three different horse gaits, using different heuristics and sensors to improve the accuracy of behavior recognition, achieving a maximum of 81%.

1. Introduction

The study and monitoring of wildlife has always been a subject of great interest since the quantitative measurement of animal behavior is an important tool for understanding their reproduction, survival, welfare and interaction with other animals. It is important to study the motion patterns of wild animals and how they may be affected by changes in weather, flora or by the introduction of non-native species. Learning such details about wildlife requires long-term activity logs as well as other biometric data such as heart rate, body temperature, movement speed and frequency of feeding. Therefore, the design and deployment of a monitoring system capable of obtaining behavioral information from animals has been the focus of several studies [1–5].

Collecting and processing relevant information from wildlife is a hard technological task [6–8] due to several factors that need to be solved: (1) the development of lightweight and lower power-consumption devices to attach to the animal, (2) the design and implementation of a wireless network to collect the information from those devices, and (3) storing the data and its further processing mechanisms.

The behavioral parameters of an animal can be measured using different types of sensors. With this data, different communication strategies can be deployed to send the collected information. The traditional Very High Frequency (VHF) radio-tracking system collapses as soon as it starts using multiple collars due to the scarcity of frequencies assigned. Using this kind of radio-tracking, the researchers have to move through the experimental area with a receiver antenna looking for collared animals. Therefore, data collection is infrequent and limited to daylight hours or to research operating hours [9]. Satellite localization mechanisms are so expensive that only migratory animals were used, as in [10]. There are recent studies that have aimed at designing wireless sensor networks that are capable of measuring specific behavioral parameters and transmitting them over a wireless network to a central base station [3,5,11]. This strategy overcomes the disadvantages of the

Table 1
Qualitative study between relevant works in the area.

	Used sensor	Collected data	Recognition method	Output classes	Real-time classification
Ref. [16]	2-axis accel.	Online	MLP ANN	5	No
Ref. [17]	GPS & 3-axis accel.	Online	Decision tree	5	No
Ref. [18]	3-axis accel.	Offline	Decision tree	3	No
This work	3-axis accel.	Online	Embedded MLP ANN	3	Yes

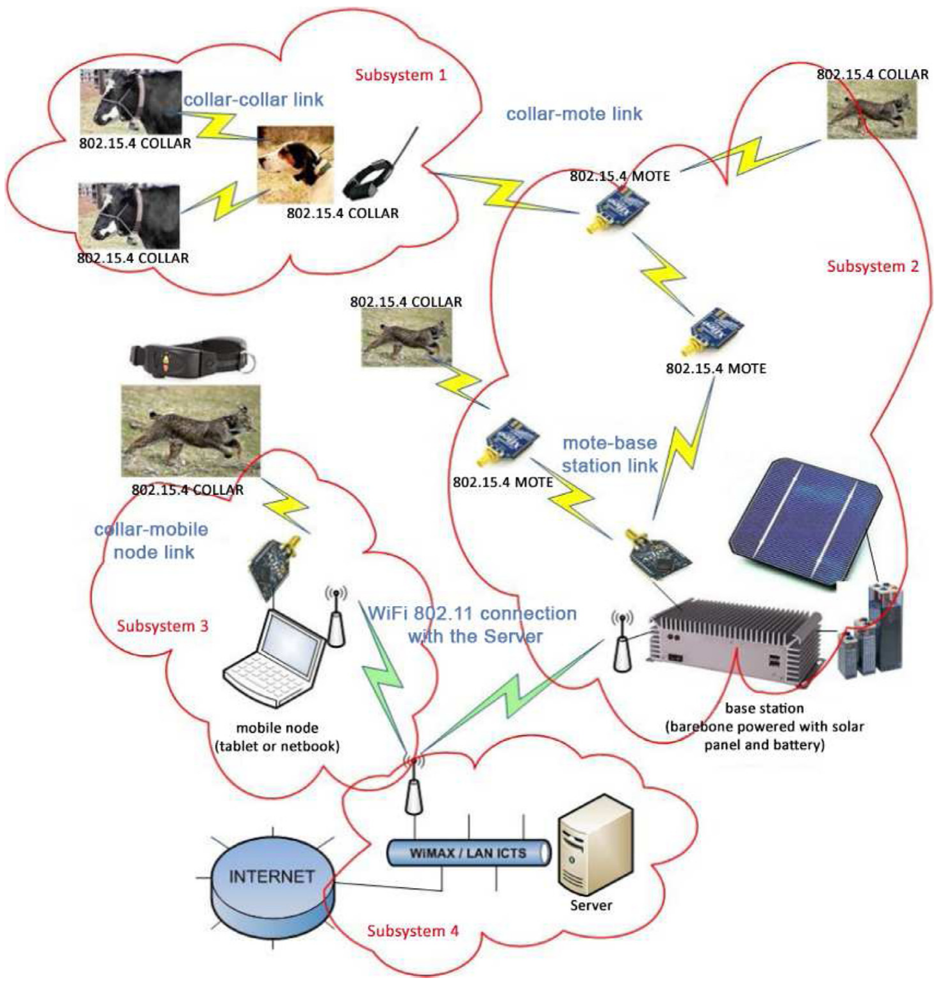

Fig. 1. Network topology architecture.

two previous ones. Global positioning systems (GPS) is the most popular sensor used in outdoor environments to estimate the temporal and spatial distribution of animals [1–3,8,9,12]. Some of these studies also infer animal activity from GPS tracking data [1–3]. In [2], the position and acceleration obtained from the GPS is used to infer different activity states from cattle. In this study, discriminant analysis classification agreed with human observation for at least 74% of the data while regression tree classification agreed with human observation for at least 84% of the data. In [1], the authors compared the classification results from human observers with those obtained from discriminant analysis of GPS data and found that the two were in agreement for 71% of the data. Both papers classify between three different behaviors: grazing, travelling and resting. In the research carried out in [3], the behavior of a herd of dairy cows is classified using the animal tracking data obtained from the GPS into two classes (active and inactive) using a*k*-means classifier.

When using the GPS sensor for animal monitoring, high capacity memory cards to store the information and long life-batteries are needed because of the power consumption, which usually are not lightweight. Therefore, high-power consumption and frequent loss of connection with the satellites in the areas of a field covered by obstacles (e.g., trees) are the main drawbacks of GPS-based monitoring systems. In addition, current studies have identified the difficulty of balancing data resolution with technical limitations, particularly the power and memory requirements of the animal-attached device [3,8].

The use of inertial sensors, like the accelerometer, gyroscope and magnetometer, overcomes the disadvantages of GPS and allows to obtain information about the entire range of the animal's body movements [4,5,13,14]. In [13] a body attitude (orientation) estimation for free ranging animals using an Inertial Measurement Unit (IMU) is described. In [14] the authors placed offline pitch-roll sensors around the neck of each sheep in a herd. The data was analyzed using three classification methods: a linear discriminant analysis, a classification tree method, and a manually developed decision tree consisting of four “if then” loops. All three methods provide very good classification predictions with more than

90% accuracy, distinguishing between two activity categories: active and inactive.

Regardless of the type of sensors used to monitor animals, large amounts of data are needed when studying their behavioral patterns, implying important analytical and interpretative steps to process the information. Algorithms that look for particular behavioral patterns based on the input data usually conduct this kind of recognition or classification. Some of these algorithms are Neural Networks (NN), Support Vector Machines (SVM) or even complex statistical methods, which can detect specific behaviors such as sleeping, running, copulating, etc. Generally, the computational costs of some of these algorithms are high enough to require specific platforms capable of parallelizing computations for this classification. Supervised neural networks, such as feed-forward networks, are particularly well suited for modeling and controlling dynamic systems, classifying noisy data, and predicting future events [15].

In [16], a 2.4-GHz ZigBee-based mobile ad hoc wireless sensor network to collect information from sheep and send it to a base station is presented; and, also, a multilayer perceptron (MLP) based artificial neural network (ANN) to obtain the corresponding behavior from the gathered data in an online way is described. The accuracy rate of the network is 76.2% (classifying between five different classes: grazing, lying down, walking, standing and others).

In this work, we propose a hierarchical wireless sensor network to collect information about animals activity using low-power consumption intelligent devices placed on them which contain a neural network implementation to classify their behavior based on sensory information. Therefore, we propose an online monitoring system capable of real-time classification of the animal behavior. The NN is designed and trained offline using a software tool and then all of its training parameters and configuration are used on an embedded version of the NN, which is implemented on a low-power microcontroller. The main novelty of this work with respect to related studies lies on the in-collar behavior classification.

Table 1 summarizes a qualitative study between the most relevant works in this area and this work, comparing the used sensor, such as methods and features, as real-time classification capability.

As can be seen, all works use an accelerometer as sensor, and most of them collect the samples online, i.e., it is not necessary to take off the device from the animal, since they are sent by wireless network. As recognition method, decision tree and MLP based ANN are commonly used because both are able to achieve a good hit average, classifying between three or five classes in these cases. Real-time classification is only achieved by this work, and it is the main novelty in comparison with the others.

MINERVA is a research project whose main aim is to study and classify wildlife behavior inside Doñana National Park [19]. Nowadays, tracking and classification systems for wildlife used in Doñana National Park obtain positional information using a GPS and transmit it via GSM (by SMS). However, to reduce the power consumption, the position is obtained between two and five times a day. These solutions are not enough for biologists interests: they need more information to recognize animal behaviors. To solve this lack of information, two solutions can be implemented: the system can be adapted to transmit information more regularly (since communications consume in average more than 80% of battery life, which makes this option inefficient); or, on the other hand, this information can be processed locally in order to classify the animal behavior and transmit only the behavior itself. Therefore, instead of sending the information after every sensor read, the communication to the network only occurs after several sensor reads; this fact increases battery life but keeps the information of animal behaviors. Viability and power consumption studies for these two approaches have been carried out by the authors in [20]. This project has the additional aim of developing an infrastructure for collecting this information and make it accessible through the internet. The pattern recognition of the sensed data is performed in real time by the microcontroller using a low-power implementation of a NN that classifies three different horse gaits [21] (motionless, walking and trotting). This information is transmitted using a mesh wireless multisensory network distributed on collars placed on some animals. This multisensory network reads data from the sensors and sends them to a network of motes, which acts as a router and retransmits these packets to a base station. This base station receives the information through the network and uploads it to a remote server database using Doñana National Park's Wi-Fi connection. Researchers can access this data using a web-based user interface and track the animal activity and its location in real time without the necessity of being in Doñana National Park [19].

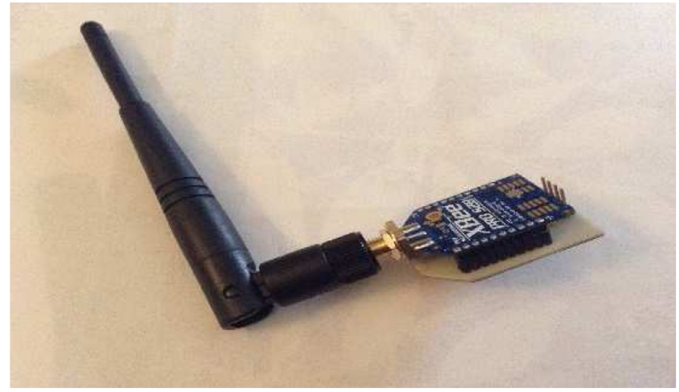

Fig. 2. Mote device without battery.

The paper is structured as follows: Section 2 presents the wireless sensor network and its architecture. Section 3 describes the collar device. Then, Section 4 presents the Fast Artificial Neural Network library. Section 5 describes the experiments that have been carried out, the testing scenario and their results. In Section 6, the authors discuss the results obtained and previous studies related with this and future works using the current state of the project as a starting point. Finally, Section 7 presents the conclusions.

2. Network topology

The main aim of the network is to obtain behavioral information from wildlife and store it in a remote database server. This way, researchers can access this information through a website for further research and studies. The classification information is collected and sent through the network by collars that are placed on the animals and that consist of several sensors, while a set of motes transmit it to the base station, which is located in the park. Due to the fact that capturing a wild animal to replace its collar is very expensive, the whole system is designed to have the lowest power consumption possible. That is why the behavioral classification is done in an embedded neural network on the collar, transmitting only the classification result instead of the obtained information from each of the sensors that the collars have. The study that the authors have carried out in [20] shows that the data transmission is, by far, the action that consumes more battery life. In this study, a 2.4-GHz ZigBee-based [22,23] wireless sensor network is used. Fig. 1 shows the WSN topology architecture.

ZigBee defines three different device types: Coordinator, Router and End Device which correspond with the ones that have been used in this network: base station, motes and collars, respectively. The main goal of the base station is to receive data packets from the collars and retransmit them to a remote database server using Doñana National Park's Wi-Fi connection. Moreover,

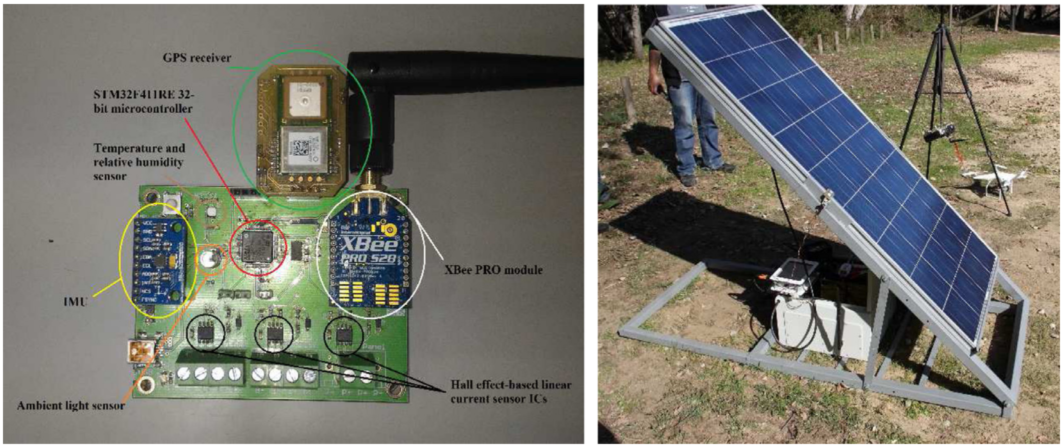

Fig. 3. BridgeBoard (left) and base station (right).

the base station has other important functionalities, focused on controlling park environmental conditions, like temperature or humidity. This device consists of several elements, such as an Intel NUC barebone,[1] a 60A battery to power the system, a solar panel (1476 $\times$ 659 $\times$ 35 mm) to charge the battery during daylight and a printed circuit board (PCB) called BridgeBoard which contains different sensors and the XBee module to receive packets from collars and motes. The battery, NUC and BridgeBoard are protected by a metallic waterproof box, which allows to place the base station outdoors without any risk. Everything has been mounted on a dedicated metallic structure designed to avoid the effects of hard weather conditions. The motes are XBee devices configured as ZigBee routers which are placed surrounding the base station. Their main goal is to expand the coverage area and provide a communication spot between collars and the base station in case that the collars are out of the coverage range of the base station. Figs. 2 and 3 show the mote device and the base station, respectively.

Although each device in this network has a very important role in the MINERVA project, in this work we will focus on the collars, which are the most relevant ones in terms of the classification of wildlife behavior. More information about all these elements can be found in [20].

3. Collar

The collar collects information from the animal on which it is placed by using different sensors. It has a MinIMU-9V2 inertial measurement unit (IMU), which consists of a LSM303DLHC 3-axis accelerometer, a L3GD20 3-axis gyroscope and a 3-axis magnetometer. An I2C interface accesses nine independent rotation, acceleration, and magnetic measurements that can be used to calculate the sensor's absolute orientation. All of these sensors have 12-bit resolution for a more precise data acquisition. The IMU is used in addition to a GPS, which provides location and time information in all weather conditions. The main aim of the collar is, using the information obtained from the IMU, to classify the animal's behavior (between three different gait patterns) using this data as an input for a feed-forward neural network implemented on the collar's microcontroller unit (MCU). The periodical measures of each sensor are carried out using a low power microcontroller (STM32L152 [ref]) with a real-time operating system (RTOS) which is powered using a four AAA battery pack (1.5 V, 1155 mAh each).

The collar prototype (see Fig. 4) has an XBee module (XBee PRO S2B [6]) that can transmit data through a wireless network.

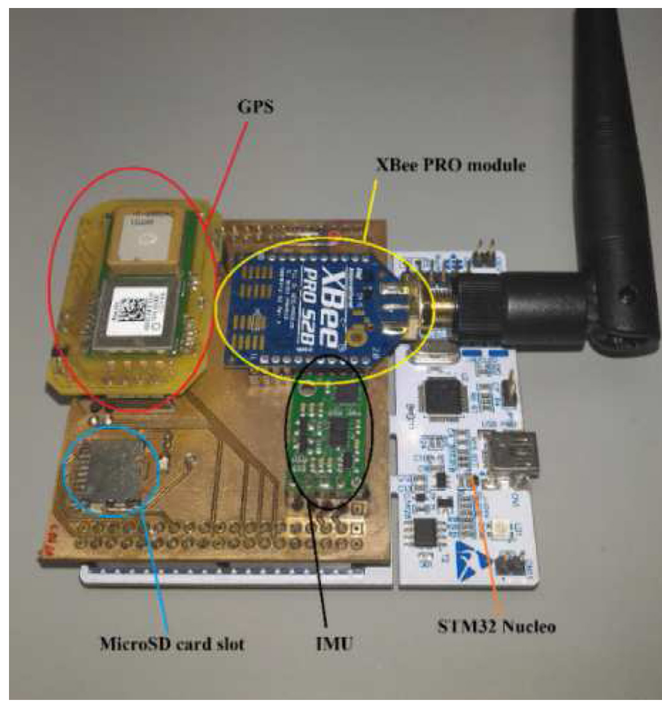

Fig. 4. Collar device prototype.

XBee modules are integrated solutions based on ZigBee, which is an open global standard of the IEEE 802.15.4 MAC/PHY [22,23]. This device family allows to implement a mesh network of motes (or routers) where collars (or devices) send information, and other elements (coordinators) of the network redirect these packets to a web server. The main objective is to transmit sensed information to the nearest router of the network, so that it can reach the coordinator and upload this information to the database server. If the signal cannot reach a valid point to transmit, i.e., the animal is out of the network coverage, the collar carries a microSD card where the information is stored, so that the animal behavioral information can be accessed later or offline, avoiding data loss. Using the XBee module the collar will send the recognized gaits to a base station that will upload this information to a database server on the internet.

Due to the fact that capturing a semi-wild animal is very expensive and difficult, the microcontroller is able to switch to sleep mode if there are no routers in the network coverage capable of

[1] http://www.intel.com/content/www/us/en/nuc/nuc-kit-d54250wykh.html.

receiving this collar's information, increasing battery life. Moreover, the measures are transmitted periodically according to a frequency value that is established and that can be modified, reducing radio transmissions and thus, reducing power consumption.

4. Fast Artificial Neural Network library

The main goal of project MINERVA is to monitor wild animals, get the behavior information based on their activity levels, which are obtained from the sensors that are placed on the collar that was described in the previous section, and then, send this information to a database server on the Internet for further research by Doñana's biologist staff. For this purpose, a classifier system needs to be used in order to predict which one of the studied behaviors (motionless, walking and trotting) is the animal performing based on the sensor information. Since neural networks are presently offering better accuracies at solving these classification problems, we have explored their use in this work by embedding them into an embedded system.

The neural network has been deployed in the collar's microcontroller using the Fast Artificial Neural Network library [24], which is a free open source neural network library that implements multilayer artificial neural networks in C programming language. It is easy to use, well documented, versatile and allows to use both floating point and fixed point numbers. It also has bindings with more than 20 programming languages and several graphical user interfaces (GUIs), although in this paper the standard C library with neither wrappers nor GUIs is used. Fixed point numbers are used in this work due to the microcontroller's lack of Floating Point Unit (FPU) because of low-power consumption requirements.

Two different versions of the library are used in this work. The first one is the full fixed point FANN library, which can be downloaded from the Github page of the project. This version is used for training the neural network and, after this step, test the dataset with the configuration that has been obtained in the training phase. This whole simulation process is done in the PC just for testing how good the classification results would be before deploying the neural network configuration on the microcontroller.

The second version of the FANN library that we have used corresponds to a modified version we have performed. In this second case, we have improved the performance and removed some parts for power saving, such as: training, floating point operations and other non-necessary classes and functionalities when it comes to obtaining the classification results. This second version fits in the collar's microcontroller. The reasons why the training functionality has been removed from the vanilla version of the FANN library are: (1) the microcontroller has limited processing capabilities and memory; (2) the computational cost of the training step (which reduces battery life); and (3) the training process will still be done in the PC due to the fact that, for this purpose, the training step does not need to be done in the collar.

In general, tests (which will be detailed in the next section) were performed using a multi-layer perceptron feedforward neural network [15]. This kind of NN is the standard algorithm in pattern recognition tasks, trained by a backpropagation algorithm. It consists of three or more layers: an input layer, an output layer, and one or more hidden layers. In this work, an input layer with three, six or nine inputs (depending on the data set used), one hidden layer with ten, twenty or thirty neurons (to see if it improves) and an output layer with three neurons (it always has three outputs, one per gait) was implemented, trained and tested using the data collected in Doñana.

The FANN library allows to choose between different activation functions: linear, threshold, threshold symmetric, sigmoid stepwise, sigmoid symmetric stepwise, linear piece and linear symmetric piece [25]. However, in this work, we have used Sigmoid Symmetric Stepwise due to the fact that it is the same as the one that we used in the experiments that were carried out in [5], so that the results can be compared using exactly the same network architecture. This activation function gives an output that is between -1 and 1. It is a stepwise linear approximation to symmetric sigmoid and faster than symmetric sigmoid, but a bit less precise. For training, FANN supports a set of training algorithms, but the default and most used one is the backpropagation algorithm [26]. Initialization algorithms, such as the Nguyen–Widrow [27], have not been used in this work due to the initialization process being provided by the FANN library. However, good results were obtained in the first tests, so this step will not be considered in the future. Some parameters like the mean squared error (MSE) and the number of epochs can be configured to improve the resulting accuracy, and they will be set experimentally as tests will be performed.

5. Experimental results and comparison

In this section, we present the experimental results obtained from the classification system using the FANN library, varying both the number of neurons in the hidden layer and the input dataset between raw (unprocessed) and different filtered sensor data. These tests were performed in three different testing scenarios: (1) training and testing the NN using the full FANN library on the PC, (2) training the NN on the PC using the same library and testing it on the embedded version of the FANN library running inside the collar, and (3) using the same training parameters that were obtained in previous experiments and testing the network on a different real scenario (the collars are placed on different horses and, after the experiment is done, the average accuracy ratio is calculated). A comparative study of the performance between the FANN library and a previous work performed by the authors in [5], in which we used the Matlab Neural Network Toolbox for classifying between the same three horse gaits that are intented to be classified in the present study, was conducted in order to determine which library behaves better using the same NN architecture, training and datasets.

5.1. Simulation tests

The first part of this study is to train the classifier system using the FANN library and to perform several simulation tests, using both a PC and a collar (which is not placed in a horse), where the same NN architectures are implemented. These architectures are MPL-based feed-forward NNs with three layers: the input layer, one hidden layer and the output layer. The input layer contains 3, 6 or 9 neurons (depending on which sensors are used from the input dataset), and the number of neurons in the hidden layer is set to 10, 20 or 30 neurons. The output layer consists of 3 neurons, corresponding to the three behaviors to be classified (motionless, walking and trotting). The transfer function used is the Sigmoid Symmetric Stepwise, as said in the previous section.

To train the NN, the widely used backpropagation algorithm is used, as it is the default training mechanism that FANN implements, which also is the same that was used in the Matlab Neural Network Toolbox in the previous work [5]. The length of the whole dataset for both the training and the test is 30,000 samples, which were obtained during different visits to Doñana's National Park from the IMU sensors inside the collar while a horse performed three different behaviors: motionless, walking and trotting. The sensors described in Section 3 gathered data every 33 ms, so the sampling frequency is 30 Hz. The data was obtained at different seasons of the year between which the weather conditions were definitely not the same. The horses used for the collection of data had approximately the same characteristics in terms of height, weight and age. The samples were randomly divided into three

Table 2
Hit rate percentages using the full version of the FANN library and unprocessed (raw) sensor data as input.

Neurons in hidden layer	Classes	Sensors used				
		Acceler. (%)	Gyro. (%)	Magnet. (%)	A&G&M (%)	A&G (%)
10	Trotting	75.58	49.6	55.74	77.04	74.62
	Motionless	81.98	69.26	55.40	82.54	83.16
	Walking	81.12	54.46	66.14	82.34	81.58
	Average	**79.56**	**57.79**	**59.12**	**80.64**	**79.79**
20	Trotting	75.00	53.18	58.08	76.68	74.98
	Motionless	81.34	69.62	58.50	86.44	86.62
	Walking	80.38	56.12	67.70	82.36	81.98
	Average	**78.91**	**59.64**	**61.43**	**81.83**	**81.19**
30	Trotting	75.84	53.48	58.42	76.32	74.74
	Motionless	81.52	70.18	58.64	87.50	86.42
	Walking	80.30	56.98	68.30	82.50	81.68
	Average	**79.22**	**60.21**	**61.79**	**82.11**	**80.95**

Table 3
Hit rate percentages using the full version of the FANN library and filtered sensor data as input.

Neurons in hidden layer	Classes	Applied filter	
		Kalman (%)	FreeIMU(%)
10	Trotting	88.84	61.54
	Motionless	97.32	53.04
	Walking	99.98	63.00
	Average	**95.38**	**59.19**
20	Trotting	89.67	62.74
	Motionless	96.59	57.90
	Walking	99.99	66.92
	Average	**95.42**	**62.52**

sets; the following cross-correlation scheme is used for every experiment in this work: 70% for training, 15% for validation and 15% for testing the network.

5.1.1. Using the full version of the FANN library

The first performance test used the raw sensor data, thus the NN has three, six or nine inputs (x, y and z for each 3-axis sensor of the IMU). To evaluate the importance of each IMU sensor when it comes to recognizing animal behaviors, the NN was trained and tested with different combinations of these sensors. On the other hand, for the second performance test, Kalman and FreeIMU preprocessing algorithms were used. These samples were obtained when applying Kalman and FreeIMU filters to the accelerometer, gyroscope and magnetometer raw data in real-time when the collar MCU collected this information [28]. In this case, three neurons are used in the input layer of the NN (these algorithms obtained three values: pitch, roll and yaw from the IMU sensors). 10, 20 and 30 neurons were used in the hidden layer in both experiments.

For each NN architecture that has been presented, several training-testing steps were performed to calculate the results in terms of average accuracy ratio. The results of these tests using both sensor raw data (unprocessed) and Kalman and FreeIMU filters are presented in Table 2 and Table 3, respectively.

The results that can be seen in Table 2 show that the accelerometer is the sensor that has the most valuable information about the horse movement, while the gyroscope and magnetometer improve the pattern definition. The classifier system has an accuracy of 82.11% using 30 neurons in the hidden layer. In Table 3, the hit rate of our classification system using Kalman filtered data is around 95.4% regardless of the number of neurons in the hidden layer.

5.1.2. Using the embedded version of the FANN library

The aim of this test is to determine if the performance for running the embedded version of the FANN library inside the collar differs from running the full version of the FANN library in a computer. For this purpose, the same experiments that were performed in the previous subsection were carried out using this library inside the collar.

The NN was trained in a computer and, after that, the weights of the connections and the rest of the training parameters that are necessary to create the NN were generated and implemented in the collar.

As can be seen in previous tests, for each architecture that has been presented, several training-testing steps were performed to calculate the results in terms of average accuracy ratio. The collar was not placed on the animal. The results of these tests using both raw sensor data (unprocessed) and Kalman and FreeIMU filtered data are presented in Table 4 and Table 5, respectively.

From Table 4, these results show that the accelerometer is the sensor with better information about the horse movement, while the gyroscope and magnetometer improve the pattern definition. The classifier system has an accuracy of 82.41% with 30 neurons in the hidden layer. In Table 5, the hit rate obtained by the NN using Kalman filtered data is 95.39% using 20 neurons in the hidden layer, which is almost the same as the one obtained when tested with 10 neurons.

5.2. Real test

After the simulations were performed, the next step was to test the embedded NN implementation into the collar placing it on a horse and obtaining classification results in real-time. By performing this test, we can obtain the hit rate accuracy of the NN and compare these results with the ones obtained on the simulations that were performed in the previous subsection.

For this test, the same NN architectures were trained and embedded into a collar. These architectures were MPL-based feedforward NNs with also three layers: the input layer, one hidden layer and the output layer. The input layer contains 3 or 9 neurons (depending on which sensors were used from the input dataset), and the number of neurons in the hidden layer is fixed to 10. The output layer consists of 3 neurons, corresponding to the three horse gaits to be classified (motionless, walking and trotting). The transfer function used is the Sigmoid Symmetric Stepwise, as said in the previous section.

The results obtained in simulation tests show that there is no great difference when between using 10, 20 or 30 neurons in the hidden layer, in terms of average hit rate. For this reason, a hidden layer with 10 neurons was used in this test to save memory and

Table 4
Hit rate percentages using the embedded version of the FANN library and unprocessed sensor data as input.

Neurons in hidden layer	Classes	Sensors used				
		Acceler. (%)	Gyro. (%)	Magnet. (%)	A&G&M (%)	A&G (%)
10	Trotting	75.37	23.77	55.31	77.92	75.52
	Motionless	88.70	58.72	59.40	87.76	86.70
	Walking	76.62	81.71	68.03	77.52	79.15
	Average	**80.23**	**54.73**	**60.91**	**81.07**	**80.46**
20	Trotting	74.01	34.86	58.23	79.48	77.40
	Motionless	89.65	57.02	58.45	84.86	85.69
	Walking	77.59	78.29	68.80	80.46	81.65
	Average	**80.42**	**56.72**	**61.83**	**81.72**	**81.58**
30	Trotting	75.45	35.66	58.06	77.64	77.76
	Motionless	89.31	57.23	59.13	86.80	87.70
	Walking	77.55	78.04	68.38	82.81	80.48
	Average	**80.77**	**56.98**	**61.86**	**82.41**	**81.98**

Table 5
Hit rate percentages using the embedded version of the FANN library and filtered sensor data as input.

Neurons in hidden layer	Classes	Applied filter	
		Kalman (%)	FreeIMU (%)
10	Trotting	98.56	58.01
	Motionless	83.86	53.47
	Walking	99.99	64.13
	Average	**94.14**	**58.54**
20	Trotting	89.46	62.12
	Motionless	96.72	57.80
	Walking	100	67.81
	Average	**95.39**	**62.57**

Fig. 5. Horse used for real tests.

reduce calculations. The datasets collected by the sensors, which were used as input to the NN, were accelerometer, accelerometer-gyroscope-magnetometer and the output of a real-time Kalman filter.

To train the NN, datasets used in the previous section were used, where the length of the whole dataset for both the training and the test are 30,000 samples. The samples were randomly divided into three sets; the following cross-correlation scheme was used for every experiment in this work: 70% for training, 15% for validation and 15% for testing the network. The collar was placed close to the jaw of the horse, as can be seen in Fig. 5.

The horse that was used in this test belongs to a different breed from that of the horses used to collect data for performing the simulation tests and training the NN.

Table 6
Hit rate values obtained in the real scenario.

Neurons in hidden layer	Classes	Sensors used		Applied filters
		Acceler. (%)	A&G&M (%)	Kalman (%)
10	Trotting	64.05	64.44	75.03
	Motionless	66.81	67.93	84.38
	Walking	66.12	67.42	83.64
	Average	**65.66**	**66.60**	**81.01**

Once the NN was trained in the PC by using the FANN library, connections' weights were obtained and loaded into the collar, allowing to perform the test, where the horse was performing the three gaits to be classified (motionless, walking and trotting) in real-time. The results of these tests using both raw sensor data (unprocessed) and Kalman filtered data are presented in Table 6.

The results obtained in these tests show that the hit rate values are about 15% lower than the ones achieved in the simulation. The best accuracy value obtained by the NN was using Kalman filtered data as input, with a hit rate percentage of 81.01%, whereas using RAW data only achieved 66.60% in the best case.

5.3. Comparative study of the obtained results

The results that were obtained from the NN simulation on the PC using the full FANN library and from the NN simulation on the collar using the embedded version of the FANN library (Sections 5.1.1 and 5.1.2) show that there is no difference between these two approaches in terms of accuracy ratio. The results from these two simulations were compared by calculating the average difference of the hit rate percentage values considering every architecture and case presented in the corresponding tables. The calculated average accuracy percentage difference is 1.00473684% and the typical deviation is 1.03962797%, proving that, with the same input dataset, the results obtained in the PC and in the collar are practically the same.

Comparing these simulations with the ones that were done in a previous work using the Matlab Neural Network Toolbox [5] and the same input dataset, it can be seen that the results obtained with FANN hardly differ from them. Therefore, it can be concluded that FANN library can, at least, obtain the same performance as the Matlab Neural Network Tookbox and thus it could be a good chance.

Since the real test was performed with a horse from a different breed than that of the ones that had been used to train the network, worse accuracy results were expected. However, even in this case, the NN achieves 81.01% hit rate when using Kalman filter, which is around 14% lower than that of the simulation

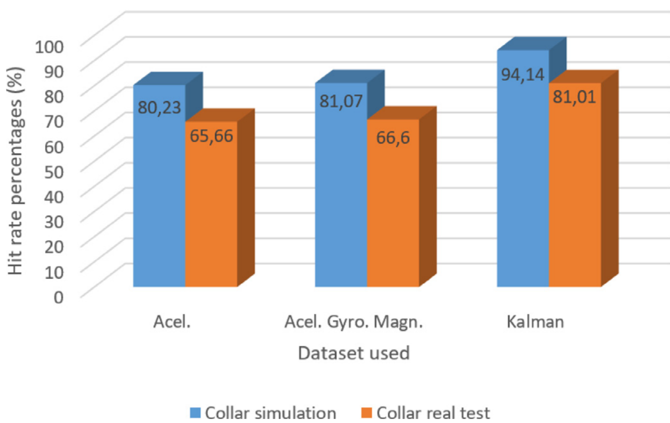

Fig. 6. Hit rate comparison between the real test and the simulations performed in the collar.

on the collar, but still a very decent accuracy ratio. Using the accelerometer and all the IMU sensors as input, the NN achieved 15% less accuracy but, in these cases, 65.66% and 64.60% are not considered to be good values when it comes to monitoring animals. Fig. 6 presents a bar graph displaying the hit rates from the simulations done in the collar and the real test performed with the horse.

As is described before, it makes sense that the real test experiments obtained worse classification results, due to the fact that the horse in which these tests were performed is from a different breed than that of the ones that were used to train the NN.

6. Discussion

The comparison between the classification results obtained from the simulation of the embedded version of the FANN library in the collar and the ones obtained from the simulation of the full FANN library on the PC shows that deploying a NN in the devices that are pltor wildlife (capturing a wild or semi-wild animal to change the battery from its collar is a difficult and expensive task). This approach is a novelty and it is not considered in the previous studies that have been discussed in this manuscript. These results are very similar to the ones obtained in [5], in which the authors carried out different simulations on a PC with the same NN architecaced on animals is not only viable, but also the best solution in terms of power consumption [20], which is a very important fact when it comes to monituring and preprocessing steps on the data input as the ones that have been done in this manuscript but using the Neural Network Toolbox from Matlab. Therefore, for this purpose, the performance of the developed embedded version of the FANN library proves to be as good in terms of results as the Matlab Toolbox or even the full version of the library, but overcoming two main disadvantages: (1) it can be implemented and deployed on a MCU and (2) it is optimized for low-power consumption.

The results obtained from the real tests in which the collar was placed on a horse were around 14% less accurate than the ones obtained in the simulations. This is not caused by the performance of the NN but by two facts: (1) the horse in which these tests were performed is from a completely different breed than that of used to train the NN, and (2) a slight change on the collar's position while placing it on the horse makes the inertial sensors gather completely different values, which affects the classification results. In the near future, the authors will focus on solving these two issues. The first one can be solved by training the network with more data from different horse breeds. We have already started working on solving the second issue, studying the best place and way to position the collar and using time windows to calculate the difference produced in the sensor information. Using this information instead of the raw data obtained by the inertial sensors as input to the NN reduces the errors that are caused by the position of the collar.

Table 7
Hit average obtained.

	Recognition method	Classified classes	Hit average (%)
Ref. [16]	MLP based ANN	5	76.2
Ref. [17]	Decision tree	5	85.5
Ref. [18]	Decision tree	3	82.2
This work	MLP based embedded ANN	3	81.1

Even though several problems happened in the real test, the classification system with an embedded ANN presented in this work not only improves the results obtained in other related works quoted in Section 1 (introduction), as can be seen in Table 7, but it performs the classification in real time.

Focusing on the communications network, previous works have used wireless sensor networks [4,7,9,11,13,16], but none of them have implemented the embedded sensor systems placed on the animals as end devices. With this configuration, these devices only communicate with the rest of the network when transmitting the gathered information instead of also working as routers (receiving information from other devices and retransmitting them to the next hop of the network). This solution dramatically increases battery life [20], which is an important factor in animal monitoring systems that previous works have not taken into account (not using low-power consumption devices or energy-harvesting techniques).

Classification results are obtained in real-time and stored in an online database server where the information can be accessed by researchers, biologists and other staff members from Doñana National Park.

In [4], over-the-air programming (OTAP) is used to adaptively modify the network sampling rate. This mechanism has a great potential and will be definitely considered in future works, where it could be used to modify the ANN training configuration and connection weights with new parameters that improve the accuracy of the classification or even allowing to classify new behaviors that are not trained yet in the current development status of the project.

7. Conclusions

In this work, we propose an embedded MLP-based ANN system placed on semi-wild animals to classify their behavior using the information collected by inertial sensors. For this purpose, several experiments have been carried out to test the classification accuracy of three different horse gaits (motionless, walking and trotting) using different NN architectures, input data and preprocessing algorithms. The results have been obtained from a simulation on the PC using the FANN library, from a simulation on the collar using a light and embedded version of the FANN library that has been developed by the authors and from real experiments where the collar was placed in different horses. Deploying a NN on an embedded device for animal monitoring in real time is a novelty and it had not been done so far for this purpose. The in-collar classification reduces the number of transmissions in the communication, which greatly increases battery life, as has been stated in previous works by the authors. In addition, the use of ANNs makes it possible to have more adaptable and configurable systems, since only the weight matrix should be changed for adapting the collar to be used with other species or detecting new gaits. Statistical methods usually work by setting thresholds manually (which is problematic and dependant on the data set), while an ANN picks up the threshold automatically.

The results obtained in the simulation tests show that the FANN library (both the full and the embedded versions) achieved a great hit rate percentage, obtaining around 82% in the best case when the raw dataset was used, and 95% when filtered sensor data was used. The hit rate difference of these two simulations was calculated, obtaining an average of 1.0047% (typical deviation of 1.0396%). These results were compared with the simulations that were performed by the authors in a previous work using the Matlab Neural Network Toolbox, showing that, in terms of accuracy ratio, there is no difference between using these two approaches.

The results obtained from a real test where the collar with the embedded NN implementation was placed on a horse show a decrease of approximately 15% in the hit rate from what it was expected after the simulations. This decrease could be caused by the fact that the horse used to perform this test was from a different breed than that of used to train the NN. Also, other external factors, like the collar's position on the horse, could be critical to the performance of the behavior classification system. Very good results were obtained when simulations tests were performed (82.41% using accelerometer, gyroscope and magnetometer as input; and 95.39% using Kalman filtered data as input). These results will be taken into account as a reference for future works where the collar positioning system and the NN training will be improved to obtain better results in real time scenarios.

Acknowledgments

This work was supported by the excellence project from Andalusia Council MINERVA (P12-TIC-1300). The authors would like to thank Rocío Benitez from Club Hípico "La Alegría" for helping us on the experiments that were done with her horses and Ramon C. Soriguer, Francisco Carro, Francisco Quirós and the EBD-CSIC for their support on the tests that were done in Doñana National Park.

References

[1] E. Schlecht, C. Hülsebusch, F. Mahler, K. Becker, The use of differentially corrected global positioning system to monitor activities of cattle at pasture, Appl. Anim. Behav. Sci. 85 (3) (2004) 185–202.
[2] E.D. Ungar, Z. Henkin, M. Gutman, A. Dolev, A. Genizi, D. Ganskopp, Inference of animal activity from GPS collar data on free-ranging cattle, Rangel. Ecol. Manag. 58 (3) (2005) 256–266.
[3] M. Schwager, D.M. Anderson, Z. Butler, D. Rus, Robust classification of animal tracking data, Comput. Electr. Agric. 56 (1) (2007) 46–59.
[4] E.S. Nadimi, R.N. Jørgensen, V. Blanes-Vidal, S. Christensen, Monitoring and classifying the behavior of a herd of sheep using ad hoc wireless sensor networks and artificial intelligence, in: Proceeding of the Seventh World Congress on Computers in Agriculture, American Society of Agricultural and Biological Engineers, Reno, Nevada, 2009, p. 1.
[5] E. Cerezuela-Escudero, A. Rios-Navarro, J.P. Dominguez-Morales, R. Tapiador–Morales, D. Gutierrez-Galan, C. Martín-Cañal, A. Linares-Barranco, Performance evaluation of neural networks for animal behaviors classification: horse gaits case study, in: Proceeding of the Thirtieth International Conference Distributed Computing and Artificial Intelligence, Springer, 2016, pp. 377–385.
[6] M. Dominguez-Morales, A. Jimenez-Fernandez, D. Cascado-Caballero, A. Linares-Barranco, R. Paz, G. Jimenez-Moreno, R. Soriguer, Technical viability study for behavioral monitoring of wildlife animals in Doñana: an 802.15.4 coverage study in a natural park, in: Proceedings of the International Conference on Data Communication Networking (DCNET), IEEE, 2011, pp. 1–4.
[7] E.S. Nadimi, H.T. Søgaard, T. Bak, ZigBee-based wireless sensor networks for classifying the behaviour of a herd of animals using classification trees, Biosyst. Eng. 100 (2) (2008) 167–176.
[8] D. Anderson, Virtual fencing – a concept into reality, in: G.J. Bishop-Hurley (Ed.), Proceedings of the Spatial Grazing Behavior Workshop, 2006, pp. 61–91.
[9] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, D. Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet, in: ACM Sigplan Notices, 37, ACM, 2002, pp. 96–107.
[10] L.D. Mech, S.M. Barber, A Critique of Wildlife radio-Tracking and Its Use in National Parks, CO Technical Report, Biological Resources Management Division, US National Park Service, Fort Collins, 2002.
[11] E.S. Nadimi, H. Søgaard, T. Bak, F.W. Oudshoorn, ZigBee-based wireless sensor networks for monitoring animal presence and pasture time in a strip of new grass, Comput. Electr. Agric. 61 (2) (2008) 79–87.
[12] Z. Butler, P. Corke, R. Peterson, D. Rus, Virtual fences for controlling cows, in: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA'04, 5, IEEE, 2004, pp. 4429–4436.
[13] H. Fourati, N. Manamanni, L. Afilal, Y. Handrich, Posture and body acceleration tracking by inertial and magnetic sensing: application in behavioral analysis of free-ranging animals, Biomed. Signal Process. Control 6 (1) (2011) 94–104.
[14] C. Umstatter, A. Waterhouse, J. Holland, An automated method of simple behavior classification as a tool for management improvement in extensive systems, in: Proceeding of the Fourth International Workshop on Smart Sensors in Livestock Monitoring, 2006, pp. 57–58.
[15] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[16] E.S. Nadimi, R.N. Jørgensen, V. Blanes-Vidal, S. Christensen, Monitoring and classifying animal behavior using ZigBee-based mobile ad hoc wireless sensor networks and artificial neural networks, Comput. Electr. Agric. 82 (2012) 44–54.
[17] L. González, G. Bishop-Hurley, R.N. Handcock, C. Crossman, Behavioral classification of data from collars containing motion sensors in grazing cattle, Comput. Electr. Agric. 110 (2015) 91–102.
[18] J.A.V. Diosdado, Z.E. Barker, H.R. Hodges, J.R. Amory, D.P. Croft, N.J. Bell, E.A. Codling, Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system, Anim. Biotelem. 3 (1) (2015) 15.
[19] Doñana National Park, http://whc.unesco.org/en/list/685, (accessed: 2017-02-04).
[20] J. Dominguez-Morales, A. Rios-Navarro, M. Dominguez-Morales, R. Tapiador–Morales, D. Gutierrez-Galan, D. Cascado-Caballero, A. Jimenez-Fernandez, A. Linares-Barranco, Wireless sensor network for wildlife tracking and behavior classification of animals in Doñana, IEEE Commun. Lett. 20 (12) (2016) 2534–2537.
[21] S.E. Harris, Horse Gaits, Balance and Movement, Maxwell Macmillan Canada, 1993. ISBN 0-87605-955-8.
[22] IEEE 802 Working Group, et al., IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks. Amendment 1: MAC sublayer, IEEE Std 802, 2011. 4–2011.
[23] IEEE 802 Working Group, et al., IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks. Amendment 2: Active Radio Frequency Identification (RFID) System Physical Layer (PHY), in: IEEE Std, 802, 2011, pp. 4–2011.
[24] S. Nissen, Implementation of a Fast Artificial Neural Network library (FANN), in: Report, 31, Department of Computer Science, University of Copenhagen (DIKU), 2003, p. 29.
[25] P. Sibi, S.A. Jones, P. Siddarth, Analysis of different activation functions using back propagation neural networks, J. Theor. Appl. Inf. Technol. 47 (3) (2013) 1264–1268.
[26] M.K.S. Alsmadi, K.B. Omar, S.A. Noah, et al., Back propagation algorithm: the best algorithm among the multi-layer perceptron algorithm, Int. J. Comput. Sci. Netw. Secur. 9 (4) (2009) 378–383.
[27] D. Nguyen, B. Widrow, Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, in: Proceeding of the 1990 IJCNN International Joint Conference on Neural Networks, 1990, IEEE, 1990, pp. 21–26.
[28] R. Tapiador-Morales, A. Rios-Navarro, A. Jimenez-Fernandez, J. Dominguez–Morales, A. Linares-Barranco, System based on inertial sensors for behavioral monitoring of wildlife, in: Proceeding of the 2015 International Conference on Computer, Information and Telecommunication Systems (CITS), IEEE, 2015, pp. 1–4.