PLOS | ONE

RESEARCH ARTICLE

Localizing Tortoise Nests by Neural Networks

Roberto Barbuti☯, **Stefano Chessa**☯, **Alessio Micheli**☯, **Rita Pucci***☯

Department of Computer Science, University of Pisa, Pisa, Italy

☯ These authors contributed equally to this work.
* pucci@di.unipi.it

Abstract

The goal of this research is to recognize the nest digging activity of tortoises using a device mounted atop the tortoise carapace. The device classifies tortoise movements in order to discriminate between nest digging, and non-digging activity (specifically walking and eating). Accelerometer data was collected from devices attached to the carapace of a number of tortoises during their two-month nesting period. Our system uses an accelerometer and an activity recognition system (ARS) which is modularly structured using an artificial neural network and an output filter. For the purpose of experiment and comparison, and with the aim of minimizing the computational cost, the artificial neural network has been modelled according to three different architectures based on the input delay neural network (IDNN). We show that the ARS can achieve very high accuracy on segments of data sequences, with an extremely small neural network that can be embedded in programmable low power devices. Given that digging is typically a long activity (up to two hours), the application of ARS on data segments can be repeated over time to set up a reliable and efficient system, called Tortoise@, for digging activity recognition.

Citation: Barbuti R, Chessa S, Micheli A, Pucci R (2016) Localizing Tortoise Nests by Neural Networks. PLoS ONE 11(3): e0151168. doi:10.1371/journal.pone.0151168

Editor: Gennady Cymbalyuk, Georgia State University, UNITED STATES

Received: May 27, 2015

Accepted: February 24, 2016

Published: March 17, 2016

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The authors have no support or funding to report.

Competing Interests: The authors of this manuscript have read the journal's policy and have the following competing interests: Italian patent application "dispositivo e procedimento di localizzazione in particolare di siti di nidificazione e simili" (device and method of localization of nest sites), Prof. Roberto Barbuti, Prof. Giuseppe Anastasi, Prof. Stefano Chessa, Prof. Alessio Micheli, number filed MI2011A002337, data of claim: 21/12/2011; Prof.

1 Introduction

The impact of the species extinction is of greater devastation than was ever expected [1]. Reptiles and amphibians seem to have been disproportionately affected, showing average population decreases of 34% and 97% respectively. The biggest threat to biodiversity historically comes from a combined impact of habitat loss, degradation, new predators, food, and the mistreatment of animals for material assets. This level and type of endangerment was reported in [2] and [3]. Given this history, the new main factor for population decline is the stark lack of diversity itself. The decrease in tortoise population is attributable to a combination of infertility and predation of eggs (at incubation, as well as young tortoises in their first years of life). The eggs are affected by environmental pollution, which inhibits hatching. The hatchlings are also easy prey for predators.

In an attempt to limit the tortoises mortality, herpetologists from many organizations have taken part in protection programs in order to monitor and support female tortoises and their hatchlings. In the Galapagos Islands, the Darwin Scientific Foundation and the Charles Darwin Foundation have been committed to protect giant tortoises (*Chelonoidis nigra*) [4]. In Jumby Bay, the population of hawksbill tortoises (Eretmochelis imbricata) is monitored by the Jumby

Stefano Chessa is a research associate with Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" (Institute of science and information technology, ISTI), at Institute of the National Research Council, via G. Moruzzi 1, 56124 Pisa, Italy; Prof. Roberto Barbuti is the director of "Museum of Natural History" of the University of Pisa, via Roma 79, 56011 Calci (Pisa), Italy.

Bay Hawksbill Project run by the Jumby Bay Company (*Eretmochelis imbricata*) is monitored by the Jumby Bay Hawksbill Project run by the Jumby Bay Company [5]. The Turtle Conservancy works to protect the population of gopher tortoises (*Gopherus polyphemus*) in southern California, and various species of turtles and tortoises in Madagascar and Mexico [6].

As part of these programs, herpetologists observe the tortoise behaviour in order to identify the nests and to retrieve the eggs. It is possible to monitor eggs in their natural environment, however the eggs are often taken to a protection centre. This ensures a lower mortality of the hatchlings during their most vulnerable period [5]. This is repeated every year during the reproduction period. Only during these periods is it possible to observe the females digging activity and to identify the location of the nests. Unfortunately this procedure cannot be applied on a large scale, as it is dependent on the direct observation of the animals. For large-scale applications an automatic system is necessary—one which is able to recognize the behaviour of the animal, and consequently promptly locate the nests.

In this context, our general objective is to design an automatic system called Tortoise@, which aims to recognize digging activity and locate the nests. This is done using a combination of a small device attached to the carapace of wild animals, and an activity recognition system (ARS) based on machine learning methods. This system is based upon, and claims the benefit of, Patent Application Number filed MI2011A002337.

The device should be worn comfortably by the animal, and it should be operational during the entirety of the breeding season (e.g. about two months in Italy for Mediterranean tortoises). Thus the device should be very light in weight, small in size, and energy efficient (i.e. a low computational power and a small memory). The device must be energy efficient because it will need a portable supply of power, i.e. it must be battery powered. In an effort to augment the operational life of the system, the device must be able to recognize the digging activity autonomously. This enables the system to be set up to avoid sending a continuous stream of data, and instead only send the current geographic coordinate when a nest digging event has been identified. According to the need of limited computational and memory capabilities, the ARS must find a good trade-off between accuracy, memory, and computational power.

Another feature of our approach is the use of an accelerometer to detect characteristic movements of the tortoises. Our observation on data collected during the initial experimental campaign show that lateral movements registered by the accelerometer during the excavation of the nest are regular, regardless of the size and species of the tortoise. These characteristics affect only the amplitude of movement, and not its type. The occurrence of a regular pattern, and the presence of the typical data noise, reinforce the decision to use a data-driven, robust, yet simple, pattern recognition approach to automatically identify such patterns.

Keeping in mind these objectives and observations, in summary, our scientific goal is to assess the possibility of introducing an efficient and effective recognition system for the digging activities of tortoises by means of accelerometers data and to support it by an experimental validation from real-data.

The use of neural networks, which have a flexible architecture, allows us to investigate models with low computational and memory requirements. Furthermore, Tortoise@ takes advantage of modular composition of very small neural network models combined with a filter of the model output stream to identify the activity of tortoises. In the paper, both the model and the filter are referred to as an ARS, with name TartaNet. Tortoise@ is part of the biologging research area because of the environmental purpose, and the technology involvement. The biologging field was introduced in [7–9] and described by Bograd et Al. in [10]. Biologging researchers investigate the behaviour of animals in the wild or in semi-free conditions. Ropert et Al. [11] use hall sensors and a logger to study the feeding behaviour in birds and koalas. Baranes et al. [12] and Kays et al. [13] apply biologging techniques to classify

animal behaviour based on oyster catchers (*Haematopus ostralegus*) and on toucans, respectively. The processing of animal activity accelerometer data, with the machine learning algorithms, has, with time, become a more accepted approach to monitor the animal behaviours.

Several works in literature present the use of machine learning algorithms for animal activities recognition with different ethological purpose, taking into consideration typically accelerometer data of daily activities, such as feeding, sleeping, and walking, among others. Recently, for example in [14] machine learning algorithms are proposed for behavioural patterns of farm animal (cows) to identify daily activities. A similar approach is used in [15] to store, annotate, and automatically recognize the activity of wild animals. In [16], the machine learning algorithms are proposed as a classification framework for recognizing dog activities. In this concern, the artificial neural networks (ANN) [17] represent a possible approach to provide recognition capabilities to embedded on devices and it is therefore considered in this paper. Focusing on ANN for animal activities recognition, it is to note some recent works. Examples of possible real world applications of this technique are described in [18, 19], and [20], wherein the ANNs are proposed to classify the daily activity of several species.

2 Materials and Methods

In the following sub-sections we describe the subjects and the device used for this research, and we present the methods proposed to reach our goal of animal activities recognition.

2.1 Ethics statement

The device was gently attached to the carapace with minimal pressure. We used one hundred tortoises in this research and observed their behaviour two to three times a day, for at most one hour. The glue pad used for the tortoises is the one described at the following URL: http://www.uhu.com/en/products/glue-pads.html. The glue pad remained on the animal for a maximum of 3 hours. The procedure is non-invasive, does not cause pain or death of animal, and therefore these studies did not require consultation from the Ethics Committee of the University of Pisa (protocol n. 86/609/CEE). The procedure was carried out in the Protection Centre for Mediterranean Tortoises, in Massa Marittima, Tuscany, Italy. This is a Centre for conservation, ex situ of fauna (CESFA). *The Museum of Natural History*, of The University of Pisa, and *The Unione dei Comuni Montani Colline Metallifere* (UCMCM), are authorities responsible for the centre and for the protection of the inhabiting animals. The owners gave the permission to conduct the procedure inside the centre. Both the UCMCM and *The Museum of Natural History* approved the procedure and the Tortoise@ research conducted by the University of Pisa. In the additional information added to the manuscript, the authorizations are available as Supporting Information files.

2.2 Subjects

Our subjects were healthy adult female tortoises from the Protection Centre for Mediterranean Tortoises. In the centre, the tortoises occupied fenced areas divided according to either species or populations of Mediterranean tortoises (Testudo hermanni, Testudo graeca, Testudo marginata in semi-free conditions). The fenced areas of adult tortoises ranged from $180m^2$ to $6.25m^2$, depending on the number and size of the tortoises. The terrain was hilly and grassy with Mediterranean shrubs. An extensive campaign of data collection was conducted in the field in the June and July of 2012. For our behavioural study, each subject was kept under surveillance throughout the experiments in order to identify its activities.

2.3 Hardware equipment and selection of sensors

For data gathering purposes, we used a system comprising two MicaZ devices [21] and a standard PC. A MicaZ is a low-power programmable device with a communication board (with a 2.4 GHz radio subsystem), a MPR2400 antenna, and a radio. The MicaZ radio is based on the IEEE 802.15.4 standard, which guarantees low energy consumption and long-lasting operations [22]. The MicaZ device used as tag attached on the animals is augmented with the MTS310 sensor board which, in turn, has a temperature, a light sensors, and a two-axis accelerometer (the ADXL202). The light and temperature sensors are intended to provide context information, describing the individual's surroundings. However, these sensors alone cannot provide sufficient information on the tortoise activities. The light and temperature sensors are intended to provide context to primary information, describing the individual's surroundings, while the accelerometers are used to capture the movements of the animals [20, 23], and [24].

In our experiments we used the following setting for the data collection: one MicaZ (the dispatcher), attached to the animal, was used to record and transmit data concerning the environment, as well as on the movements of the tortoises. The second MicaZ (the collector), which was connected to the PC via USB cable, received the data transmitted by the other MicaZ and provided aggregated data to the PC for storage and off-line analysis. The transmitted data was used to identify the different activities. The dispatchers recorded light levels, temperature, and the accelerometer signals at a frequency of 4Hz (every 250 ms) and stored them in the internal memory of the MicaZ. In fact, this frequency proved afterwards to be excessive, and the sampling could be reduced without impacting the behaviour of the system.

The accelerometer signal was $\pm 2g$, and described the movements of the carapace along the x and y axes of the sensor. As shown in Fig 1, the x described the movements of carapace along the short side, and the y axis described the inclination along the long side of carapace.

For practical use in the wild, the prototype will require a further engineering step to reduce its size and be more comfortable for the animals. The engineered system would need to be unobtrusive and follow the design of the carapace of the tortoises as proposed in the patent and shown in Fig 2.

2.4 Data collection protocol

Experiments were performed, with the tortoises, in a semi natural habitat. This environment enabled us to identify the characteristics of the main activities of interest for female tortoises. The data collection protocol was supervised by a human operator who observed the behaviour of the tortoises. Data collection was initiated at the behest of the supervisor, subsequent to the identification of one of three main activities: eating, walking, and digging.

The collection procedure consisted of three phases: *placement, monitoring, and storage*. Throughout these phases, the supervisor recorded and stored each sequence of sensor data in a dataset. Each sequence was labelled with the corresponding activity tag.

This dataset was used in the off-line analysis for training and validating the ARS.

- *Placement*: The supervisor attached the device with a non-toxic and re-usable glue pad onto the carapace between the second and third vertebral plate, without interrupting the activity observed. The location chosen for the device was non-invasive and non-restrictive. The position required consistency in order to guarantee a reasonable homogeneity of the collected data. The device used during the data collection is illustrated in Fig 1.
- *Monitoring*: In this phase, the MicaZ recorded accelerometer data with the given sampling frequency, and stored it in its flash memory. The operator recorded the specific activity being

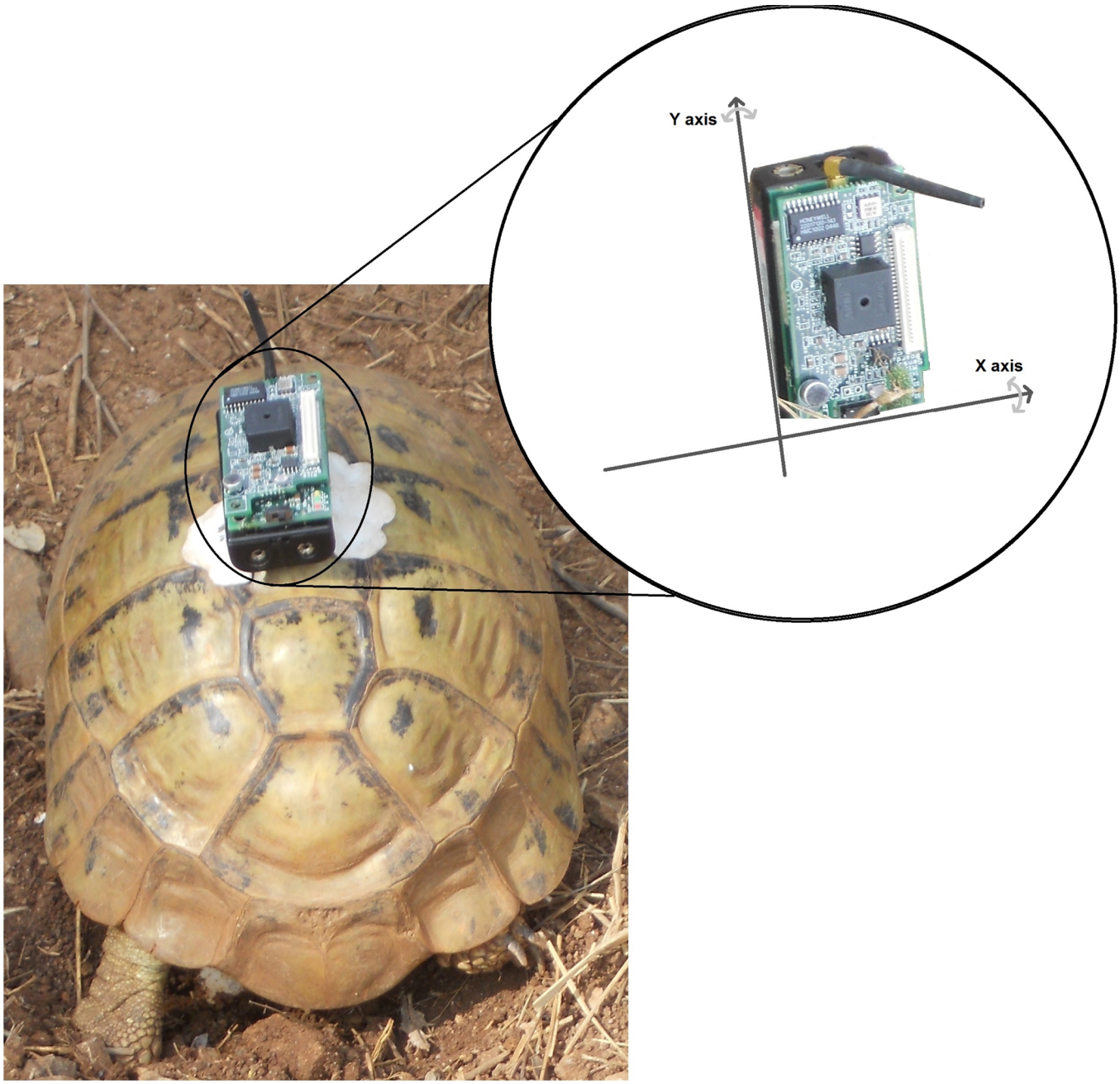

Fig 1. Accelerometer axis. The device is positioned on the carapace with a great care by the supervisor at the beginning of each data collection procedure.

doi:10.1371/journal.pone.0151168.g001

performed by the animal in the tortoise activity diary in order to associate this activity with the sampled data.

- *Storage*: At the end of the activity, the supervisor retrieved the sensor from the carapace, for another recording, and downloaded the signal stored in the flash memory of the mote onto the computer.

Note that it was not possible to establish the duration of each activity in advance. For this reason, a specific time slot was not used during the recording phase for the storyboard of the

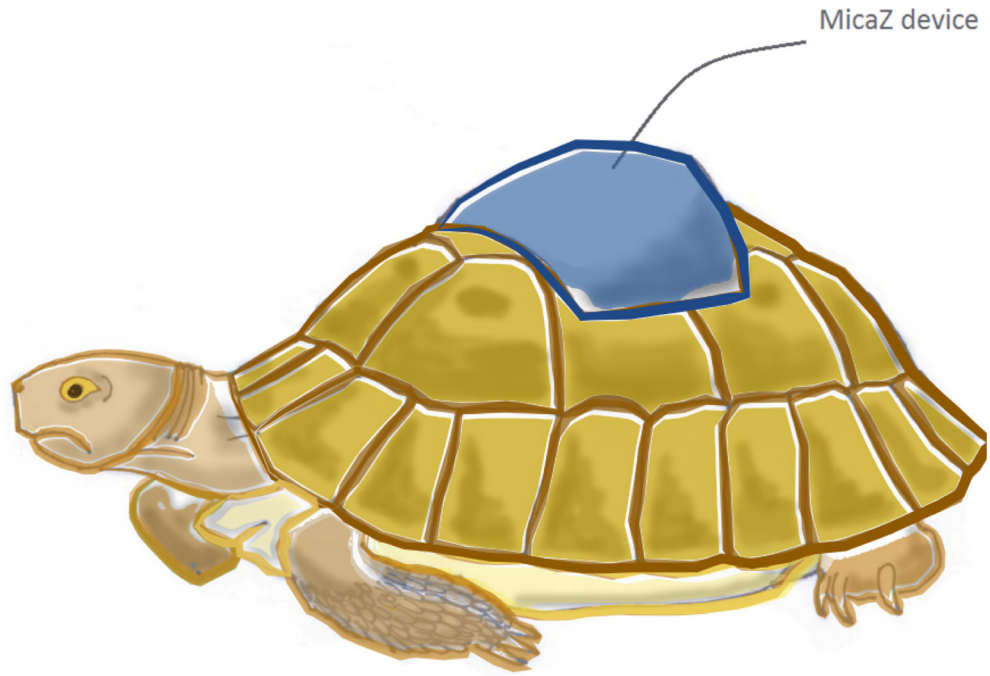

Fig 2. Final device. Detail of the position of device on the tortoise carapace.

doi:10.1371/journal.pone.0151168.g002

actions, and the length of each recorded signal was based on the duration of each individual activity, with an upper limit of two hours.

2.5 Data Analysis

In the data collection campaign, sequences of sensor data (accelerometers, light and temperature) were collected. Each sequence refers to one nest excavation, walking, or eating activity performed by an individual. The dataset obtained with this dataset is composed by a temporal series of accelerometer raw data.

2.5.1 Pre-processing methods. Pre-processing techniques were used to reduce the noise of the signal and to normalize the values in a uniform range. We also down-sampled the sequences in order to experiment with different sampling rates, for the sake of memory parsimony.

- *Filtering*: in order to filter out noise, we applied a moving average filter, which is effective in the case of non-uniform noise [25]. The moving average filter is a type of Finite Impulse Response (FIR) [26]. This filter operates by averaging on an interval of five points from the original sequence to produce each point in the filtered sequence.
- *Normalization*: the integer values produced by the accelerometers were scaled down by the average of all the observed values—note that this is not a proper normalization. The reason for using this scheme, rather than a more conventional normalization, was to obtain an input stream for the neural network in the form of integer values (instead of float values). This helps a substantial saving in memory, and enables the implementation of the system on memory constrained devices.

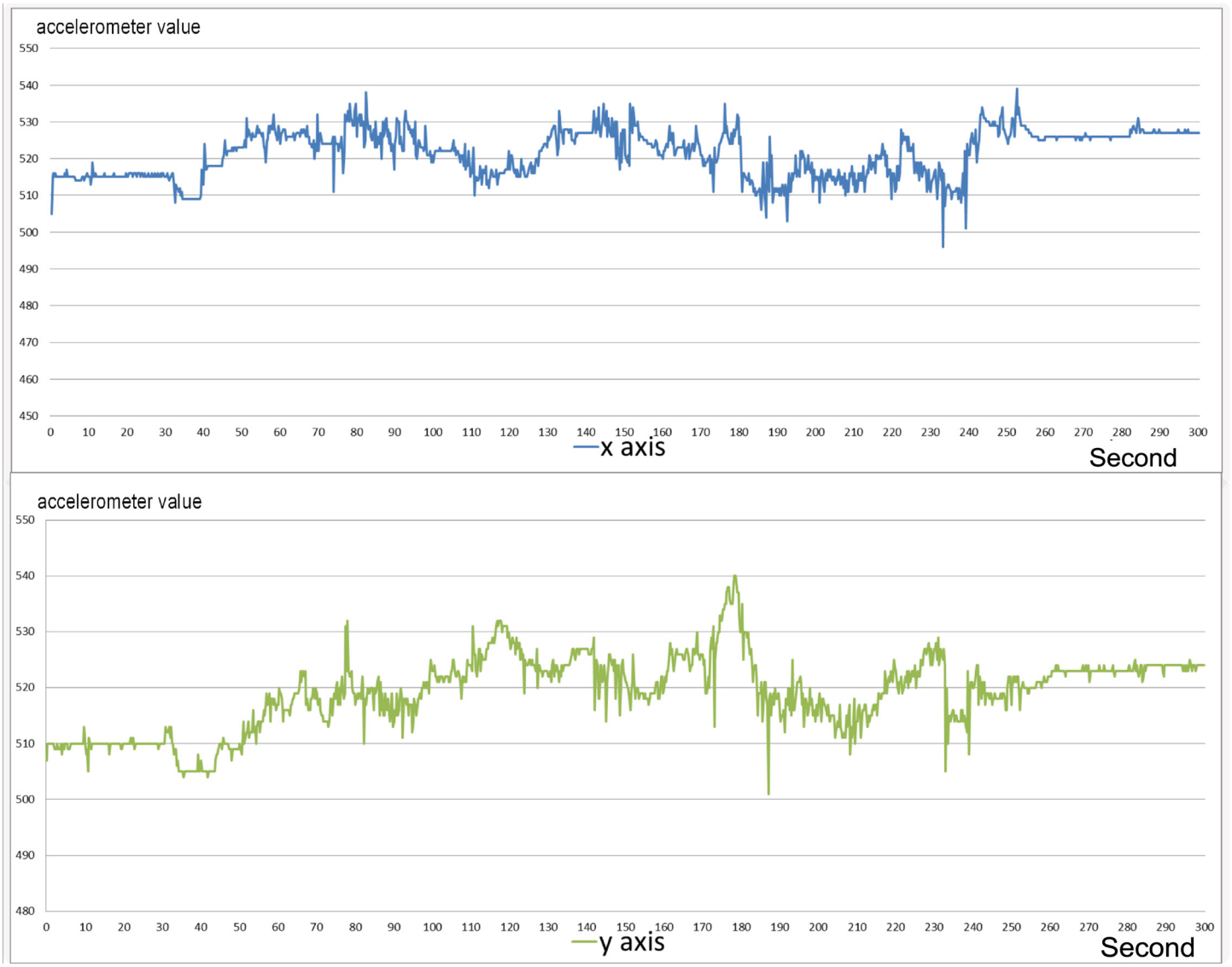

Fig 3. Sequence of a walk phase from the accelerometer. Walking activity, as represented by the blue signal (x axis), is characterized by chaotic oscillation of values, representative of the oscillation of the carapace. A similar behaviour is observed in the green signal (y axis).

doi:10.1371/journal.pone.0151168.g003

- *Down-sampling*: Down-sampling involves reducing the sampling rate of a sequence of data. For our purposes, we adopted the simple mechanism of excluding samples at a given rate from the original sequence. As mentioned before, we used this filter to test the system with different sampling rates, by reducing the number of inputs of the system. The down-sampling is applied on the sequences filtered with the moving average filter.

2.5.2 Input of the neural network. Since the accelerometer has two axes, an accelerometer sequence is in the form of paired sequences: one for each axis.

Fig 3 shows an example of a walking sequence. In this case the accelerometer sequence is chaotic and hardly predictable. Fig 4 shows a sequence recorded during eating, which is characterized by the complete absence of movement. Fig 5 shows a sample of the digging sequence.

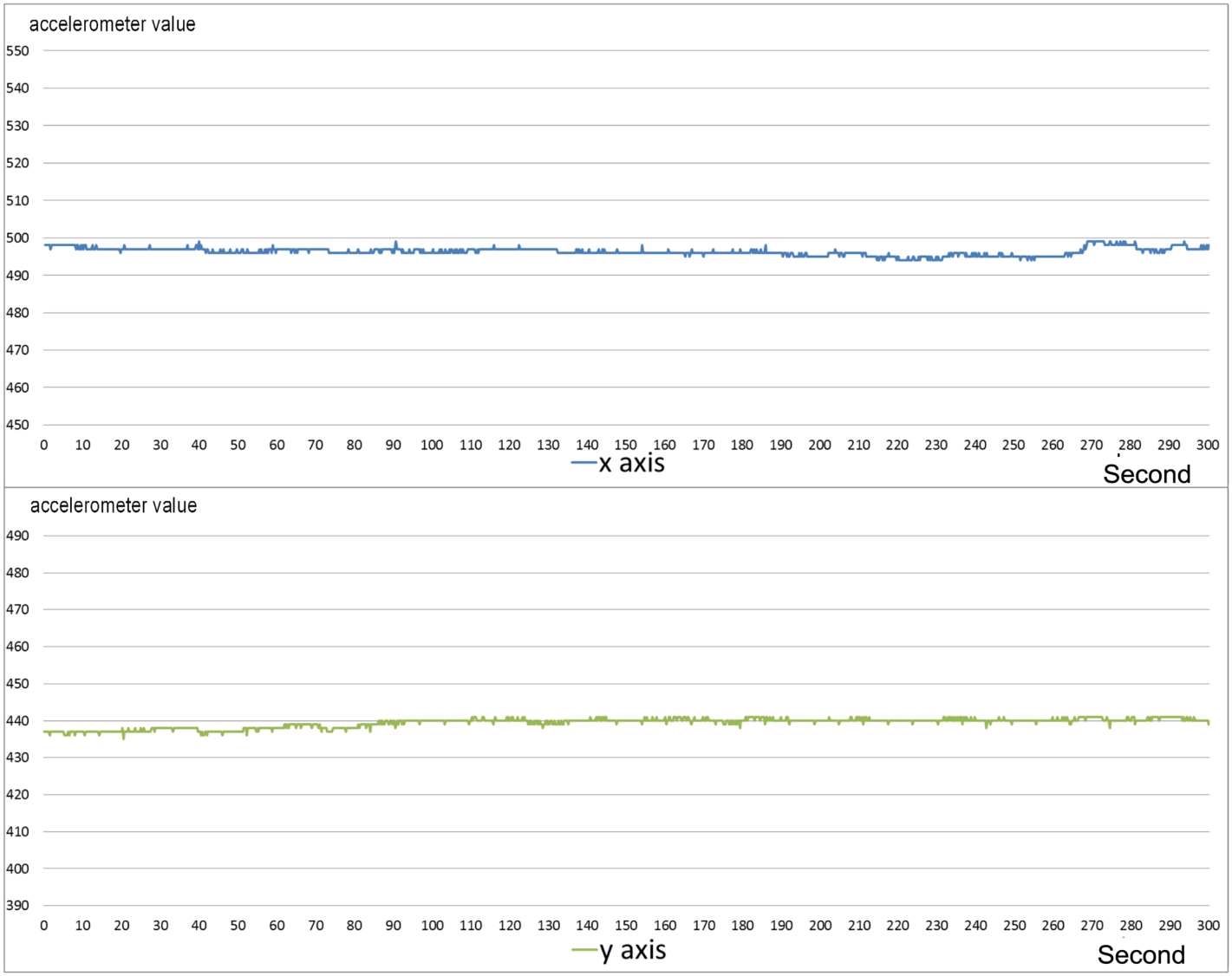

Fig 4. Sequence of a eating phase from the accelerometer. In this phase, the carapace moves only to let the tortoise to reach food outside the reach of its head. Both x and y axes show relatively insignificant variation.

doi:10.1371/journal.pone.0151168.g004

The x axis sequence obtained during a digging sequence approximates a periodic square wave. This periodicity is due to the turnover of posterior paws during the excavation.

The accelerometer sequence on the y axis provides negligible information for the discriminative aim, therefore the rest of the paper, is focussed on the analysis of the x axis data, which contributes to reduce the system load for data storage and processing. The database of this study consists of a set of 83 labelled accelerometer sequences. The set is divided into 15 walk sequences, 6 eating sequences, and 62 digging sequences, of which 13 terminated with an egg deposition. The other 49 excavation sequences which were not terminated with a deposition is a result of normal tortoise behaviour in captivity.

The sequences of the accelerometer data are indexed over the time dimension by describing the tortoise activity as a set of successive samples. Each sample corresponds to a quarter of a

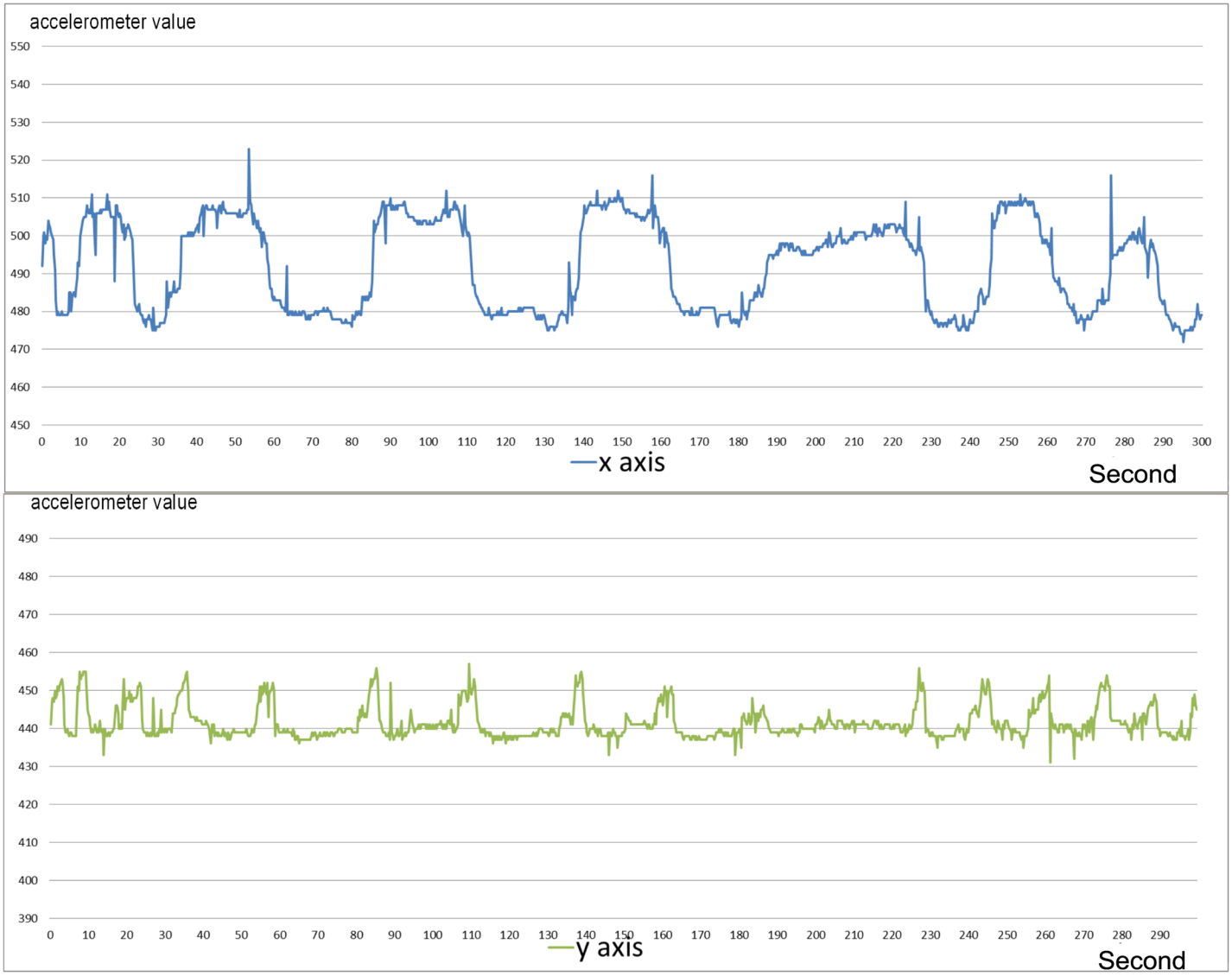

Fig 5. Sequence of a digging phase from the accelerometer. The blue signal (x axis) shows a periodic movement of the carapace along the x axis that resembles a square wave. Each square wave observed on the signal of the x axis corresponds to a peak of the signal of the y axis (the green signal).

doi:10.1371/journal.pone.0151168.g005

second related to the frequency of sampling (4 Hz). As mentioned in Section 2.3, to reduce the size of the sequence, we down-sampled it to 1 Hz, with a one second interval (a *time step*).

Each time step is classified by an ARS, which is fed with a window of n previous steps (here after, the *input window*). At each time step, we shift the input window by one interval of a quarter of window and obtain a new output value. This particular use of the input windows can be allowed by a neural network model, known as the Input Delay Neural Network (IDNN). This approach is popular in the literature of activity recognition as described in [27].

Given the main role of the input window, we identified a characteristic pattern of the excavation phase that can be recognized by the neural network. The pattern occurs in a window containing a repetition of the squared wave, as shown in Fig 6, and plays a key role in training the ARS. The period of time chosen for the pattern is the mean duration of two squared waves

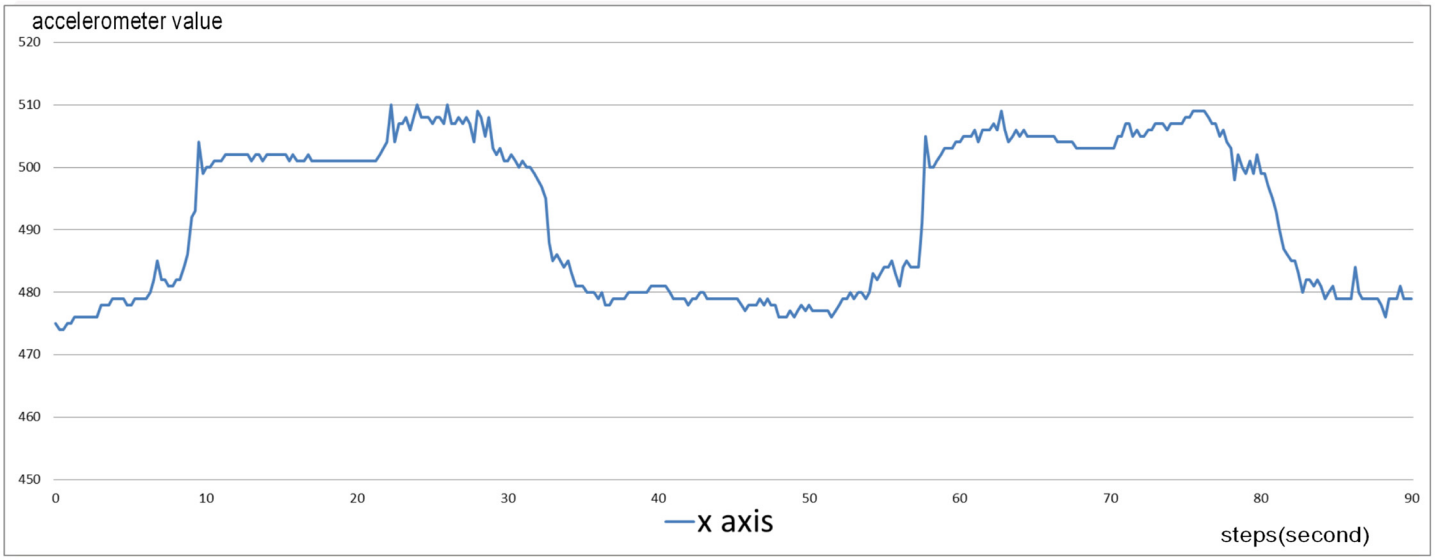

Fig 6. The characteristic pattern extracted from the x axis accelerometer data of a digging sequence. This pattern shows the characteristic periodic movement performed by the tortoise in the digging activities. It is composed of two squared waves which distinguish the excavation sequences. The signal of characteristic pattern is shown in figure with a frequency of 4Hz. The time interval shown in the histogram corresponds to approximately 90 seconds.

doi:10.1371/journal.pone.0151168.g006

in the excavation sequences. This corresponds to approximately 90 time steps (90 seconds) in the real data at hand for Mediterranean tortoises. We therefore set the input window to have the same dimension—90 time steps. The dimension of this pattern may vary, depending on the specific species of tortoise under study. In those cases, the size of the window should be changed accordingly in order to include a minimum of two repetitions of squared waves. Note that the variance of frequency was one of the main factors in choosing the pattern dimensions. In our case, to guarantee a generalization of characteristic pattern for Mediterranean tortoises, we took into consideration samples collected by 3 different families of *Testudo*. The great difference between these species allows us to analyse various frequencies of the same type of movement, and then to deal with these variations automatically, through learning, over the different sampled cases in the gathered data set.

The ARS is trained to identify the characteristic pattern and can be utilised to recognize it in long sequences. To find a trade-off between the ARS responsiveness, the memory overhead, and the need for providing enough information to identify the digging activity, we introduce the concept of a sub-sequence extracted by the long sequences, named segment of sequence in the rest of the paper. Taking into account this trade-off, the size of the segment is 5 minutes (300 intervals). Given the initial delay due to the input window (of 90 time steps) and the time interval chosen for the shifting window (one quarter, that is 22 time steps), the neural network provides an output stream of 10 classification values for each segment. Table 1 summarizes the length of the input window, pattern, segment of sequence, and output stream.

The basic idea beyond this setting is to use these patterns rather than sequences in the training of the neural network to keep a low memory overhead and to improve the performance of the classification prediction by focusing the activity recognition algorithm on the most specific information available about the digging phase.

According to the characteristic pattern, the neural network is trained with samples of 90 seconds labeled with positive and negative classifications. The samples with positive classification

Table 1. Lengths and descriptions of the input window, pattern, segment of sequence, and output stream.

Input window (buffer)	90 values	input to the NN
Pattern	90 time steps	sub-sequence used to train the neural network
Segment of sequence	300 time steps	accelerometer sequence used for the classification
Output stream	10 values	output values of neural network over the course of a segment

doi:10.1371/journal.pone.0151168.t001

are patterns extracted from nest excavation sequences. The samples with negative classification are obtained with patterns sampled by eating and walking sequences.

2.6 TartaNet activity recognition system(TartaNet)

The TartaNet activity recognition system (TartaNet: TARTAruga, which means tortoise in Italian, and neural NETwork) integrates a machine learning (ML) model (which is fed with the input windows), and a filter of the model output stream. The machine learning model identifies, within each segment, the windows similar to the characteristic pattern. The TartaNet filter classifies the output stream of machine learning model as a digging or non-digging activity.

2.6.1 Machine learning model of TartaNet. To build the ML model of TartaNet, we focus on the Artificial Neural Network (ANN) in the form of the well-known feedforward ANNs/ Multi-Layer Perceptron (MLP) architecture [17]. The robustness and the ability for universal approximation of this method provides the basis for the flexibility of the approach for the approximation of arbitrary classification functions from experimental data, despite not having a theory of the pattern characteristics.

The MLP is a network of computational units organized into one input layer, one or more hidden layers, and an output layer. The possibility to set up different numbers of units allows us to easily investigate different configurations to find a good tradeoff between model complexity and accuracy of the results, which is particularly demanding for this application, as discussed in the previous sections regarding efficiency constraints.

The MLP supervised learning algorithm allows adapting free parameters used to weight the connections of the model, in order to obtain the best approximation of the desired outputs (*targets*). This is typically realized in terms of minimization of an error (or loss) function on the training dataset. In particular, in the classical least-mean-square approach, the error is computed as the square of the difference between the output of the neural network (computed by a Sigmoid Tanh function in our setting) and the target values over all the training samples. For MLP, the backpropagation technique [28] is the most popular among the supervised training algorithms. The backpropagation learning algorithm is based on a gradient descent of the loss function through the weight space in order to reduce the error, allowing tuning the model free-parameters for the training data. Variants including regularization approaches are considered (see Section 3.1). The gradient descent is repeated until the improvement of the error is no longer significant. Once the model is trained, we evaluate the prediction on new sequences of activities.

In order to deal with sequence (of sensor) data by MLP, we focus on the shifting window models. In particular, we consider the Time Delay Neural Network models developed for sequential data (e.g. in [17, 29, 30], and [31]), models which are based on a MLP architecture, and specifically on two models: an Input Delay Neural Network (IDNN) [17], and IDNNs inspired by characteristics of the Convolutional Neural Network (CNN) [32].

Input Delay Neural Network: The dynamic nature of the accelerometer sequences needs a model that can recognize a movement pattern, irrespective of the precise location in time. Thus

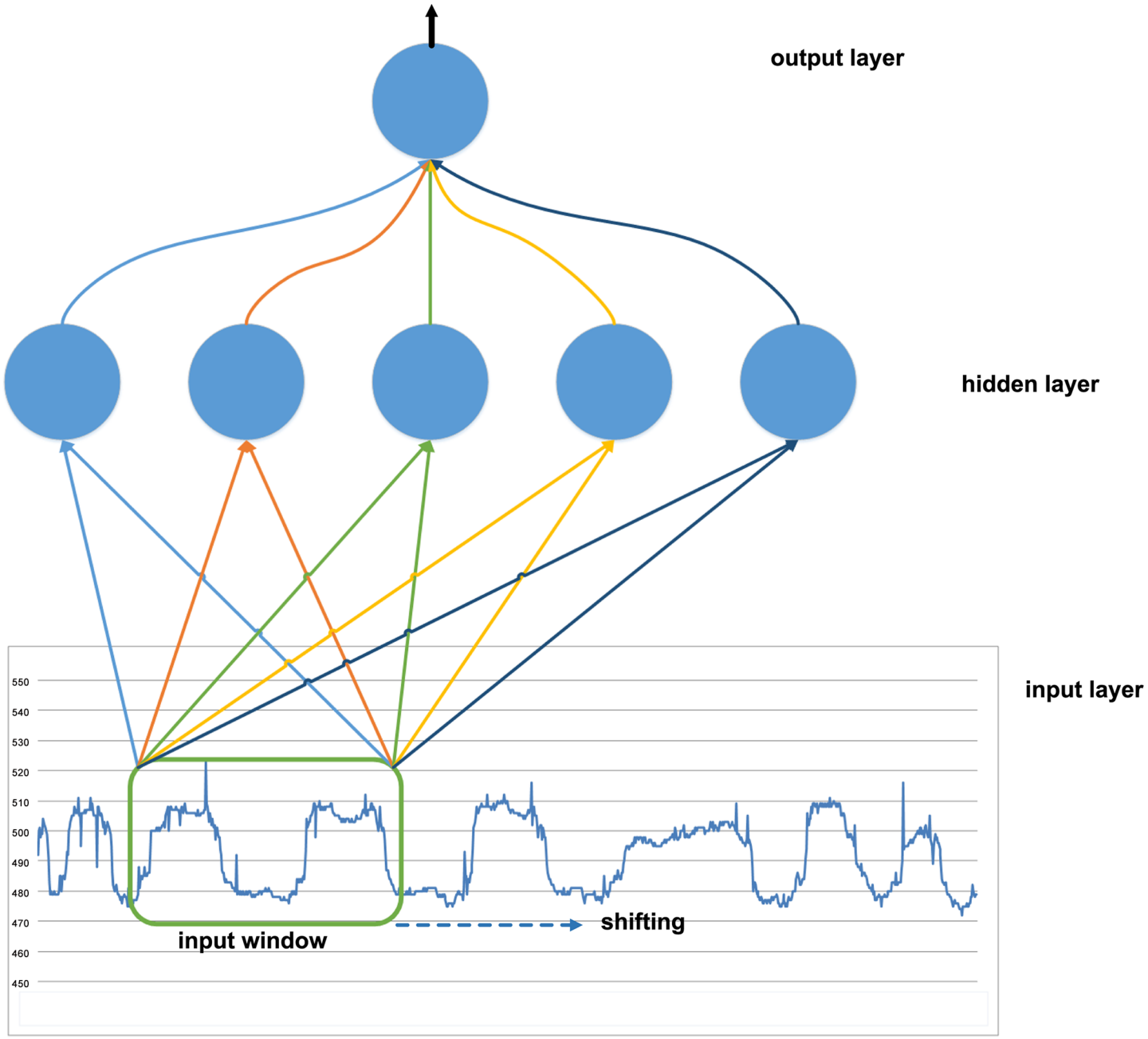

Fig 7. Input delay neural network (IDNN) with an input sequence. An input window of the signal sequence is used to feed each hidden unit. The input window is shifted on the input sequence in order to analyze the entire sequence.

doi:10.1371/journal.pone.0151168.g007

the neural network needs to represent the relationship between events in time, without a particular temporal alignment, and the Input Delay Neural Network (IDNN) model satisfies this requirement. Fig 7 shows the IDNN model. This model exploits a MLP architecture (all connections are feedforward) with sequential inputs. In particular, in the IDNN model, the inputs to hidden units consist of the outputs of the input units not only during the current time step, but also during previous time steps. As shown in Fig 7, this is implemented using the input

window which is shifted on the segment of sequence, which equivalently corresponds to a delay in the units reading the history on the inputs.

The shifting window is highlighted over the segment of sequence, and it feeds the hidden units of the IDNN. Different weights are used for each hidden unit, which in turn feeds the output unit.

The window takes into account the temporal structure of the pattern defined in Section 2.5.2 to train the IDNN. Once trained, the IDNN is used to read the segments of sequences, over time, using the same trained model. In this way the model implements the property of translation invariance [33], identifying the characteristic pattern regardless of its position in the sequence.

Typically, due to the use of different weights for each hidden unit, the IDNN network requires memory occupation for the storage of the weights that can easily exceed the memory available in low-power devices (see details in Section 3.3). This prevents us from using the IDNN with the original architecture in the Tortoise@ application. For this reason we also consider IDNN inspired by the CNN, as discussed in the next section.

Local receptive fields in input window and weights sharing: The CNN is a biologically-inspired trainable architecture that learns invariant features from data [32]. The model is introduced in [34, 35] and explained in [36, 37]. Partially inspired by the CNN, we propose to modify the IDNN as described below.

The first variation is to exploit the local receptive fields (LRF) concept from CNN and applying it inside the input window of the IDNN approach. The main idea of LRF is to connect units in a layer to receive input from a set of units in a small neighbourhood of the previous layer. Each hidden neuron scans the input using their local receptive field. A unit with local receptive field obtains an inventory of features to represent the characteristics of the sub-window (as described by Lang et al. in [30]), without being burdened by an excessive number of free parameters and reducing the memory space [17, 38]. Specifically, we identify four input sub-patterns in the characteristic pattern (shown in Fig 6). Two of these sub-patterns identify the ascent phases of the squared waves and the other two identify the descent phases, as shown in Fig 8.

According to the split of the characteristic pattern, the sub-patterns are then mapped on the input window of the neural network. We then divide the input window into four sub-windows (each sub-windows is formed by 22 inputs). As a result of this specialization (namely IDNN LRF), the hidden layer is composed of four hidden units, each of which is connected to a sub-window by independent weights, and consequently is focused on a specific phase of the characteristic pattern. As a result, each hidden unit takes 22 inputs and therefore the model requires the storage of 22 weights for each of the four hidden units. This reduces the memory occupation for the weights of the classical IDNN, which require 90 weights for each hidden unit.

The second idea inspired by CNN is weight sharing (WS) among LRF hidden units, which constrains the weight vectors of some hidden units to be equal. This further reduces the number of free parameters from the large amount of units sharing the same weight vector and obtains a certain level of shift invariance (the detection of features regardless of their position). We implement weight sharing between the sub-windows that make up the characteristic pattern, i.e. two sub-windows of ascent and two sub windows of descent. This reshapes the IDNN structure by reducing it to two hidden units: one for each phase, as shown in Fig 9.

In this way the memory occupation accounts for 22 weights of two hidden units each (half of the memory occupation obtained with the sole LRF method). Table 2 shows the reduction of input weight vector introducing sub-windows.

2.6.2 TartaNet filter for neural network output stream. The models described in the previous section classify each input window that makes up a segment of sequence. Hence, for each segment, each model provides an output stream made up of classification values defined in the

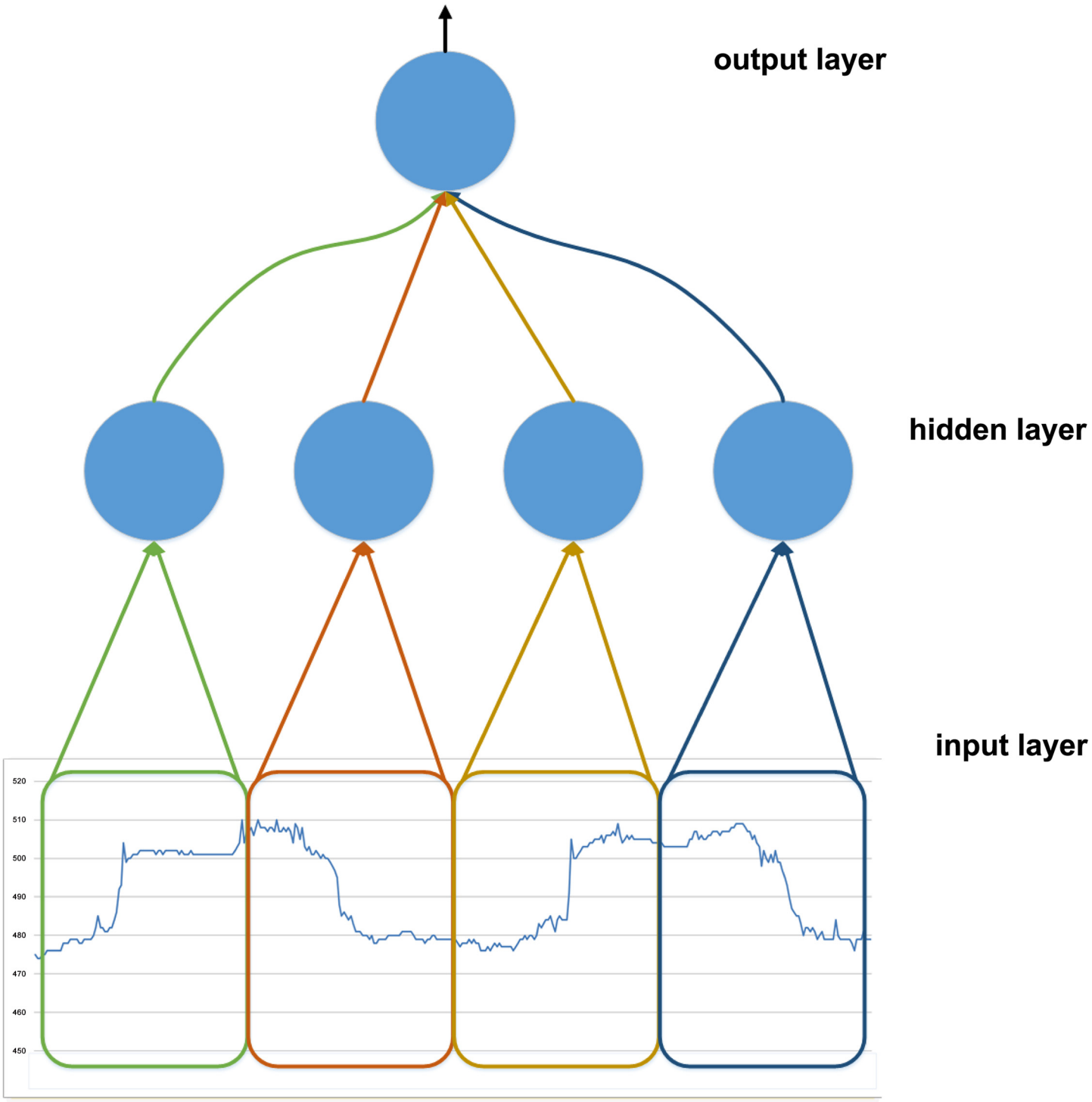

Fig 8. Input delay neural network with local receptive fields (IDNN LRF). The input window is divided in four sub-windows. Each sub-window is a LRF of a hidden unit.

doi:10.1371/journal.pone.0151168.g008

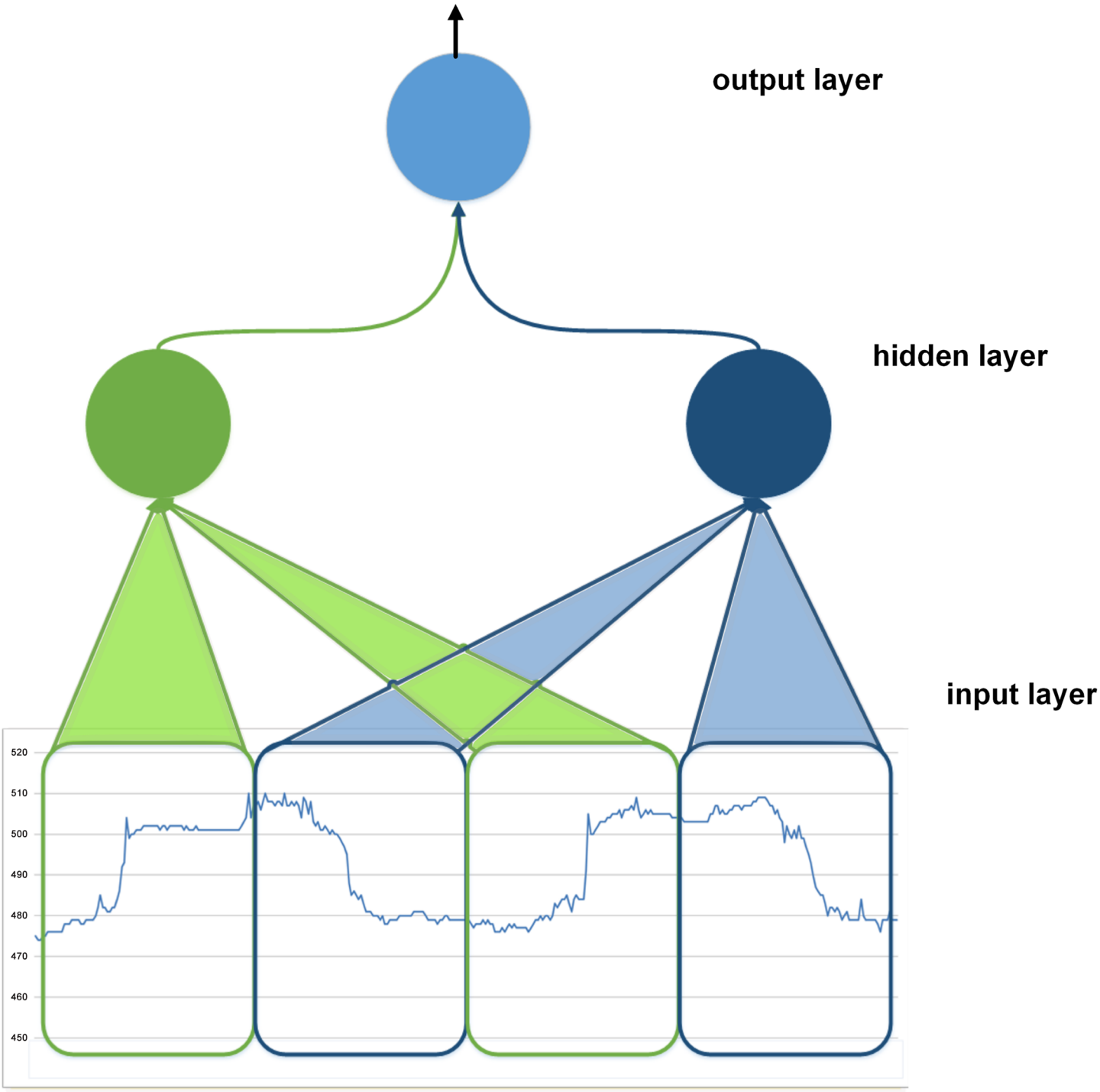

Fig 9. Input delay neural network with LRF and weight sharing (IDNN LRF WS). The input window relates the four sub-windows identified for IDNN LRF. With characteristic pattern divided in sub-windows, both the first and the third sub-windows are in an ascent phase of the squared signal, whilst the second and the fourth sub-windows are in the descent phase. The colours of the connections highlight the sharing of weights between hidden units due to the match between sub-windows. Weights are shared between the two sub-windows that make up each phase: i.e. each unit uses the same set of weights for both the sub-windows of the ascending and descending phase (we identify this model as IDNN LRF WS).

doi:10.1371/journal.pone.0151168.g009

Table 2. The column # subwindow lists the sub-division of the input window related to the different models. The column # weights per hidden unit lists the number of weights necessary for each hidden unit (including one more for the bias) related to the different models according the dimension of their input field.

Model	# sub-windows	# weights per hidden unit
IDNN	no sub-window	91
IDNN LRF	4	23
IDNN LRF + WS	2	23

doi:10.1371/journal.pone.0151168.t002

range [−1,+1], where a positive output value corresponds to an input window similar to the characteristic pattern.

Fig 10a shows an example of output stream of IDNN LRF obtained with a segment of digging sequence. In this case many output values are in the range [0,+1]. Fig 10b shows the output stream of the same neural network with a segment of walking sequence. In this case, most of the output values are negative since the input windows do not correspond to the characteristic pattern.

According to the description above, we use an output filter to classify the output stream of the neural network as positive or negative over a segment. The TartaNet filter exploits three features computed on the ANN output stream:

- *NP*(Number Positive): Number of positive values.
- *MP*(Mean Positive): Mean value of all positive values compared to all the output values: *sum of positive values / output stream length*.
- *MPT*(Mean Positive Threshold): Mean value of the positive values above a threshold compared to all the output: *sum of the positive values above* τ / *output stream length*.

Where:

- The threshold τ was computed as mean of output positive values obtained with a validation set. The value obtained for the cases discussed in Fig 10 is 0.2.
- The TartaNet filter evaluates each feature individually against a *feature bound*, to assess it as a positive or negative response through a threshold.

The bounds of *MP* and *MPT* are useful to distinguish the output stream of one digging sequence from one obtained with a walking sequence. The bounds of MP and MPT are identified by observing the results on a separate dataset, the validation set used in the model selection, explicated in Section 3.1. These results are shown in the histograms in Figs 11 and 12. The histograms show the *MP* and *MPT* values obtained during the validation phase using fifteen negative segments (non digging sequences) and fifteen positive segments (digging sequences). Note in Fig 11 that the *MP* values of the output streams have a gap between the higher *MP* value obtained with the negative segments and the lower *MP* value obtained with the positive segments, that is between values for segments 15 and 16 in this case. Taking this gap into account, we set the threshold for the the *MP* using the mean value between the two bounds. Using the same approach, we found the bounds for *MPT* in Fig 12.

The value of NP identifies how many positive values are identified in the output stream. The use of NP in the filter enables us to also identify output streams with a few positive responses (which may have a low score over the average values used by *MP* and *MPT*) i.e. even cases in which the segment include few neat positive patterns can be recognized as digging

Fig 10. Output of IDNN LRF. a) The light blue signal corresponds to the movement of carapace collected during a digging phase. The orange signal is the response of the IDNN LRF to each (shifting) input window. The IDNN LRF outputs peaks when it finds windows with high similarity to the characteristic pattern. **b)** The blue signal refers to non-digging (walking) movements. The orange signal (the IDNN LRF output) has negative values with one small positive pick due to a window similar to the characteristic pattern by chance. In both graphs the yellow rectangle comprises values used for MP filter, and the green rectangle comprises values used for MPT filter.

doi:10.1371/journal.pone.0151168.g010

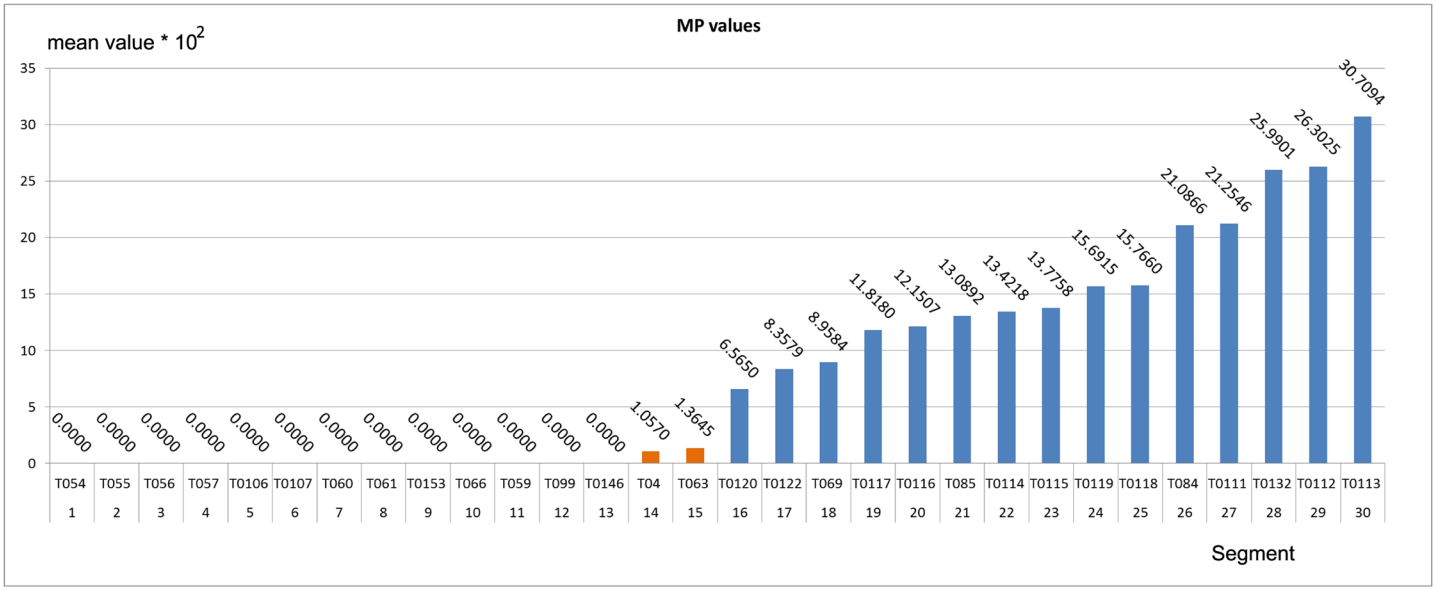

Fig 11. Histogram of the *MP* values of IDNN LRF output stream. Each column relates to the average of the positives values in the corresponding segment. The blue columns are the mean values obtained with digging segments whereas the orange (or zero) ones are the mean values of non-digging segments.

doi:10.1371/journal.pone.0151168.g011

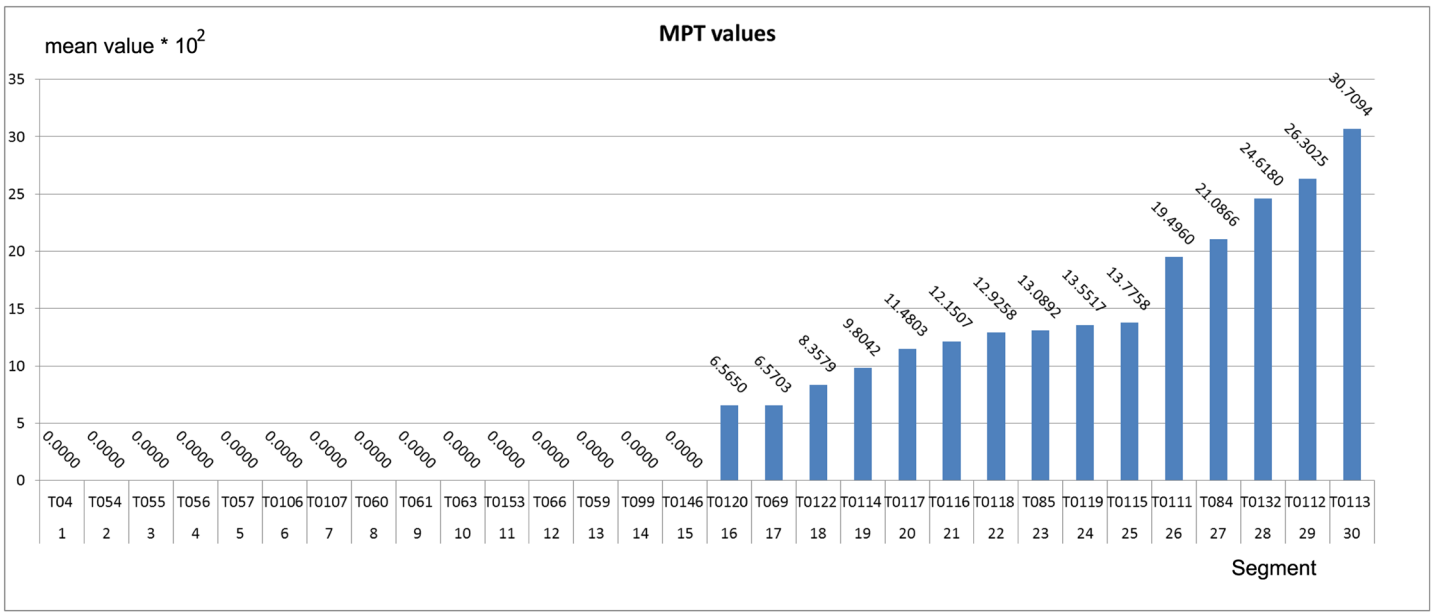

Fig 12. Histogram of the *MPT* values of IDNN LRF output stream. Each column relates to the average of the positives values above a threshold τ in the corresponding segment. The blue columns are the mean values obtained with digging segments whilst the orange(or zero) ones are the mean values of non-digging segments.

doi:10.1371/journal.pone.0151168.g012

activity. We observed that, with the digging sequences, at least a quarter of values are positive and above the selected τ, suggesting a threshold of 2 positive values. The filter then classifies the segment of sequence as positive if and only if at least two features are positive (i.e. above their threshold).

The TartaNet filter provides the classification of the neural network outputs, obtaining a final result to classify the accelerometer data. It is worth noting that the reason for using a modular design combining an NN and an output filter applied to a segment of the sequences lies in the requirement for efficiency, which underlies the aim of our approach. In particular, it allows us to simplify the model and reduce the memory request for the ARS. Specifically, the use of the output filter allows the reduction of the complexity of the NN model (up to only 2/4 hidden units), and to avoid the need of a complex output layer by moving part of the recognition process to the filter. Furthermore, the introduction of the segment of sequence (Section 2.5) allows us to reduce the ARS memory overhead.

Finally, this design makes it possible to produce a result for each input segment (every 300 seconds), thus giving the opportunity to repeat the classification process several times on disjointed segments (the entire digging sequence may last up to 2 hours). As a result we can reach high global classification accuracies without requiring the learning over entire sequences and therefore without additional memory costs in the implementation of the model. In other words, the reliability of the classification (that can be confirmed over disjointed segments) is not, in this way, at the cost of the dimension of the model architecture.

3 Results and Discussion

We performed the evaluation of the proposed models, IDNN, IDNN LRF, and IDNN LRF WS, for TartaNet with the aim to find a good trade-off between their performance and applicability.

3.1 Model validation

For models training, validation and evaluation, we split the segment dataset into four parts:

- ANN training set: these segments are used to generate the positive and negative patterns to train the neural networks. The resultant training set is composed of 134 balanced input patterns of 90 seconds each (67 positive and 67 negative).
- ANN validation set: these segments are used to generate the validation set of patterns to configure the neural networks (in our case 10 positive and 10 negative patterns of 90 seconds).
- Output filter validation set: these segments are applied to configure the thresholds of the output filters. This dataset is composed of 30 segments of five minutes (15 positive and 15 negative), which are disjoint from the training and validation sets described above.
- Test set: the segments of this external dataset are used to check the overall performance of the final TartaNet system on a set disjoint from the training and the validation sets. This final evaluation is useful to estimate future behvaiour of the whole system. The test set is composed of 56 segments of five minutes each (28 positive and 28 negative).

Each model is trained with the ANN training set, composed in equal parts of positive and negative patterns. The ANN validation set is used to select the best configuaration (model selection) on the basis of the classification accuracy with the distinct patterns. Then the output filter is configured on its validation set to evaluate its behavior for the entire period of the segments. After the validation phases, we evaluate the performance of the selected models with the test set (using new data, unseen in the training and validation phase).

Table 3. Classification accuracy of the models.

Model	Training(ANN) (standard deviation)	Test (standard deviation)
IDNN	92.59% (± 0.003)	96.24% (± 1.68)
IDNN LRF	90.74% (± 0.015)	95.51% (± 0.68)
IDNN LRF WS	90.74% (± 0.006)	94.34% (± 0.84)

doi:10.1371/journal.pone.0151168.t003

Note that the corpus of ANN training and validation sets accounts for approximately 30% of the entire data set, while the rest of the data is used for output filter validaton and test set (the latter is approximately 42% of the data).

For the ANN training phase, we assume two different initialisation ranges for the initial weights. The weights connections between the input layer and hidden layer are initialized within the range of [−0.00001,0.00001]. The weights connections between the hidden layer and the output layer are initialized within the range of [−0.01,0.01]. The number of units for the input layer is fixed to 90, according to the dimension of the pattern, for all the ANN models.

The validation phases of the ANN allow us to configure the *hyperparameters* values, specifically the number of hidden units, learning rate, momentum, and a parameter of weight decay [17]. The number of hidden units determines the size of the neural network and its learning capability. This is a hyperparameter for the IDNN model, whereas for the other two models there is a fixed structure related to sub-windows. For IDNN we considered values in the range [1, 10]. In particular, we assessed IDNN as 5 units, which is the best trade-off for this model. The learning rate controls the size of the change of weights during the iterations of the learning phase. The momentum is used to stabilize the network convergence. We explored the learning rate and the momentum in range of [0.0001,0.1] and [0,0.1], respectively. To include regularization in the ANN learning, which is useful to control the complexity of the model and hence to improve the generalization capability of models, we use a *weight decay* approach. This corresponds to the addition of a penalty term for complexity (based on the weight vector space norm) to the square error in the loss function used in the model training phase [17] [39]. In particular, this penalty term controls the weights magnitude, specifically the weight matrix squared L2 norm, i.e. the sum of squares of weights values, penalizing models with extreme parameter values. We explored the weight-decay hyperparameter in the range [0,0.001].

3.2 Performance analysis

The performance analysis asseses the classification accuracy measuered in terms of percentages related to the size of the sets and the relative confusion matrix for each model.

The performance measurements of the three models take into account the averages of errors computed on five different initializations of their weights for the ANN.

The accuracy obtained with each model is shown in Table 3. For the sake of completeness, the table reports the statistics on the configuration phase on ANN training of patterns (column labeled Training ANN), showing that all the models achieved more than 90% accuracy. Test results correspond to the evaluation on external data for the whole of TartaNet. The table does not report the validation accuracy of the output filter as it was 100% for each model, as per the results from Subsection 2.6.2.

The IDNN model achieves an accuracy of 96% (with a standard deviation of approximately 1.7 among different model initilizations) in the test phase with 4% of error due to misclassified segments showing as false positives, as shown in the confusion matrix Table 4.

With the IDNN LRF model we experience slightly inferior performance (95.5%), but it is worth mentioning that the memory requirements are less than IDNN model (as we show in

Table 4. Confusion matrix of IDNN on the test set.

Real / Predicted	Digging	Non-digging
Digging	28	0
Non-digging	2	26

doi:10.1371/journal.pone.0151168.t004

the section below). The analysis of errors shown in the confusion matrix (Table 5) shows 2 false positives, and 1 false negative.

The IDNN LRF WS model has a different behavior due to the high specialization of the hidden units. The 7% of failed classifications are mainly false positivies as in the other two models (as shown in Table 6). The IDNN LRF WS accuracy (94%) is slightly below the other models.

As explained in Section 2.6, the classification on segments (300 seconds) can be repeated several times (as digging needs a long period of continuous activity, up to a couple of hours). Because of this we can increase the accuracy of the global Tortoise@ system in recognizing the digging activity and virtually completely avoid the few false positives observed in the results. The few false positive classifications (i.e. false digging) shown in the confusion matrices in all the proposed models are obtained with walking sequences. Considering the IDNN LRF model and the two false positives in Table 6 (segments T01 and T03 in the provided data set), we tested the repetition of the classification after 10 minutes (two times the length of a segment) that resulted in a non-digging outcome (segments T04 and T05 from the same walking sequence that are classified as negative by the model) allowing us to avoid a Tortoise@ false identification of the nesting phase. The confusion matrix in (Table 5) also shows a false negative in one segment: T098. This misclassification of a single segment does not avoid the possibility of the recognition of the sequence as a digging sequence. Indeed, the rest of the sequences are composed of positive patterns which can activate in any other instance over the course of nesting the Tortoise@ identification of the nesting phase.

3.3 Applicability analysis

In this section we evaluate the three models according to their applicability in low-power devices such as the one used in our data collection. The following discussion holds true for more general cases, including the use of different devices, as it is related to efficiency, which is a central aim for this study, as explained in the introduction. The main concern is the memory

Table 5. Confusion matrix of IDNN LRF on the test set.

Real / Predicted	Digging	Non-digging
Digging	27	1
Non-digging	2	26

doi:10.1371/journal.pone.0151168.t005

Table 6. Confusion matrix of IDNN LRF WS on the test set.

Real / Predicted	Digging	Non-digging
Digging	27	1
Non-digging	3	25

doi:10.1371/journal.pone.0151168.t006

Table 7. Memory footprint. Each row in table relates bytes needed to memorize the input weight vectors of hidden units and output units.

Model	Memory footprint
IDNN	1844 byte
IDNN LRF	388 byte
IDNN LRF + WS	196 byte

doi:10.1371/journal.pone.0151168.t007

occupation of each model, since these devices have a small memory (typically a few kilobytes). We evaluate the memory occupation of the neural network by considering the integer representation of the inputs and the float representation of the weights.

Each of the three models need 90 integer values for the representation of the input window. For the IDNN model, 90 float weights are needed for each hidden unit, and five float weights for the output unit (including one more per each unit for the bias). The IDNN LRF and IDNN LRF WS models, which use sub windows, need 22 float weights for each hidden unit. Note that IDNN LRF has four hidden units, whereas IDNN LRF WS has two hidden units, and the weights for the output unit are four and two, respectively. Table 7 shows the memory footprint of the three models. IDNN has the highest memory footprint. IDNN LRF WS has a very low memory footprint which, however, penalizes its classification performance, while IDNN LRF has a slightly larger memory footprint than IDNN LRF WS, but it offers the best compromise between memory footprint and classification performance.

4 The Tortoise@ System

TartaNet is part of Tortoise@, which is an autonomous system for large scale applications aimed at identifying tortoise nests and rescuing eggs. Tortoise@ is a biologging system preliminary presented in [40] and [41], which enables an autonomous and assistive observation of tortoise digging behavior. It is a custom-made system based on sensors used to monitor movement, and to locate nests. The system is organized into four steps:

- *Environment monitoring*: monitors the environment (light and temperature) to identify the suitable conditions for the deposit eggs. This provides useful contextual information [42].
- *Movement monitoring*: data acquired by the accelerometer sensor are analyzed in order to recognize the digging movements.
- *Extended movement monitoring*: at regular intervals and for a defined short period of time, this phase is repeated to confirm the detection of digging activities, and thus to improve the overall accuracy of Tortoise@, allowing us to set up a reliable system for digging recognition.
- *Data communication*: sends geographic coordinates to a remote user through a base station.

The steps of Movement monitoring and Extended movement monitoring are implemented by the TartaNet activity recognition system. The four steps and the transition rules between steps are described in Fig 13. All these steps are implemented within a single device, using local processing for Tartnet to limit communications with the remote user only to the notification of the position of the identified nests. Note that this design meets the requirement of energy efficiency. The analysis of environmental conditions performed in step 1 limits the activation of the neural network implemented in step 2 only to the cases in which the environmental conditions are suitable for nesting. This also limits the recording of the acceleration data stream, which occurs only when step 2 is active and lasts for 5 minutes. When step 2 identifies a digging

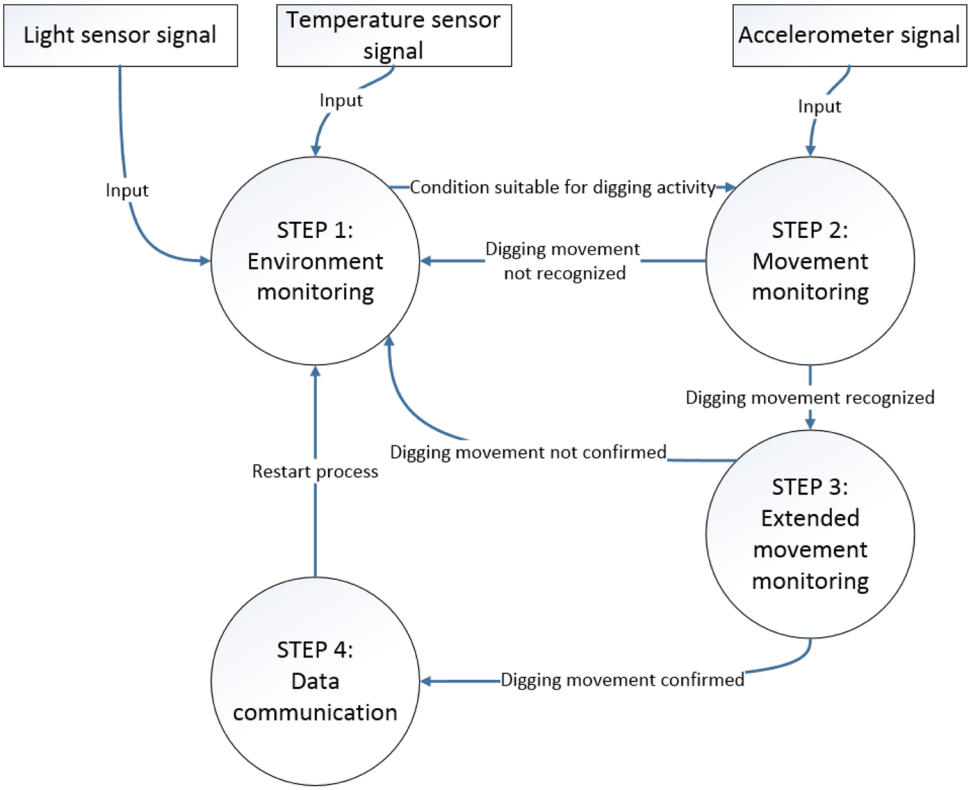

Fig 13. A scheme of the Tortoise@ system. The three input signals (light, temperature and accelerometer), the four steps (in the circular boxes) and the transitions between steps (labelled lines).

doi:10.1371/journal.pone.0151168.g013

activity, it then activates step 3 to extend the recognition of the activity in successive time slots (see Section 2.6).

This helps improve the overall reliability of the classifier. If the digging activity is confirmed by step 3 then the system promptly sends the information about the position of the nest to a remote user in step 4. This local processing avoids the burden of having a continuous transmission of data for remote processing, thus granting a longer operational lifetime to the device.

In terms of ethological research the Tortoise@ system provides the opportunity to make automatic an important procedure to assist tortoise populations. The high level of accuracy of the Tortoise@ system makes it particularly suitable for protection programs. The collection of eggs from the wild, the raising of hatchlings in captivity and the return of them to the wild when they are big enough to have a high probability of surviving, is an experimental methodology in conservation management. Examples of this methodology applied to birds and tortoises are reported in [43–46].

The system also identifies information on suitable environmental conditions for nesting and the behavioral characteristics of tortoises during the digging phase. It represents a new possibility for herpetologists to monitor and assist endangered tortoise populations.

Conclusions

This work was inspired by a concrete applicative problem concerning the protection of tortoises, and, in particular, by the need of biologging devices that may localize tortoises during

the escavation of their nest to enable protection programs aiming at retrieving the eggs before the hatching and at protecting the hatchlings in their first year of life.

Although simple in principle, the strong applicative constraints concerning the small size of the device and of its batteries, and the long operational time (the nesting period may last a few months), require innovative and very efficient activity recognition mechanisms that may fit a class of low-power and low-memory devices.

With this requirement in mind, we addressed the scientific goal of finding efficient solutions for the recognition of the digging activities of tortoises, leveraging on data produced by simple sensors like accelerometers.

Our findings, based on real-data gathered in an experimental data collection campaign, show that one dimensional accelerometer (appropriately oriented) is sufficient to catch the characteristic pattern of movement executed by the tortoises during digging, that this pattern is quite short (fits well in 90 samples of accelerometer for the species of tortoises of interest in this study) and that this pattern can be recognized with high accuracy by relying only on few neurons ANN based on the IDNN with Local Receptive Fields model.

This result exploits the modular organization of the proposed activity recognition method (called TartaNet), which combines a very small ANN and a simple output filter. In particular, this design finds a tradeoff between the accuracy and size of the ANN, while the output filter leverages on a few repetitions of the pattern recognition process to achieve a very high overall accuracy of the method.

The assessment of the scientific goal was achieved in a constructive way providing an effective and efficient model specialized for the digging recognition and supported by the experimental results on real-data for the Mediterranean tortoises.

The promising results obtained with TartaNet will enable us to develop a low cost prototype with reduced size, weight and battery pack that can be used on wild tortoises in a natural environment with negligible invasiveness and on large-scale.

In terms of models development, in our future work we will investigate the use of the convolutional neural network model by including further hidden (sub-sampling) layers in the neural network. In this way, the classification procedure and the recognition algorithm would be integrated together. At the cost of increasing of memory occupation, we would expect to further reduce the sensitivity of the output to shifts and distortions trough the sub-sampling of such approach and to achieve a full automatic tuning of the classification procedure.

Future interesting endeavors regard the possible comparison with alternative approaches such as those based on Fourier transform.

Supporting Information

S1 File. Italian law about the use of animals for experimentation.
(PDF)

S2 File. Authorization of the museum of natural history.
(PDF)

S3 File. Authorization of Regional office.
(PDF)

S4 File. Authorization of UCMCM.
(PDF)

S1 Dataset. Dataset used for experimentation.
(ZIP)

Author Contributions

Conceived and designed the experiments: AM RP RB SC. Performed the experiments: RP AM. Analyzed the data: RP AM. Contributed reagents/materials/analysis tools: SC. Wrote the paper: RB SC AM RP.

References

1. Milligan H, Deinet S, McRae L, Freeman R Protecting Species: Status and Trends of the Earth's Protected Areas. Preliminary Report. Zoological Society of London, UK. 2014. Available from: www.zsl.org/sites/default/files/media/2014–11/protected_species_final_20141112.pdf
2. Zylstra ER, Steidl RJ, Jones CA, Averill-Murray RC. Spatial and temporal variation in survival of a rare reptile: a 22-year study of Sonoran desert tortoises. Oecologia. 2013. pii: 173(1):107–116 doi: 10.1007/s00442-012-2464-z PMID: 23011852
3. Gibbon JW, Scott DE, Ryan TJ, Buhlmann KA, Tuberville TD, Metts BS, et al. The Global Decline of Reptiles, Dj Vu Amphibians. BioScience. 2000. pii: 50(8):653–666 doi: 10.1641/0006-3568(2000)050%5B0653:TGDORD%5D2.0.CO;2
4. Thornton IWB. Darwin's islands: a natural history of the Galpagos. American Museum of Natural History; 1971
5. Levasseur K, Tilley D, Hein R. Monitoring Eretmochelys imbricata: Tagging and Nesting Research on the Hawksbill Turtle on Long Island, Antigua, West Indies. Wider Caribbean Sea Turtle Conservation Network; 2011. Available from: http://www.jbhp.org/wp-content/uploads/2011/02/JBHP-2011-Annual-Report.pdf
6. Conservancy T. Project Report: Ploughshare Tortoise, conservation efforts. Durrell Wildfire Conservation Trust and Madagascar National Parks and Turtle Conservancy and Andrew Sabin Family Foundation and united states fish wildlife service; 2013. Available from: http://www.turtleconservancy.org/projects/ploughshare-tortoise-report-web.pdf
7. Rutz C, Hays GC. New frontiers in biologging science. Biology letters. 2009. pii: 5:289–292 doi: 10.1098/rsbl.2009.0089 PMID: 19324624
8. Kooyman GL. Genesis and evolution of bio-logging devices: 1963–2002. Mem Natl Inst Polar Res. 2004. pii: 58:15–22.
9. Ropert-Coudert Y, Beaulieu M, Hanuise N, Kato A. Diving into the world of biologging. Endangered Species Research. 2009. pii: 10:21–27. doi: 10.3354/esr00188
10. Bograd SJ, Block BA, Costa DP, Godley BJ. Biologging technologies: new tools for conservation. Introduction. Endangered Species Research. 2010. pii: 10:1–7. doi: 10.3354/esr00269
11. Ropert-Coudert A Y and Kato, Grmillet D, Crenner F. Bio-logging: recording the ecophysiology and behaviour of animals moving freely in their environment. Sensors for Ecology: Towards Integrated Knowledge of Ecosystems. 2012. pii: 17–41.
12. Shamoun-Baranes J, Bom R, van Loon EE, Ens BJ, Oosterbeek K, Bouten W. From Sensor Data to Animal Behaviour: An Oystercatcher Example. PLoS ONE. 2012. pii: 7(5):e37997. doi: 10.1371/journal.pone.0037997 PMID: 22693586
13. Kays R, Jansen PA, Knecht EMH, Vohwinkel R, Wikelski M. The effect of feeding time on dispersal of Virola seeds by toucans determined from GPS tracking and accelerometers. Acta Oecologica. 2011. pii: 37(6):625–631. doi: 10.1016/j.actao.2011.06.007
14. Martiskainen P, Jrvinen M, Skn JP, Tiirikainen J, Kolehmainen M, Mononen J. Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. Applied Animal Behaviour Science. 2009. pii: 119(1):32–38. doi: 10.1016/j.applanim.2009.03.005
15. Gao L, Campbell HA, Bidder OR, Hunter J. A Web-based semantic tagging and activity recognition system for species' accelerometry data. Ecological Informatics. 2013. pii: 13:47–56. doi: 10.1016/j.ecoinf.2012.09.003
16. Ladha C, Hammerla N, Hughes E, Olivier P, Pltz T. Dog's life: wearable activity recognition for dogs. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM; 2013. pii: 415–418.
17. Haykin S. Neural networks: a comprehensive foundation. 3th edition. Prentice Hall PTR; 2009.
18. Bidder OR, Campbell HA, Gmez-Laich A, Urg P, Walker J, Cai Y, et al. Love thy neighbour: automatic animal behavioural classification of acceleration data using the k-nearest neighbour algorithm. PloS one. 2014. pii: 9(2):e88609. doi: 10.1371/journal.pone.0088609 PMID: 24586354

19. Oczak M, Viazzi S, Ismayilova G, Sonoda LT, Roulston N, Fels M, et al. Classification of aggressive behaviour in pigs by activity index and multilayer feed forward neural network. Biosystems Engineering. 2014. pii: 119:89–97. PLOS 20/23 doi: 10.1016/j.biosystemseng.2014.01.005

20. Nathan R, Spiegel O, Fortmann-Roe S, Harel R, Wikelski M, Getz WM. Using tri-axial acceleration data to identify behavioral modes of free-ranging animals: general concepts and tools illustrated for griffon vultures. The Journal of experimental biology. 2012. pii: 215(6):986–996. doi: 10.1242/jeb.058602 PMID: 22357592

21. Technology C. MICAz: wireless measurement system. Crossbow; 2007. Available from: http://www.openautomation.net/uploadsproductos/micaz-datasheet.pdf

22. Baronti P, Pillai P, Chook VW, Chessa S, Gotta A, Hu YF. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. Computer communications. 2007. pii: 30 (7):1655–1695. doi: 10.1016/j.comcom.2006.12.020

23. Soltis J, Wilson RP, Douglas-Hamilton I, Vollrath F, King LE, Savage A. Accelerometers in collars identify behavioral states in captive African elephants Loxodonta africana. Endangered Species Research. 2012. pii: 18(3):255–263. doi: 10.3354/esr00452

24. Grnewlder S, Broekhuis F, Macdonald DW, Wilson AM, McNutt JW, Shawe-Taylor J, et al. Movement activity based classification of animal behaviour with an application to data from cheetah (Acinonyx jubatus). PLoS ONE. 2012. pii: 7(11):e49120. doi: 10.1371/journal.pone.0049120

25. Rogoza V, Sergeev A. The Comparison of the Stochastic Algorithms for the Filter Parameters Calculation. In: Advances in Systems Science. Springer. 2014. pii: 241–250.

26. Smith SW. The Scientist Engineer's Guide to Digital Signal Processing. California Technical Publishing. 1997.

27. Lara OD, Labrador MA. A Survey on Human Activity Recognition using Wearable Sensors. Communications Surveys Tutorials. 2013. pii: 15(3):1192–1209. doi: 10.1109/SURV.2012.110112.00192

28. Rumelhart DE, Hinton GE, Williams RJ. Learning Internal Representations by Error Propagation. Proceedings of the IEEE. 1985. pii: 1:318–362.

29. Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ. Phoneme recognition using time-delay neural networksn. Acoustics, Speech and Signal Processing. 1989. pii: 37(3):328–339. doi: 10.1109/29.21701

30. Lang KJ, Waibel AH, Hinton GE. A time-delay neural network architecture for isolated word recognition. Neural Networks. 1990. pii: 3(1):23–43. doi: 10.1016/0893-6080(90)90044-L

31. Sejnowski TJ, Rosenberg CR. Parallel networks that learn to pronounce English text. Complex systems. 1987. pii: 1(1):145–168.

32. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998. pii: 86(11):2278–2324. doi: 10.1109/5.726791

33. Waibel A. Modular construction of time-delay neural networks for speech recognition. Neural computation. 1989. pii: 1(1):39–46. doi: 10.1162/neco.1989.1.1.39

34. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1995. pii: 3361:255–258.

35. LeCun Y, Jackel LD, Bottou L, Brunot A, Cortes C, Denker JS, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. International conference on artificial neural networks. 1995. pii: 60:261–276. PLOS 21/23

36. Sermanet P, Chintala S, LeCun Y. Convolutional neural networks applied to house numbers digit classification. 21st International Conference on Pattern Recognition (ICPR). 2012. pii: 3288–3291.

37. LeCun Y, Kavukcuoglu K, Farabet C. Convolutional networks and applications in vision. International Symposium on Circuits and Systems (ISCAS). 2010. pii: 253–256.

38. Lpez-Risueo G, Grajal J, Haykin S, Daz-Oliver R. Convolutional neural networks for radar detection. In: Artificial Neural Networks. Springer. 2002. pii: 1150–1155.

39. Hastie T, Tibshirani R, Friedman J, Hastie T, Friedman J, Tibshirani R. The elements of statistical learning. vol. 2. Springer. 2009.

40. Barbuti R, Chessa S, Micheli A, Pallini D, Pucci R, Anastasi G. Tortoise@: a system for localizing tortoises during the eggs deposition phase. Atti Societa Toscana Scienze Naturali, memorie B. 2012. pii: 119:89–95. Available from: www.stsn.it/Atti2012/Barbuti.pdf

41. Barbuti R, Chessa S, Micheli A, Pucci R. Identification of nesting phase in tortoise populations by neural networks. Extended Abstract. The 50th Anniversary Convention of the AISB, selected papers. 2014. pii: 62–65.

42. Bettini C, Brdiczka O, Henricksen K, Indulska J, Nicklas D, Ranganathan A, et al. A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing. 2010. pii: 6(2):161–180. doi: 10.1016/j.pmcj.2009.06.002

43. Kuehler C, Lieberman A, Oesterle P, Powers T, Kuhn M, Kuhn J, et al. Development of restoration techniques for Hawaiian thrushes: collection of wild eggs, artificial incubation, hand-rearing, captive-breeding, and re-introduction to the wild. Zoo Biology. 2000. pii: 19(4):263–277. doi: 10.1002/1098-2361(2000)19:4%3C263::AID-ZOO4%3E3.0.CO;2-G

44. Colbourne R, Bassett S, Billing T, McCormick H, McLennan J, Nelson A, et al. The development of Operation Nest Egg as a tool in the conservation management of kiwi. Science and Technical Pub. Department of Conservation. 2005.

45. Mitrus S. Headstarting in European pond turtles (Emys orbicularis): Does it work? Amphibia-Reptilia. 2005. pii: 26(3):333–341. doi: 10.1163/156853805774408504

46. Aguilera WT, Mlaga J, Gibbs JP. Giant tortoises hatch on Galapagos island. strategies. 2013. pii: 497:306–308.