



Research article

A stacking ensemble learning for Iberian pigs activity prediction: a time series forecasting approach

Federico Divina*, Miguel García-Torres, Francisco Gómez-Vela and Domingo S. Rodríguez-Baena

Division of Computer Science, Universidad Pablo de Olavide, E. S. 41013, Seville, Spain

* **Correspondence:** Email: fdivina@upo.es; Tel: +34954977592.

Abstract: Automatic determination of abnormal animal activities can be helpful for the timely detection of signs of health and welfare problems. Usually, this problem is addressed as a classification problem, which typically requires manual annotation of behaviors. This manual annotation can introduce noise into the data and may not always be possible. This motivated us to address the problem as a time-series forecasting problem in which the activity of an animal can be predicted. In this work, different machine learning techniques were tested to obtain activity patterns for Iberian pigs. In particular, we propose a novel stacking ensemble learning approach that combines base learners with meta-learners to obtain the final predictive model. Results confirm the superior performance of the proposed method relative to the other tested strategies. We also explored the possibility of using predictive models trained on an animal to predict the activity of different animals on the same farm. As expected, the predictive performance degrades in this case, but it remains acceptable. The proposed method could be integrated into a monitoring system that may have the potential to transform the way farm animals are monitored, improving their health and welfare conditions, for example, by allowing the early detection of a possible health problem.

Keywords: animal behavior prediction; machine learning; forecasting methods; ensemble learning

Mathematics Subject Classification: 68T05, 68T07, 68T09, 92D50

1. Introduction

Smart farming [1–4] involves the use of machine learning (ML) methods to improve results regarding different aspects, such as animal health and welfare. Within this domain, smart livestock management is an important component [5]. In fact, it helps in monitoring the health, welfare, productivity and reproduction of animals throughout their life cycle. Usually, sensors and cameras are used to monitor the activity of animals, and this data can then be analyzed in order to obtain an

automatic way to make decisions, for example for early treatments or for stopping the spread of a disease.

Within livestock production and management, we can identify two sub-categories: animal welfare and livestock production. In production the aim is to use ML methods for obtaining predictions of the production of livestock and thus increase the economical benefits of farmers. On the other hand, when ML is used for animal welfare, the objective is to improve the health of animals, for example with an early detection of diseases or pain situations [6,7]. This would facilitate a timely application of proper treatment, which could yield a faster recovery and reducing loss of production [8–10]. However, the continuous observation of livestock by breeders is not a feasible solution in a commercial setting.

However, nowadays, the use of technologies such as the internet of things (IoT) sensors can assist in this task. In fact, sensors and cameras can produce a massive amount of valuable data. However, such data need to be collected and analyzed in order to extract useful information. It follows that an automatic way of detecting abnormal changes in activity is required.

This is because an automated system that can monitor several behavioral or activity patterns at once would be beneficial in animal production as an aid to assess health and welfare [11]. Such a system would allow one to estimate whether an animal follows a regular activity pattern, and it may be a useful tool to assess the health welfare status of the animal [12]. For this purpose, one should be able to recognize behavioral signatures associated with normal and abnormal behavior [13].

Many studies proposed in the literature (see Section 2) use data generated from camera and/or sensors located on the animals in order to predict a given behavior. Such data are then, generally, manually annotated and classification algorithms are applied in order to obtain predictive models.

In this work, we had a more general objective, i.e., the prediction of animal activity, and not just of a specific behavior. The main motivation is that data collected by sensors and/or cameras should be manually annotated in order to obtain a dataset that can be used to acquire a predictive model for specific behaviors. Such annotations could be a potential source of errors, and thus could compromise the validity of the models obtained. Moreover, if animals are bred in wide open spaces with the possible presence of medium and high vegetation, as it is usually the case with Iberian pigs, such a manual annotation phase would be nearly impossible given the characteristics of the environment, which is the case for the animals considered in this work. One can easily imagine that it would be very difficult for a person to follow even a single animal in such an environment. The use of cameras, which is common practice in closed environments, would not be possible either, since the environment is too vast to be fully covered, and the vegetation present in the breeding area would block the view creating many blind spots, where the behavior of an animal could not be recorded. In order to address this problem, some alternative ways of collecting data should be used; for instance in [14] the authors proposed the use of drone-recorded videos and computer vision approaches to automatically track the location and body posture of free-roaming animals. However such an approach would require the involvement of a human expert in drone flying. On the other hand, the activity could be directly collected by the sensors located on the animals, e.g., accelerometers. This would eliminate the errors due to the manual annotation and such a solution can be used in vast environments, even in the presence of vegetation. In this way, we address the problem of predicting the activity of animals as a time series forecasting model rather than a classification problem. To the best of our knowledge, this work is the first attempt to tackle this problem as a time series forecasting problem.

In order to achieve our objective, a forecasting model was developed, and if the activity of a specific

animal does not fit such a model, then it can be evidence of an early sign of a disease; thus, the breeders could focus on the animal that presented such anomalous conduct.

Toward this aim, we analyzed data collected by sensors located on the animals, in particular on Iberian pigs, which recorded data relative to the movements. A predictive model was then extracted for each animal monitored. Our approach does not involve the use of cameras, as this allowed us to monitor the animal continuously, without the need to be in the range of cameras. In addition to this, cameras may face demanding conditions, e.g., the occlusion of one pig by another, that are not present when sensors are used. Moreover, for our specific case, the pigs are bred in a large open space, which would make it difficult to perform the visual monitoring of animals.

In this paper we will propose a comparative experimentation scheme for different ML and statistical methods for the prediction of animal activity. Toward this aim, data produced by triaxial (X, Y, Z) accelerometers located on twenty Iberian pigs was used. In particular, we will propose a novel stacking ensemble learning strategy, where a base layer of predictors will provide the input to a second layer of aggregation methods, which will in turn produce the final predictions.

Moreover, we also wanted to check if the model designed and trained for one animal could be used to predict the activity of other animals. This would considerably reduce the computational costs of possible monitoring systems, as only one predictive model should have to be obtained and possibly updated.

We summarize the main contributions of this work as follows:

- Introduction of a novel stacking ensemble strategy for the forecasting of Iberian pigs activity.
- Consideration of the problem of prediction of animal activity as a time series forecasting problem rather than a classification problem.
- Experimental comparison of different ML and statistical methods for the prediction of Iberian pig activity.
- Experimental verification that the model trained on one animal's activities can be transferred to other animals.

The rest of the paper is organized as follows. In Section 2 an overview of recent related works is provided. In Section 3 we will provide animal's activities of the data used in this paper and the different methods used for the prediction of animal activity. Results are reported in Section 4, and in particular results regarding the study on transfer learning are reported in this section. Finally, in Section 5 we will draw the main conclusions and identify possible future works.

2. Related works

As mentioned before, the incorporation of IoT technologies and data processing techniques into the farming sector plays a very important role in the improvement of production and animal health and welfare. In fact, such advances enabled the ability to predict livestock activity and/or behavior, since a huge amount of activity data may be available. It follows that the use of both statistical and ML techniques has become essential in the analysis of such data in order to extract useful information [15] from it, for instance for the construction of behavior prediction models.

Some works use statistical data analysis to establish relationships between data and animal behavior. For example, in [12], the authors use data produced by accelerometers to distinguish different types of gait (walking, trotting, and galloping) through the use of Spearman correlation coefficient analysis. Another example is presented in [16], where linear regression has been applied to data originating from two different types of sensors, i.e., pedometers and accelerometers. The results obtained show that the lying time measured by the pedometer and the number of lying bouts measured by the accelerometer are closely related. A quadratic discriminant analysis was applied to classify sheep's lame behavior based on a change in the acceleration signals in [17]. Finally, in [18], the authors used a two-way factorial analysis in order to predict calving time in dairy cattle.

ML techniques are also used in the prediction of cattle behaviors. For instance, in [11], a support vector machine strategy was applied to data originating from accelerometers and visual observations, with the aim of classifying behavioral patterns of dairy cows, like standing, lying, ruminating or feeding. Convolutional neural networks (CNNs) were used in [19] to identify postures and drinking behaviors of group-housed pigs, while in [20] four ML strategies were used to predict different behaviors and body posture in sheep. CNNs were also used in [21] to classify animal behavior in situ and in real time.

Ensemble methods, e.g., random forests (RFs), are also gaining more and more attention. For instance, RFs, linear discriminant analysis and neural networks were used for calving prediction in [22]. RFs were also used by Schmeling et al. [23], where decision tree, support vector machine and naive Bayes were applied for the prediction of lying behavior in dairy cows.

In [24], an offline k-nearest neighbor (KNN) algorithm was used to predict walking, standing or lying behaviors. Then, an online k-means technique obtains clusters over these results. Finally, a combined algorithm that is based on decision rules from prior historical data is applied to the previous outputs, producing an overall classification label. Other scholars that used ensemble strategies are, for instance, Keceli et. al. [25], who applied the random undersampling boosted (RUSBoosted) tree classifier, which is an ensemble method that uses the RUSBoost data balancing algorithm [26], to predict the eight hours before calving. In [27], an adaptive boosting (AdaBoost) ensemble learning algorithm was used to classify postures and activities in dairy calves. As mentioned in the introduction, a stacking ensemble is based on different levels, where predictions of different base learning algorithms are used by a meta-learner to produce the final predictions. To the best of our knowledge, such a strategy was only used in [28], where it was applied to classify behaviors. In particular, grazing, resting and walking behaviors were classified by using GPS data. To achieve this a two-layer system was proposed. The first layer consists of 13 ML algorithms, while two different meta-learners were used in two stacking ensemble strategies. Another work that addresses the problem of estimating the bite rate of grazing cattle from accelerometer data by using semi-supervised linear regression is [29].

Other works address farming production, such as [30], where a genetic programming symbolic regression approach was used to predict the production of calves weaned per cow, or [31] where Xtreme gradient boosting (XGBoost) was used to predict milking frequency, daily milk yield, protein content and the fat content of milk.

As it can be observed, most of the works were focused on classifying different behaviors, i.e., addressing the problem as a classification problem, as opposed to predict activity in general. For a recent survey on the subject, we refer the reader to [32]. As far as we are aware, only in [33] have active and inactive states been taken into consideration in the first phase of a hierarchical classification

model. Later, in a second phase, the active behavior obtained in the first phase is further classified as grazing or walking.

3. Material and methods

The complete process of the present work is summarized in Figure 1. In this figure, two phases can be clearly distinguished; on the one hand, there was data collection, which was carried out in real time, and on the other hand, there are data processing and analysis, which was carried out offline. In the first phase, the movement data is collected by the loggers, and once collected, it is downloaded and processed. In the second phase the complete process, which will be detailed in this section, is carried out. In what follows, we will especially address the offline phase, and we refer the reader to [34] for further details on the real time phase.

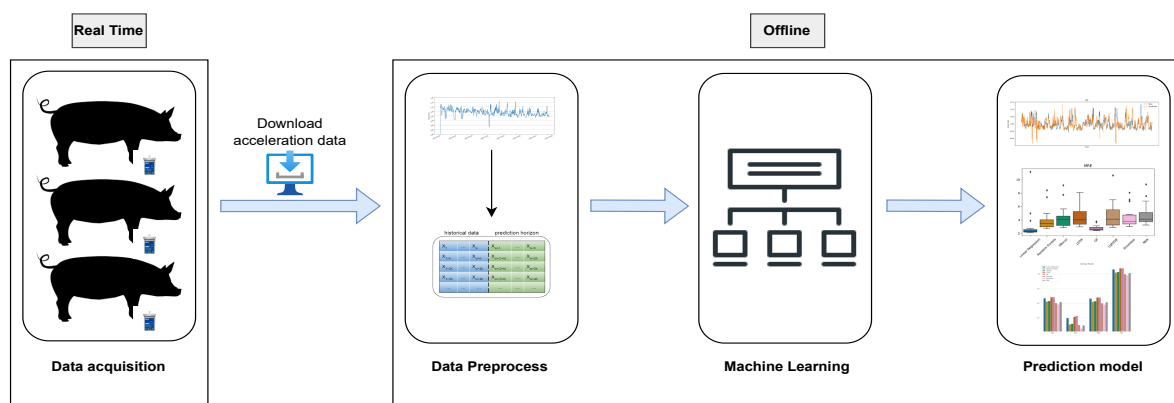


Figure 1. The entire proposed process is detailed in the figure. The data collection is carried out in real time. Downloading and analysis of the data is performed offline.

3.1. Data

The data employed in this work were first proposed in [34] and consists of acceleration data collected on 20 Iberian pigs by using a triaxial accelerometer. In particular the HOBO® Pendant G Data Logger (UA-004-64) [35] was used and it was physically positioned on the animals. This logger is capable of simultaneously recording acceleration and tilt by measuring an analog signal in each of its three axes (X , Y and Z), and it has 64 KB of data memory. The acceleration is stored using gravity (g) as the unit of measurement, so it actually stores m/s^2 . The time required to fill the memory depends on how many channels are being recorded and the recording rate.

The specific characteristics of the data collection process for this dataset are summarized as follows. As already mentioned, the data collection involved twenty pigs and lasted twelve days. The animals were 5 months old, weighed approximately 40 kg, and were located in an outdoor area of about 2500 m^2 . The device was placed in the left ear of the pigs and set to have a recording interval of 30 seconds. For more details, we refer the reader to [34].

The data described were processed in order to obtain a time series for each pig. The resulting time series were used as input to the forecasting algorithms that will be detailed later. First, we aggregated

the three forces (the X , Y and Z axes) read by the devices into a single signal called the sum vector (S):

$$S = \sqrt{X^2 + Y^2 + Z^2}.$$

After this first phase, the data presented missing values, which could have been caused by failures in the devices, or by the time elapsed since the devices were placed in the different animals. Therefore, such gaps were filled by using interpolation to try to minimize the impact on the final results [36].

Finally, the data were aggregated every 10 minutes. Aggregating time series values is a common practice in the literature [37–39]. It is usually employed to find more general results in the studies performed or to reduce large datasets.

An example of a resulting time series is shown in Figure 2, where the activity value of animal 2 was reported over the entire period of collection of the data. We can observe that in this period, the activity registered for this particular animal was usually within the range of 1 and 1.1, with some peaks that could indicate a very active or a very inactive state. In order to test whether or not the so obtained time series were stationary, we have derived the autocorrelation and the partial autocorrelation graphs of each time series. In Figure 3 an example of such graphs, relative to animal 2, is shown. From this analysis, we concluded that all graphs presented a few significant lags but these died out quickly; thus, we can conclude that our series is stationary. In order to confirm this result, we have also applied the augmented Dickey-Fuller test to each time series [40]. The results confirm that all time series were stationary.

In order to use the time series we transformed them as in [41,42]. Each time series was transformed into a matrix, whose dimension depends on a historical window, w , and a prediction horizon, h , which are two parameters. The historical window (w) represents the number of previous entries that are used to train an algorithm in order to obtain a model that will be used to predict the subsequent values (h). Figure 4 graphically shows this process.

In this work, the prediction horizon (h) was set to 24, corresponding to a period of 4 hours. On the other hand, the size of the historical window was varied, and as we will see in Section 4, we tested various values of w (going from 8 up to 28 hours with steps of 4 hours) in order to estimate the optimal amount of historical data to be used.

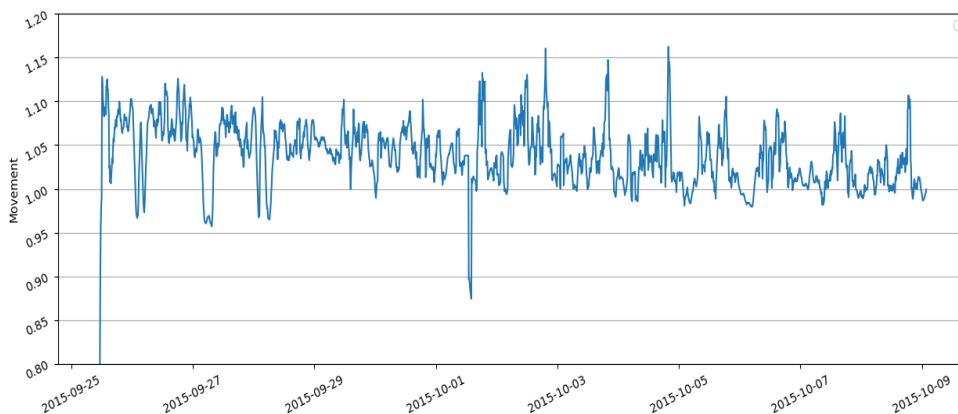


Figure 2. Example of a resulting time series for animal 2. In this figure the sum vector is shown.

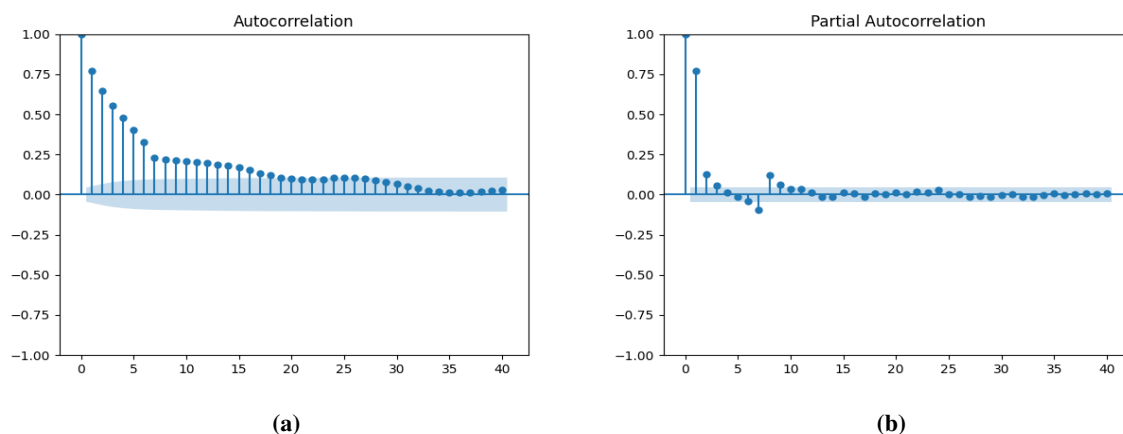


Figure 3. Correlation plots for the time series relative to animal 2: (a) Autocorrelation function; (b) Partial Autocorrelation function.

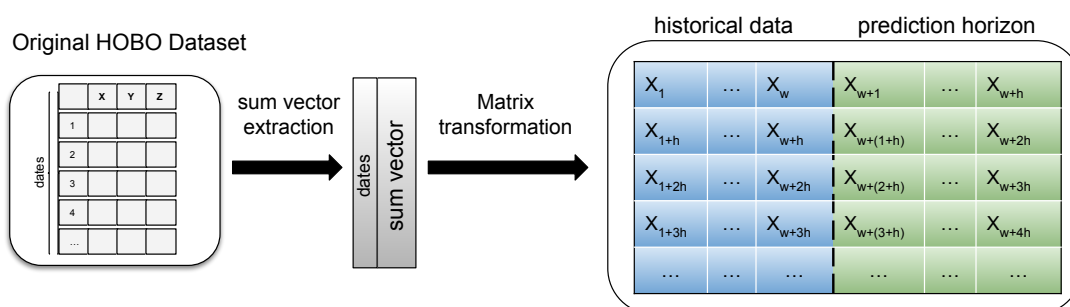


Figure 4. Preprocessing of the data. w determines the amount of the historical data used, while h determines the prediction horizon. X_i represents a value of the sum vector.

3.2. Methods

This section describes the methods used in this work, which were selected due to their popularity as well as for their good performance. We also provide a description of the stacking ensemble method that we propose.

Linear regression (LR) [43] models linear relationships between a response variable Y and one or more explanatory variables $X_i, i = 1, \dots, n$ through the equation

$$Y = a + \sum_{i=1}^n b_i X_i + \varepsilon.$$

This equation also considers the error term ε that adds noise to the relationship. This approach is referred to as multiple LR when the number of explanatory variables is 2 or more. We have used the implementation provided by the SciKit-learn package [44].

KNN [45] is a non-parametric procedure that involves estimating the value of a new point based on the interpolation of its k nearest neighbors. The interpolation uses uniform weights so that each point in the local neighborhood contributes uniformly to the calculation of a query point. The search of neighbors involves the brute-force computation of distances between all pairs of points in the dataset. To speed up the computation of nearest neighbors, in this work we used the KD-Tree [46], which is a tree-based space partitioning data structure for organizing points in a k -dimensional space. This structure improves the efficiency of the nearest neighbourhood search. In this work we have used 8 neighbors in order to produce the predictions.

Long short-term memory (LSTM) [47] is a deep learning approach that uses a recurrent neural network [48] architecture. LSTM can process sequences of data based on long-short term memory. It uses short term memory to remember the previous information and use it to process the current input. The long-term memory allows one to learn long-term temporal dependencies even in the presence of long time lags. After having tested several configurations, in this work we used two LSTM layers with 48 and 24 units, using a batch size of 24, 1000 epochs, the Adam optimizer with a learning rate of 0.001 and the hyperbolic tangent activation function, as well as the Keras package [49].

RFs [50] constitute a bagging ensemble method that builds a set of decision trees during the training step. Each tree makes its own prediction and the final value is computed as the average over all trees. The RF builds the models by using a bagging approach. For each tree a random sample with replacement is selected and fits the trees to these samples. Compared to a decision tree, it decreases the variance without increasing the bias. In this study, we have used the implementation provided by the SciKit-learn package [44], with 50 estimators with a maximum depth of 8, a minimum number of samples to split an internal node of 5 and a learning rate of 0.01.

Symbolic regression [51] is a generalization of LR, which aims to identify an underlying mathematical expression that best fits to the data. In contrast to classical regression, this approach avoids any prior assumption since it infers the model structure and parameters from the data. In this work, the search is performed by using a genetic programming (GP) approach. GP first builds a population of naive random formulas to represent the relationship between the input variables and the class targets. Then, such relationships evolve by applying the crossover and mutation operators to random individuals from the current generation. The best individuals are selected to produce the next generation. This process continues until a stopping criterion is reached, which is usually a convergence criterion and/or when the maximum number of generations has been reached. We have used the following function set: (*add, sub, mul, div, sqrt, log, abs, neg, inv, max, min*), a population of 600 individuals and a maximum of 20 generations. As for the fitness function we have used the mean squared error (MSE) function, while the probabilities of the subtree crossover and mutation were set to 0.7 and 0.1, respectively. Hoist and point mutation were applied with 0.05 and 0.1 probabilities, respectively. We have used the implementation provided by the GPLearn package [52].

Gradient boosting (GBoost) decision trees [53] constitute a boosting ensemble strategy that generates a set of decision trees. In a nutshell, GBoost combines a series of weak base classifiers into a strong one. It uses an iterative approach to learn from the mistakes of weak classifiers, so that it fits a decision tree on the residuals error (gradient) of the previous tree. Finally, the global convergence is achieved by following the direction of the negative gradient. The SciKit-learn implementation [44] was used under the conditions of 100 trees with a maximum depth of 4 and a learning rate of 0.05.

GBDT with piecewise linear trees [54] is an extension of GBoost, as it uses piecewise linear (PL)

regression trees, instead of piecewise constant regression trees, as base learners. This difference helps the algorithm to accelerate the convergence and improve the accuracy. In this study, we have used the implementation included in the light gradient boosting (LightGB) machine package [55], and we will refer to this method as LightGB in the remainder of the paper. As for GBoost the learning rate was set to 0.05; however, the number of estimators was set to 200 in this case.

Stacking ensemble [42] techniques are applied to construct a predictive model by using different learning algorithms; then, a meta-learner algorithm is used to produce the predictions, which are based on the results generated by the base algorithms. More formally, a stacking ensemble strategy can be defined as follows.

Let \mathcal{L}_k , $k = 1, \dots, N$ be a set of learning algorithms, and the pair is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$, where

$$\mathbf{x} = (x_1, \dots, x_w)$$

represents the w recorded values and

$$\mathbf{y} = (x_{w+1}, \dots, x_{w+h})$$

denotes the h values to predict. Let m_{kj} , $k = 1, \dots, N$, $j = 1, \dots, h$ be the model generated by the learning algorithm \mathcal{L}_k on \mathbf{x} to predict x_{w+j} , and let f_j be the generalized function responsible for producing the final predictions. f_j can be a standard function, e.g., the average, or a model induced by a learning algorithm. Then, the prediction \hat{x}_{w+j} value is given by the following expression:

$$\hat{x}_{w+j} = f_j(m_{1j}, \dots, m_{Nj}).$$

Stacking ensembles are different from other ensemble methods, such as bagging or boosting. In fact, in bagging methods, like RFs, different learners are obtained by using a different, randomly selected, subsample of training data and features. The predictions of the learners are then used in a voting scheme in order to produce the final results. Boosting methods, such as AdaBoost [56] or GBoost methods, are similar, but they obtain a sequence of predictive models that focus each time on examples for which the previously obtained learners performed poorly. On the other hand, as described above, stacking ensemble learners use a set of base learners, whose predictions are combined by a meta learner in order to produce the final results. The base learners are trained on the same data. Stacking ensembles are successful due mainly to three reasons [57]. First, predictive models can be seen to search a hypothesis space H to identify the best hypothesis. However, since, usually the datasets are limited, we can find many different suboptimal hypotheses in H . It is not possible to establish a priori which hypothesis will generalize better, i.e., which will perform the best on unseen data. Ensemble methods can perform better since they can use various models in order to obtain a good approximation of the optimal hypothesis.

Another reason is that many search strategies may get stuck in local optima. Ensemble strategies combine different search mechanisms, where the search may start from different points. This may provide a better approximation for the optimal function to apply.

Finally, the unknown optimal hypothesis may not be included in H . However, a combination of several hypotheses drawn from the search space H can expand the space of possible hypotheses, which could include the optimal hypothesis.

Our proposal consists of a two-layer based stacking ensemble approach. The strategy considers several weak learners that are applied independently before the individual results are combined by

using a meta-learner to output the final prediction. In the first layer, RF, KNN and LightGB algorithms are used, while in the second layer, LR is used to produce the final predictions. Notice that the input to the first layer consists of the original data, whose results are then the input of the meta learner, which produces the final predictions. Extensive experimentation was performed in order to select both the base learners and the meta learner, and different combinations of algorithms were considered. Results of this experimentation yielded the selection of the above mentioned methods since their combination produced the best results. Notice that for each learner used in the proposed stacking scheme, we have used the same parameter settings as when the single learners were used. In order to select the parameter settings for the methods, a grid search strategy was used, which considered the combination of various parameter values. This grid search was performed in order to optimize the parameters of the base learner as well, as those of the meta learner.

Figure 5 shows a graphical representation of the ensemble strategy proposed in this paper. We can see that the training data are used by the three base learners, and their output represents the input for the meta-learner. This last algorithm uses the predictions produced by the first layer in order to produce the final predictions.

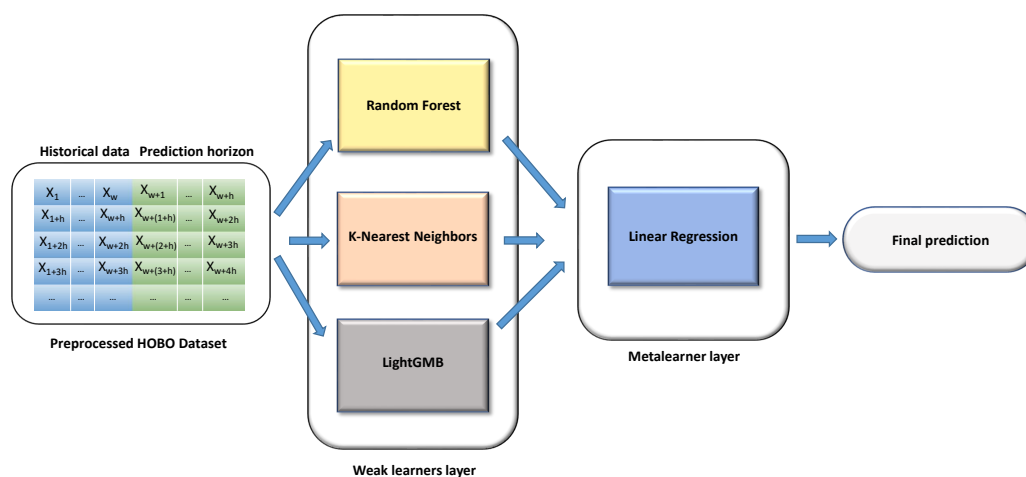


Figure 5. Graphical scheme of the stacking ensemble strategy used in this work.

In the remainder of the paper we will refer to this method simply as an “ensemble”.

4. Results

In order to test our proposed method and the other ML methods, we used different historical window sizes w . More specifically we used historical windows of 168, 144, 120, 96, 72 and 48 reads, corresponding to 28, 24, 20, 16, 12 and 8 hours, respectively. The prediction horizon was kept fixed to four hours, i.e., to 24 reads. After generating the matrices corresponding to the different values of w (see Section 3.2 for details), the first 70% of the rows were used for training, and the remaining 30% were used for testing. This assured that the predictions would be based exclusively on past observations. The aim was twofold: find the best value for w and compare our ensemble proposal to the other strategies.

Figure 6 shows the average results obtained on the 20 time series by the various algorithms, using

the above mentioned values of w (reported on the x axis). In particular, Figure 6a presents the average values of the mean relative error (MRE), while Figure 6b–d report the average results for mean absolute error (MAE), mean squared error (MSE) and the mean absolute percentage error ($MAPE$), respectively.

These measures are defined as follows:

$$MRE = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i + \hat{Y}_i|}{Y_i}, \quad MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|,$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i - \hat{Y}_i|}{Y_i}, \quad MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

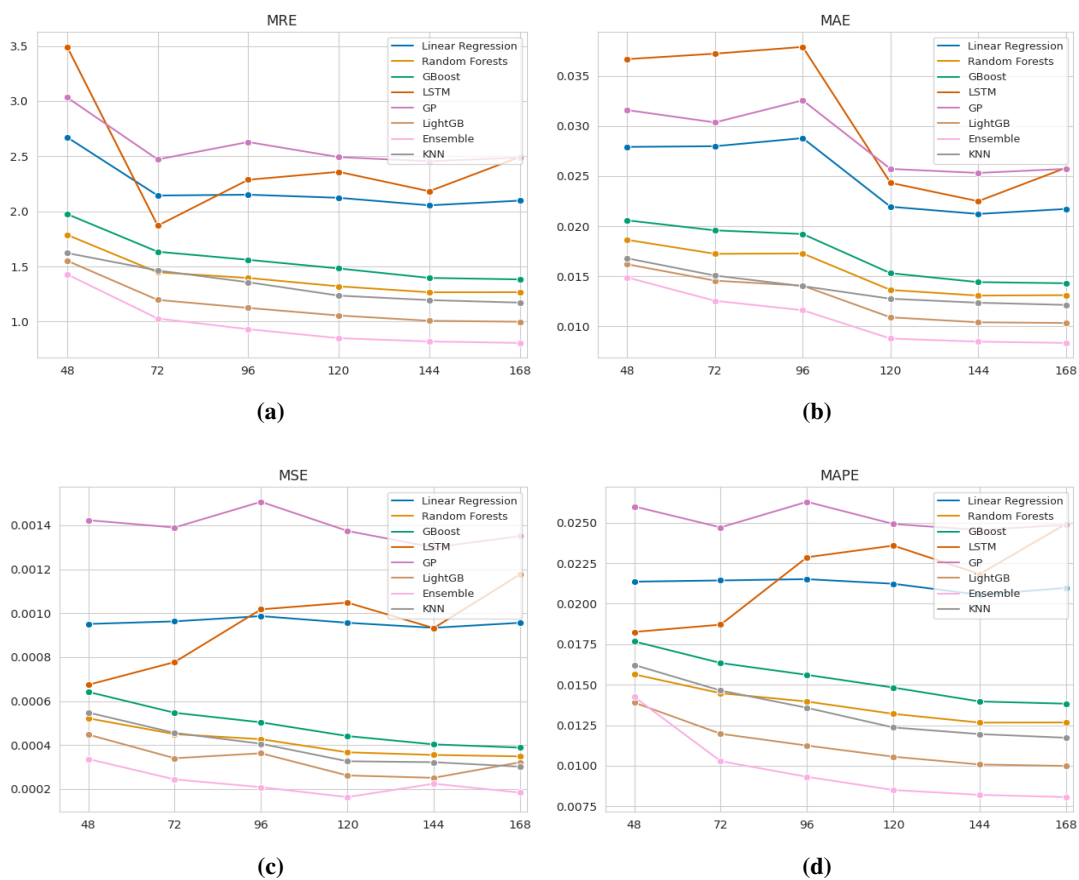


Figure 6. Average results obtained for each value of w used.

In the above equations, \hat{Y}_i is the predicted value and Y_i the real value. Since these measures provide an estimation of the prediction errors, they have to be minimized.

The first thing that we can notice is that, in general, the higher the values of w , the better the average results. There is little difference for historical window sizes of 144 and 168. In fact the results obtained by the ensemble approach proposed in this paper with these two values were not significantly different, according to a Wilcoxon signed-rank test with a confidence level of 95%. Only the LSTM algorithm

yielded noticeably better results when w was set to 144; however, this difference is not statistically significant, according to the same test as before. From these graphs, we can also observe that the ensemble strategy proposed is the one that obtained the best results for all values of w , and we can notice that the tendency of the other ensemble approaches (i.e., RF, GBoost and LightGB) varied in a similar way according to the value of w . Based on these results, we can conclude that the best sizes of historical window to apply to predict the pigs' activity in the next 4 hours are 24 and 28 hours.

In Figure 7 the average *MRE* values obtained by each algorithm for each value of w are shown.

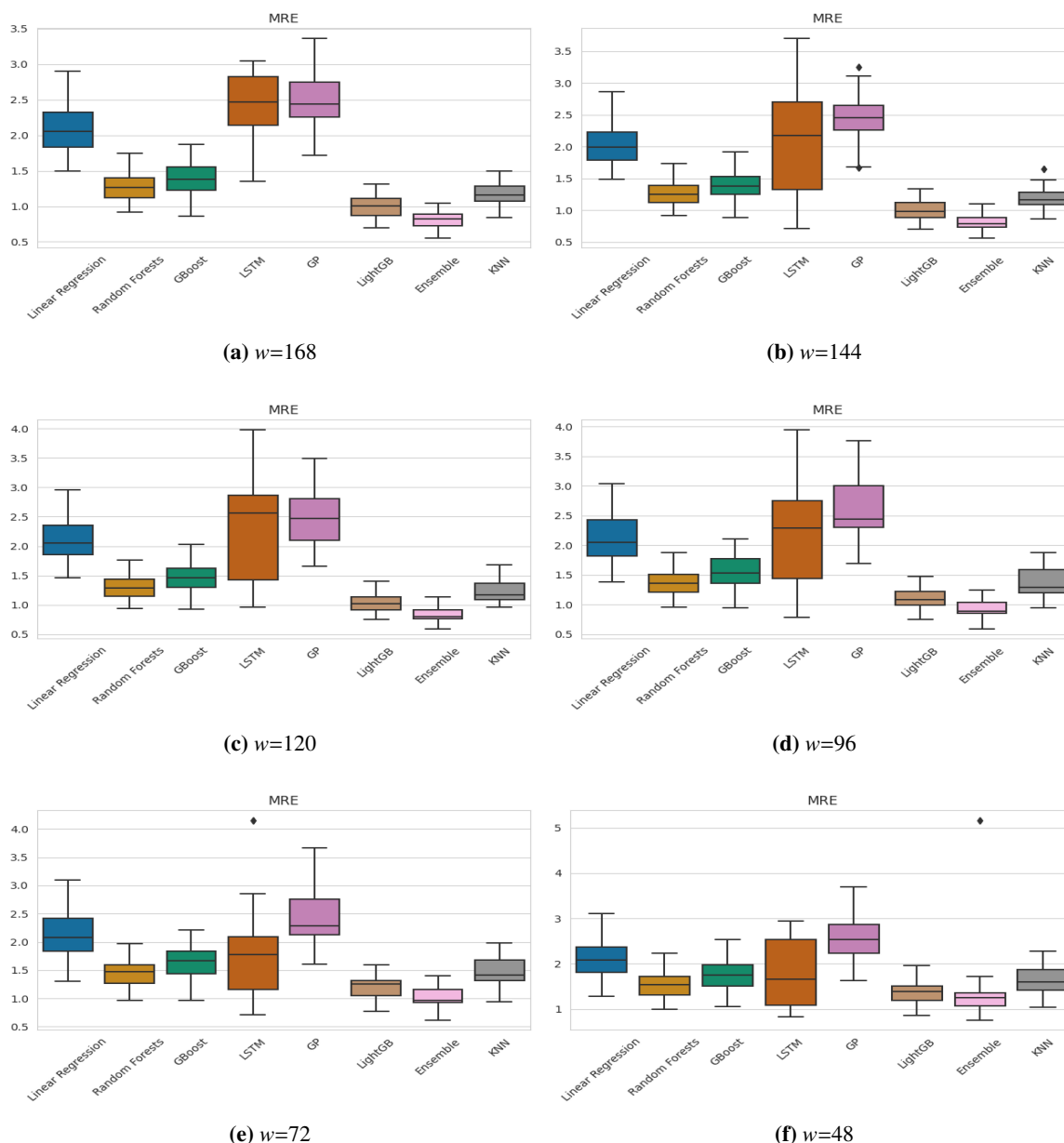


Figure 7. Comparison of the *MRE* obtained by various algorithms for each value of w used.

We can see that our proposed method clearly outperforms the other algorithms. LightGB is the

strategy that achieved the second best results. We can also notice that in general the forecasting performances of the algorithms are more similar with lower values of w . For instance, for $w=48$, the results of the ensemble approaches are very close to each other. However the differences are still significantly different according to a Wilcoxon signed-rank test with a confidence level of 95%. We can also notice the presence of an outlier for the proposed ensemble strategy, when $w=48$. In particular, this outlier is due to animal 15, whose time series exhibited more accentuated behavioral peaks. The proposed strategy does not seem to be able to capture such peaks with a small window size.

It is also worth noticing that LSTM yielded better results when a window size of 48 was used. This is probably due to the fact that higher values of w correspond to fewer rows of the data matrix, and the LSTM could not train satisfactorily on many time series; thus, it yielded poor forecasting performances on test data. On the other hand, GP yielded very similar results on all values of w , even if these results were, in general, worse than those of the other algorithms considered; however, we can consider GP as a robust method in terms of the historical window size. We can also notice that the ensemble approaches are those that obtained the best results. A simple approach like KNN also yielded satisfactory predictions.

The results regarding the *MAE* obtained by the various algorithms when using different window sizes are reported in Figure 8. We can see that these results also confirm the superiority of our proposed method, as, also in this case, it achieved the best results. In this case it is easier to check the superiority of our strategy, as the *MAE*, for instance with $w = 168$, was determined to be about 0.01. This allows us to conclude that the strategy's predictions are precise, considering that the average value of the reads was about 1.02 with a standard deviation of 0.05.

Figures 9 and 10 also confirm the above mentioned considerations. We can observe that, for higher values of w , the performance level of GP was almost the same as that of LSTM, as far as the *MAE* and the *MAPE* are considered.

In Figure 11, we can see an overview of the results obtained. In the graph, a logarithmic scale was used in order to scale the results. From this figure, we can clearly see that our proposed method outperformed all other strategies for all values of w considered.

Finally, we show two examples of predictions produced by our proposed method with $w = 168$. Figure 12 shows the best predictions, according to the *MAE*, produced by our proposed method with $w = 168$. In particular these predictions were produced for animal 1, and Figure 12a shows the forecasting for the whole test set, while, for a better appreciation of the results, Figure 12b shows only the forecasting for a subset of the test data.

On the other hand, in Figure 13, the worst predictions, according to the *MAE*, produced by our ensemble method are shown for $w = 168$. In this case, the results are relative to animal 18. We can see that in both cases, the method was able to accurately predict the activity of animals. It is interesting to notice that the method is able to predict peaks of activity. This is especially important, for instance, for predictions in the case of animal 18, whose activity time series was characterized by many peaks. These results confirm that our proposed method could help breeders to predict the activity of Iberian pigs, and, in this way, to obtain a normal activity pattern.

If such a pattern is not matched, then it can be an early indication of a health or welfare problem, which could help farmers to take timely action in order to prevent further problems.

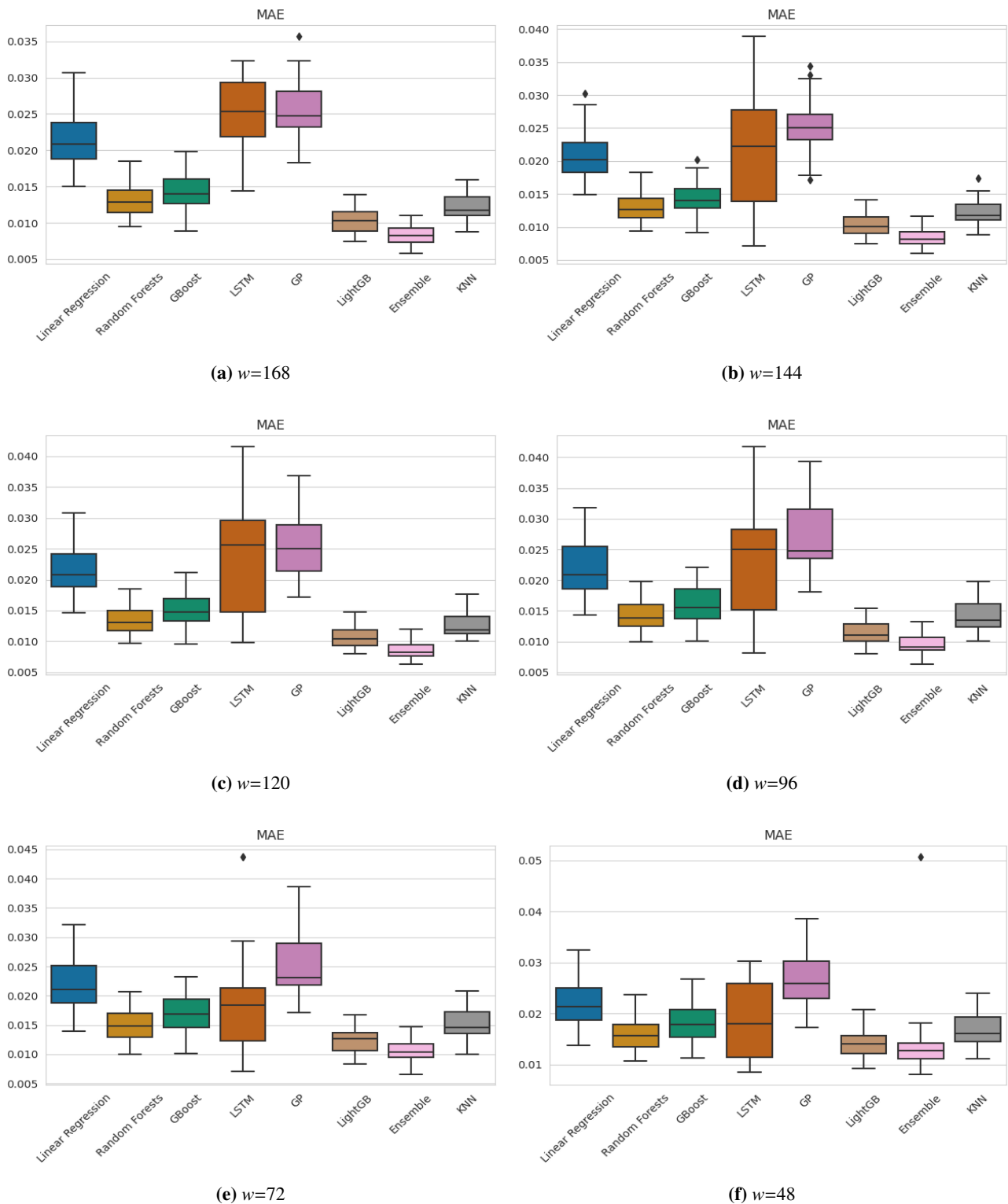


Figure 8. Comparison of the *MAE* obtained by various algorithms for each value of w used.

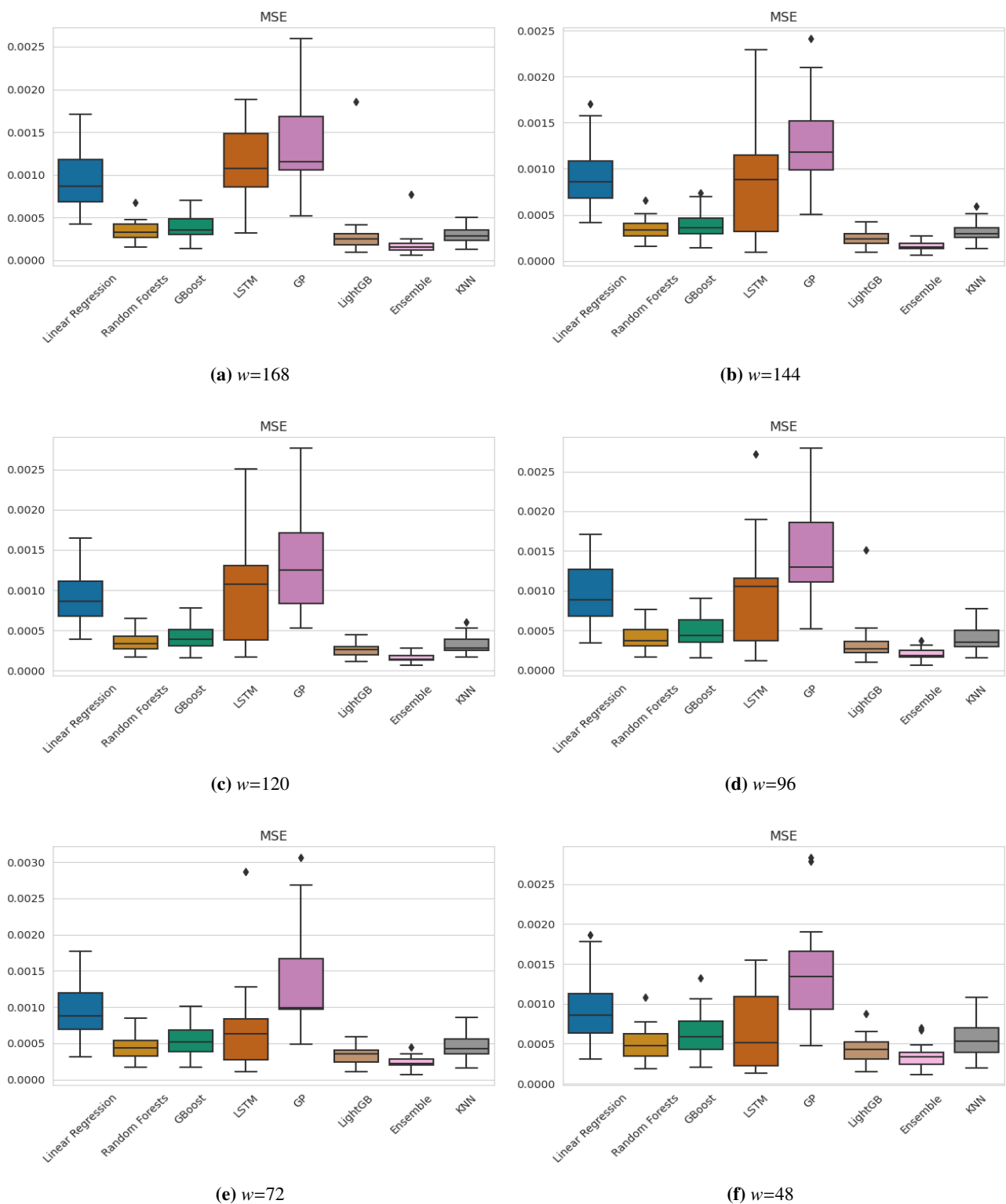


Figure 9. Comparison of the *MSE* obtained by various algorithms for each value of w used.

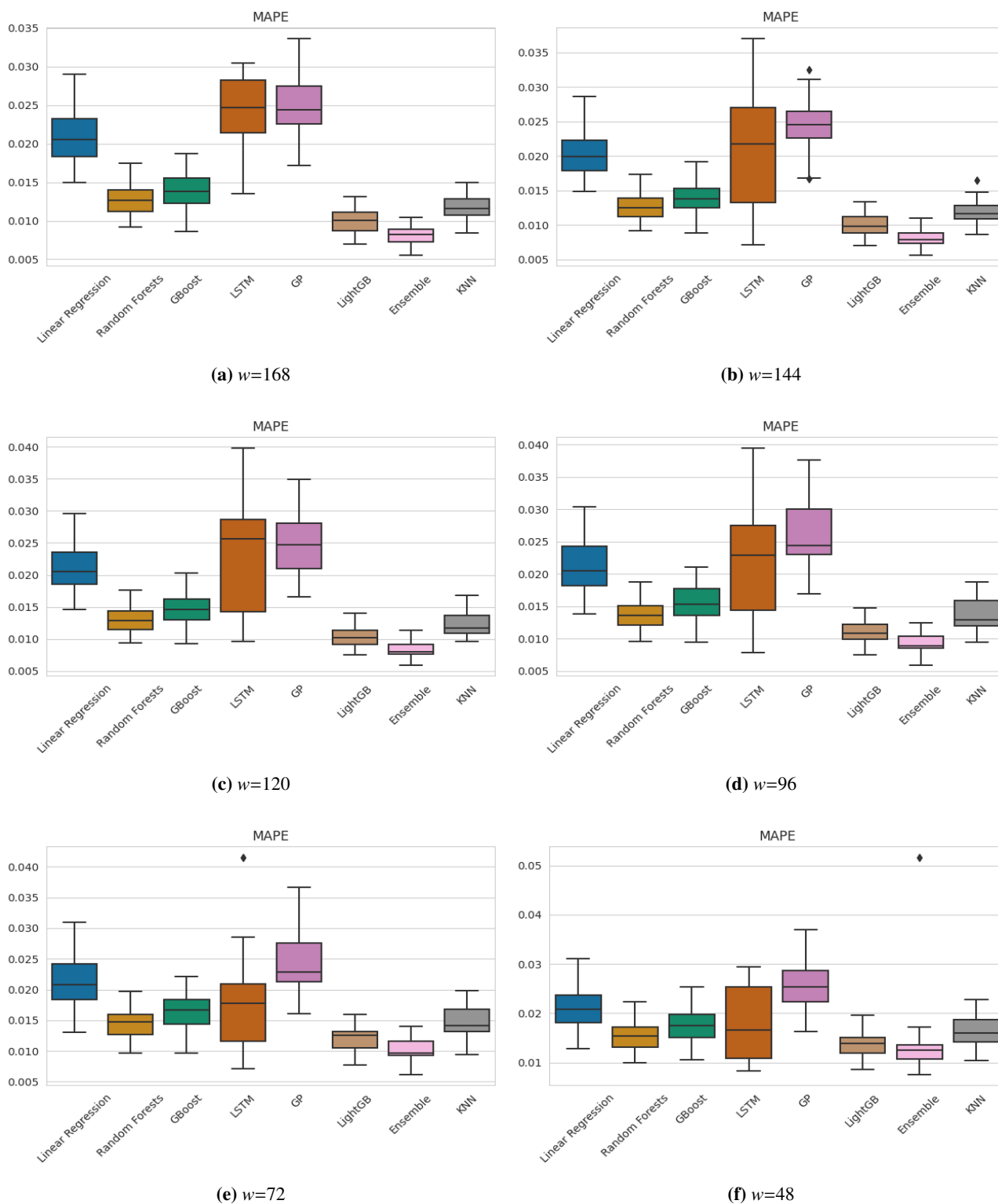


Figure 10. Comparison of the *MAPE* obtained by various algorithms for each value of w used.

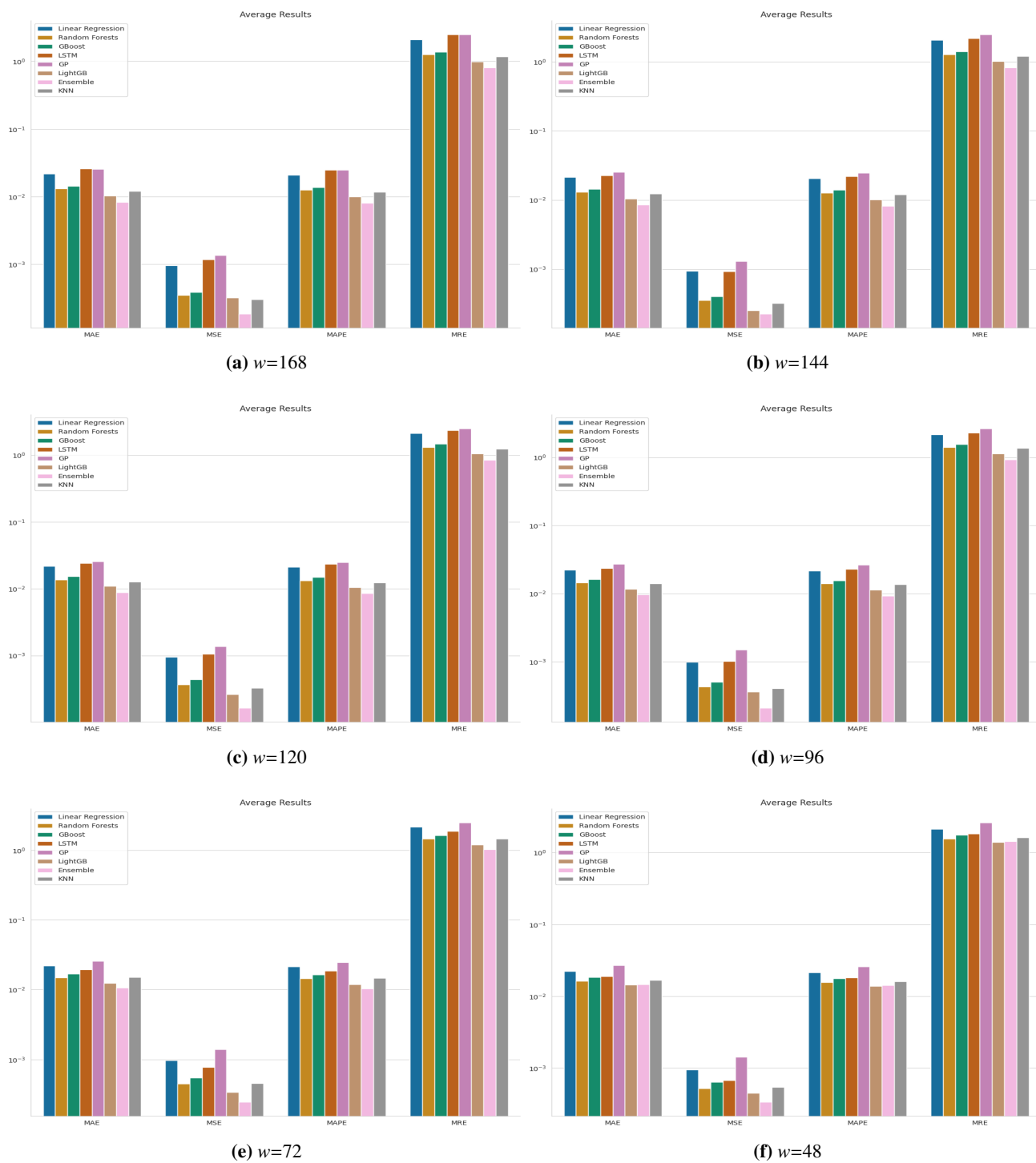
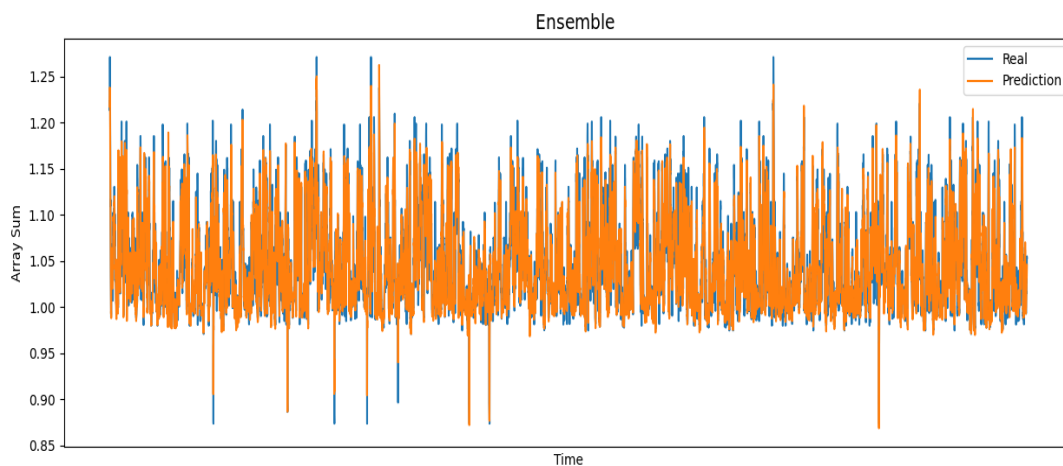
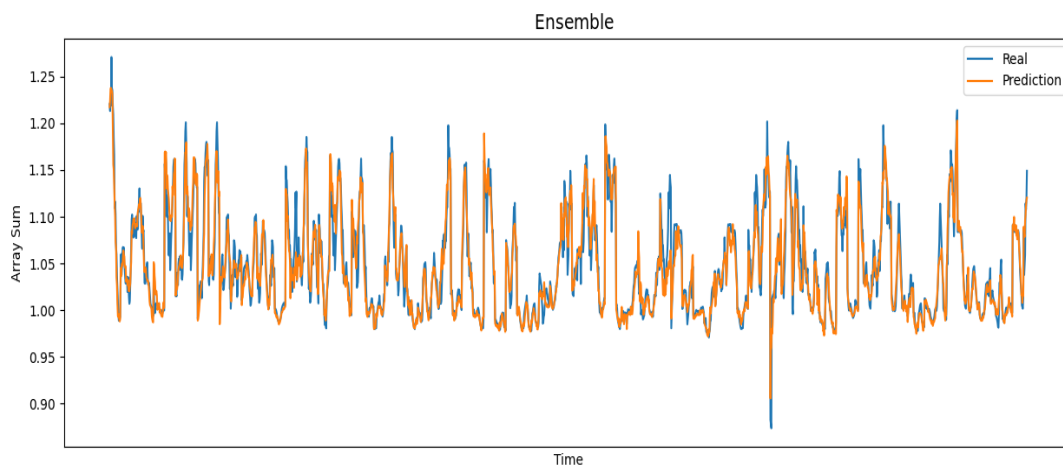


Figure 11. Average results obtained by various algorithms for each value of w used. The measures presented are MAE , MSE , $MAPE$, MSR , respectively.

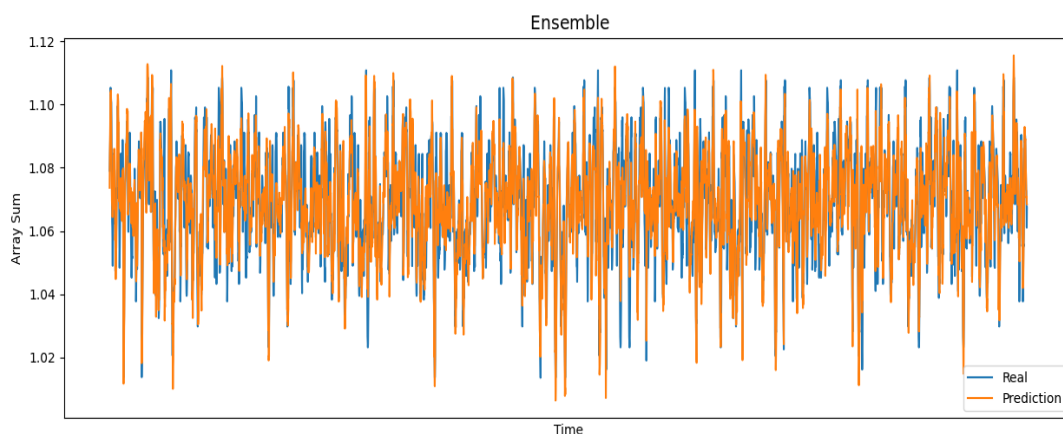


(a) All

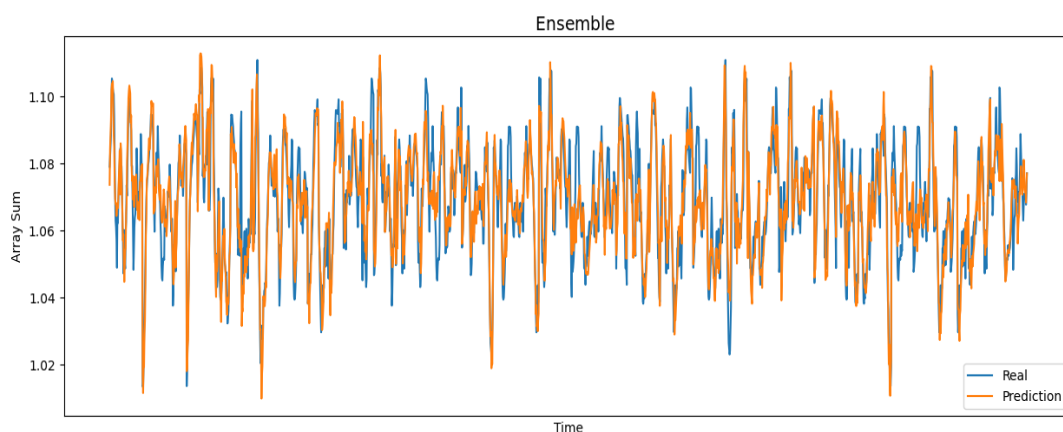


(b) Subset

Figure 12. Examples of best prediction over the real values proposed ensemble model on the animal 1. The predictions were obtained on the test set. The blue line represents the real data and the orange the predicted data.



(a) All



(b) Subset

Figure 13. Examples of worst prediction over the real values for the proposed ensemble model on the animal 18. The predictions are obtained on the test set. The blue line represents the real data and the orange the predicted data.

In order to determine whether if the strategies were vulnerable to overfitting, Table 1 was constructed; it reports the average results obtained on training data by all strategies tested with a historical window size of 168. We can conclude that the strategies did not overfit on the training data since, basically, the best performing methods on the training data were the same as those on the testing data.

Table 1. Average results obtained on training data for $w = 168$.

	LR	RF	GBoost	LSTM	GP	LightGB	Ensemble	KNN
MAE	0.01800	0.00500	0.00885	0.02603	0.02546	0.00038	0.00435	0.00974
MSE	0.00066	0.00005	0.00014	0.00122	0.00132	0.00001	0.00004	0.00020
MAPE	0.01744	0.00484	0.00858	0.02507	0.02466	0.00037	0.00421	0.00942
MRE	1.74409	0.48436	0.85842	2.50666	2.46599	0.03740	0.42098	0.94227

As stated in the introduction, one of the objectives of this study was to verify whether or not a model applied to an animal could be used in order to predict the activity of other animals, as this could minimize computational costs, which may facilitate the incorporation of an activity monitoring system in real livestock farms.

In order to test this hypothesis, we applied the models obtained for animal i on all the other animals j , with $1 \leq j \leq 20, j \neq i$. For these experiments we only selected the model with a window size of 168 since, as observed in Section 4, this was the windows size yielding the best results.

Figure 14 shows the results of this experiment, where the MRE , MAE , MSE and $MAPE$ are reported.

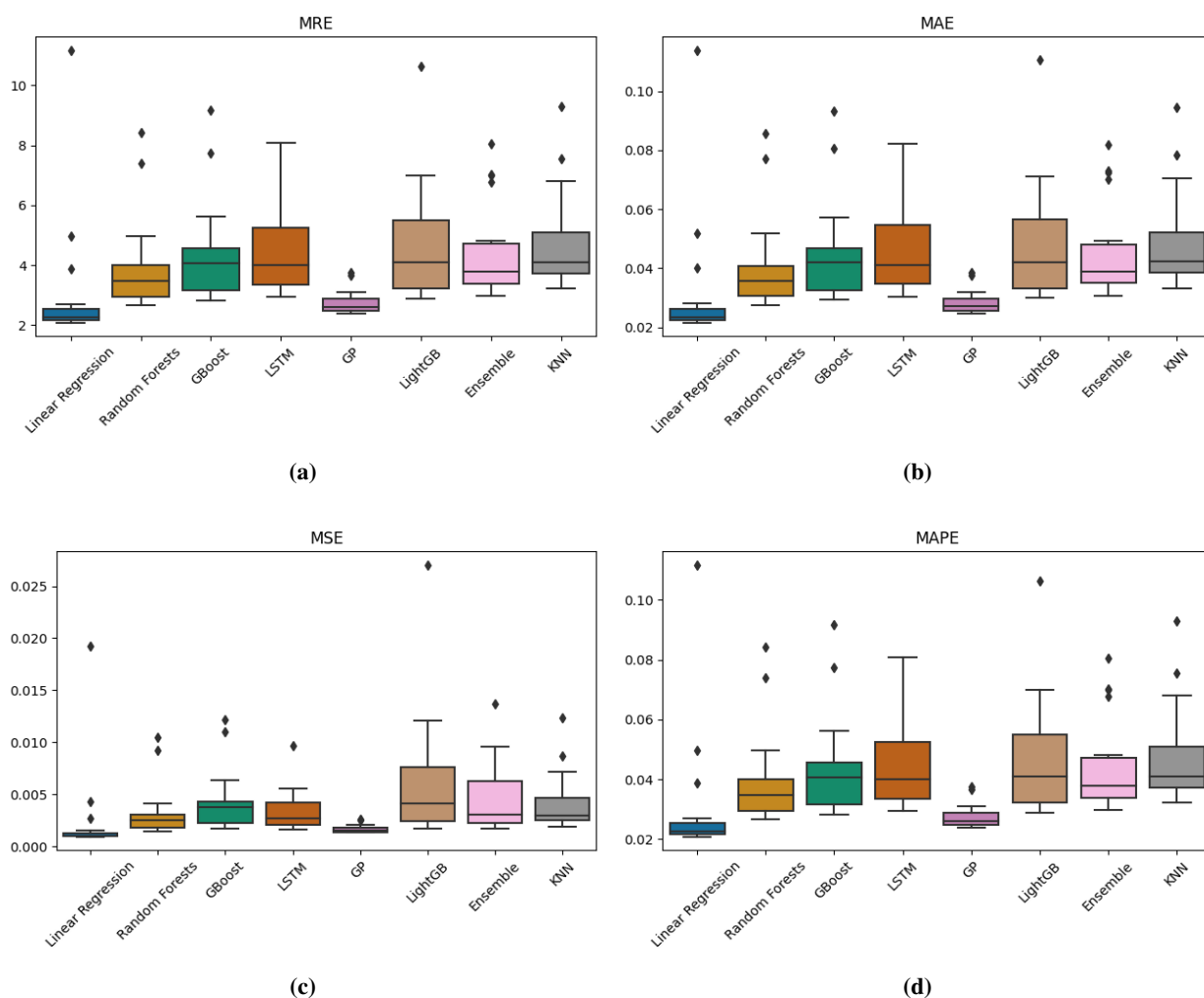


Figure 14. Average results obtained on various algorithms with $w = 168$.

The results obtained were not what we expected, since the strategies that yielded the best results in this case were GP and LR, with GP obtaining slightly better results. The average MRE of the predictions produced by GP was about 2.7, and that of LR was 2.9, while the average MAE was about 0.02 for GP and 0.03 for LR. The MAE can help one to see that these results could justify the use of

a single predictive model if the computational capacity of the farm was reduced; this is because the average value of the time series was about 1.02, and an average absolute error of 0.02 is acceptable. The results obtained by the other methods were similar, with RFs being the strategy that produced more accurate predictions with an average MRE of 3.9 and MAE of 0.04.

The differences between the results of GP and LR, relative to the other methods, were significantly different according to a Wilcoxon two-paired test with a significance level of 5%. The other algorithms yielded results that were not significantly different from each other, but were different for the MSE in the case of RFs and LightGB.

As an example of the predictions obtained when applying a model that was trained on one on a animal to another, in Figure 15, we show four examples. In particular, we show the predictions obtained for animal 10 based on the model obtained on animal 7 by using our ensemble strategy (Figure 15a), GP (Figure 15b), LR (Figure 15c) and LSTM (Figure 15d). Most notable are the predictions of the LSTM algorithm, which were not accurate; furthermore, it was unable to predict peaks of activity.

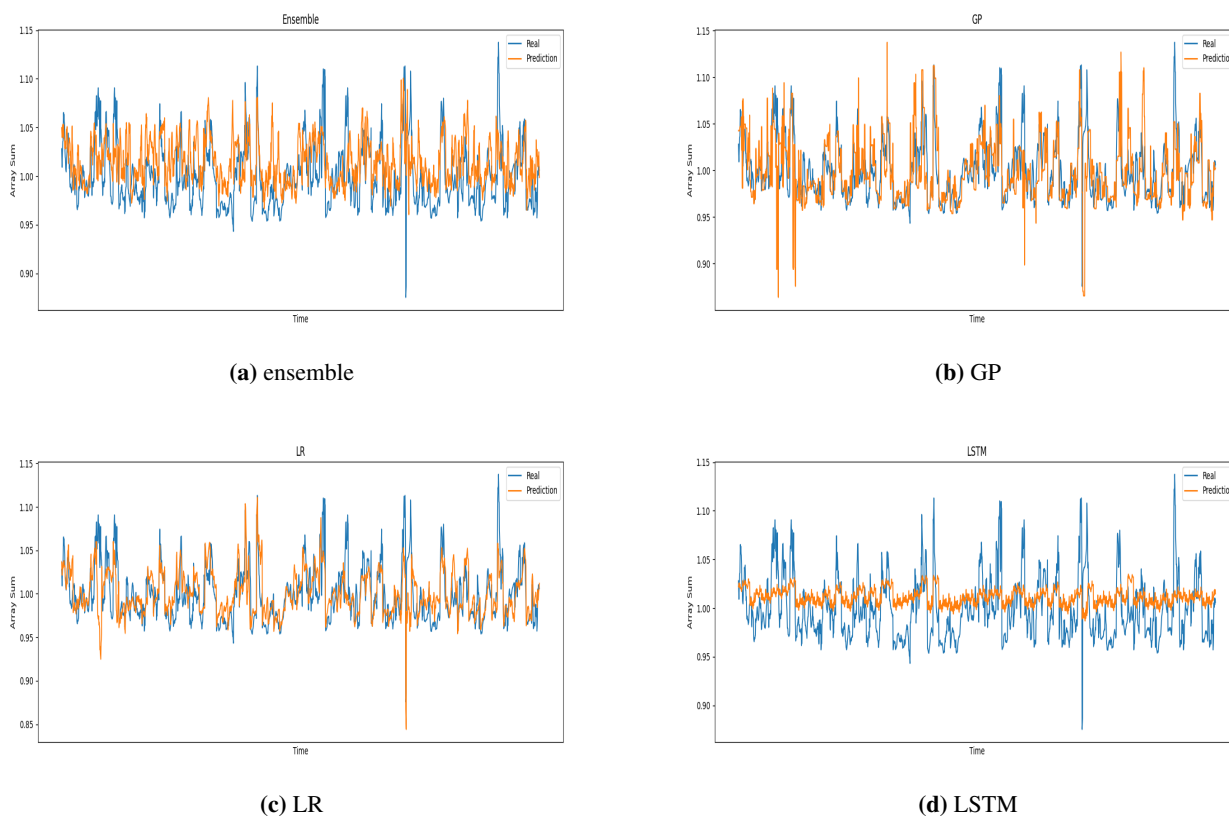


Figure 15. Examples of predictions on a subset of the testing data. The predictions were obtained on the test set. Again the blue line represents the real data and the orange the prediction.

Instead, the predictions obtained by using the other three models, and especially those obtained by using GP and LR, were close to the actual values. Just to give an idea of the performance for this case, the MRE obtained was about 2.8% for the ensemble and the GP strategies, 1.9% for the LR and 3.1%

for LSTM.

5. Conclusions and future work

In this work we have tackled the problem of predicting the activity of Iberian pigs. Having a monitoring system that could precisely predict the activity of animals based on past behaviors would allow one to obtain a normal activity pattern for each animal. If a specific animal does not follow this pattern, an alarm could be risen and preventive actions can be taken; this would allow farmers to increase the welfare and health conditions of the animals by, for example allowing the timely detection of a disease. In this way the illness may be treated in such a way that it does not spread to the other animals.

Toward this aim we tested different ML and statistical methods on data collected by sensors positioned on twenty animals. Unlike other proposals, we have addressed this problem as a time series forecasting problem. In this way, the manual annotation phase of the data is not needed, which allows one to save human resources. Moreover, we have proposed a novel stacking ensemble method based on two layers, where the predictions of three base learners are used by a meta learner in order to produce the final output. Results confirm that our proposed method outperforms the other methods, providing the best forecasting for all animals.

We have also evaluated whether the model acquired on one animal could be used to predict the activity of other animals. When this was done, the results were worse. However, if the computational infrastructure available to a breeder is limited, this strategy could be used, as it produces acceptable results, especially when LR or symbolic regression based on GP is used. As for running times, LSTM requires more time to train. The rest of the methods have comparable running times, in the range of one to five minutes.

As for future developments, we intend to explore the use of metaheuristics in order to find a better configuration of the deep neural networks since, in this work, such LSTM was the worst performing strategy of those tested, and this may be due to a non-optimal configuration of parameters, or the architecture of the network. We also intend to test other deep learning models, such as Seq2Seq and transformers. We also plan to use data obtained on different animals, such as cows, in order to check if our proposed method is valid in this case. Another future development would be to include data regarding time information in addition to the accelerometer data, as it could be that the activity of animals could be related to the time of the day. The development of a ML-based solution to monitor and predict livestock behavior requires ne to address some challenges before it can be deployed. For example, it should include encryption techniques and regulatory frameworks, such as “General Data Protection Regulation” in Europe, to deal with sensitive information such as, for example, livestock health data. In addition, it would also require one to integrate the proposed models into a big data architecture to address the scalability. In addition, we intend to test a similar strategy for the classification of some specific given behaviors, incorporating the manual annotation of behaviors.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors would like to thank the Spanish Ministry of Science and Innovation for the support under the project PID2020-117954RB-C21 and the European Regional Development Fund and Junta de Andalucía under projects PY20-00870 and UPO-138516. This work was a result of a collaboration between Pigchamp Pro Europa, SL and Pablo de Olavide University, under contract 2015/00111/001.

Conflict of interest

All authors declare no conflicts of interest.

The animals involved in this study suffered no harm, since for collecting the data used in this work only the placement of a sensor mounted on a collar was needed.

References

1. N. Zhang, M. Wang, N. Wang, Precision agriculture-a worldwide overview, *Comput. Electron. Agric.*, **36** (2002), 113–132. [https://doi.org/10.1016/S0168-1699\(02\)00096-0](https://doi.org/10.1016/S0168-1699(02)00096-0)
2. R. Gebbers, V. I. Adamchuk, Precision agriculture and food security, *Science*, **327** (2010), 828–831. <https://doi.org/10.1126/science.1183899>
3. H. Auernhammer, Precision farming-the environmental challenge, *Comput. Electron. Agric.*, **30** (2001), 31–43. [https://doi.org/10.1016/S0168-1699\(00\)00153-8](https://doi.org/10.1016/S0168-1699(00)00153-8)
4. S. Wolfert, L. Ge, C. Verdouw, M. J. Bogaardt, Big data in smart farming-a review, *Agric. Syst.*, **153** (2017), 69–80. <https://doi.org/10.1016/j.agry.2017.01.023>
5. M. J. Kim, C. Mo, H. T. Kim, B. K. Cho, S. J. Hong, D. H. Lee, et al., Research and technology trend analysis by big data-based smart livestock technology: a review, *J. Biosyst. Eng.*, **46** (2021), 386–398. <https://doi.org/10.1007/s42853-021-00115-9>
6. A. Prunier, L. Mounier, P. L. Neindre, C. Leterrier, P. Mormède, V. Paulmier, et al., Identifying and monitoring pain in farm animals: a review, *Animal*, **7** (2013), 998–1010. <https://doi.org/10.1017/S1751731112002406>
7. R. Relić, S. Hristov, M. Joksimović-Todorovlć, V. Davidović, J. Bojkovski, Behavior of cattle as an indicator of their health and welfare, *Bull. Univ. Agric. Sci. Vet. Med. Cluj Napoca*, **69** (2012), 1–14. <https://doi.org/10.15835/BUASVMCN-VM:69:1-2:8847>
8. G. Marchesini, D. Mottaran, B. Contiero, E. Schiavon, S. Segato, E. Garbin, et al., Use of rumination and activity data as health status and performance indicators in beef cattle during the early fattening period, *Vet. J.*, **231** (2018), 41–47. <https://doi.org/10.1016/j.tvjl.2017.11.013>
9. D. Weary, J. Huzzey, M. Von Keyserlingk, Board-invited review: using behavior to predict and identify ill health in animals, *J. Anim. Sci.*, **87** (2009), 770–777. <https://doi.org/10.2527/jas.2008-1297>
10. S. G. Matthews, A. L. Miller, T. Plötz, I. Kyriazakis, Automated tracking to measure behavioral changes in pigs for health and welfare monitoring, *Sci. Rep.*, **7** (2017), 17582. <https://doi.org/10.1038/s41598-017-17451-6>

11. P. Martiskainen, M. Järvinen, J. P. Skön, J. Tiirikainen, M. Kolehmainen, J. Mononen, Cow behavior pattern recognition using a three-dimensional accelerometer and support vector machines, *Appl. Anim. Behav. Sci.*, **119** (2009), 32–38. <https://doi.org/10.1016/j.applanim.2009.03.005>
12. A. de Passillé, M. Jensen, N. Chapinal, J. Rushen, Technical note: use of accelerometers to describe gait patterns in dairy calves, *J. Dairy Sci.*, **93** (2010), 3287–3293. <https://doi.org/10.3168/jds.2009-2758>
13. P. L. Greenwood, P. Valencia, L. Overs, D. R. Paull, I. W. Purvis, New ways of measuring intake, efficiency and behavior of grazing livestock, *Anim. Prod. Sci.*, **54** (2014), 1796–1804. <https://doi.org/10.1071/AN14409>
14. B. Koger, A. Deshpande, J. T. Kerby, J. M. Graving, B. R. Costelloe, I. D. Couzin, Quantifying the movement, behavior and environmental context of group-living animals using drones and computer vision, *J. Anim. Ecol.*, **92** (2023), 1357–1371. <https://doi.org/10.1111/1365-2656.13904>
15. R. García, J. Aguilar, M. Toro, A. Pinto, P. Rodríguez, A systematic literature review on the use of machine learning in precision livestock farming, *Comput. Electron. Agric.*, **179** (2020), 105826. <https://doi.org/10.1016/j.compag.2020.105826>
16. G. Mattachini, A. Antler, E. Riva, A. Arbel, G. Provolo, Automated measurement of lying behavior for monitoring the comfort and welfare of lactating dairy cows, *Livest. Sci.*, **158** (2013), 145–150. <https://doi.org/10.1016/j.livsci.2013.10.014>
17. J. Barwick, D. Lamb, R. Dobos, D. Schneider, M. Welch, M. Trotter, Predicting lameness in sheep activity using tri-axial acceleration signals, *Animals*, **8** (2018), 12. <https://doi.org/10.3390/ani8010012>
18. W. Shinada, N. Gakumazawa, S. Koshikawa, T. Ito, T. Fujiwara, M. Takahashi, et al., Precalving behavior in dairy cattle with different calving times, *Anim. Sci. J.*, **94** (2023), e13833. <https://doi.org/10.1111/asj.13833>
19. A. Alameer, I. Kyriazakis, J. Bacardit, Automated recognition of postures and drinking behavior for the detection of compromised health in pigs, *Sci. Rep.*, **10** (2020), 13665. <https://doi.org/10.1038/s41598-020-70688-6>
20. E. S. Fogarty, D. L. Swain, G. M. Cronin, L. E. Moraes, M. Trotter, Behavior classification of extensively grazed sheep using machine learning, *Comput. Electron. Agric.*, **169** (2020) 105175. <https://doi.org/10.1016/j.compag.2019.105175>
21. R. Arablouei, L. Wang, L. Currie, J. Yates, F. A. Alvarenga, G. J. Bishop-Hurley, Animal behavior classification via deep learning on embedded systems, *Comput. Electron. Agric.*, **207** (2023), 107707. <https://doi.org/10.1016/j.compag.2023.107707>
22. M. Borchers, Y. Chang, K. Proudfoot, B. Wadsworth, A. Stone, J. Bewley, Machine-learning-based calving prediction from activity, lying, and ruminating behaviors in dairy cattle, *J. Dairy Sci.*, **100** (2017), 5664–5674. <https://doi.org/10.3168/jds.2016-11526>
23. L. Schmeling, G. Elmamooz, P. T. Hoang, A. Kozar, D. Nicklas, M. Sünkel, et al., Training and validating a machine learning model for the sensor-based monitoring of lying behavior in dairy cows on pasture and in the barn, *Animals*, **11** (2021), 2660. <https://doi.org/10.3390/ani11092660>

24. J. A. Vázquez-Diosdado, V. Paul, K. A. Ellis, D. Coates, R. Loomba, J. Kaler, A combined offline and online algorithm for real-time and long-term classification of sheep behavior: novel approach for precision livestock farming, *Sensors*, **19** (2019), 3201. <https://doi.org/10.3390/s19143201>
25. A. S. Keceli, C. Catal, A. Kaya, B. Tekinerdogan, Development of a recurrent neural networks-based calving prediction model using activity and behavioral data, *Comput. Electron. Agric.*, **170** (2020), 105285. <https://doi.org/10.1016/j.compag.2020.105285>
26. C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: a hybrid approach to alleviating class imbalance, *IEEE Trans. Syst. Man Cybern.*, **40** (2010), 185–197. <https://doi.org/10.1109/TSMCA.2009.2029559>
27. C. Carslake, J. A. Vázquez-Diosdado, J. Kaler, Machine learning algorithms to classify and quantify multiple behaviors in dairy calves using a sensor: Moving beyond classification in precision livestock, *Sensors*, **21** (2020), 88. <https://doi.org/10.3390/s21010088>
28. M. L. Williams, W. P. James, M. T. Rose, Variable segmentation and ensemble classifiers for predicting dairy cow behavior, *Biosyst. Eng.*, **178** (2019), 156–167. <https://doi.org/10.1016/j.biosystemseng.2018.11.011>
29. S. Hu, R. Arablouei, G. J. Bishop-Hurley, A. Reverter, A. Ingham, Predicting bite rate of grazing cattle from accelerometry data via semi-supervised regression, *Smart Agric. Technol.*, **5** (2023), 100256. <https://doi.org/10.1016/j.atech.2023.100256>
30. F. Abbona, L. Vanneschi, M. Bona, M. Giacobini, Towards modelling beef cattle management with genetic programming, *Livest. Sci.*, **241** (2020), 104205. <https://doi.org/10.1016/j.livsci.2020.104205>
31. B. Ji, T. Banhazi, C. J. Phillips, C. Wang, B. Li, A machine learning framework to predict the next month's daily milk yield, milk composition and milking frequency for cows in a robotic dairy farm, *Biosyst. Eng.*, **216** (2022), 186–197. <https://doi.org/10.1016/j.biosystemseng.2022.02.013>
32. A. da Silva Santos, V. W. C. de Medeiros, G. E. Gonçalves, Monitoring and classification of cattle behavior: a survey, *Smart Agric. Technol.*, **3** (2023), 100091. <https://doi.org/10.1016/j.atech.2022.100091>
33. H. Suparwito, K. W. Wong, H. Xie, S. Rai, D. Thomas, A hierarchical classification method used to classify livestock behavior from sensor data, *International Conference on Multi-disciplinary Trends in Artificial Intelligence*, 2019, 204–215. https://doi.org/10.1007/978-3-030-33709-4_18
34. D. S. Rodriguez-Baena, F. A. Gomez-Vela, M. García-Torres, F. Divina, C. D. Barranco, N. Daz-Diaz, et al., Identifying livestock behavior patterns based on accelerometer dataset, *J. Comput. Sci.*, **41** (2020), 101076. <https://doi.org/10.1016/j.jocs.2020.101076>
35. A. A. Rayas-Amor, E. Morales-Almaráz, G. Licona-Velázquez, R. Vieyra-Alberto, A. García-Martínez, C. G. Martínez-García, et al., Triaxial accelerometers for recording grazing and ruminating time in dairy cows: an alternative to visual observations, *J. Vet. Behav.*, **20** (2017), 102–108. <https://doi.org/10.1016/j.jveb.2017.04.003>
36. M. Lepot, J. B. Aubin, F. H. Clemens, Interpolation in time series: an introductory overview of existing methods, their performance criteria and uncertainty assessment, *Water*, **9** (2017), 796. <https://doi.org/10.3390/w9100796>

37. H. Teichgraber, A. R. Brandt, Time-series aggregation for the optimization of energy systems: goals, challenges, approaches, and opportunities, *Renew. Sustain. Energy Rev.*, **157** (2022), 111984. <https://doi.org/10.1016/j.rser.2021.111984>
38. D. Leite, I. Škrjanc, Ensemble of evolving optimal granular experts, owa aggregation, and time series prediction, *Inf. Sci.*, **504** (2019), 95–112. <https://doi.org/10.1016/j.ins.2019.07.053>
39. S. L. Wickramasuriya, G. Athanasopoulos, R. J. Hyndman, Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization, *J. Amer. Stat. Assoc.*, **114** (2019), 804–819. <https://doi.org/10.1080/01621459.2018.1448825>
40. Y. W. Cheung, K. S. Lai, Lag order and critical values of the augmented dickey-fuller test, *J. Bus. Econ. Stat.*, **13** (1995), 277–280. <https://doi.org/10.1080/07350015.1995.10524601>
41. J. F. Torres, A. M. Fernández, A. Troncoso, F. Martínez-Álvarez, Deep learning-based approach for time series forecasting with application to electricity load, *International Work-Conference on the Interplay Between Natural and Artificial Computation*, 2017, 203–212. https://doi.org/10.1007/978-3-319-94120-2_12
42. F. Divina, A. Gilson, F. Gómez-Vela, M. García Torres, J. F. Torres, Stacking ensemble learning for short-term electricity consumption forecasting, *Energies*, **11** (2018), 949. <https://doi.org/10.3390/en11040949>
43. J. Neter, M. H. Kutner, C. J. Nachtsheim, W. Wasserman, *Applied linear statistical models*, Irwin, 1996.
44. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: machine learning in python, *J. Mach. Learn. Res.*, **12** (2011), 2825–2830. <https://doi.org/10.5555/1953048.2078195>
45. W. Härdle, O. Linton, Applied nonparametric methods, *Handb. Econometrics*, **4** (1994), 2295–2339.
46. J. L. Bentley, Multidimensional binary search trees used for associative searching, *Commun. ACM*, **18** (1975), 509–517. <https://doi.org/10.1145/361002.361007>
47. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
48. A. Robinson, F. Fallside, *The utility driven dynamic error propagation network*, Department of Engineering Cambridge, University of Cambridge, 1987.
49. F. Chollet, Keras-team/Keras, 2015. Available from: <https://github.com/fchollet/keras>.
50. T. K. Ho, Random decision forests, *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995. <https://doi.org/10.1109/ICDAR.1995.598994>
51. D. A. Augusto, H. J. Barbosa, Symbolic regression via genetic programming, *Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks*, 2000, 173–178. <https://doi.org/10.1109/SBRN.2000.889734>
52. T. Stephens, Gplearn-genetic programming in python, 1859. Available from: <https://gplearn.readthedocs.io/en/stable/index.html>.
53. J. H. Friedman, Greedy function approximation: a gradient boosting machine, *Ann. Stat.*, **29** (2001), 1189–1232. <https://doi.org/10.1214/aos/1013203451>

54. Y. Shi, J. Li, Z. Li, Gradient boosting with piece-wise linear regression trees, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, 3432–3438. <https://doi.org/10.5555/3367471.3367518>
55. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, et al., Lightgbm: a highly efficient gradient boosting decision tree, *Adv. Neural Inf. Process. Syst.*, **30** (2017), 3149–3157. <https://doi.org/10.5555/3367471.3367518>
56. Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, In: P. Vitányi, *Computational learning theory*, Springer Berlin Heidelberg, 1995, 23–37. https://doi.org/10.1007/3-540-59119-2_166
57. T. G. Dietterich, Ensemble methods in machine learning, In: J. Kittler, F. Roli, *Multiple classifier systems*, Springer-Verlag, 2000, 1–15. https://doi.org/10.1007/3-540-45014-9_1



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)