Data Driven Broiler Weight Forecasting using Dynamic Neural Network Models

Johansen, Simon Vestergaard; Bendtsen, Jan Dimon; Riisgaard-Jensen, Martin; Mogensen, Jesper

Published in:
IFAC-PapersOnLine

DOI (link to publication from Publisher):
10.1016/j.ifacol.2017.08.1073

Publication date:
2017

Document Version
Accepted author manuscript, peer reviewed version

Data Driven Broiler Weight Forecasting using Dynamic Neural Network Models

Simon V. Johansen[1,2] **Jan D. Bendtsen**[2] **Martin R.-Jensen**[1] **Jesper Mogensen**[1]

[1] *SKOV A/S, Glyngøre, Denmark*
[2] *Department of Electronic Systems, Aalborg University, Denmark*
Email: sjo@skov.dk, dimon@es.aau.dk, mrj@skov.dk and jmo@skov.dk

Abstract: In this article, the dynamic influence of environmental broiler house conditions and broiler growth is investigated. Dynamic neural network forecasting models have been trained on farm-scale broiler batch production data from 12 batches from the same house. The model forecasts future broiler weight and uses environmental conditions such as heating, ventilation, and temperature along with broiler behavior such as feed and water consumption. Training data and forecasting data is analyzed to explain when the model might fail at generalizing. We present ensemble broiler weight forecasts to day 7, 14, 21, 28 and 34 from all preceding days and provide our interpretation of the results. Results indicate that the dynamic interconnection between environmental conditions and broiler growth can be captured by the model. Furthermore, we found that a comparable forecast can be obtained by using input data from the previous batch as a substitute for future input data.

Keywords: Artificial neural nets in agriculture

1. INTRODUCTION

Due to the growing middle class in many developing countries, the increase in meat consumption is predicted to continue rising (OECD, 2015). The highest growth in meat production is foreseen in poultry meat, of which we expect broiler meat to represent the majority. The reason being that broiler production gives the highest yield per feed unit among land animals, making chicken meat a relatively inexpensive source of animal protein. The poultry industry on a world scale is predicted to steadily increase, from an average of 107.6 billion kg of meat in 2012-2014 to 134.8 in 2024 (OECD, 2015, pp. 136) – a significant predicted increase of 24%.

Available scientific literature on dynamic broiler models focuses mainly on active broiler weight control by regulating feed uptake and composition, which traditionally favors simplistic models (Wathes et al., 2008), (Aerts et al., 2003). Feed is generally considered the biggest expense in broiler production, and correct climate control is known to yield superior feed utilization for broiler growth. However, no scientific literature has been found that studies the complex dynamic interconnection between broiler weight and broiler house environmental conditions, and will thus be the subject of this work. Such a model has the potential of allowing active broiler weight control by regulating the broiler environment, which is nonexistent in industry today. Especially considering that most broiler houses use ad libitum feeding regimes, which excludes the regulation of feed uptake.

Traditionally, empirically motivated nonlinear growth curve models have been used to determine evolution of broiler weight. It has been extensively studied in scientific literature such as (Aggrey, 2002), (Ahmadi and Mottaghitalab, 2007) and (Elerolu et al., 2014). A growth curve in this context is a static curve fitted to old broiler weight data. It has a fixed structure with few parameters that offers biologically intuitive interpretations. Common models include the Richerds Model and Gompertz-Laird Model that are described by 4 and 3 parameters respectively, where the parameters have biologically intuitive interpretations – such as time and size of maximum growth rate (Aggrey, 2002). In (Lopes et al., 2008), the relation between broiler house environment and production performance was investigated using a neural network. However, it is not clear how to extract temporal performance (i.e. growth prediction) from this type of model.

Dynamic broiler growth models are an extension of growth curves and has primarily been developed for control synthesis in scientific literature. In (Aerts et al., 2003) a time-variant online parameter estimation of a dynamic model was successfully applied to predict future weight up to 7 days without a priori information. The model use feed intake as input and weight as an output and was used for model predictive control (MPC) in (Cangar et al., 2007). This control algorithm was tested in a laboratory setting with a stocking density of 5.3 birds/m^2 and 20 birds/m^2, the later to emulate farm scale density. The mean relative weight control error was 2.7% and 7.3% for the low and high-density experiments respectively – suggesting that farm scale broiler production is harder to both predict and control.

A similar result was obtained in (Demmers et al., 2010), where a small differential recurrent neural network was used to model the feed quantity and control the broiler weight using nonlinear MPC. In (Stacey et al., 2004) a

dynamic broiler weight model was developed and used information about feed uptake and composition of two feed types with known nutritional value. The model was successfully used to control broiler weight, it was tested on farm scale with 30,000-40,000 broilers per house and achieved results comparable to that of a stockman. However, these studies do not take environmental conditions into account.

In this paper we use neural network class models for forecasting, which is a soft computing technique – a group of inexact methods that is capable of dealing with uncertainty. More specifically, we will be using dynamic neural networks, which has been successfully applied to model complex biological processes. Recent applications include algae growth prediction in a laboratory setting (Wang et al., 2015), prediction of bioethanol production in a bioreactor (Grahovac et al., 2016), bioreactor prediction (Nair et al., 2016), yeast fermentation modeling in a bioreactor (Nasimi and Irani, 2014) and state estimation in a continuous bio reactor (Hernandez et al., 2013).

In the present effort, we present a data-driven dynamic neural network broiler batch forecasting model. In particular, we show that dynamic interaction between environmental conditions and chicken behavior in ad libitum feed broilers is present and can be captured by a model. We do this in a data driven framework on real farm scale production data from a state of the art broiler house.

We have to emphasize that state of art broiler production is mainly empirically driven. For this reason, we use a broiler expert employed by SKOV A/S to interpret both the input variables and the results presented in this work.

The remainder of the article is structured as follows. In section 2 the model, training and validation method is described. In section 3 the data collection and analysis, model configuration and forecasting results are described. In section 4.1 we discuss the obtained experimental results followed by concluding remarks in section 5.

2. MODEL, TRAINING AND VALIDATION METHOD

2.1 Model

Since the multilayer perceptron (MLP) model and its variations have been extensively researched in scientific literature we will only give a brief introduction and describe how we apply it to our specific problem. For a thorough introduction to the subject we redirect the reader to (Du and Swamy, 2014) and (Haykin, 1994).

The particular type of Dynamic Neural Network (DNN) model we use in this work can be classified as a discrete nonlinear ARX model of the form

$$\hat{y}\left[k+1 \mid \mathcal{W}\right] = \mathcal{N}\left(\hat{Y}[k], U[k+1] \mid \mathcal{W}\right) \tag{1}$$

with

$$U[k+1] = \left[u[k-n_1+1]^T \ \cdots \ u[k-n_i+1]^T\right]^T \text{ and} \tag{2a}$$

$$\hat{Y}[k] = \left[\hat{y}\left[k-m_1 \mid \mathcal{W}\right]^T \ \cdots \ \hat{y}\left[k-m_j \mid \mathcal{W}\right]\right]^T, \tag{2b}$$

where $\mathcal{N}$ is a MLP model, $U[k]$ is delayed values of the input vector $u[k] \in \mathbb{R}^N$ corresponding to the i elements of $n = \{n_1, \cdots, n_i\}$, $\hat{Y}[k]$ is delayed values of the previous output vector $\hat{y}[k] \in \mathbb{R}^M$ corresponding to the j elements of $m = \{m_1, \cdots, m_j\}$ and $\mathcal{W}$ is an abstract representation of all the model weights in a vector. This particular structure has been adopted to accommodate potentially long propagation delays, while still aiming to keep the number of weights relatively low.

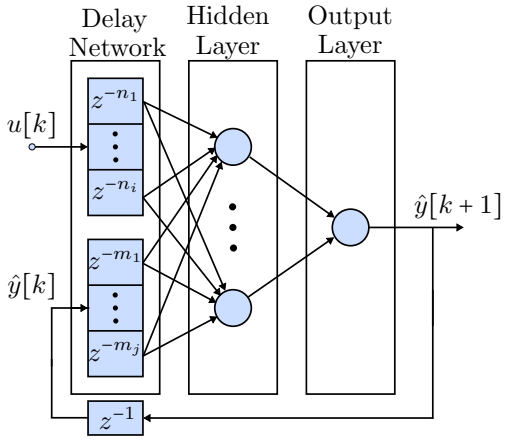


Fig. 1. Visual representation of (3) with one input and one output for simplicity. Neurons are represented with blue circles, which contain an activation function and bias each. Note that all incoming signals to a neuron are multiplied by a weight. The operator z^{-a} produce a delay of a samples.

The DNN model $\mathcal{N}$ is selected with one hidden layer with hyperbolic tangent activation function in the hidden layer and linear activation function in the output layer. It can be shown that $\mathcal{N}$ is an universal function approximator, meaning that it has the capacity for approximating any system to any accuracy (Du and Swamy, 2014, chap. 4.2). In matrix-vector representation, (1) equals

$$\hat{y}\left[k+1 \mid \mathcal{W}\right] = W^{\mathrm{o}} \tanh\left(\mathcal{X}\right) + \theta^{\mathrm{o}} \text{ with} \tag{3}$$

$$\mathcal{X} = \sum_{a=1}^{j} W^{\mathrm{h}}_{y,a} \hat{y}\left[k-m_a+1 \mid \mathcal{W}\right] + \sum_{b=1}^{i} W^{\mathrm{h}}_{u,b} u\left[k-n_b+1\right] + \theta^{\mathrm{h}},$$

where W^{o} is the output weights, $W^{\mathrm{h}}_{y,a}$ is the delayed output weights, W^{h}_{b} is the delayed input weights, θ^{h} is the hidden layer bias and θ^{o} is the output bias. A visual representation of (3) is depicted on Fig. 1.

2.2 Model Validation

We partition the available batches into three categories, namely training, prediction, and evaluation. In ascending order from the latest batch is the evaluation batch, prediction batch, and the remaining batches are training batches. The training batches are used to find the network weights $\mathcal{W}$ through training. The latest training batch is used in an early stopping setting to avoid over-fitting by selecting the model that generalizes best to this batch.

We generate 64 models with randomly initialized weights and train each one individually on the same training data. We then use the ensemble mean and standard deviation of all the individual model forecast runs to represent the "true" model forecasting output in open loop from day 7 to 34. We will investigate the impact of using old input data from the prediction batch to represent the best guess for future input $U[k]$ in the evaluation batch. To

demonstrate the model's ability to forecast broiler weight throughout the batch, we forecast the weight from all preceding samples to day 7, 14, 21, 28 and 34 – denoted *weight on day forecasting*.

2.3 Model Training

The dynamic neural network is trained by minimizing the cost function

$$\underbrace{\frac{1}{B}\sum_{b=1}^{B}\sum_{k=\kappa}^{K_b}\frac{\|y\left[k\mid b\right]-\hat{y}\left[k\mid\kappa,b,\mathcal{W}\right]\|_2^2}{N_y\left(K_b-\kappa+1\right)}}_{E_D}+\bar{\alpha}\underbrace{\|\mathcal{W}\|_2^2}_{E_\mathcal{W}}, \quad (4)$$

where B is the number of training batches, κ is the index of the first output, and K_b is the number of samples in batch b. The target output at sample k given batch b is denoted $y\left[k\mid b\right]$, while the predicted network output at sample k, initialized at sample κ, using future input values from batch b with model weights $\mathcal{W}$, is denoted $\hat{y}\left[k\mid\kappa,b,\mathcal{W}\right]$. In this context, initialization refers to the initial values of (2). $\|\cdot\|_2$ is the vector 2-norm.

Both the inputs and outputs are normalized to a mean of 0 and standard deviation of 1 during training. The network weights $\mathcal{W}$ are initialized using the Nguyen-Widrow algorithm as explained in (Nguyen and Widrow, 1990).

The last term is a scalar regularization term punishing the size of the $N_\mathcal{W}$ system weights $\mathcal{W}$, where $\bar{\alpha}$ is the regularization weight. The regularization weight $\bar{\alpha}$ is determined iteratively through Bayesian Regulation as described in (Burden and Winkler, 2008). We normalize the regularization weight to make the cost functions comparable between runs, and is calculated according to $\bar{\alpha}=\alpha/\beta$ with:

$$\alpha=\frac{\gamma}{2E_\mathcal{W}} \quad \beta=\frac{N_D-\gamma}{2E_D} \quad \gamma=N_\mathcal{W}-\alpha\,\mathrm{trace}\left(G^{-1}\right) \quad (5)$$

Where $E_\mathcal{W}$ and E_D originate from (4) and γ, $0\leq\gamma\leq N_\mathcal{W}$, is a measure of how many of the $N_\mathcal{W}$ parameters are used. Lastly, G is the hessian matrix of the joint cost $\beta E_D+\alpha E_W$ according to (6a), where I is the identity matrix of appropriate dimensions. It is partially approximated through the jacobian J_D of E_D in (6b), similar to the Levenberg Marquard algorithm. The scalar constants α and β are iteratively updated between training epochs, with initial conditions $\alpha=0$ and $\beta=1$.

$$G=\frac{d^2\left(\beta E_D+\alpha E_\mathcal{W}\right)}{d\mathcal{W}\,d\mathcal{W}^T}=\frac{d^2\left(\beta E_D\right)}{d\mathcal{W}\,d\mathcal{W}^T}+2\alpha I \quad (6a)$$

$$\approx J_D^T J_D+2\alpha I \text{ with } J_D=\beta\frac{dE_D}{d\mathcal{W}^T} \quad (6b)$$

The cost function (4) is minimized using the Levenberg-Marquardt optimization algorithm by means of the Ceres Solver library (Sameer Agarwal et al., 2015). It is an algorithm that efficiently solves large scale least square problems.

3. EXPERIMENT

This work is based on data gathered from a ≈20 year old state of the art broiler house located in the northern Denmark. We use data from 12 batches, collected over a period of 19 months. Each batch contains roughly 40,000 broilers.

3.1 Data Analysis

The inputs to the model consist of the available environmental variables: the measured temperature, humidity and CO_2, and the references of the light intensity, ventilation level, and heating level. The model outputs consist of the available broiler behavior indicators: the measured weight along with feed and water consumption per bird.

We intentionally distinguish between reference, demand and measured variables. Reference variables are independent, demand variables are determined by a deterministic entity like a controller outside the model and measured variables are dependent on the model process.

To elaborate, humidity, temperature, and CO_2 measurements are measured during closed loop operation and is controlled through both the ventilation and heating demands. For this reason, these variables are not independent, which has been known to cause problems in closed loop identification (Rajamani Doraiswami and Stevenson, 2014). We do not consider this an issue as all closed loop controlled variables are considered inputs to the model and therefore no deterministic output feedback is present. Keep in mind that reference inputs for controlled variables is determined by the broiler farmer, whom we consider a non-deterministic entity, making it an open loop identification problem with restricted input space. Hence, it still remains to be confirmed if enough excitation of the correlated input variables are present to accurately represent the desired operational input space.

In order to understand the results we give our interpretation of the input and output variables, which are depicted on Fig. 2. For this reason, we emphasize the difference between what the model "knows" through training data, and the "unknown" validation and prediction data, which can help explain if the model has difficulties at generalizing. We also point out known sources of error that can influence the model's forecasting ability.

Input variables First we give an example of the aforementioned correlated input variables, which is most clearly illustrated on the evaluation batch. Consequently, the ventilation demand is lower and both heating demand and measured CO_2 is higher throughout the batch. As this can be considered significantly different from the training data this will test the models ability to generalize.

As an extension of this, we note that the ventilation demand is negatively correlated with both the measured CO_2 and positively correlated with the measured temperature. Furthermore, the measured humidity appears correlated with the ventilation demand, but this is in fact caused by a larger physical organic mass in the broiler house as the broilers grow throughout the batch.

We note a low degree of variation in temperature across all available batches, which we attribute to it being a controlled variable with tighter margins than e.g. CO_2. The prediction batch has a lower than average temperature near the batch end, and for this reason we expect to see deviations if the model fails at generalizing. Consequently the model is only valid for tightly temperature controlled broiler houses. Lastly, we note that the light intensity ref-

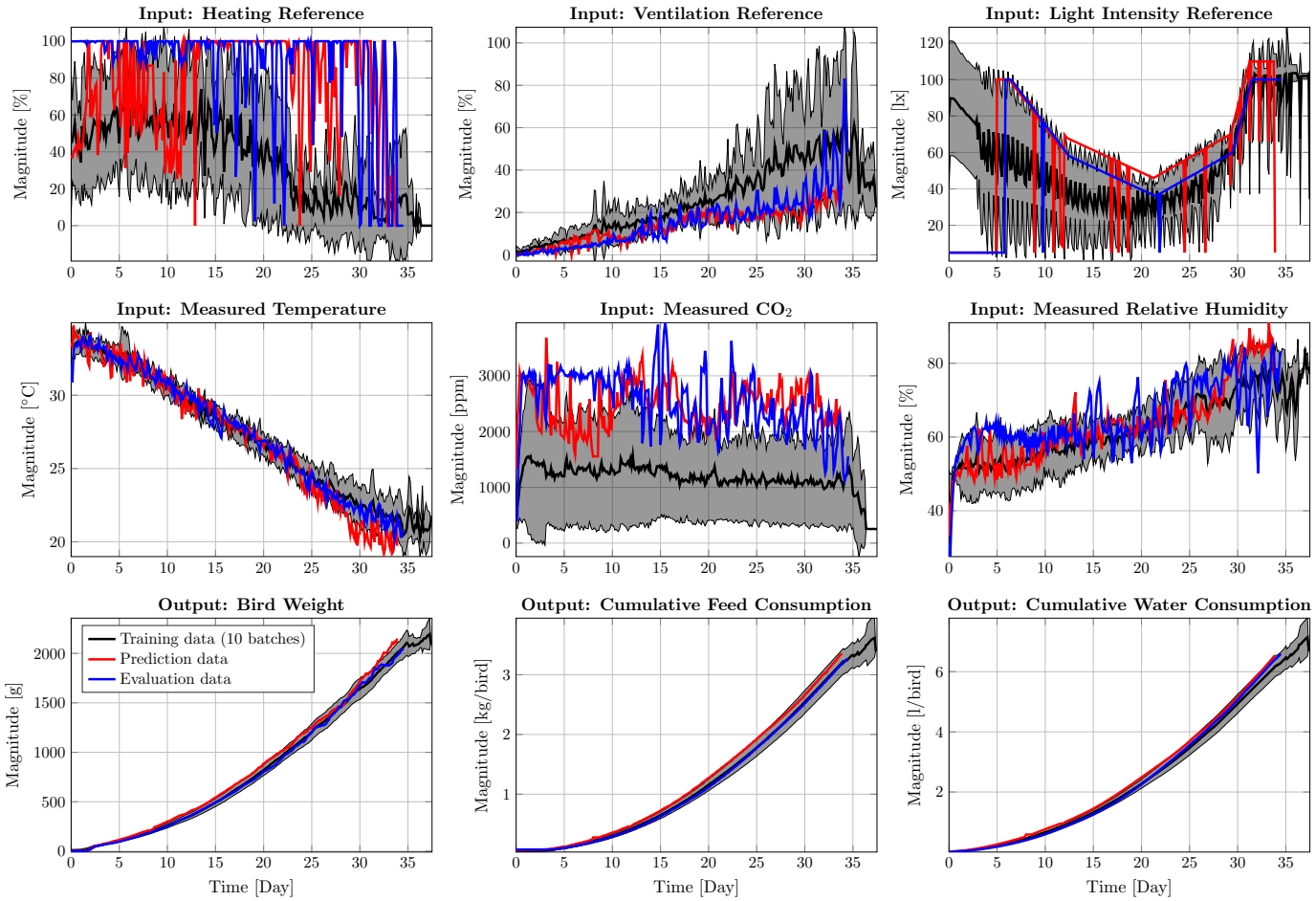


Fig. 2. The inputs and output data from the available batches. The black shaded area denotes one standard deviation from the mean value.

erence is quite spiky, which is caused by manual overriding by the farmer.

Output variables First off, broiler weight is highly positively correlated with both the cumulative feed and water consumption, which makes sense as the two are necessary for both survival and growth of the broilers. We note that bird weight has been reported by our customers to be highly predictable from feed and water consumption, which can be illustrated by the fact that the three have similar variations and behavior. Furthermore, it is unknown if and when the farmer has changed feed type for any of the batches.

It is a common problem within broiler production that the measured weight is negatively biased from day 15 and onwards. We believe that this among others is due to the broilers outgrowing their skeletal capacity, resulting in a reluctance for the heavier broilers to jump up on the weight pads placed in the broiler house. This has been corrected by multiplying a manually configured and time varying "behavior" constant with the measured weight. We only have the corrected weight which naturally leads to some uncertainty. The projected slaughterhouse weight on day 34 are 2138g and 2091g for the prediction and evaluation batch respectively – a difference of 47g. The weighing pad measured a weight of 2007g and 2140g on this day, resulting in a difference of -84g and 2g compared to the slaughterhouse weight, respectively.

3.2 Model Configuration and Results

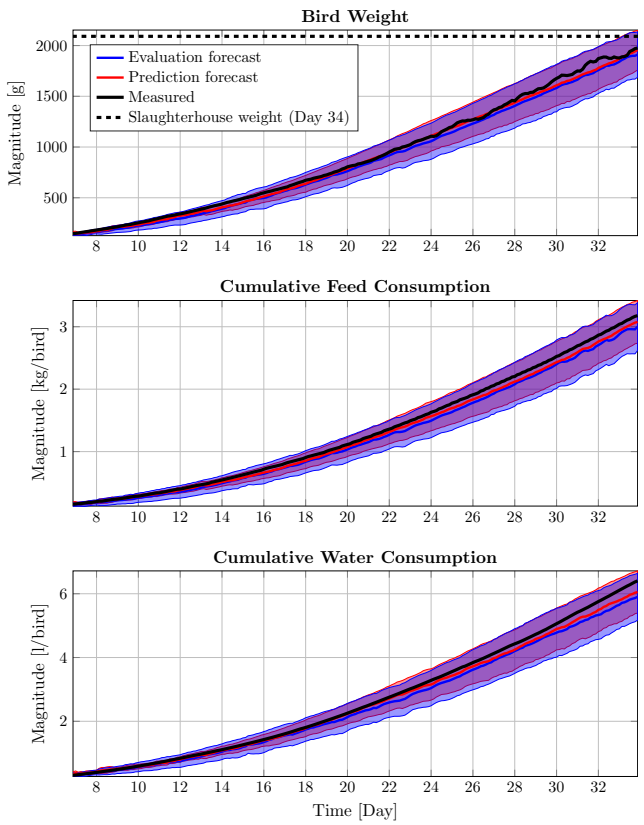
For this work we use $h = 7$ hidden neurons with input delays of $n \in \{0, 1, 2\}$ samples and output delays of $m \in \{0, 1, 9\}$ samples in (1), resulting in a model with 220 free parameters. We find that the output delay of ≈ 1 day has a stabilizing effect on the forecasting. We have good experience with a sampling interval of $T_s = 3$ hours and first output training sample $\kappa = 10$ (day 1.25) in (4).

On Fig. 3a a forecasting example from day 7 to 34 is presented, while weight on day forecasting on day 7, 14, 21, 28 and 34 from all preceding samples that is depicted on Fig. 3b.

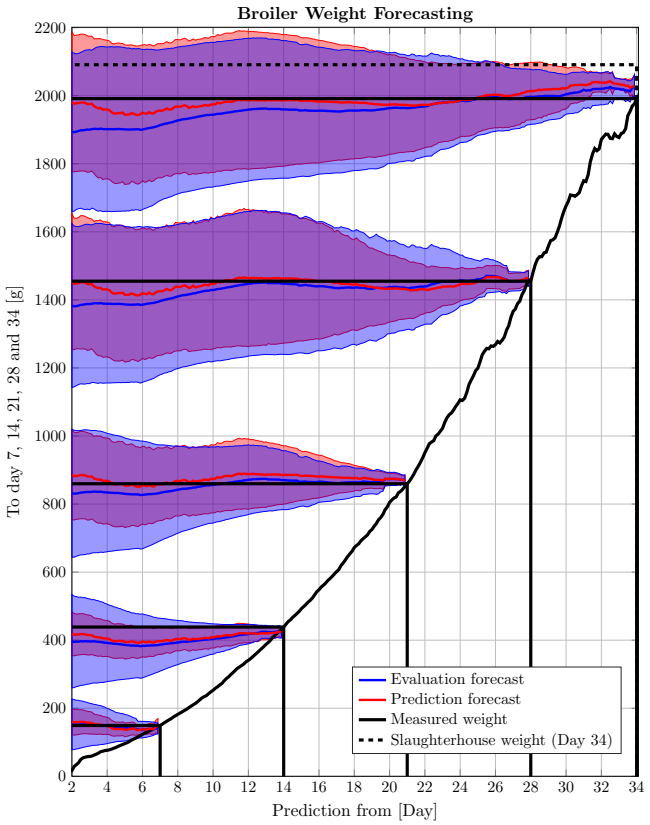
4. DISCUSSION

4.1 Forecasting example

In this subsection we discuss Fig. 3a. We see that both prediction and evaluation forecasts produce good overall forecasting capabilities, in the sense that it tracks the overall target weight throughout the batch. Comparing the use of prediction and evaluation forecast, we see a slightly higher mean weight of 45g on day 34 for the



(a) Example of forecasting from day 7 to 34.



(b) Weight on day forecasting to day 7, 14, 21, 28 and 34.

Fig. 3. The precision of the individual models is illustrated through both the ensemble mean and standard deviation. The solid blue evaluation forecasts are the ensemble mean generated with models initialized with evaluation batch data and future inputs from the evaluation batch. The solid red prediction forecasts are generated with the same procedure, but with future inputs from the prediction batch. To be specific, the blue traces uses future input data from the current batch, where the red traces uses future input data from a previous batch. The blue and red shaded areas are the ensemble standard deviation.

prediction forecast. This is close to the slaughterhouse weight difference of 47g on this day, despite the weight being ≈100g lower (≈9% of the slaughter weight). This indicates that the model captures at least some of the dynamic interconnection between the two forecasts, as the climate conditions are the only difference.

The model uses future prediction of the cumulative feed and water consumption to support the weight forecast. We see that the model underestimates both the feed and water consumption at day 34 by ≈ 6% and ≈ 8% respectively for the evaluation forecast, which can explain part of the ≈9% underestimated weight. It supports that broiler weight can be inferred from the cumulative feed and water consumption, as these figures are quite close to the underestimated slaughter weight.

4.2 Weight on day forecasting

In this subsection we discuss Fig. 3b. Again, we see that both prediction and evaluation weight on day forecasts produce good overall forecasting capabilities, in the sense that the forecast is close to the measured weight for the target days.

Common for the weight on day forecasts is that the ensemble standard deviation tends to decrease over time – which is expected as the forecasting horizon diminish. Furthermore, keep in mind that differences between the two forecasts are caused by different future inputs, as they are initialized with the same data – as seen around day 13, where the humidity and CO_2 is exited in the evaluation batch. If both forecasts react, it is caused by the model initialization with the evaluation data – as observed around day 31, where the weight measurement plateaus.

The weight on day 34 prediction forecast tends to be higher than the prediction forecast, which is also the case for the two batches. This difference diminishes gradually from day 7 and onwards, which the model attributes to the overall climate differences in the respective batches. We see that the measured weight on day 34 differs significantly from the projected slaughterhouse weight. The model appears to be able to pick this up just before the broiler weight plateaus at around day 31. This suggests that the model has learned the weighing pad behavior from the training data, and not the "actual" broiler weight.

Only the ensemble standard deviation around day 14 prediction is consistently deviating from the weight measurement, where the others tend to increase throughout the batch. Furthermore, models simulated from around day 8-15 appear to increase the forecasting ensemble standard deviation for both models, indicating that an unfamiliar

combination of weight, feed and water consumption is observed in the evaluation data. Alternatively, both the prediction and evaluation batch contains different CO_2 from the training data for this time interval. Regardless, it indicates that the model is having difficulties at generalizing correctly for this interval in particular.

Comparing the evaluation and prediction forecast, we see that they resemble each other well in terms of ensemble standard deviation as the shaded area is similar for the two forecasts, indicating that input data from the previous batch can be used to represent the ensemble standard deviation of the current batch despite different performance.

5. CONCLUDING REMARKS

In this present work DNN forecasting models have been trained on farm scale broiler batch production data from 12 batches. We have produced and interpreted a forecast from day 7 to 34 and a weight on day forecast to day 7, 14, 21, 28 and 34 from all preceding days along with displaying the prediction ensemble standard deviation. We found an overall good agreement between measured broiler weight and the weight forecasts, but limited to the measured weight, which is known to be negatively biased onwards of day 15. Most importantly, we found that a dynamic interconnection between environmental broiler house conditions and broiler weight is present, and that it can be captured at least partially by the developed forecasting model. We analyzed the training data and forecasting data to explain the underlying reasons for some of these deviations. Additionally, we found that environmental input data from the previous batch can represent the ensemble standard deviation of the current batch, and is considered a reasonable substitute for future environmental input data.

Future work includes: Investigating state estimation as an alternative to reinitializing the model at every time step in the weight on day forecasting. Establishing a measure of how much the model has to generalize from known training data to forecasting data, as this was an intricate part of evaluating the quality of this work. Determine the optimal number of neurons, input/output delays and input/output variables.

SKOV A/S has submitted a patent application covering the presented concept and intends on implementing the developed forecasting algorithm in existing equipment.

ACKNOWLEDGEMENTS

The authors would like to thank broiler manager Leif Barsballe for permitting access to the data used in this work, and broiler expert Martin Rishøj at SKOV A/S for assisting in the analysis of the results.

REFERENCES

Aerts, J.M. et al. (2003). Recursive prediction of broiler growth response to feed intake by using a time-variant parameter estimation method. *Poultry Science*, 82, 40–49.

Aggrey, S. (2002). Comparison of three nonlinear and spline regression models for describing chicken growth curves. *Poultry Science*, 81(12), 1782–1788.

Ahmadi, H. and Mottaghitalab, M. (2007). Hyperbolastic models as a new powerful tool to describe broiler growth kinetics. *Poultry Science*, 86(11), 2461–2465.

Burden, F. and Winkler, D. (2008). Bayesian regularization of neural networks. In *Methods in Molecular Biology*, 23–42. Springer Science + Business Media.

Cangar, O. et al. (2007). Online growth control as an advance in broiler farm management. *Poultry Science*, 86, 439–443.

Demmers, T. et al. (2010). Neural predictive control of broiler chicken growth. *IFAC Proceedings Volumes*, 43(6), 311–316.

Du, K.L. and Swamy, M.N.S. (2014). *Neural Networks and Statistical Learning*. Springer Science + Business Media.

Elerolu, H. et al. (2014). Comparison of growth curves by growth models in slowgrowing chicken genotypes raised the organic system. *International Journal of Agriculture and Biology*, 16(3), 529–535.

Grahovac, J. et al. (2016). Modelling and prediction of bioethanol production from intermediates and byproduct of sugar beet processing using neural networks. *Renewable Energy*, 85, 953–958.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company.

Hernandez, S.C. et al. (2013). State estimation by artificial neural networks in a continuous bioreactor. *IFAC Proceedings Volumes*, 46(31), 215–220.

Lopes, A.Z. et al. (2008). Modeling productive performance of broiler chickens with artificial neural network. In *Livestock Environment VIII, 31 August - 4 September 2008, Iguassu Falls, Brazil*. American Society of Agricultural and Biological Engineers (ASABE).

Nair, V.V. et al. (2016). Artificial neural network based modeling to evaluate methane yield from biogas in a laboratory-scale anaerobic bioreactor. *Bioresource Technology*, 217, 90–99.

Nasimi, R. and Irani, R. (2014). Identification and modeling of a yeast fermentation bioreactor using hybrid particle swarm optimization-artificial neural networks. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 36(14), 1604–1611.

Nguyen, D. and Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *1990 IJCNN International Joint Conference on Neural Networks*. Institute of Electrical & Electronics Engineers (IEEE).

OECD (2015). OECD-FAO agricultural outlook 2015. `http://dx.doi.org/10.1787/agr_outlook-2015-en`.

Rajamani Doraiswami, C.D. and Stevenson, M. (2014). Closed loop identification. In *Identification of Physical Systems*, 357–378. Wiley-Blackwell.

Sameer Agarwal, K.M. et al. (2015). Ceres solver. `http://ceres-solver.org`.

Stacey, K. et al. (2004). An automatic growth and nutrition control system for broiler production. *Biosystems Engineering*, 89(3), 363–371.

Wang, H. et al. (2015). Chlorophyll-a predicting model based on dynamic neural network. *Applied Artificial Intelligence*, 29(10), 962–978.

Wathes, C. et al. (2008). Is precision livestock farming an engineer's daydream or nightmare, an animal's friend or foe, and a farmer's panacea or pitfall? *Computers and Electronics in Agriculture*, 64(1), 2–10.