Proceedings of the 10th World Congress on New Technologies (NewTech'24)
Barcelona, Spain - August 25-27, 2024
Paper No. ICERT 102
DOI: 10.11159/icert24.102

Hourly Hydropower Production Forecasting with Machine Learning: A Case Study in Linköping, Sweden

Linus Kåge[1], Vlatko Milic[1,2], Maria Andersson[1], Magnus Wallén[1]
[1]Linköping University
Division of Energy Systems, Department of Management and Engineering, Linköping University, 581 83 Linköping, Sweden
linus.kage@liu.se; vlatko.milic@liu.se; maria.h.andersson@liu.se; magnus.wallen@liu.se
[2]University of Gävle
Division of Building, Energy and Environment Technology, Department of Technology and Environment, University of Gävle, 801 76 Gävle, Sweden
vlatko.milic@liu.se

Abstract – Machine Learning (ML) is frequently utilized in prediction tasks; however, its applications in hydropower forecasting, particularly in forecasting hourly power production, has not been thoroughly investigated. In this paper, two Deep Learning (DL) models, namely an autoregressive neural network and Long Short-Term Memory, are compared to a seasonal autoregressive moving average (SARIMA) model to forecast the hourly power production at a hydropower station situated in Linköping, Sweden. Hyperparameter optimization algorithms are used to identify suitable DL models and algorithms for automatic model identification of SARIMA models are utilized. The three models are evaluated using a rolling origin strategy on a test dataset that consists of 10 months (January – October 2023) of hourly power production. The DL models provided similarly accurate forecasts as the SARIMA model according to mean squared error and mean absolute error. However, the DL models are poorly calibrated, resulting in lower coverage compared to the SARIMA model. Furthermore, the models are using a univariate time series (i.e., using historical power production to forecast future power production) and future studies need to explore additional variables that may be useful in providing a more accurate forecast.

***Keywords*:** Machine learning, deep learning, forecasting, time series, hydropower, power production, uncertainty estimation

1. Introduction

Hydropower constitutes an important role in the renewable energy mix. In 2022, hydropower accounted for 37% global renewable energies capacity [1], and in Sweden, hydropower generated 45% of all electricity in 2020 [2]. As the expansion of intermittent electricity production, such as wind and solar power, continues, power sources with controllable production will become increasingly important for balancing the power grid. Increased digitalization in the energy sector allows for large amounts of data to be collected which can be used for Machine Learning (ML) and Deep Learning (DL) applications. Examples of ML and DL within the energy sector include forecasting energy use and demand, fault detection, predicting power output from intermittent power sources such as wind and solar power [3]. In their systematic review, Krechowicz et al. [4] investigated 262 peer-reviewed articles employing ML application to forecast renewable energies and the authors found that only 13 articles were related to hydropower. Bernardes et al. [5] investigated 73 peer-reviewed papers of ML application in hydropower and concluded that research on ML applications with hourly time resolution is less common compared to research conducted with annual or monthly time resolution. Polprasert et al. [6] used autoregressive moving average (ARIMA) models to forecast the monthly power production at a hydropower station in Vietnam. The authors concluded that the ARIMA model performed well, and the accuracy of the forecast was satisfactory, but caution needs to be considered due to environmental and climate condition which may have an impact on the hydropower production. However, the authors did not explore alternative models in addition to the ARIMA model. DL has also been studied in forecasting hydropower production. Barzola-Monteses et al. [7] compared multi-layer perceptron (MLP), Long Short-Term Memory (LSTM) models and ARIMA to forecast the monthly gross power hydropower production. The authors utilized a simple grid search to find a good selection for hyperparameters in the DL models. It was concluded that the MLP was an appropriate choice for the given dataset and that finding a good choice for hyperparameters is important for predictive performance.

Numerous scientific investigations have been conducted on prediction of the power production from renewable energy sources. However, to the best of the authors' knowledge, there has been insufficient research conducted on ML applications in hydropower and more specifically on forecasting hourly power production. This research gap is also highlighted by [4, 5]. The first objective of this paper is to compare the predictive performance of neural networks and LSTM with a seasonal ARIMA model (SARIMA) according to mean squared error and mean absolute error. The second objective is to study the estimated model uncertainty in the forecast and compare the accuracy and quality of the estimated forecast intervals by assessing the width and the coverage of the intervals.

1.2 Data

The data set consists of hourly measurements of power production from the Odensfors hydropower station located in the river Svartån. Svartån runs from the lake Sommen to lake Roxen in Linköping, Sweden. Fig. 1 presents the hourly power production from June 2020 to October 2023. The power production has a yearly seasonality with low power production during the summer and an increased power production during fall, winter, and spring where precipitation, snowmelt and surface runoff increases which subsequently affects the water flow.

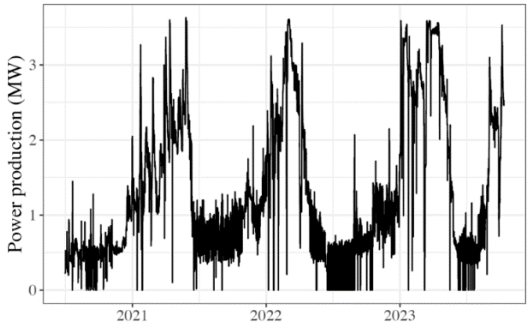


Fig. 1: Hourly power production from June 2020 to October 2023.

The time series is split into three disjoint subsets, a training, a validation, and a test data set. The training set is used to estimate the models, while the validation set is used to adjust model configurations in attempt to improve the predictive performance. The final evaluation is performed on the test set, where the models are compared. The training set contains measurements between 2020-07-01 and 2022-07-31 ($n_{training} = 18265$), the validations set from 2022-08-01 to 2022-12-31 ($n_{valid} = 3673$), and the testing set from 2023-01-01 to 2023-10-11 ($n_{test} = 6817$).

2. Theory

A time series consists of measurement over time, and if the magnitude of the observations in the time series steadily increase or decrease over time, then the time series is said to have a trend. The time series may exhibit a seasonal pattern, which is a pattern that occurs due to seasonal factors and the seasonal effects are fixed for the time series [8]. Forecasting in time series analysis means to predict the future as accurately as possible [8]. Let $y_{\{1:n\}}$ be a time series which consists of n measurements, then the objective when forecasting is to correctly predict $y_{\{n+1:n+H\}}$ where H is the number of time steps into the future.

2.1 Autoregressive Moving Average models

The ARIMA model that consists of two components, the moving average (MA) part of order q and the autoregressive (AR) part of order p [8]. Eq (1) presents the formula of $ARIMA(p, d, q)$ and is written as

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t, \quad (1)$$

where y'_t is the d:th degree of first differencing of the original time series and c is the estimated intercept of the model. $\phi_1, \ldots, \phi_p$ and $\theta_1, \ldots, \theta_q$ are the estimated coefficients for the AR and MA part respectively. Seasonality in the data can be handled by extending the ARIMA model into a SARIMA model by including AR and MA terms at the seasonal lags [8] as presented in Eq (2). Let m be the determined seasonality in the time series, then the $SARIMA(p,d,q)(P,D,Q)_m$ model is expressed as

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \Phi_1 y'_{t-m} + \cdots + \Phi_P y'_{t-mP} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \Theta_1 \epsilon_{t-m} + \cdots + \Theta_Q \epsilon_{t-mQ} + \epsilon_t, \quad (2)$$

where $\Phi_1, \ldots, \Phi_P$ and $\Theta_1, \ldots, \Theta_Q$ are the estimated cofficients of the seasonal AR and MA parts. The autocorrelation function (ACF) and partial autocorrelation function (PACF) can be used to identify a SARIMA model in Eq. (2). However, identifying a SARIMA model using ACF and PACF is a subjective choice and may be more difficult to interpret. Algorithms to identify SARIMA models have been introduced to make model identification easier (e.g., [9]). The algorithm introduced by Hyndman and Khandakar [9] is initialized by estimating four models, and the best model according to the information criteria AIC is kept and set to the "current model". After identifying the best model, variations to p, d, P and Q in the current model are performed to explore additional models that may be a better alternative according to AIC. This extra step is then repeated until convergence where a model with a lower AIC cannot be found.

2.2. Neural networks and Deep Learning models

Neural networks (NN) form the foundation of DL, where the network are used to approximate the unknown function that generates the data [10]. The network is parametrized by the weights which are estimated by minimizing a cost function to better approximate the unknown function [10]. NN allows for non-linearity to be modelled which is achieved by introducing non-linear activation functions in the units of the network (e.g., the rectified linear unit, sigmoid function, or the hyperbolic tangent function). The activations functions and the corresponding graphs are visualized in Fig. 2. The rectified linear unit takes the input and sets the negative values to zero, otherwise it is linear. The sigmoid function squeezes the input into the range [0,1], and similarly the hyperbolic tangent takes the input and squeezes it into range [-1,1].

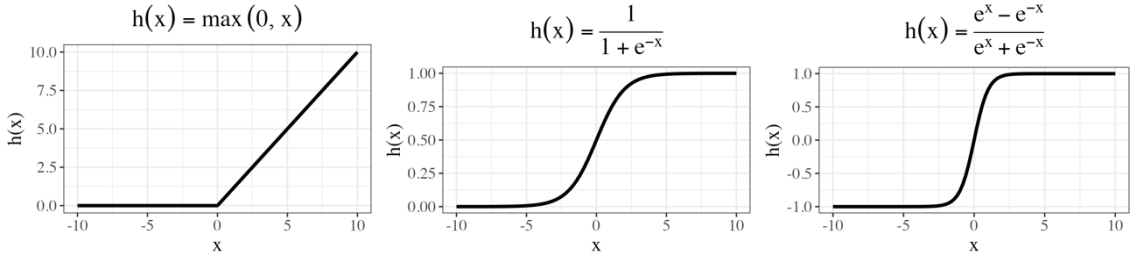


Fig. 2: From left to right, rectified linear units (ReLU), sigmoid and hyperbolic tangent (tanh).

NN can be used for time series forecasting by feeding the network previous values in the time series as inputs, that is creating an autoregressive neural network. The autoregressive neural networks (AR-net) allow a flexible neural network to process a time series and forecast the future time steps. The inputs of AR-net are previous values of the time series of any order p which needs to be determined prior to training the model. AR-net can easily fit a high order of the auto-regressive part [8, 11].

The LSTM model was originally developed by Hochreiter and Schmidhuber [12] to handle computational difficulties that Recurrent Neural Networks (RNN) have with training on longer sequences. RNN is a type of network used to process sequential data but often have problems with either vanishing or exploding gradients, resulting in unstable training on longer sequence [13, 14]. The idea of LSTM is to create paths which controls the flow of information which allows the gradient to neither vanish nor explode, resulting in more stable training. LSTM consists of three types of gates: the forget gate, the

external input gate and the output gate [10]. The forget gate allows for parts of the memory of the LSTM to be forgotten, the external input gates controls how much new additional information should be added onto the LSTM memory and finally the output gate controls how information from the LSTM memory should be used in the output [10].

2.5. Monte Carlo Dropout

Dropout is a technique used to regularize the NN to prevent it from overfitting to the training data. With dropout, units in the NN are randomly removed during training preventing the weights connected to the corresponding unit from being updated during backpropagation [15]. Although dropout is used to prevent overfitting, enabling dropout during prediction allows for model uncertainty in the forecast to be estimated by performing several forward passes [16]. Allowing dropout during prediction is called Monte Carlo (MC) dropout. The forward passes will become stochastic depending on which set of weights that remain at each forward pass resulting in different forecasts for each forward pass. By performing several stochastic forward passes, the marginal predictive distribution can be approximated in which the model uncertainty can be estimated.

2.6. Forecasting strategies

There are different strategies in how a model can be designed to forecast the future, such as the recursive strategy and the multi-input multi-output (MIMO). Given an input sequence, the recursive forecasting strategy performs one step predictions and then use the prediction as input to predict the next time step. An arbitrary long horizon can easily be forecasted with the recursive strategy due to predictions being re-used in the input sequence to the model [17]. However, due to the recursion of using predictions as inputs, errors made by the model may be accumulated resulting in inaccurate forecast due to the previous erroneous forecasts. Using MIMO, the model is designed to forecast the whole horizon in one step meaning that the forecast is no longer a scalar but a vector of length H [17]. MIMO does not suffer the problem with accumulated errors that the recursive strategy has due to the model is optimized to predict the whole horizon in one step.

2.7. Hyperparameter optimization

NNs have adjustable configurations called hyperparameters (HP) which influence the predictive performance of the model. The HPs are set prior to training the model and identifying the HPs is a time-consuming task due to the numerous combinations of HPs to evaluate. Furthermore, the selection process of HP may also introduce problems with replication due to the subjective choices made by the researcher in selecting the hyperparameters [18]. Hyperparameter optimization (HPO) aims to assist in finding a good selection of hyperparameters which results in small predictive errors. Hyperband, introduced by Li et al. [19], is built upon the idea of successive halving. Successive halving (SH) explores different set of hyperparameters and continuously discard half of the combinations of HPs which results in poor predictive performance. This is repeated until convergence where only one combination of HPs remain [20]. Hyperband applies successive halving and a grid search over the number of configurations of hyperparameters, allowing for a faster exploration of hyperparameters [19].

3. Methods

The model building procedure can be summarized in four steps, see Fig. 3. Initially, the data is pre-processed by taking the first order difference of the time series for the DL models, described in section 3.1. For the SARIMA model, differences may be taken in the first order but also in the seasonal order. After pre-processing the time series, models are built on the training set, using hyperparameter optimization for DL and automated model identification of SARIMA model. After estimating the models, the uncertainty in the forecast is estimated for each model as described in section 3.2. Finally, all models are compared according to the evaluating metrics described in section 3.3.

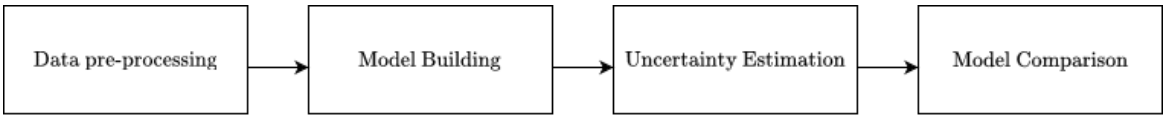


Fig. 3: Flowchart of methodology in the performed experiments.

3.1 Data pre-processing

Prior to training the DL models, the data has been pre-processed by taking the first order difference $z_t = y_t - y_{t-1}$. Instead of predicting the actual power production at any given time point, the expected difference in production between two consecutive time points is instead predicted. After forecasting the horizon of the differenced time series, the predictions must be post-processed to represent power production as expressed in Eq (3). For any given time point $t + \ell$, the predicted power production is estimated as

$$\hat{y}_{t+\ell} = y_t + \sum_{i=1}^{\ell} \hat{z}_{t+i}, \tag{3}$$

where $\ell \in [1, H]$. This means that the forecast $\hat{y}_{t+\ell}$ is the last observed value y_t in the day before and the cumulative sum of the forecasts of the differences between two time points up until the time point of interest $t + \ell$.

3.2 Estimation of model uncertainty

To estimate the model uncertainty in the forecast, several stochastic forward passes need to be performed, and the results stored. The number of stochastic forward passes is considered a hyperparameter and needs to be chosen prior to training the model uncertainty. In this paper 100 stochastic forward passes are performed to estimate the model uncertainty. Furthermore, the prediction intervals with 95% confidence are constructed by evaluating the 2.5th and 97.5th percentiles of the MC dropout simulation. The lower and upper bound of the prediction intervals are evaluated before transforming back the forecast into the original time series, according to Eq (3).

3.3 Model evaluation

Evaluation of the models consider both the accuracy of the point estimates in Eq (3) but also the uncertainty in the forecast given by the models. To assess the forecast accuracy two metrics are used, mean squared error (MSE) in Eq (4) and mean absolute error (MAE) in Eq (5). Let $\hat{y}_{t+\ell}$ be the point prediction made by the model at time point $t + \ell$. Furthermore, let the observed value of the test point be $y_{t+\ell}$, then MSE is defined as

$$\text{MSE} = \frac{\sum_{\ell=1}^{H} \left(y_{t+\ell} - \hat{y}_{t+\ell}\right)^2}{n_H}, \tag{4}$$

where n_H is the length of the forecast horizon. Similar, MAE is defined as

$$\text{MAE} = \frac{\sum_{\ell=1}^{H} \left|y_{t+\ell} - \hat{y}_{t+\ell}\right|}{n_H}. \tag{5}$$

Assessing the model uncertainty, the metrics prediction interval coverage probability (PICP) and interval score (IS) are used. PICP in Eq (6) measures how many of the data points are covered by the prediction interval [21]. Let L_t and U_t be the estimated lower and upper bound of the prediction interval estimated by MC Dropout, then the PICP is defined as

$$\text{PICP} = \frac{\sum_{\ell=1}^{H} \mathbb{1}\{y_{t+\ell} \in [L_{t+\ell}, U_{t+\ell}]\}}{n_H}, \tag{6}$$

where $\mathbb{1}\{\cdot\} = 1$ if the condition is true, otherwise it is 0. Although PICP evaluates the percent of data points covered by the prediction interval it does not consider the uncertainty of the model (i.e., the width of the interval) in the forecast at the time point. IS in Eq (7) assesses how uncertain the model is in the its forecast by considering the width of the interval at any time point [22]. If (1-α) $\cdot 100\%$ prediction intervals are constructed where $1 - \alpha$ is the confidence level, then the interval score is defined as

$$\mathrm{IS}_{t+\ell} = (U_{t+\ell} - L_{t+\ell}) + \frac{2}{\alpha}(L_{t+\ell} - y_{t+\ell})\mathbb{1}\{y_{t+\ell} < L_{t+\ell}\} + \frac{2}{\alpha}(y_{t+\ell} - U_{t+\ell})\mathbb{1}\{y_{t+\ell} > U_{t+\ell}\}. \quad (7)$$

Again, $\mathbb{1}\{\cdot\}$ is an indicator variable encoded as 1 if the condition is true, otherwise it is 0, $U_{t+\ell}$ and $L_{t+\ell}$ are the estimated upper and lower bound of the prediction interval at time point $t + \ell$. The interval score is evaluated at each time point, and to assess the interval score for the whole forecast horizon, the summation of $\mathrm{IS}_{t+\ell}$ is divided by n_H, like the metrics in Eq. (4) – (6). The three models are evaluated on the whole test set by using rolling origin. The origin of the forecast is moved along the time series allowing the model to test on more data without the consequences of performing one long forecast from a fixed origin [23]. However, the forecast horizon remains 24 ($n_H = 24$) hours, and therefore the metrics will be divided by $n_H \cdot number\ of\ days$ in the test set. The DL models are trained using MIMO meaning that the rolling origin evaluation happens naturally but with the SARIMA model, the origin will be moved by concatenating more days onto the training set without re-estimating the coefficients of the model. This means that the model will always have the true observed power production as inputs to the model, and not use forecasted values as inputs, when performing a forecast.

4. Results and analysis

Prior to training the DL models, a grid of values for the Hyperband algorithm needs to be determined and the settings for the Hyperband algorithm are presented in Tables 1 and 2. Moreover, the Hyperband algorithm is performed by monitoring MSE on the validation data for each set of HPs. The number of hidden layers that the Hyperband algorithm will search over is between 1 and 6 with an increment of 1. Similarly, the number of hidden units in each hidden layer will vary between 16 and 256 with a step size of 16. Lastly, the learning rate in the optimization algorithm when training the networks will test three different values, as presented in Table 1.

Table 1: Defined settings for Hyperband algorithm of AR-net.

Hyperparameter	Values
Number of hidden layers	1 to 6, with step size = 1
Number of units in hidden layers	16 to 256 with step size = 16
Learning rate	0.01, 0.001, 0.0001

The final model resulted in one single hidden layer with 240 units ReLU as activation function. In connection with the hidden layer, a dropout layer is added which is activated when forecasting to enable the model uncertainty to be estimated. The HP configuration for hyperband of the LSTM model is presented in Table 2. It uses the same settings as the AR-net, with an additional tuning of the number of units in the LSTM layer of the model. The final LSTM model resulted in 224 units in the LSTM layer and a single hidden layer with 224 units with ReLU as activation function.

Table 2: Defined settings for Hyperband algorithm of LSTM.

Hyperparameter	Values
Number of units in LSTM	16 to 256 with step size = 16
Number of hidden layers	1 to 6, with step size = 1
Number of units in hidden layers	16 to 256 with step size = 16
Learning rate	0.01, 0.001, 0.0001

The identified SARIMA model using the algorithm presented in section 2.1 is $SARIMA(1,1,3)(2,0,0)_{24}$. Table 3 presents the forecasting accuracies of the three models. All three models have similar forecasting accuracy according to MSE and MAE. However, the PICP is much higher and IS is lower for the SARIMA model in comparison to the DL models. The set confidence levels in the prediction intervals are 95% and the PICP does not correspond to the confidence level for the DL models.

Table 3: Model evaluation on the test set. The best results are presented in bold.

Model	MSE	MAE	PICP	IS
SARIMA	0.08	0.16	**0.93**	**1.89**
AR-net	0.079	**0.14**	0.27	3.56
LSTM	**0.073**	0.14	0.26	3.61

Fig. 4 presents the forecast of the three models using the moving origin. All three models produce forecasts close to the observed production, however, none of the models are not capable of predicting the increased variation during summer months.

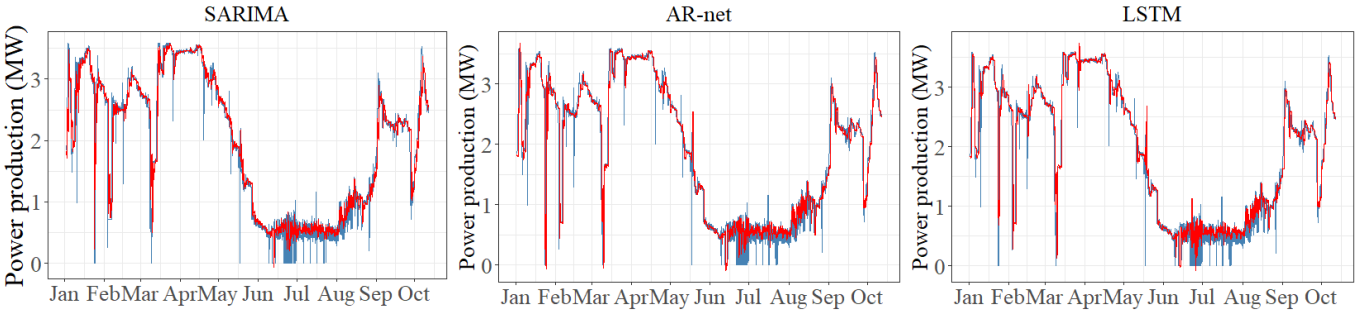
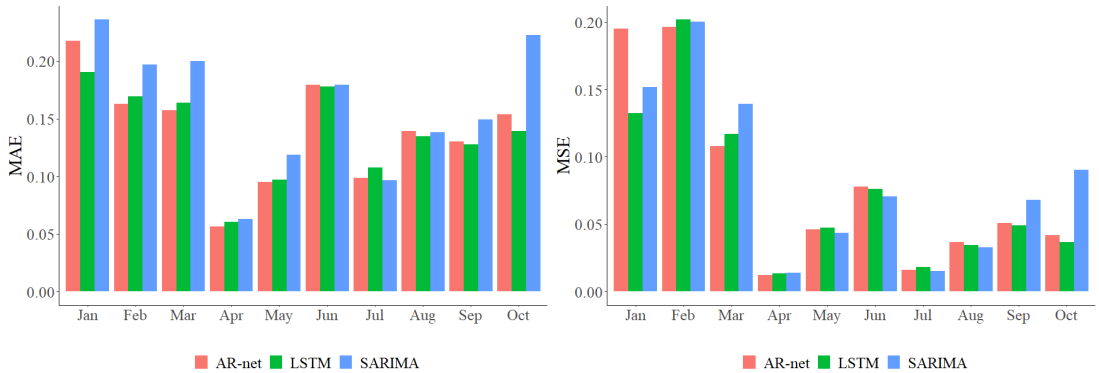


Fig. 4: Forecast on the test set. Red line corresponds to model forecast and blue line corresponds to observed power production.

Fig. 5 presents the model performance for every month to further evaluate where the models make erroneous forecasts. Once again, all three models perform similarly according to MSE and MAE for each month. However, the SARIMA model has larger PICP and lower IS for all months in comparison to the DL models. The largest errors for all models occur during the spring (January to March). The smallest errors occur during the summer months and then they increase during fall. The pattern of smaller forecasting errors in the summer and increased forecasting errors during spring and fall is similar for all three models.



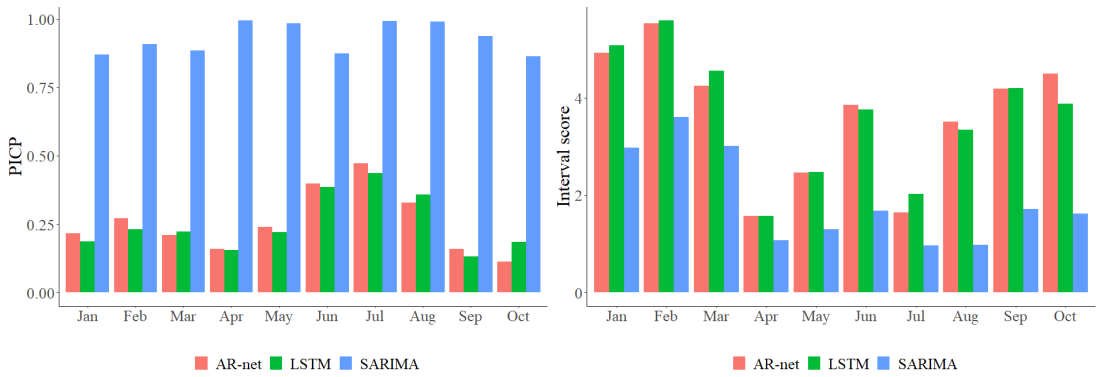


Fig. 5: Model performance for each month according to the evaluation metrics

4. Concluding discussion

Both AR-net and LSTM show low coverage of 27% and 26% respectively, suggesting that the models are poorly calibrated. If the estimated prediction intervals are created with a confidence of $1 - \alpha$, then roughly (1-α)% of the data points are covered by the prediction interval. If the coverage exceeds the confidence level, then the model is under-confident, and similarly, if the coverage is smaller than the confidence level it is said that the model is overconfident [24, 25]. The three models have similar forecasting accuracy according to MSE and MAE. However, the coverage is much smaller than the SARIMA model but also smaller than the given confidence level, suggesting that the DL models are overconfident in the forecast.

Odensfors power station has water storage capabilities, and the daily power production is manually planned by operators. The manual planning of the production poses a challenge when forecasting future power production. In the data set, the plan of tomorrow's production in Odensfors hydropower station is not measured, suggesting that future studies may need to explore this variable in more detail to assess whether it improves the accuracy or not. Furthermore, local weather conditions affect the water flow. Weather variables such as precipitation, ground moisture, temperatures, and evaporation and their interdependencies may be important to include in the model and worth examining in future studies. Recent advances in ML-based weather forecasting show promising results of providing accurate weather forecasts globally [26]. Utilizing these ML-based weather forecasts could potentially improve the accuracy of the production forecasts.

An alternative to MC dropout for estimating the model uncertainty is to explore Bayesian neural networks (BNN), which are a type of stochastic neural network. The weights in a BNN are assigned to a probability distribution which is determined using Bayes' rule. The uncertainty can then be quantified by doing similar MC estimation as done in this study, where several samples from the posterior probability distribution is drawn and the prediction is computed for each random sample [27]. Furthermore, BNNs provide regularization in the prior probability distribution to prevent the model from overfitting [27]. It is important to emphasize that the performance of a NN is dependent on the choice of hyperparameters. Further investigation into HPO may be necessary to further improve the performance of the neural networks. In this paper, the length of input sequence, which can be interpreted as a hyperparameter, has been fixed to 24 hours (i.e., one day of production to forecast the production the next day). Further sensitivity analysis needs to be conducted to better understand how the length of input data affects the forecasting accuracy of the models.

In conclusion, all three models have similar forecasting accuracy according to MSE and MAE. However, the DL models are poorly calibrated according to PICP which suggests that additional work on developing the models is required to further improve the model performance.

Acknowledgements

We would like to thank Tekniska verken i Linköping AB for funding the project and providing the dataset.

References

[1] REN21, "Renewables 2023 Global Status Report Collection," 2023.

[2] Swedish Energy Agency, "Energy in Sweden 2022 An Overview," Eskilstuna, Sweden, 2022. Accessed: 2024-01-03. [Online]. Available: https://energimyndigheten.a-w2m.se/Home.mvc?ResourceId=208766

[3] M. M. Forootan, I. Larki, R. Zahedi, and A. Ahmadi, "Machine Learning and Deep Learning in Energy Systems: A Review," *Sustainability*, vol. 14, no. 8, 2022, doi: 10.3390/su14084832.

[4] A. Krechowicz, M. Krechowicz, and K. Poczeta, "Machine Learning Approaches to Predict Electricity Production from Renewable Energy Sources," *Energies*, vol. 15, no. 23, 2022, doi: 10.3390/en15239146.

[5] J. Bernardes, M. Santos, T. Abreu, L. Prado, D. Miranda, R. Julio, P. Viana, M. Fonseca, E. Bortoni, and G. S. Bastos, "Hydropower Operation Optimization Using Machine Learning: A Systematic Review," *Ai*, vol. 3, no. 1, pp. 78-99, 2022, doi: 10.3390/ai3010006.

[6] J. Polprasert, V. A. Hanh Nguyen, and S. Nathanael Charoensook, "Forecasting Models for Hydropower Production Using ARIMA Method," presented at the 2021 9th International Electrical Engineering Congress (iEECON), 2021.

[7] J. Barzola-Monteses, J. Gómez-Romero, M. Espinoza-Andaluz, and W. Fajardo, "Hydropower production prediction using artificial neural networks: an Ecuadorian application case," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13253-13266, 2021, doi: 10.1007/s00521-021-06746-5.

[8] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, 3 ed.: OTexts: Melbourne, Australia, 2021. [Online]. Available: OTexts.com/fpp3. Accessed on: 2023-11-23.

[9] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R," *Journal of statistical software*, vol. 27, pp. 1-22, 2008.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[11] O. Triebe, N. Laptev, and R. Rajagopal, "Ar-net: A simple auto-regressive neural network for time-series," *arXiv preprint arXiv:1911.12436*, 2019.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[13] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157-166, 1994.

[14] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013: Pmlr, pp. 1310-1318.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[16] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016: PMLR, pp. 1050-1059.

[17] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067-7083, 2012, doi: 10.1016/j.eswa.2012.01.039.

[18] B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A. L. Boulesteix, D. Deng, and M. Lindauer, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, 2023, doi: 10.1002/widm.1484.

[19] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The journal of machine learning research*, vol. 18, no. 1, pp. 6765-6816, 2017.

[20] K. Jamieson and A. Talwalkar, "Non-stochastic best arm identification and hyperparameter optimization," in *Artificial intelligence and statistics*, 2016: PMLR, pp. 240-248.

[21] Y. Xie, C. Li, M. Li, F. Liu, and M. Taukenova, "An overview of deterministic and probabilistic forecasting methods of wind energy," *iScience*, vol. 26, no. 1, p. 105804, Jan 20 2023, doi: 10.1016/j.isci.2022.105804.

[22] T. Gneiting and A. E. Raftery, "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359-378, 2007, doi: 10.1198/016214506000001437.

[23] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," *International journal of forecasting*, vol. 16, no. 4, pp. 437-450, 2000.

[24] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. S1, pp. 1513-1589, 2023, doi: 10.1007/s10462-023-10562-9.

[25] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *International conference on machine learning*, 2018: PMLR, pp. 2796-2804.

[26] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, and W. Hu, "Learning skillful medium-range global weather forecasting," *Science*, p. eadi2336, 2023.

[27] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on Bayesian neural networks—A tutorial for deep learning users," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 29-48, 2022.