An Optimised Ensemble Approach for Multivariate Multi-Step Forecasts Using the Example of Flood Levels

Michel Spils[a] and Sven Tomforde[b]

Department of Computer Science, University of Kiel, Kiel, Germany

Keywords: Machine Learning, Time-Series Forecasting, Ensemble Deep Learning, Flood Forecasting.

Abstract: Deep Learning methods have become increasingly popular for time-series forecasting in recent years. One common way of improving time-series forecasts is to use ensembles. By combining forecasts of different models, for example calculating the mean forecast, it is possible to get an ensemble that performs better than each single member. This paper suggests a method of aggregating ensemble forecasts using another neural network.The focus is on multivariate multi-step ahead forecasting. Experiments are done on 5 water levels at small to medium-sized rivers and show improvements on naive ensembles and single neural networks.

1 INTRODUCTION

Flood and disaster protection are typically sovereign tasks of local authorities and include not only coastal regions but also areas of influence of flowing waters of different sizes. For example, the State Office for the Environment in Schleswig-Holstein, Germany's northernmost federal state, operates 182 gauging stations close to in-land rivers where water levels and flow behaviour are determined and forecasts are made. On the basis of these forecasts, disaster control measures are taken if necessary. One challenging aspect of forecasting water levels is the lack of training data. It is not possible to simply simulate the environment, the only way to gather more data for a gauging station is to wait.

The application scenario requires very reliable forecasts of water levels, as protective measures are taken on this basis. At the same time, this places special demands on forecasting methods because, on the one hand, conditions are not static due to climate change and changes in topology and, on the other hand, the focus is on extreme events. Thus, a deviation of a few percent in normal behaviour is absolutely uncritical, whereas this is essential in the case of floods. Hence, the overall goal is to investigate a self-adaptive and self-learning forecast system with a continuous assessment of uncertainty, reliability and impact of the determined forecasts.

[a] https://orcid.org/0000-0002-6431-6085
[b] https://orcid.org/0000-0002-5825-8915

In this paper, we present a novel approach to generate and use ensembles for water level prediction. Based on previous work on the forecast quality of single predictors, we investigate to what extent an ensemble of optimised models improves the forecast quality. This is combined with an approach to optimise the weighting within the ensemble in order to achieve a further gain in prediction quality and robustness. We analyse the impact influence factors of the observed behaviour and provide the considered data of the experiments.

The remainder of this paper is organised as follows: Section 2 describes the current state-of-the-art from a technical and an application point of view. Section 3 introduces our approach for an ensemble-based forecasting of flood levels. Afterwards, Section 4 introduces the underlying data and metrics, followed by a description of base models and execution time. The analysis of the experimental results as well as the insights and findings are discussed in Section 5. Finally, Section 6 summarises the paper and describes future work.

2 RELATED WORK

In the first subsection 2.1 we give a short overview of flood forecasting using Machine learning in general, in subsection 2.2 we focus on different ways of using ensembles for forecasting.

Spils, M. and Tomforde, S.
An Optimised Ensemble Approach for Multivariate Multi-Step Forecasts Using the Example of Flood Levels.
DOI: 10.5220/0012396000003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 2*, pages 388-396
ISBN: 978-989-758-680-4; ISSN: 2184-433X

2.1 Flood Forecasting

In (Kratzert et al., 2021) Long Short-Term Memory networks (LSTMs) are used to simulate flow rates based on meteorological observations. They find that it is possible to (pre-)train a single model to predict flow rates at multiple basins. The authors of (Hu et al., 2018) study the performance of LSTMs compared to standard neural networks for flood forecasting and find that they perform significantly better. In (Kao et al., 2020) the authors explore LSTM based Encoder-Decoder architectures for multi-step-ahead flood forecasting.

In (Grundmann et al., 2020) the use of precipitation forecast ensembles for flood forecasting is investigated. The German Weather Service provides an ensemble with 20 members of high-resolution precipitation forecasts, which is useful for rare events with strong rainfall that are not represented in the main forecast. (Morgenstern et al., 2022) enrich their LSTM input with statistical information such as area maximum, minimum and standard deviation of precipitation intensity. (Wee et al., 2021) and (Mosavi et al., 2018) give a general overview of water level and flood forecasting using machine learning. Most research on flood forecasting focuses on large rivers and on either very short forecast horizon, up to six hours ahead or on daily forecast. We attempt to forecast water levels with a forecast horizon up to 48 hours at small to medium sized rivers with a catchment area of up to $600km^2$.

2.2 Ensemble-Based Forecasting

In (Sommer et al., 2016), a first study has been presented that aims at re-weighting the ensemble based on current conditions using an evolutionary reinforcement learning paradigm (XCSF/eXtended Classifier System for Function approximation). Moreira-Matias et al. (Moreira-Matias et al., 2013) use the error of their base models on a sliding window to directly weigh their ensemble. In a more recent approach (Choi and Lee, 2018) change the ensemble weights at each step based. This is again based on the base models error on a sliding window, but is also parametrized by a learning rate λ and a discount factor γ which weights more recent errors higher. (Cerqueira et al., 2017) train a second model for each base model that predicts the expected error, this error is then used to weight the ensemble. Later they extended this approach to also consider model diversity (Cerqueira et al., 2018). In (Saadallah and Morik, 2021) a Deep Deterministic Policy Gradient (DDPG) model is used to combine univariate single horizon forecasts. The RL uses a sliding window of past forecasts as state and the rank of the ensemble model compared with the base models as the reward. One approach explicitly designed for multi-horizon forecasting approximates optimal weights by calculating a weighted linear regression on either the training data or a sliding window of past data (Galicia et al., 2019). In (Gheyas and Smith, 2011) a set of base learners is trained on pairwise disjoint subsets of available features. The output of the base learners is used as input to another model, after undergoing feature selection. In (Casanova and Ahrens, 2009) the authors evaluate equal weighting, simple skill-based weighting and Bayesian model averaging for weather forecasting.

Most existing approaches do not take the current input into account and instead rely on recent or historic performance. We supply the input to the base models to our weighting approach so that very recent changes in the situation such as a unexpected rainfall can be considered when weighting the ensemble.

3 PROPOSED ENSEMBLE WEIGHTING APPROACH

In this section we present our approach to combine forecasting ensembles. The basic idea of our approach is to directly weight the ensemble forecast using a neural network, instead of the usual approach of using the ensemble forecast as input to another neural network or alternative machine learning algorithm. First, a set of arbitrary base learners is trained. This has the advantage that each model is trained independently, which allows us to take advantage of hyperparameter optimization, unlike dependent frameworks like AdaBoost.

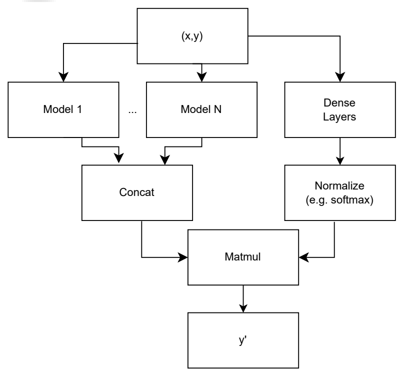

Figure 1: Ensemble weighting architecture.

The ensemble model is a neural network with two paths. One path with an arbitrary but constant number of base learners and the second path with a small number of layers to predict the weighting matrix. Currently, a number of dense layers is used to predict the optimal weighting, but the exact architecture is arbitrary. To calculate the final output the model does an element-wise multiplication of the concatenated outputs of the base models and the weight matrix.

Figure 1 shows this architecture.

The ensemble model has the task of optimizing the weighting for n base models and the current input data x. The ensemble model calculates a matrix of weights w which is normalized so that $w_{i,j} \geq 0$ and $\sum_{i=1}^{n} w_{i,j} = 1$ for each forecast step $j \in \{1, \ldots, h\}$. This guarantees that the forecast of the ensemble model is bounded by the minimum and maximum of the base forecasts. Whether this is a positive or a negative constraint depends on the quality and diversity of the base models.

This can be done in different ways, the optimal normalization function likely depends on the distribution of model quality. We tested linear normalization, which is defined as:

$$w' = \frac{w - min(w)}{\sum_{i=1}^{n}(w_i - min(w))} \tag{1}$$

And the softmax function:

$$w' = \frac{exp(w)}{\sum_{i=1}^{n}(exp(w_i))} \tag{2}$$

To calculate the final output the ensemble model does an element-wise multiplication of the concatenated outputs of the base models and the weight matrix. The meta-learner is trained on the same data as the base models. During the training process of the meta-learner, the base models are frozen. It is not technically necessary to implement this with the base models as part of the meta-learner and it may be more computationally efficient to cache the forecasts made by the base models.

4 DATASETS AND BASE MODELS

In this section, we first introduce our datasets and evaluation metrics. We then describe our base models or learners and how long the training took.

4.1 Datasets

The 5 datasets used in this paper represent the different water levels that we attempt to forecasts and contain publicly available data. They all contain sensor and radar data and values derived from the sensor data. The target value of each dataset is the water level of a river in Schleswig-Holstein, Germany. Additionally, they contain precipitation forecasts that are synthesized by shifting the calculated precipitation 48 hours into the past. In a production environment, this synthetic forecast would be replaced by the ICON D2 forecast (Reinert et al., 2020). The input data for each gauge is a subset of the nearest available sensor stations measuring air temperature, air pressure, air humidity, soil moisture, evaporation, and up- and downstream water levels. The aggregate data and all used source code will be made available on request. The goal is to predict the water levels for the next 48 hours from the past water levels and other sensor data.

Figure 2 shows a map of the $458.6 km^2$ catchment of the water level Foehrden-Barl at the river Stoer. The green dot represents the location of the target water level, the black dots represent the location of upstream water levels and the blue dot the location of the nearest air temperature and air humidity sensor. The three marked areas represent subcatchments for which precipitation was calculated. For the Foehrden-Barl dataset soil moisture and air pressure data was used, but the sensor stations are outside the area on the map.

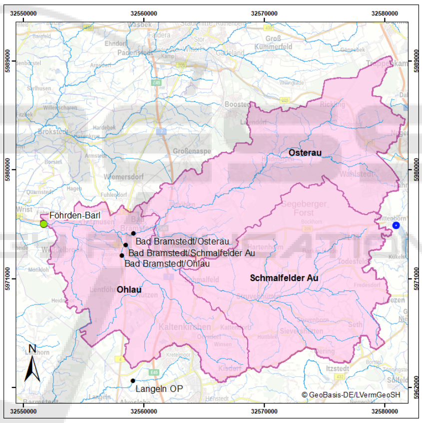

Figure 2: Map of the catchment Foehrden-Barl.

All features are sampled hourly, but some derived values like soil moisture are upsampled from daily calculations. For training all data was standardized to a mean of zero and a standard deviation of one.

Table 1: The five datasets used in the experiments.

Dataset	Features	Length
Foehrden-Barl	13	61360
Hollingstedt	9	71542
Halstenbek	5	83816
Poetrau	5	77699
Willenscharen	10	77434

4.2 Evaluation Metrics

We track several metrics for our experiments, but for the sake of brevity and readability we only include the Weighted Average Percentage error (WAPE) and the Nash–Sutcliffe model efficiency coefficient (NSE) in this paper. The WAPE of each forecast horizon is defined as

$$WAPE = \frac{\sum_{t=1}^{T} |y_t - \hat{y}_t|}{\sum_{t=1}^{T} |y_t|} \quad (3)$$

with T the number of observations, y_i the true value and $\hat{y}_i$ the predicted value.

The NSE is a standard metric for hydrological models. A perfect model would result in a NSE of 1, a model that just predicts the mean observed value in a NSE of 0.

$$NSE = 1 - \frac{\sum_{t=1}^{T} (y_t - \hat{y}_t)^2}{\sum_{t=1}^{T} (y_t - \bar{y})^2} \quad (4)$$

with $\bar{y}$ being the mean observed value.

4.3 Base Learner

For each dataset, we trained 24 neural networks as ensemble members. All models were trained on the first 70% of each dataset, with the following 15% parts as validation and test set. To reduce overfitting training was stopped whenever the validation loss did not reduce for three epochs. On average this happened after 23 epochs. Table 2 shows the (hyper-)parameters of all base models. We exclusively used LSTM networks, because preliminary testing showed that they outperform other architectures in both performance and training speed for our datasets. On average the MAE of models using for example a Transformer architecture was 25% higher and models using the Autoformer architecture ((Wu et al., 2022)) were 50% higher, while taking 3-4 longer to train.

4.4 Execution Time

Training neural networks can be very expensive, so it is important not to ignore the computational cost of new approaches. All training was done on a single Nvidia A100 GPU with 80GB memory. Training the base models took on average 55s, with a minimum of 12s and a maximum of 93s. Training the ensemble models took on average 40s, with a minimum of 10s and a maximum of 129s. The cost of our approach of training a neural network to combine an ensemble of models is in the same order of magnitude as the the cost of training a single neural network.

Table 2: Base learner parameters.

Parameter	Value
Input window	144 hours
Output window	48 hours
Learning rate	0.001
Loss function	MSE
Optimizer	Adam
Batch size	4096
Max Epochs	100
Dropout	0.25
LSTM Layers	[1,2]
LSTM Units	[64,128,256]
Hidden layers	[1,2]
Hidden layer units	[128,256]

5 EXPERIMENTS AND RESULTS

This section contains descriptions of our experiments and their results.

5.1 Experiment 1: Static Model Set, Hyperparameter Optimization

For this experiment, we did a hyperparameter optimization for the ensemble model with a fixed set of base models. We used Tree-structured Parzen Estimator (TPE) for sampling and trained 100 models, minimizing the validation loss. Table 3 shows the hyperparameter constraints. In Figure 3 the NSE values of each ensemble model, the base model and the mean forecast of the base model set is displayed. The ensemble model has a higher thus better NSE than the base models with nearly all tester hyperparameters, except at the water level Poetrau, where the base models perform better with small forecast horizons. Figure 4 shows the results of the hyperparameter-optimization. The linear/minmax normalization results in a lower validation loss for all datasets except Poetrau. The optimal found hyperparameters differ between the datasets. There is no clear optimal amount of hidden layers and hidden units. There is a trend towards a low learning rate being better, but for Foehrden-Barl a larger learning rate leads to better results. The range of the model quality of the ensemble model is much smaller than the range of quality for base models.

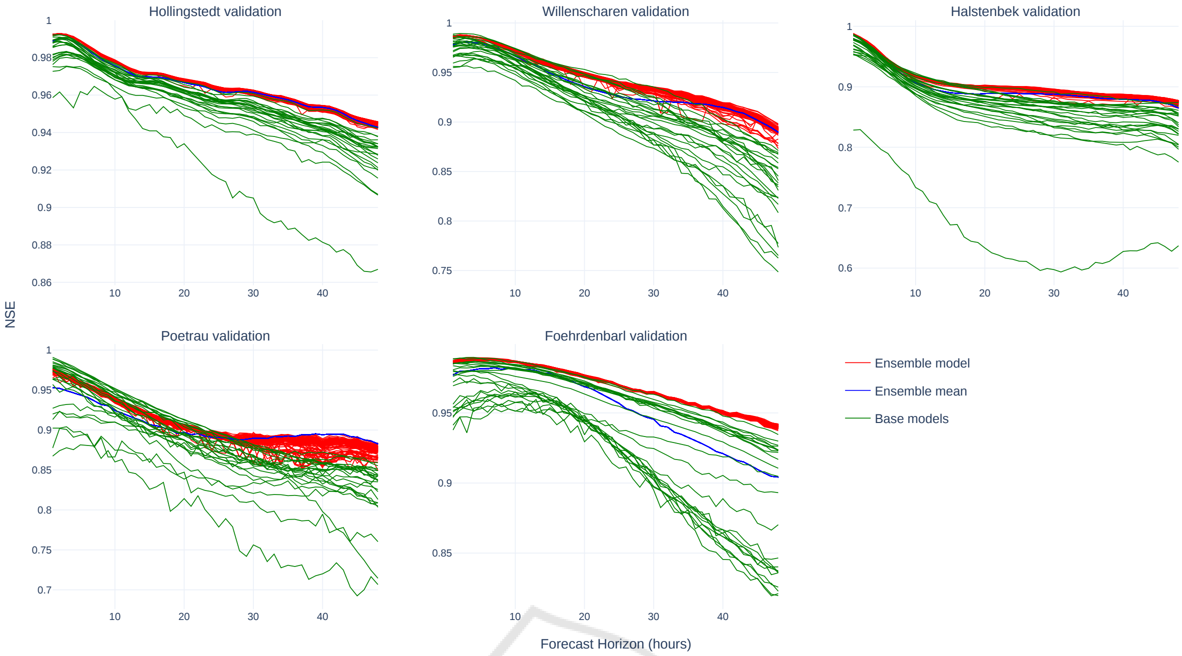

Figure 3: NSE for ensemble models with fixed base model set, mean of these sets and base models.

Table 3: Constraints hyperparameter optimization.

Parameter	Value range
Learning rate	$0.00001 \leq 0.01$
Hidden layers	$0 \leq n \leq 2$
Hidden layer units	$32 \leq n \leq 512$
Normalization function	Linear normalization or Softmax

5.2 Experiment 2: Random Model Set, Static Hyperparameters

For this experiment, we trained 100 ensemble models with a random set of 10 base models each. The hyperparameters for our ensemble models can be seen in Table 4. Gradient Clipping was used for improved learning.

Table 4: Ensemble model fixed parameters.

Parameter	Value
Learning rate	0.002
Loss function	MSE
Optimizer	Adam
Batch size	2048
Max Epochs	100
Dropout	0.25
Hidden layers	2
Hidden layer units	512

Figure 5 shows the WAPE for all five datasets at different forecast horizons. The mean WAPE of the ensemble models is lower than the mean WAPE of the base models for all datasets and forecasts horizons and lower than the mean WAPE of the naive model ensembles for all datasets and horizons except Poetrau with 48h. In most cases, the mean WAPE of the ensemble models is better or comparable to the best base model.

5.3 Discussion

Both experiments show that our approach outperforms both the base models and the naive ensemble for most datasets and forecast horizons. The first experiment shows that there is no clear best normalization function. This combined with the comparatively bad results for dataset Poetrau in the second experiment is a sign that some hyperparameter optimization may be necessary to reliably outperform naive ensembles for some datasets. LSTM-Ensembles strongly outperform single LSTM models.

While there are several studies that investigate flood forecasting with neural networks (Tripathy and Mishra, 2023), it is not easy to directly compare these studies. There are two main reasons for this. First the used data. Authors tend to study catchments and rivers geographically close to them. Since rivers can behave very different depending on for example size and mean catchment slope many metrics are not directly comparable. A low MAE at a large stream is

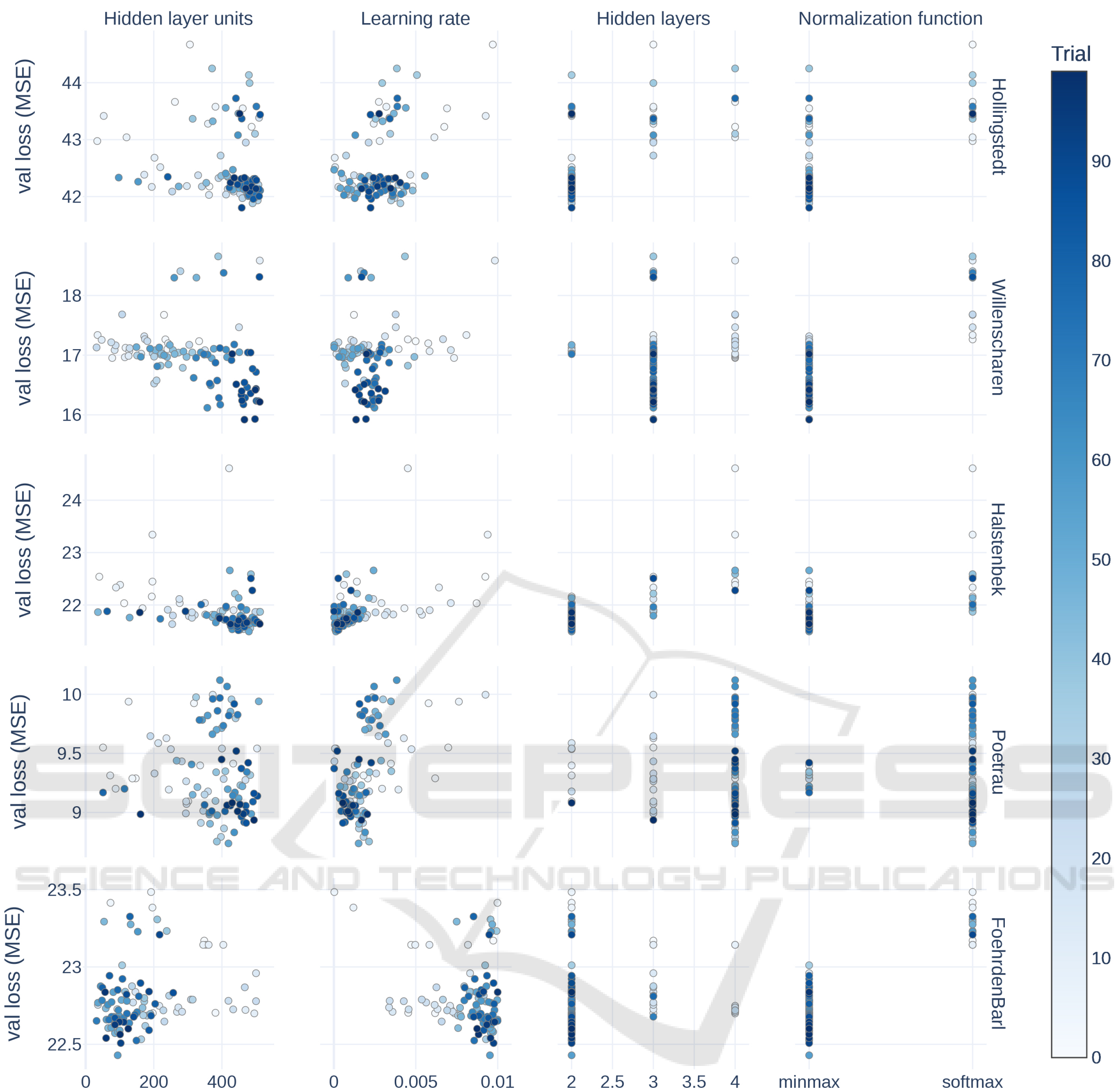

Figure 4: Validation loss for different datasets and hyperparameters.

much easier to achieve than in a small river. Additionally, the used data differs. Some only learn on events with heavy rainfall, other on the whole dataset. Many papers simply use precipitation and discharge stations to predict the future discharge, others include evaporation data and precipitation forecasts. Some make daily forecasts, other hourly. The second obstacle when comparing studies is the variety of metrics that are used to evaluate model quality, there is no single metric that is used in every paper. One study found that Spatio-Temporal Attention LSTM models (STA-LSTM) worked best for discharge forecasting (Ding et al., 2020). They reported coefficient of determination (R^2) values of 0.92, 0.75 and 0.84 for three different catchments. These values interestingly are the mean R^2 of lead times $1h$ to $6h$. Using this metric our ensembles with a static model set have an average of value of:

- Hollingstedt: 0.990
- Willenscharen: 0.988
- Halstenbek: 0.967
- Poetrau: 0.965
- Foehrdenbarl: 0.986

Another study, which investigated a $329km^2$ large basin in Texas, found that synced sequence input and output (SISO) LSTM is the best performing LSTM architecture for flood prediction (Li et al., 2020). They

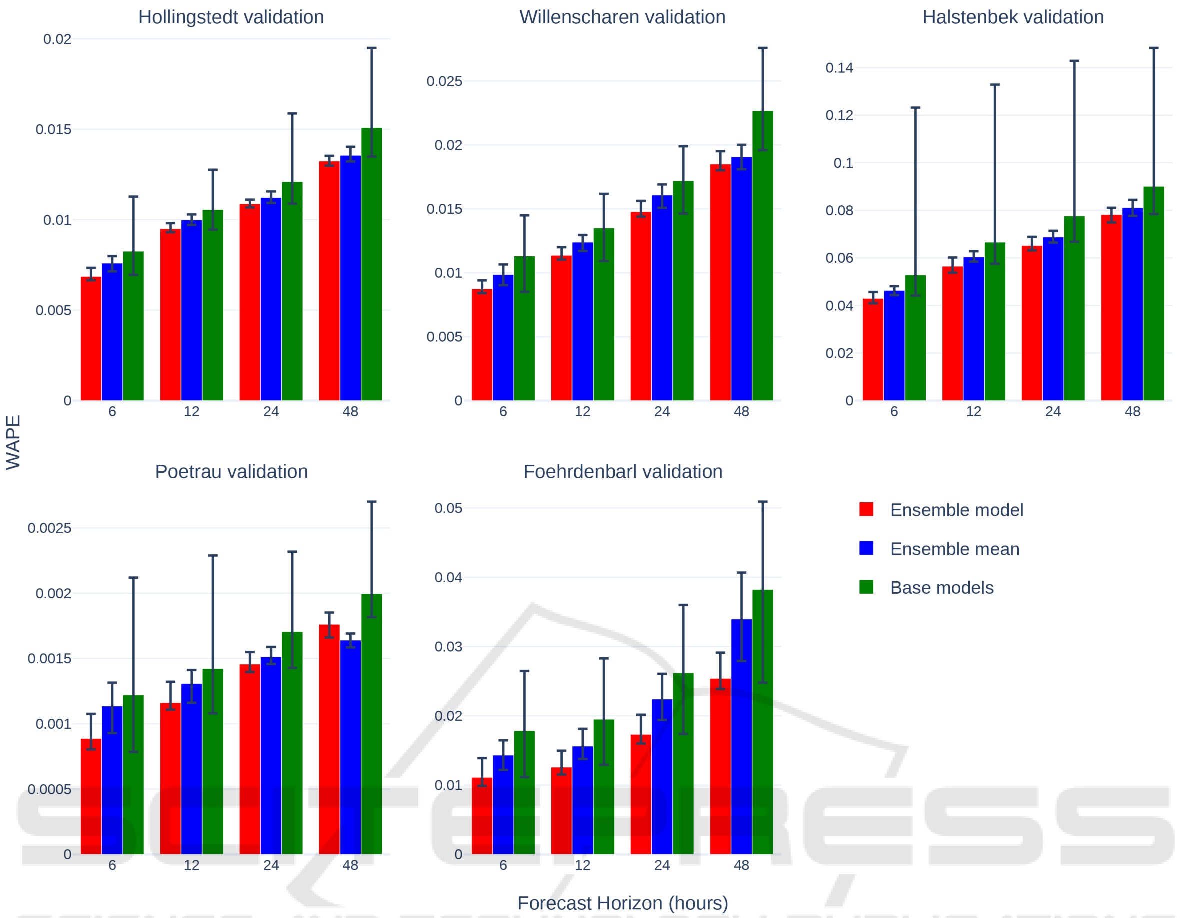

Figure 5: Mean, minimum and maximum WAPE for ensemble models with random sets of base models, mean of these sets and base models.

use rainfall and discharge gauges sampled at 15 minutes as input and report an NSE of .943, but the lead time of their forecast regrettably stays unclear. One study, which also used precipitation forecasts, found that context-aware attention LSTM (CA-LSTM) outperform regular LSTM, but are outperformed by fully connected neural networks (FCN) for lead times up to $2h$. The used data was taken from 40 flood events at a single river. They report an average root-mean-square error (RMSE) of 59.95 over up to $6h$ lead time, a metric that we did not track.

6 CONCLUSION AND FUTURE WORK

This section recaps our results and suggests some directions for further research.

6.1 Conclusion

In this paper, we developed a new method of combining multivariate multi-step forecast ensembles and tested our method on water level data. Our method outperformed naive ensembles and base learners on most datasets and forecast horizons, but we have not compared our approach to other state-of-the-art ensemble approaches, because very few investigate multivariate multi-step forecasts. The first experiment showed that the choice of hyperparameters, especially the normalization function, has some influence on the ensemble model quality but the variance is much lower than for the base models. The second experiment showed that our approach works random base model sets and usually works without any hyperparameter optimization. Training the ensemble model is very fast, taking about as long as training a single base model with an LSTM architecture. This, combined with the fact that the base models are trained independently from the ensemble model, allowing us to very efficiently build ensemble models that outperform naive ensembles and can offset bad-performing base models. Since modern machine learning approaches often undergo a hyperparameter optimization resulting in many decent, but not optimal model we can use our approach to improve from those models a nearly no cost.

6.2 Future Work

Currently, both the weighting method and the model sets are fairly naive. In future work we plan to investigate more sophisticated methods. The base models currently share the same architecture and training data and only differ in hyperparameters. Modern ensemble approaches often consider model diversity when selecting base models. Doing the same in our approach could result in a fairly large improvement since the ensemble model can only forecasts correctly if the ground truth is between minimum and maximum base forecast. We also have not yet investigated the influence of the size of our ensembles. Larger ensembles could potentially perform even better but may need different normalization functions and weighting architectures. Another direction we would like to investigate are adaptive model sets. Using the performance of base models on different benchmark data as additional input would allow us to change the set of base models. This would be useful if there is a drift in our data. With long-term drifts being common in hydrological data we intend to also extend our approach towards training and retraining models at run time instead of just weighting a static model set, thus adapting to changed environments.

ACKNOWLEDGEMENTS

The used data is mostly publicly available from DWD (German Meteorological Service) and the LfU-SH (Landesamt für Umwelt Schleswig-Holstein), kindly aggregated by the LfU-SH. This research was supported by the Federal State of Schleswig-Holstein in the context of the “KI-Förderrichtlinie” under grant 220 22 05 (project KI-WaVo).

REFERENCES

Casanova, S. and Ahrens, B. (2009). Oq. *Monthly Weather Review*, 137(11):3811–3822.

Cerqueira, V., Torgo, L., Pinto, F., and Soares, C. (2017). Arbitrated ensemble for time series forecasting. In *Machine Learning and Knowledge Discovery in Databases*, pages 478–494. Springer International Publishing.

Cerqueira, V., Torgo, L., Pinto, F., and Soares, C. (2018). Arbitrage of forecasting experts. *Machine Learning*, 108(6):913–944.

Choi, J. Y. and Lee, B. (2018). Combining LSTM network ensemble via adaptive weighting for improved time series forecasting. *Mathematical Problems in Engineering*, 2018:1–8.

Ding, Y., Zhu, Y., Feng, J., Zhang, P., and Cheng, Z. (2020). Interpretable spatio-temporal attention lstm model for flood forecasting. *Neurocomputing*, 403:348–359.

Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., and Martínez-Álvarez, F. (2019). Multi-step forecasting for big data time series based on ensemble learning. *Knowledge-Based Systems*, 163:830–841.

Gheyas, I. A. and Smith, L. S. (2011). A novel neural network ensemble architecture for time series forecasting. *Neurocomputing*, 74(18):3855–3864.

Grundmann, J., Six, A., and Philipp, A. (2020). Ensemble hydrological forecasting for flood warning in small catchments in saxony, germany.

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., and Lou, Z. (2018). Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water*, 10(11):1543.

Kao, I.-F., Zhou, Y., Chang, L.-C., and Chang, F.-J. (2020). Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, 583:124631.

Kratzert, F., Gauch, M., Nearing, G., Hochreiter, S., and Klotz, D. (2021). Niederschlags-abfluss-modellierung mit long short-term memory (lstm). *Österreichische Wasser-und Abfallwirtschaft*, 73(7-8):270–280.

Li, W., Kiaghadi, A., and Dawson, C. (2020). Exploring the best sequence lstm modeling architecture for flood prediction. *Neural Computing and Applications*, 33(11):5571–5580.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., and Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402.

Morgenstern, T., Grundmann, J., and Schütze, N. (2022). Flood forecasting with LSTM networks: Enhancing the input data with statistical precipitation information.

Mosavi, A., Ozturk, P., and Chau, K.-w. (2018). Flood prediction using machine learning models: Literature review. *Water*, 10(11):1536.

Reinert, D., Prill, F., Frank, H., Denhard, M., Baldauf, M., Schraff, C., Gebhardt, C., Marsigli, C., and Zängl, G. (2020). Dwd database reference for the global and regional icon and icon-eps forecasting system. *DWD 2023Available online: https://www. dwd. de/DWD/forschung/nwv/fepub/icon_database_main. pdf (accessed on 27 January 2023)*.

Saadallah, A. and Morik, K. (2021). Online ensemble aggregation using deep reinforcement learning for time series forecasting. In *2021 IEEE 8th International*

Conference on Data Science and Advanced Analytics (DSAA). IEEE.

Sommer, M., Stein, A., and Hähner, J. (2016). Local ensemble weighting in the context of time series forecasting using xcsf. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE.

Tripathy, K. P. and Mishra, A. K. (2023). Deep learning in hydrology and water resources disciplines: concepts, methods, applications, and research directions. *Journal of Hydrology*, page 130458.

Wee, W. J., Zaini, N. B., Ahmed, A. N., and El-Shafie, A. (2021). A review of models for water level forecasting based on machine learning. *Earth Science Informatics*, 14(4):1707–1728.

Wu, H., Xu, J., Wang, J., and Long, M. (2022). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting.