

Multidimensional cascade bagging metamodel and its online learning

Sergiy Popov^{1,*†}, Iryna Pliss^{1,†} and Yevgeniy Bodyanskiy^{1,†}

¹ Kharkiv National University of Radio Electronics, Nauky av. 14 61166, Kharkiv, Ukraine

Abstract

This paper introduces a novel, multidimensional cascade bagging system designed to work in online mode. The key feature is a cascade of simple, independently adjustable submetamodels, replacing a complex, monolithic metamodel. This approach provides computational efficiency, rapid processing, and online adaptability, allowing the system to respond to changing data characteristics without extensive retraining. Simulation results demonstrate forecasting accuracy gain and optimization potential.

Keywords

multidimensional ensemble, cascade bagging, online learning, short-term electric load forecasting

1. Introduction

The landscape of information processing has been dramatically reshaped in recent years, largely due to the widespread adoption of artificial neural networks, particularly deep learning architectures. These networks have demonstrated remarkable capabilities across a vast spectrum of applications, encompassing areas such as audio and video processing, natural language understanding, and the analysis of complex biological data. The driving force behind this success lies in their inherent universal approximating and extrapolating properties.

However, the deployment of these powerful deep learning systems brings also significant challenges. They are quite slow to set up and require very large datasets for effective training. These training data requirements can be a major obstacle when addressing real-world problems where data acquisition is costly, time-consuming, or simply unavailable in sufficient quantities. Traditional, shallower neural networks offer a more rapid development cycle, particularly those employing bell-shaped kernel activation functions. However, they are often hampered by the “curse of dimensionality,” a phenomenon where performance degrades exponentially as the number of input features increases.

Recognizing these limitations, researchers have explored hybrid approaches that combine the strengths of different computational intelligence paradigms. Neuro-fuzzy systems, for instance, offer a promising avenue for tackling a wide class of Data Stream Mining tasks. These systems integrate the learning capabilities of neural networks with the interpretability and knowledge representation capabilities of fuzzy logic. Despite their potential, neuro-fuzzy systems also face their own set of shortcomings, which can restrict their applicability.

Often it is the case that the same problem can be addressed using several different computational intelligence systems. Selecting the most suitable system for a specific application can be a non-trivial endeavor, particularly when dealing with data streams – sequences of data arriving one observation at a time. These data streams are frequently non-stationary, meaning their underlying statistical properties change over time, further complicating the selection process.

* *AISSLE-2025: International Workshop on Applied Intelligent Security Systems in Law Enforcement, October, 30–31, 2025, Vinnytsia, Ukraine*

^{1*} Corresponding author.

[†] These authors contributed equally.

✉ serhii.popov@nure.ua (S. Popov); iryna.pliss@nure.ua (I. Pliss); yevgeniy.bodyanskiy@nure.ua (Y. Bodyanskiy)

ORCID 0000-0002-1274-5830 (S. Popov); 0000-0001-7918-7362 (I. Pliss); 0000-0001-5418-2143 (Y. Bodyanskiy)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Under these conditions, the ensemble approach [1–4] becomes a particularly attractive solution. This methodology involves tackling a single problem using multiple, diverse computational intelligence systems, and then combining their individual outputs to produce a final, refined result. Here the problem arises of how to combine the ensemble members outputs in order to obtain a final result that is optimal in some sense.

From a mathematical perspective, the bagging approach [5] is often considered the most effective technique for this combining process. In bagging, the output signals from all ensemble members are fed into the inputs of a so-called metamodel. This metamodel then processes these signals, generating a final result that represents a synthesis of the individual member contributions. A key advantage of this approach is the possibility of its implementation in online mode [6–9], allowing for real-time processing of sequential data with potentially non-stationary characteristics. However, when the output signals of the ensemble members are multidimensional sequences, this approach becomes complicated. In such cases, the signal presented to the metamodel can have a very high dimension, making it difficult to tune the metamodel and potentially slowing down its processing speed.

To mitigate this complexity, the cascade systems ideas [10–12] can be employed to simplify the implementation and functioning of the metamodel. Cascade systems break down the processing into a series of interconnected stages, where each stage operates on a smaller number of input signals. This modular approach reduces the overall computational burden and improves the efficiency of the system. Furthermore, the cascade approach has proven successful in the development of several hybrid computational intelligence systems, demonstrating its versatility and effectiveness.

Consequently, within the framework of this cascade bagging approach, the ensemble metamodel is structured as a set of submetamodels. Each submetamodel is fed with the output signal from the preceding cascade along with the output signal from one of the individual ensemble members. This cascading arrangement allows for a progressive refinement of the combined output, ultimately leading to a more accurate and robust final result.

2. Architecture of multidimensional cascade bagging system

Fig 1. shows the architecture of the proposed multidimensional ensemble bagging system consisting of p ensemble subsystems $ES_1, ES_2, \dots, ES_r, \dots, ES_p$ connected in parallel and $p-1$ bagging submetamodels $SMM_2, \dots, SMM_r, \dots, SMM_p$.

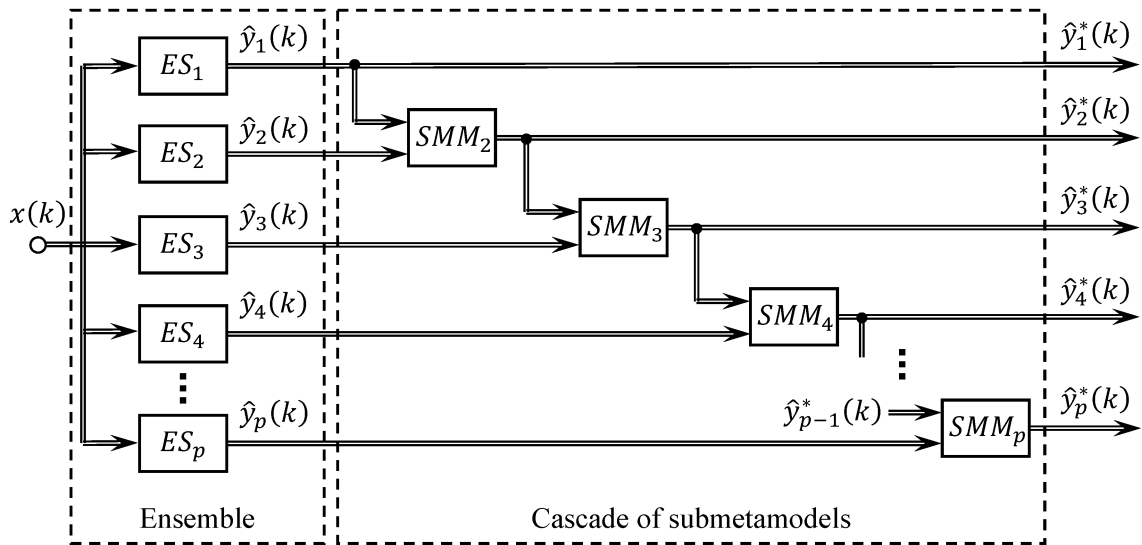


Figure 1: Architecture of multidimensional cascade bagging system

The ensemble contains p subsystems (ensemble members) solving the same problem. They receive the same vector input signal $x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T \in R^n$ (here $k=1,2,\dots$ is the current discrete time) and output a set of vector output signals $\hat{y}_1(k), \dots, \hat{y}_r(k), \dots, \hat{y}_p(k)$, where $\hat{y}_r(k) = (\hat{y}_{r1}(k), \dots, \hat{y}_{rj}(k), \dots, \hat{y}_{rm}(k))^T \in R^m$. These signals are fed to a set of cascaded submetamodels $SMM_2, \dots, SMM_r, \dots, SMM_p$ which form the bagging system outputs $\hat{y}_2^*(k), \dots, \hat{y}_r^*(k), \dots, \hat{y}_p^*(k)$.

In the simplest case, if the output signal of the first ensemble member ES_1 meets the *a priori* given accuracy target, submetamodels are not utilized and the ES_1 output is used as the ensemble output $\hat{y}_1^*(k) = \hat{y}_1(k)$. If the accuracy of $\hat{y}_1(k)$ is not satisfactory, then this signal together with the output of ES_2 $\hat{y}_2(k)$ are fed into submetamodel SMM_2 which forms the signal $\hat{y}_2^*(k)$ as

$$\hat{y}_2^*(k) = c_2 \hat{y}_2(k) + (1 - c_2) \hat{y}_1^*(k) = c_2 \hat{y}_2(k) + (1 - c_2) \hat{y}_1(k), \quad (1)$$

where $0 \leq c_2 \leq 1$ is a tuned parameter.

The signal $\hat{y}_2^*(k)$ should be more accurate than $\hat{y}_1(k)$. The same cascade principle applies to all subsequent submetamodels until the signal of the r -th submetamodel SMM_r

$$\hat{y}_r^*(k) = c_r \hat{y}_r(k) + (1 - c_r) \hat{y}_{r-1}^*(k), \quad 0 \leq c_r \leq 1 \quad (2)$$

meets the *a priori* given accuracy target or all ensemble members p are used.

The advantage of this approach is that each cascade has only one tuned parameter c_r , $r=2,3,\dots,p$ that greatly simplifies the whole metamodel tuning process.

3. Fast online cascade metamodel learning

The cascade bagging metamodel learning process involves continuous calculation of the parameters

$c_2, \dots, c_r, \dots, c_p$ to achieve and maintain the desired accuracy level for the task at hand

Let's introduce a vector learning error of the r -th submetamodel SMM_r

$$e_r(k) = y(k) - \hat{y}_r^*(k) = y(k) - \hat{y}_{r-1}^*(k) - c_r (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)) = e_{r-1}(k) - c_r (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)), \quad (3)$$

where $y(k)$ is a m -dimensional reference signal, and its squared norm

$$\|e_r(k)\|^2 = \|e_{r-1}(k)\|^2 - 2c_r e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)) + c_r^2 \|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2. \quad (4)$$

Summing up squared errors over the whole dataset

$$\sum_k \|e_r(k)\|^2 = \sum_k \|e_{r-1}(k)\|^2 - 2c_r \sum_k e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)) + c_r^2 \sum_k \|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2, \quad (5)$$

and solving for

$$\frac{\partial \sum_k \|e_r(k)\|^2}{\partial c_r} = -2 \sum_k e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)) + 2c_r \sum_k \|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2 = 0, \quad (6)$$

we arrive at

$$c_r = \frac{\sum_k e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k))}{\sum_k \|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2}. \quad (7)$$

For nonstationary situations, the submetamodels parameters calculations should be performed on a sliding window. Let s be a sliding window size, then

$$c_r(k, s) = \frac{\sum_{\tau=k-s+1}^k e_{r-1}^T(\tau) (\hat{y}_r(\tau) - \hat{y}_{r-1}^*(\tau))}{\sum_{\tau=k-s+1}^k \|\hat{y}_r(\tau) - \hat{y}_{r-1}^*(\tau)\|^2}. \quad (8)$$

When $s = 1$, we obtain a single-step learning rule in the following form

$$c_r(k) = \frac{e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k))}{\|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2}. \quad (9)$$

It is easily seen that

$$\begin{aligned} \|e_r(k)\|^2 &= \|e_{r-1}(k)\|^2 - 2 \frac{(e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)))^2}{\|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2} + \\ &+ \frac{(e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)))^2 \|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2}{\|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^4} = \|e_{r-1}(k)\|^2 - \frac{(e_{r-1}^T(k) (\hat{y}_r(k) - \hat{y}_{r-1}^*(k)))^2}{\|\hat{y}_r(k) - \hat{y}_{r-1}^*(k)\|^2}, \end{aligned} \quad (10)$$

hence

$$\|e_r(k)\|^2 \leq \|e_{r-1}(k)\|^2. \quad (11)$$

This means that each submetamodel SMM_r in the cascade system is not inferior in terms of accuracy than the preceding one SMM_{r-1} .

The process of adding more cascades can continue until the desired accuracy is achieved or the maximum number of submetamodels p is reached. If the desired accuracy is achieved with a smaller number of submetamodels than currently employed r , the “excess” submetamodels can be removed to conserve computational resources. Both these processes of adding and removing submetamodels don’t require retraining the preceding submetamodels.

4. Simulation results

To evaluate the efficacy of the proposed cascade bagging system, we applied it to the challenging problem of short-term electric load forecasting (STLF). STLF is a critical task for regional power system operators, enabling efficient resource allocation and grid stability. To demonstrate the system’s capabilities, we utilized a real-world dataset comprising hourly electric load data collected over a one-year period at $m=4$ geographically distinct nodes within a regional power system in Ukraine. This dataset consists of $N=8760$ observations, each represented as a 4-element vector, representing four interdependent time series reflecting the electric load at each node.

The inherent complexity of the STLF problem stems from the stochastic nature of electricity demand. Inspection of the dataset (Fig. 2) revealed several key characteristics that pose significant forecasting challenges. The individual vector components – representing the load at each node – exhibit a combination of common trends, unique temporal patterns, and abrupt shifts in behavior.

The data is also prone to outliers, representing unexpected surges or drops in demand, and is inherently subject to noise, arising from random influences. These complexities necessitate a robust forecasting methodology capable of capturing the nuanced relationships within the data. We believe an ensemble approach, particularly our cascade bagging system, offers a promising solution to mitigate these challenges.

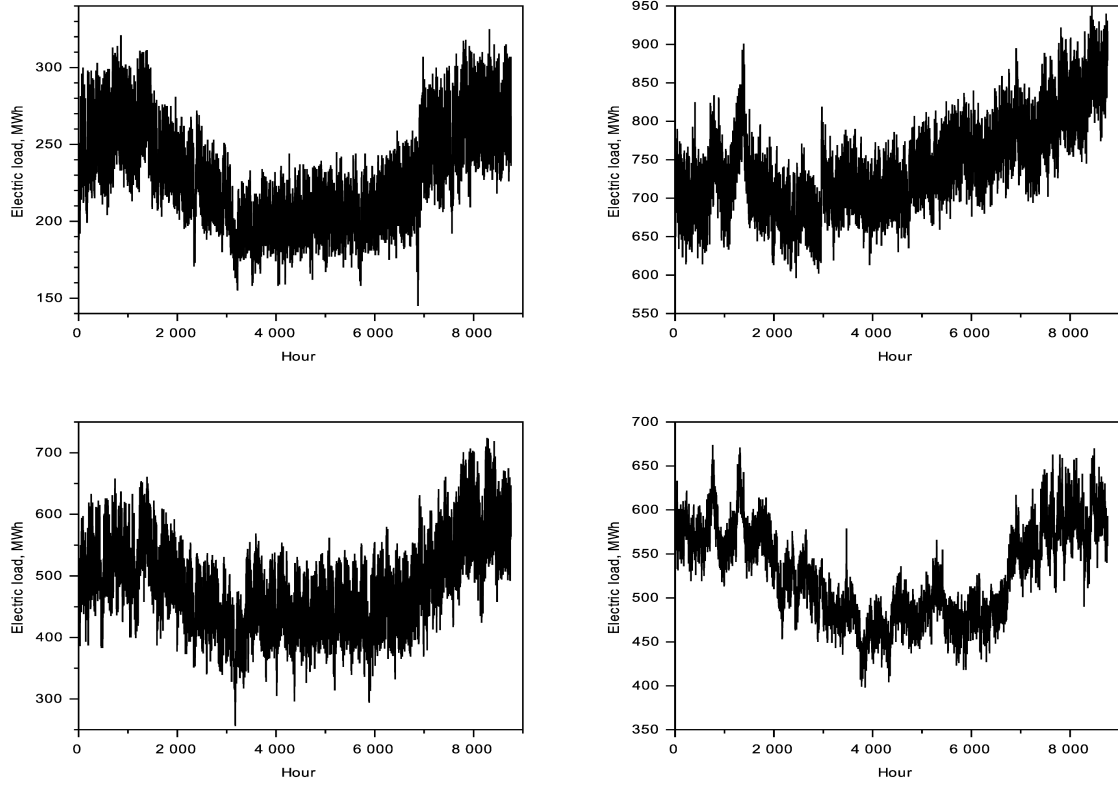


Figure 2: Hourly electric load recorded at 4 power system nodes during 1 year

For the 24-hour ahead forecasting task, we deployed $p=3$ specialized Neuro-Fuzzy Network (NFN) models as ensemble subsystems. NFNs are a class of Hybrid Systems of Computational Intelligence, known for their ability to combine the strengths of neural networks (learning complex patterns) and fuzzy logic (handling uncertainty and expert knowledge) [13]. Each of these ensemble subsystems was carefully tailored for STLTF problems, featuring distinct architectures and hyperparameter settings, we refer to them as ensemble subsystems ES_1 - ES_3 for clarity. This intentional variation in model structure and parameters was implemented to encourage diversity in the ensemble, enabling each subsystem to potentially capture different aspects of the dataset's intricacies and ultimately contributing to a more comprehensive forecast. The specifics of the NFN design and parameter optimization for the STLTF problem are beyond the scope of this paper, we refer readers to the corresponding sources [14, 15].

Fig. 3–5 present the 24-hour ahead forecasting results for a 168-hour period, generated by each of the corresponding ensemble subsystems. As anticipated, the individual subsystems exhibited distinct forecasting behaviors, reflecting their differing structures and parameter settings. A quantitative comparison of forecasting errors, detailed in Table 1, further confirms this diversity. To assess forecasting accuracy, we employed Mean Absolute Percentage Error (MAPE), a standard error measure widely used in STLTF applications, calculated over the entire dataset. The diversity observed in the individual subsystem performance highlights the potential for the cascade bagging system to leverage these differences to generate a more accurate and robust forecast.

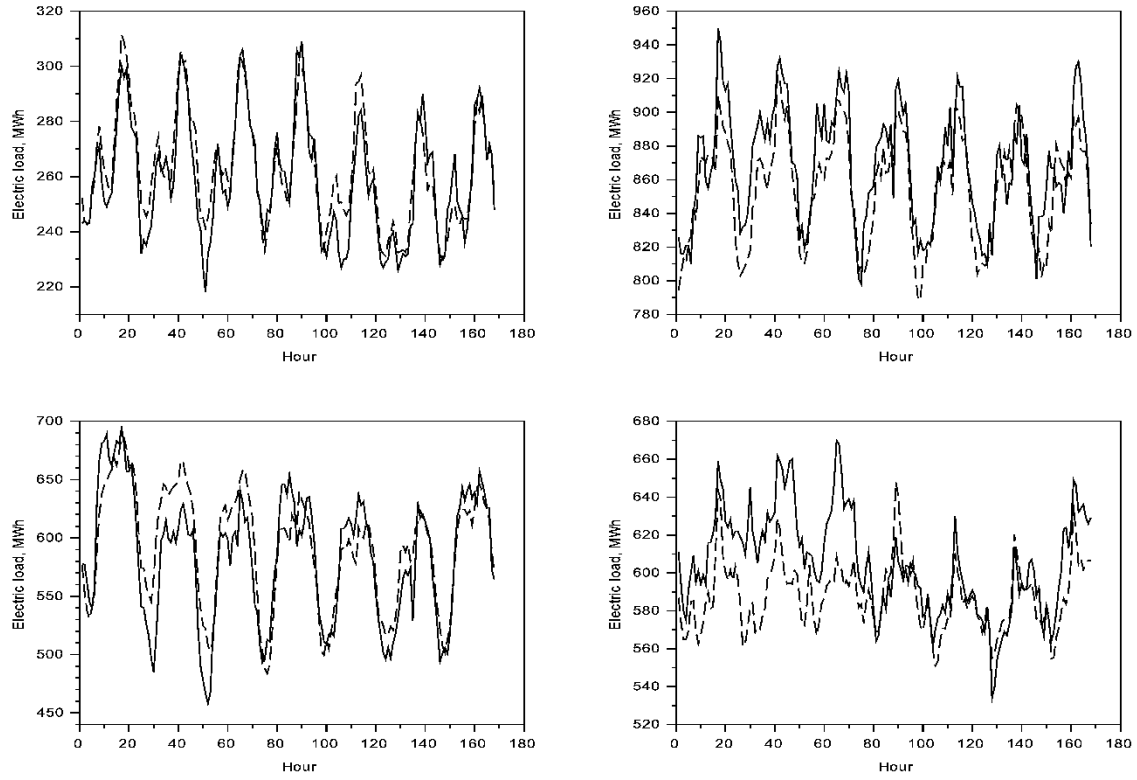


Figure 3: True electric load (solid line) and its 24 hours ahead forecast (dashed line) produced by ensemble subsystem ES_1 for 4 power system nodes over a 168 hours period

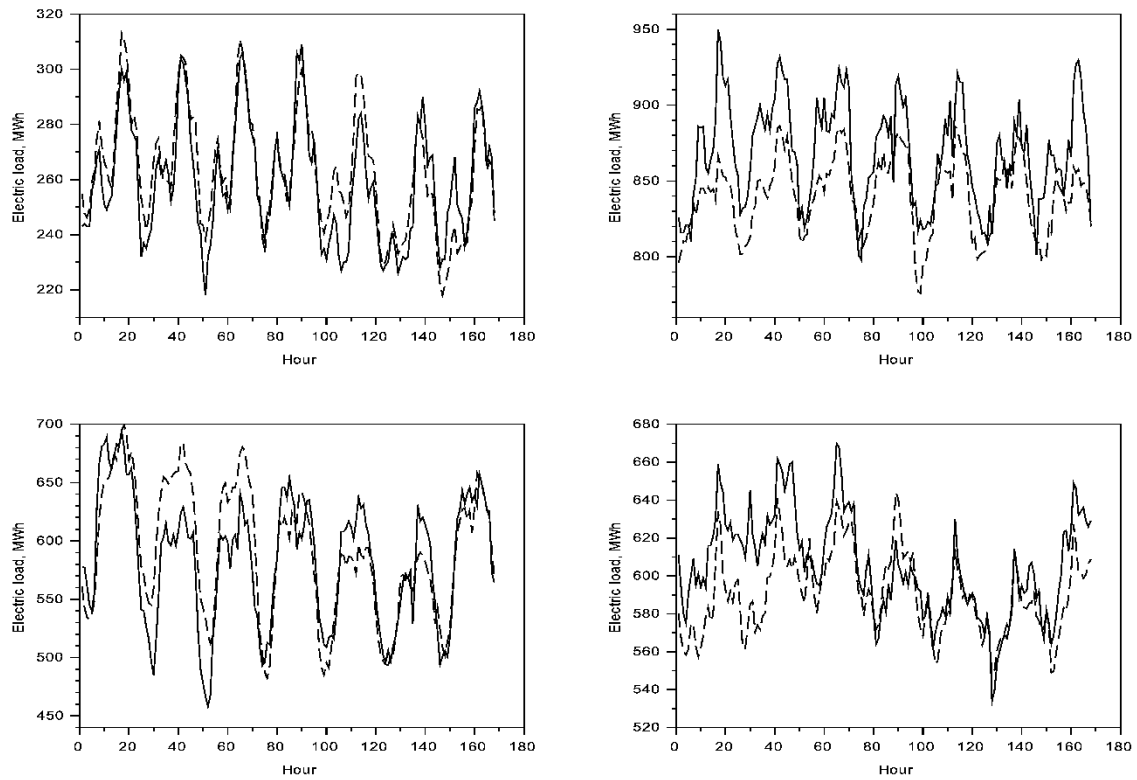


Figure 4: True electric load (solid line) and its 24 hours ahead forecast (dashed line) produced by ensemble subsystem ES_2 for 4 power system nodes over a 168 hours period

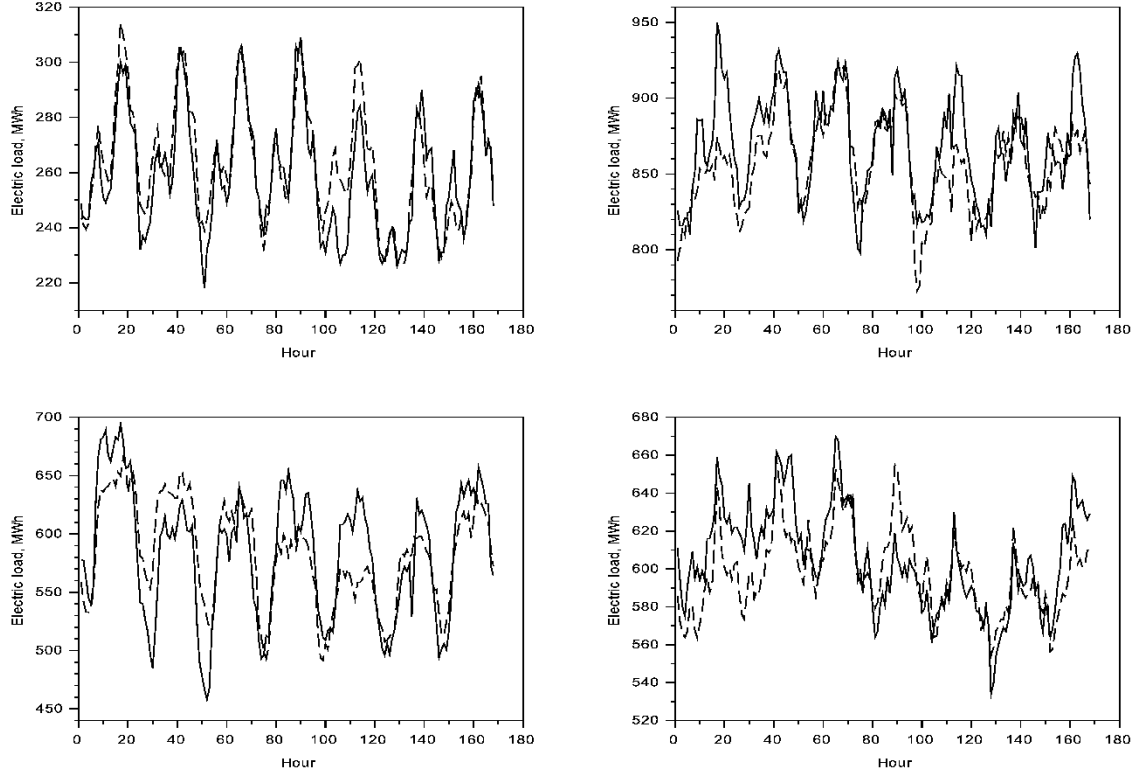


Figure 5: True electric load (solid line) and its 24 hours ahead forecast (dashed line) produced by ensemble subsystem ES_3 for 4 power system nodes over a 168 hours period

We applied the proposed cascade approach to combine the ensemble subsystems' outputs. As we have 3 ensemble subsystems, 2 submetamodels are sufficient in the cascade structure according to the architecture shown in Fig. 1. The output signals are produced using relation (2). The data is treated in online mode, i.e. data vectors are processed sequentially, one-by-one and only once, as a data stream. This eliminates the need to divide the dataset into training, validation, and test sets.

In order to be able to adapt the cascade combining process to changing properties of the signals over time (trend shifts, unexpected fluctuations, etc.), we used the sliding window version of the proposed learning algorithm (8). Choosing a proper value for the sliding window size s , provides a reasonable tradeoff between the smoothing and following properties of the learning process. It should be noted that a suitable meta-algorithm could be implemented to dynamically adjust this parameter during the learning process, based on real-time monitoring of learning errors, further optimizing the system's responsiveness and accuracy.

The outputs of submetamodels SMM_2 and SMM_3 are displayed in Fig. 6 and Fig. 7 respectively. The corresponding errors are presented in Table 1. As we are dealing with vector signals, the first 4 rows of Table 1 represent errors for each of the 4 components, and the last row contains the errors averaged over the whole vector as a cumulative metric. And finally, for the reference purpose, the last column in Table 1 lists errors for a simple averaging ensemble approach widely used in bagging procedures.

Let's now examine the data and make some observations.

While ES_3 exhibits a lower average error compared to ES_2 , which in turn is better than ES_1 , this ranking does not universally hold for all individual vector components. Notably, ES_1 demonstrates the highest accuracy for the first component; ES_3 outperforms the others for the second and third components; and ES_2 is the most accurate for the fourth component. This confirms the diversity and complementary strengths of the individual ensemble subsystems.

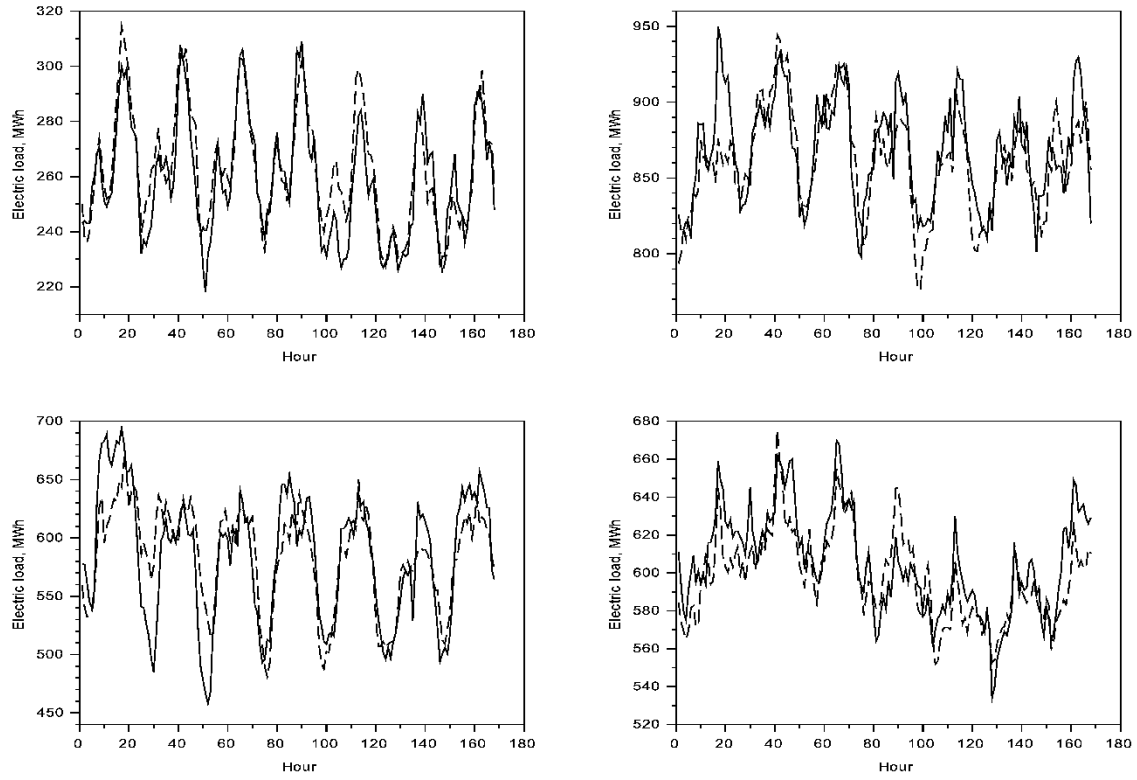


Figure 6: True electric load (solid line) and its 24 hours ahead forecast (dashed line) produced by submetamodel SMM_2 for 4 power system nodes over a 168 hours period

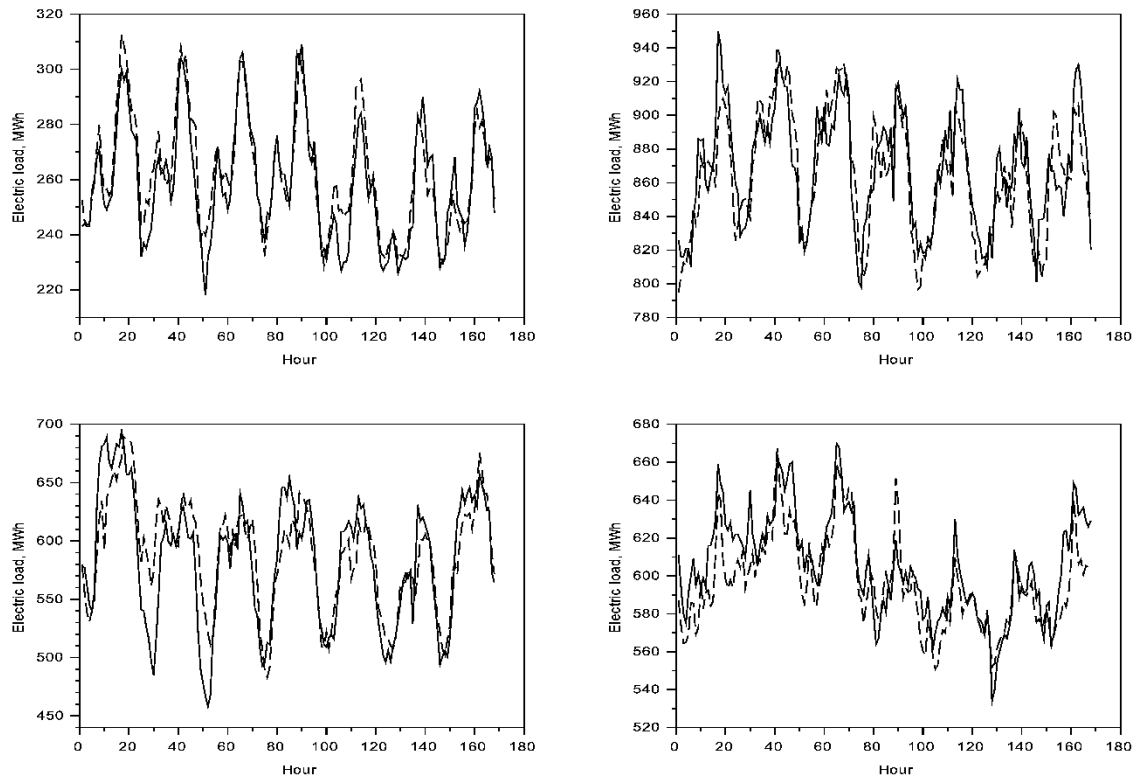


Figure 7: True electric load (solid line) and its 24 hours ahead forecast (dashed line) produced by submetamodel SMM_3 for 4 power system nodes over a 168 hours period

Table 1

24 hours ahead forecasting errors for all ensemble subsystems and submetamodels

	ES_1	ES_2	ES_3	SMM_2	SMM_3	Simple avg.
$MAPE_1$	3.6932%	3.9689%	3.7272%	3.5387%	3.4826%	3.5942%
$MAPE_2$	2.8699%	2.7779%	2.7703%	2.5458%	2.4422%	2.6663%
$MAPE_3$	5.3788%	5.3016%	5.1585%	4.7630%	4.6069%	5.0321%
$MAPE_4$	3.0220%	2.6665%	2.7657%	2.4238%	2.3439%	2.6565%
Avg. MAPE	3.7410%	3.6787%	3.6054%	3.3178%	3.2189%	3.4873%

Submetamodel SMM_2 (which combines the outputs of ES_1 and ES_2) consistently outperforms all ensemble subsystems both in terms of the average error and for each component individually. In case this level of accuracy would be sufficient for the task at hand, the ES_3 and SMM_3 could be removed from the system to reduce computational load.

Submetamodel SMM_3 (which combines the outputs of SMM_2 and ES_3) further reduces all error metrics in comparison to SMM_2 , validating the theoretical prediction outlined in relation (11). This incremental improvement underscores the effectiveness of the cascading structure.

The simple averaging approach (used here only for reference) provides more accurate forecasts than any of the ensemble subsystems alone. However, the proposed adaptive cascade procedure consistently outperforms the simple averaging starting already from the first submetamodel. This demonstrates the effectiveness of the adaptive learning algorithm.

Therefore, the simulation results conclusively confirm the effectiveness of the proposed adaptive cascade ensemble approach, showcasing its ability to generate superior forecasts compared to individual models and a basic averaging ensemble. The ability to adapt to changing signal characteristics and the potential for computational optimization further increase the practical utility of this approach.

5. Conclusions

Traditional bagging systems, particularly those dealing with multidimensional outputs, often face challenges in scalability and adaptability. The complexity of a single metamodel tasked with integrating the outputs of numerous ensemble members can become computationally expensive, especially when dealing with high-dimensional, non-stationary data streams. To address these limitations, we introduced a novel multidimensional bagging system that leverages a cascade of simple, independently configurable submetamodels. This approach prioritizes computational simplicity, high processing speed, and online adaptability.

The core innovation of our system lies in replacing the monolithic metamodel with a sequence of interconnected, computationally lightweight submetamodels arranged in a cascade structure. Each submetamodel is configured online, i.e. its single parameter can be adjusted dynamically in response to changes in the data stream. The independent learning of each submetamodel allows them to adapt to changing data characteristics without requiring global retraining of the entire system. The computational simplicity and high speed make the proposed cascade system particularly well-suited for real-time applications where rapid adaptation to changing conditions is paramount. Furthermore, the modularity of the system facilitates easy expansion and modification, allowing for seamless integration of new ensemble members or the incorporation of more sophisticated submetamodels as computational resources become available. This adaptability allows the system to remain effective even as data characteristics and application requirements evolve.

Simulation results conclusively proved the proposed method's ability to generate more accurate forecasts and offer potential for computational optimization, enhancing its practical utility.

Namely, SMM_2 consistently outperformed the individual subsystems, and SMM_3 further reduced forecasting errors, validating the cascade structure's effectiveness. The adaptive learning algorithm, incorporating a sliding window, allows the system to respond to changing signal characteristics, while keeping the balance between tracking and smoothing behavior.

Future work will focus on exploring various online learning algorithms for submetamodels tuning and investigating the optimal cascade depth for different data characteristics.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

- [1] C. Zhang, Y. Ma, Ensemble machine learning: Methods and applications, Springer, 2012.
- [2] D. Sarkar, V. Natarajan, Ensemble Machine Learning Cookbook, Packt Publishing Limited, 2019.
- [3] G. Kunapuli, Ensemble methods for machine learning, Simon and Schuster, 2023.
- [4] X. Dong, Z. Yu, W. Cao, et al, A survey on ensemble learning. Front. Comput. Sci. 14 (2020) 241-58. <https://doi.org/10.1007/s11704-019-8208-z>
- [5] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 126–140. <https://doi.org/10.1007/BF00058655>
- [6] A. Bifet, G. Holmes, B. Pfahringer, R. Gavaldà, Improving Adaptive Bagging Methods for Evolving Data Streams, in: Z. H. Zhou, T. Washio (Eds) Advances in Machine Learning. ACML 2009. Lecture Notes in Computer Science, vol 5828. Springer, Berlin, Heidelberg, 2009.. https://doi.org/10.1007/978-3-642-05224-8_4
- [7] E. Lughofer, M. Pratama, I. Skrjanc, Online bagging of evolving fuzzy systems, Information Sciences 570 (2021) 16–33. <https://doi.org/10.1016/j.ins.2021.04.041>
- [8] Ye. Bodyanskiy, P. Otto, I. Pliss, S. Popov, An Optimal Algorithm for Combining Multivariate Forecasts in Hybrid Systems, in: V. Palade, R.J. Howlett, L. Jain (Eds) Knowledge-Based Intelligent Information and Engineering Systems. KES 2003. Lecture Notes in Computer Science, vol 2774, Springer, Berlin, Heidelberg, 2003. https://doi.org/10.1007/978-3-540-45226-3_132
- [9] Ye. Bodyanskiy, S. Popov, Fuzzy Selection Mechanism for Multimodel Prediction, in: M.G. Negoita, R.J. Howlett, L.C. Jain (Eds) Knowledge-Based Intelligent Information and Engineering Systems. KES 2004. Lecture Notes in Computer Science, vol 3214. Springer, Berlin, Heidelberg, 2004. https://doi.org/10.1007/978-3-540-30133-2_101
- [10] F.A. Shah, M.A. Khan, M. Sharif et. al., A cascaded design of best features selection for fruit diseases recognition, Comput. Mater. Contin 70(1) (2022) 1491-1507. <https://doi.org/10.32604/cmc.2022.019490>
- [11] Ye. Bodyanskiy, O. Tyshchenko, A Hybrid Cascade Neuro-Fuzzy Network with Pools of Extended Neo-Fuzzy Neurons and its Deep Learning, International Journal of Applied Mathematics and Computer Science 29(2) (2019) 477-488. <https://doi.org/10.2478/amcs-2019-0035>
- [12] J. Rao, B. Li, Z. Zhang, D. Chen, W. Giernacki, Position Control of Quadrotor UAV Based on Cascade Fuzzy Neural Network, Energies 15(5) (2022) 1763. <https://doi.org/10.3390/en15051763>
- [13] N. Talpur, S.J. Abdulkadir, H. Alhussian et al. A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods, Neural Comput & Applic 34 (2022) 1837-1875 <https://doi.org/10.1007/s00521-021-06807-9>
- [14] P. Chernenko, O. Martyniuk, S. Popov, Ye. Bodyanskiy, Comparative analysis of two approaches to solving the problem of short-term forecasting of the total electrical load of a power system, Technical Electrodynamics 3, (2013) 61–72.

- [15] Ye. Bodyanskiy, S. Popov, M. Titov, Robust Learning Algorithm for Networks of Neuro-Fuzzy Units, in: T. Sobh (Eds) Innovations and Advances in Computer Sciences and Engineering. Springer, Dordrecht, 2010. https://doi.org/10.1007/978-90-481-3658-2_59