Adaptive neo-fuzzy bagging metamodel and its fast learning

Yevgeniy Bodyanskiy[1†], Ivan Izonin[2†], Iryna Pliss[1†], Olha Chala[1†] and Sergiy Popov[1*†]

[1] *Kharkiv National University of Radio Electronics, 14 Nauky av., Kharkiv, 61166, Ukraine*
[2] *Lviv Polytechnic National University, 12 Stepana Bandery St, Lviv, 79000, Ukraine*

Abstract

Adaptive nonlinear neo-fuzzy bagging metamodel and its fast online learning procedure are proposed intended for optimal combining of the results of different computational intelligence systems that simultaneously solve the same problem. It is assumed that the data is processed by the members of the ensemble and by the metamodel online in real time. The proposed metamodel is intended for solving a wide class of Data Stream Mining problems under the conditions of data non-stationarity, and when the processing speed is of the utter importance. Simulation based on the short-term electric load forecasting problem confirmed theoretical results. The metamodel demonstrated significant improvement over the results of the member models.

Keywords

adaptive nonlinear bagging, neo-fuzzy metamodel, optimal combining, fast online learning

1. Introduction

Recently, computational intelligence has emerged as a powerful tool across various fields, particularly in Data Mining. It encompasses a variety of systems such as artificial neural networks, fuzzy systems, neuro-fuzzy systems, and neo-fuzzy systems. These systems are employed to tackle an extensive array of tasks. Just to mention a few:

- Pattern Recognition: Computational intelligence is widely used in image recognition, where systems can identify features and classify images with high accuracy.
- Time Series Processing: These systems excel in forecasting future trends based on historical data, which is crucial in financial markets, weather prediction, and economic planning.
- Natural Language Processing (NLP): From sentiment analysis to machine translation, these systems process human language effectively, enhancing communication technologies.
- Diagnostics and Fault Detection: In healthcare and engineering sectors, computational intelligence aids in identifying anomalies and predicting potential failures in machinery or patient health conditions.

One of the key strengths of computational intelligence systems lies in their universal approximating capabilities. These properties enable them to model complex relationships within data effectively. Furthermore, their ability to learn by adjusting parameters based on training data makes them highly adaptable to various tasks.

ISW-2025: Intelligent Systems Workshop at 9th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2025), May 15–16, 2025, Kharkiv, Ukraine

[*] Corresponding author.
[†] These authors contributed equally.

yevgeniy.bodyanskiy@nure.ua (Ye. Bodyanskiy); ivan.v.izonin@lpnu.ua (I, Izonin); iryna.pliss@nure.ua (I. Pliss); olha.chala@nure.ua (O. Chala); serhii.popov@nure.ua (S. Popov)
0000-0001-5418-2143 (Ye. Bodyanskiy); 0000-0002-9761-0096 (I, Izonin); 0000-0001-7918-7362 (I. Pliss); 0000-0002-7603-1247 (O. Chala); 0000-0002-1274-5830 (S. Popov)

On the other hand, every task can be addressed using various computational intelligence systems, each with distinct architectures, training principles, data requirements, and operational speeds, which leads to different data processing results. For instance:

- Artificial Neural Networks (ANNs): Known for their high accuracy, especially deep neural networks, they require vast volumes of training samples (which may not always be feasible due to data scarcity) and computational resources, i.e. a lot of time and computing power for their training in multi-epoch mode.
- Fuzzy Systems: These systems are known for their ability to handle imprecise or incomplete data, making them suitable for real-world problems where data may be uncertain.
- Neuro-Fuzzy Systems: Combining the strengths of neural networks and fuzzy logic, these systems offer both learning capabilities and robustness to noisy data.

Selecting the appropriate computational intelligence system for a given problem is an intricate task that lacks a formal solution, primarily relying on the empirical knowledge of researchers and occasional intuition. This process is not trivial due to the vast array of systems available, each with its own strengths and weaknesses. To address this challenge, one promising approach is bagging [1], where an ensemble of different systems [2, 3] is employed. These systems operate concurrently but independently to solve the same problem. Their outputs are then combined through a metamodel – a higher-level model that integrates these signals into an optimal solution. Typically, this integration is achieved through a linear combination of individual system outputs.

However, nonlinear approaches to bagging have been largely overlooked, with only limited research available in this domain [4]. Most existing solutions operate in batch mode, processing data offline, which can be restrictive in dynamic environments where real-time decision-making is crucial. Adaptive systems such as [5, 6] that function online offer a potential solution to this limitation, capable of responding and adapting as new data arrives.

The aggregation of system outputs poses another challenge. While linear combinations have been extensively studied [7-9], achieving optimal adaptive combinations remains an open area for research. Introduction of adaptive fast-acting nonlinear metamodels could significantly enhance the quality of results, yet such innovations are still in their infancy and not widely adopted.

In light of these considerations, we propose a novel approach using the neo-fuzzy methodology [10-12], tailored to meet the demands of adaptive nonlinear bagging. This method is designed for real-time environments where both adaptability and nonlinearity are essential. By modifying the neo-fuzzy framework specifically for this purpose, we aim to bridge existing gaps in system selection and integration, offering a more robust solution to complex problems.

In summary, while choosing the right computational intelligence system remains challenging due reliance on experience and intuition, innovative approaches like bagging ensembles and adaptive metamodels offer promising pathways. The exploration of nonlinear solutions, particularly through neo-fuzzy approaches, holds the potential to advance our ability to effectively combine diverse systems for improved problem-solving in dynamic environments.

2. Architecture of the adaptive neo-fuzzy bagging metamodel

The architecture of the adaptive neo-fuzzy bagging metamodel is illustrated in Figure 1. This system represents an enhanced version of the traditional neo-fuzzy neuron [10-12], which is distinguished by its strong approximation capabilities and the ability to adjust its parameters dynamically, or "online." At the core of this architecture lies a layer composed of nonlinear synapses, which serve as the primary element within the neo-fuzzy neuron. These synapses essentially perform an operation known as the F-transform [13], i.e. it is a universal approximator.

The metamodel receives input signals from q ensemble members that work simultaneously to solve the same problem. Each member provides its output in the form of a time sequence:

$\hat{y}_1(k), \dots, \hat{y}_p(k), \dots, \hat{y}_q(k)$, where k represents the discrete time index $k = 1,2, \dots$ These sequences are organized into a vector format denoted as $\hat{y}(k) = \left(\hat{y}_1(k), \dots, \hat{y}_p(k), \dots, \hat{y}_q(k)\right)^T$, which is then passed to the layer of nonlinear synapses.

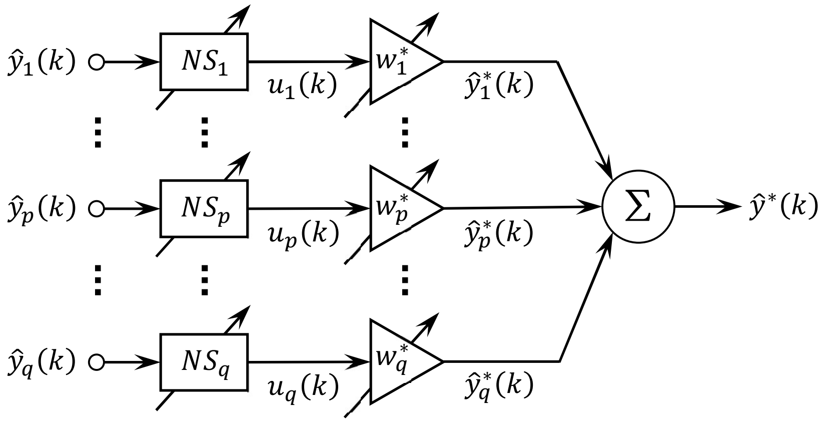

Figure 1: Adaptive neo-fuzzy bagging metamodel

Each nonlinear synapse, shown in Figure 2, consists of n nonlinear membership functions $\mu_{lp}\left(\hat{y}_p(k)\right)$, where $l = 1,2, \dots, n$ and p represents different ensemble members. These membership functions adhere to the Ruspini unity partition conditions. Additionally, each nonlinear synapse features n tunable synaptic weights w_{lp}, which are determined through a learning process tailored to optimize performance.

At the output of each of the nonlinear synapses NS_p a signal is formed

$$u_p(k) = \sum_{l=1}^{n} w_{lp}\mu_{lp}\left(\hat{y}_p(k)\right) = w_p^T \mu_p\left(\hat{y}_p(k)\right), \tag{1}$$

where $w_p = \left(w_{1p}, \dots, w_{lp}, \dots, w_{np}\right)^T$, $\mu_p\left(\hat{y}_p(k)\right) = \left(\mu_{1p}\left(\hat{y}_p(k)\right), \dots, \mu_{lp}\left(\hat{y}_p(k)\right), \dots,\right.$ $\left.\mu_{np}\left(\hat{y}_p(k)\right)\right)^T, \forall p = 1, \dots, q$.

Triangular functions in the following form are most often used as membership functions in nonlinear synapses.

$$\mu_{lp}\left(\hat{y}_p(k)\right) = \begin{cases} \dfrac{\hat{y}_p(k) - c_{l-1,p}}{c_{l,p} - c_{l-1,p}}, \text{if } \hat{y}_p(k) \in \left[c_{l-1,p}, c_{l,p}\right], \\ \dfrac{c_{l+1,p} - \hat{y}_p(k)}{c_{l+1,p} - c_{l,p}}, \text{if } \hat{y}_p(k) \in \left[c_{l,p}, c_{l+1,p}\right], \\ 0 \text{ otherwise,} \end{cases} \tag{2}$$

where $c_{l-1,p}, c_{l,p}, c_{l+1,p}$ are coordinates of the membership functions centers, which are usually distributed evenly along the coordinate axes.

One significant benefit of utilizing triangular functions within fuzzy logic systems lies in their efficiency during the learning process. When processing incoming observations – denoted as $\hat{y}_p(k)$ – these triangular functions ensure that only two neighboring membership functions are activated at any given time. This means that, rather than recalculating or adjusting all relevant weights across the system, which would be computationally intensive, only a specific subset of synaptic weights needs to be updated.

By having only two neighboring functions triggered at each step of the learning process, we minimize the number of adjustments required – specifically, adjusting $2q$ synaptic weights per moment. This targeted approach not only reduces computational load but also enhances efficiency, making real-time processing more feasible. Moreover, this property contributes to scalability since it limits the complexity that can arise as data volume increases.

In essence, the use of triangular functions streamlines the learning process by confining adjustments to a localized set of weights and membership functions, thereby optimizing performance in dynamic or large-scale applications. This efficiency is crucial for maintaining responsiveness and accuracy in systems where resources are constrained or rapid data processing is necessary.

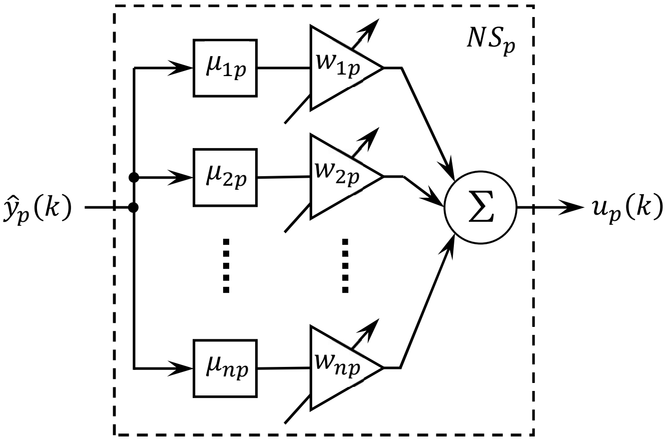

Figure 2: Nonlinear synapse

If approximation accuracy is your primary concern, then using B-splines [14] as membership functions can be highly effective within neo-fuzzy systems [15]. This approach has demonstrated its ability to deliver robust performance across a wide range of applications. However, it's important to note that when implementing this method at each time step k, nq synaptic weights of the neo-fuzzy neuron are adjusted simultaneously. The decision to use B-splines hinges on their unique properties as smooth basis functions capable of capturing intricate patterns in data while maintaining flexibility. This makes them particularly suitable for scenarios where high precision is required, such as in complex or nonlinear systems. However, it's worth considering that adjusting all nq weights at once can increase computational demands, which may impact real-time performance if not carefully managed.

The output signals of nonlinear synapses NS_p – $u_1(k), \dots, u_p(k), \dots, u_q(k)$ are fed to an additional layer of tuned weights $w_1^*, \dots, w_p^*, \dots, w_q^*$ (which is absent in the standard neo-fuzzy neuron), that are present in most known bagging systems and are subject to unbiasedness constraints:

$$\sum_{p=1}^{q} w_p^* = I_q^T w^* = 1, \tag{3}$$

where I_q^T – $(q \times 1)$ vector formed by ones.

Signals $u_p(k)$ after passing through synaptic weights w_p^* are combined in the output adder, forming the optimal output signal

$$\hat{y}^*(k) = \sum_{p=1}^{q} w_p^* u_p(k) = u^T(k) w^*, \tag{4}$$

where $w^* = \left(w_1^*, \dots, w_p^*, \dots, w_q^*\right)^T$, $u(k) = \left(u_1(k), \dots, u_p(k), \dots, u_q(k)\right)^T$.

Thus, in the process of its training, two sets of parameters are configured in the proposed metamodel: $(q \times 1)$ vector w^* and q $(n+1)$ vectors of parameters of nonlinear synapses $w_p, p = 1,2,\dots,q$.

3. Adaptive neo-fuzzy bagging metamodel learning

When the signal $\hat{y}(k)$ is fed to the metamodel, the following signals are formed at the outputs of nonlinear synapses

$$u_p(k) = w_p^T(k-1)\mu_p\left(\hat{y}_p(k)\right), \tag{5}$$

which, after passing through the set of weights w_p^*, are combined in the output adder in the form

$$\hat{y}^*(k) = u^T(k)w^*(k-1), \tag{6}$$

where $w_p(k-1),\ w^*(k-1)$ – estimates obtained based on $k-1$ previous observations.

Adaptive metamodel learning can be implemented based on errors backpropagation. First, the parameter vector w^* is tuned, and then all the parameter vectors of nonlinear synapses $w_p, p = 1,2,\dots,q$ are tuned.

A standard quadratic learning criterion can be used to tune the vector w^*

$$E^*(k) = \frac{1}{2}\left(y(k) - \hat{y}^*(k)\right)^2 = \frac{1}{2}(y(k) - u^T(k)w^*)^2 \tag{7}$$

subject to the unbiasedness constraints

$$I_q^T w^* = 1, \tag{8}$$

where $y(k)$ – external reference signal, also used for all ensemble members training.

To solve the problem, quadratic programming can be used to find the saddle point of the Lagrange function

$$L(w^*,\lambda) = \frac{1}{2}(y(k) - u^T(k)w^*)^2 + \lambda\left(I_q^T w^* - 1\right), \tag{9}$$

where λ is an undefined Lagrangian multiple, which in this case is also a tuned parameter.

To find a saddle point, it is convenient to use the Arrow-Hurwitz algorithm, which in this case takes the form

$$\begin{cases} w^*(k) = w^*(k-1) - \eta_w(k)\nabla_{w^*}L(w^*,\lambda), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\,\partial L(w^*,\lambda)/\partial\,\lambda, \end{cases} \tag{10}$$

or

$$\begin{cases} w^*(k) = w^*(k-1) + \eta_w(k)\left(e(k)u(k) - \lambda(k-1)I_q\right), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\left(I_q^T w^*(k) - 1\right), \end{cases} \tag{11}$$

where $e(k) = y(k) - u^T(k)w^*(k-1)$ – learning error, $\eta_w(k), \eta_\lambda(k)$ – search step parameters that determine the convergence rate of the search process.

The process of tuning the parameter vector w^* can be optimized for speed by appropriately choosing the step $\eta_w(k)$. By specifying this step in the form [16]

$$\eta_w(k) = \frac{e(k)}{e(k)\|u(k)\|^2 - \lambda(k-1)I_q^T u(k)}, \tag{12}$$

we get the optimal learning procedure in terms of speed

$$\begin{cases} w^*(k) = w^*(k-1) + \dfrac{e(k)\big(e(k)u(k) - \lambda(k-1)I_q\big)}{e(k)\|u(k)\|^2 - \lambda(k-1)I_q^T u(k)}, \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\big(I_q^T w^*(k) - 1\big). \end{cases} \tag{13}$$

It is easy to see that in the absence of unbiasedness constraints, this procedure completely coincides with the optimal one-step Kaczmarz-Widrow-Hoff algorithm [17, 18].

After adjusting the output parameters vector $w^*(k)$, we can proceed to adjusting the nonlinear synapses parameters based again on the quadratic criterion

$$\begin{aligned} E(k) &= \frac{1}{2}\big(y(k) - u^T(k)w^*(k)\big)^2 = \frac{1}{2}\left(y(k) - \sum_{p=1}^{q} w_p^*(k)u_p(k)\right)^2 = \\ &= \frac{1}{2}\left(y(k) - \sum_{p=1}^{q} w_p^*(k)w_p^T\mu_p\left(\hat{y}_p(k)\right)\right)^2, \end{aligned} \tag{14}$$

in this case, during the learning process, nq parameters of nonlinear synapses $NS_p, p = 1,2,\dots,q$ must be determined.

Introducing $(nq \times 1)$ vector of tuned parameters of all nonlinear synapses $w(k) = \left(w_{11}(k),\dots,w_{l1}(k),\dots,w_{n1}(k),w_{12}(k),\dots,w_{n2}(k),\dots,w_{lp}(k),w_{nq}(k)\right)^T$ and a modified vector formed by all membership functions and adjusted output weights $\mu^*\big(\hat{y}(k)\big) = \big(w_1^*(k)\mu_{11}(\hat{y}_1(k)),\dots,w_1^*(k)\mu_{n1}(\hat{y}_1(k)),w_2^*(k)\mu_{12}(\hat{y}_2(k)),\dots,w_2^*(k)\mu_{n2}(\hat{y}_2(k)),\dots,$ $w_p^*(k)\mu_{lp}\left(\hat{y}_p(k)\right),\dots,w_q^*(k)\mu_{nq}\left(\hat{y}_q(k)\right)\Big)^T$, it is easy to write the transform implemented by the metamodel with the adjusted parameters vector $w^*(k)$ in the form

$$\hat{y}^*(k) = w^T(k-1)\mu^*\big(\hat{y}(k)\big). \tag{15}$$

To adjust the parameters vector $w(k)$, the same optimal Kaczmarz-Widrow-Hoff algorithm can be used in the form

$$w(k) = w(k-1) + \frac{y(k) - w^T(k-1)\mu^*\big(\hat{y}(k)\big)}{\left\|\mu^*\big(\hat{y}(k)\big)\right\|^2}\mu^*\big(\hat{y}(k)\big) \tag{16}$$

or its modification with additional smoothing properties, which has proven effective in training neo-fuzzy neurons [19]

$$\begin{cases} w(k) = w(k-1) + r^{-1}(k)\left(y(k) - w^T(k-1)\mu^*\big(\hat{y}(k)\big)\right)\mu^*\big(\hat{y}(k)\big), \\ \alpha(k) = \alpha r(k-1) + \left\|\mu^*\big(\hat{y}(k)\big)\right\|^2, 0 \le \alpha \le 1, \end{cases} \tag{17}$$

where α is a smoothing parameter defining a compromise between the filtering and tracking properties of the learning algorithm.

The proposed training procedure for the adaptive neo-fuzzy nonlinear bagging metamodel is designed for online information processing when data is received by the system in real time.

4. Simulation results

We applied the proposed bagging approach to the short-term electric load forecasting problem (STLF), specifically focusing on 1-step ahead forecasting of daily electric load in one of Ukraine's regional power systems. To see, why bagging is the approach of choice to solve STLF problems, let's start with a short overview of STLF itself.

4.1. Overview of short-term electric load forecasting

Electric load forecasting is a critical task for utility companies, enabling them to manage electricity generation and distribution efficiently. Among various forecasting horizons, one-day ahead (one-step) forecasting stands out as particularly challenging due to its dynamic nature and the need for real-time accuracy.

Here are the most common challenges inherent to short-term electric load forecasting.

4.1.1. Data complexity

The first challenge lies in the sheer volume and diversity of data that must be processed to generate accurate forecasts. Historical consumption patterns provide a foundation for predictions, but this dataset is augmented by numerous other variables like weather conditions, economic indicators, calendar events, grid conditions, etc. The integration of these diverse data points is essential for accurate predictions but presents a significant challenge due to their different measurement scales, varying nature, and potential inconsistencies.

4.1.2. Dynamic nature of demand

Electricity demand fluctuates continuously, shaped by human activities such as turning on appliances, working schedules, and leisure time consumption patterns. This dynamic behavior makes it difficult to predict with high precision even just one day ahead. Difficulties arise from time-of-day variations, weekday vs. weekend pattern differences, and seasonal changes.

4.1.3. Nonlinear relationships

The relationship between various factors influencing electricity demand is often nonlinear and complex. Traditional statistical models, which rely on linear relationships, may struggle to capture these nuances effectively. The impact of temperature on load isn't always directly proportional; there can be saturation points beyond which further changes in temperature don't significantly affect demand [20]. The combined effect of weather and economic conditions might not be merely additive but could interact in more complex ways. There are other sources of nonlinearity as well.

This complexity necessitates the use of advanced modeling techniques capable of handling nonlinear relationships, such as machine learning algorithms like Artificial Neural Networks and Support Vector Machines.

4.1.4. External uncertainties

External factors beyond immediate control can disrupt even the most sophisticated forecasting models. These include, but are not limited to sudden weather changes, unforeseen events (e.g. sporting events, strikes, or other unexpected occurrences), technical problems in the grid, etc.

These uncertainties require forecasting models to be robust and adaptable, capable of incorporating real-time adjustments as new information becomes available.

4.1.5. Computational demands

The need for real-time processing and high-frequency updates imposes significant computational demands. To keep forecasts accurate, data must be processed quickly enough to reflect the latest conditions. Advanced models require substantial computational resources for training and updating as new data comes in.

This challenge is compounded by the need for scalability, ensuring that forecasting systems can handle increased data loads without compromising performance or accuracy.

4.1.6. Model selection and validation

Selecting the right forecasting model is a significant challenge due to the dynamic nature of load data. Different models perform better under various conditions, requiring careful selection based on historical performance and expected future scenarios. Ensuring that chosen models remain effective over time requires ongoing validation and adjustment as patterns evolve.

4.1.7. Strategies to address the challenges

To overcome these challenges, experts employ a variety of strategies, including but not limited to:

- The use of various classes of models – from linear regression to deep neural networks to find a better match to a particular forecasting task.
- Online data processing – using real-time data feeds to update forecast as new information is received.
- Ensemble forecasting – combining predictions from multiple models leverages diverse strengths and reduces reliance on any single model's potential biases. This is the focus of this paper.

Hence, short term electric load forecasting is a multifaceted challenge requiring advanced techniques, robust computational infrastructure, careful model selection and tuning, dynamic data processing capabilities, and ongoing validation. The complexity stems from the intricate interplay of numerous variables, the nonlinear relationships between factors influencing demand, and the need for real-time accuracy. Applying bagging approaches can help cope with at least some of the mentioned challenges.

4.2. Test problem details

The original time series consisted of $N = 337$ samples. We generated six forecast series ($q = 6$), each with 337 samples, using six different independent computational intelligence models. In this setup, we treated the time series as a data stream, meaning that both forecasting and metamodel operations were performed in an online mode. This approach ensured that the entire dataset was processed sequentially – sample by sample ($k = 1,2,\dots,N$) – without requiring multiple passes or divisions into training, validation, and test sets.

The original time series (see Figure 3) exhibited several distinct trends corresponding to different seasons, periodic patterns (primarily weekly), sudden changes, and outliers. The presence of these features made the forecasting task particularly challenging. Additionally, there was a significant random component in the data because electric load in large power systems depends on numerous external factors, many of which are inherently unpredictable or chaotic – for example, weather conditions [20]. This randomness contributed to the nonstationary nature of the time series, meaning its statistical properties changed over time. Furthermore, the series contained noise and was highly variable, making it a difficult target for forecasting models.

Given these characteristics – non-stationarity, noise, and complexity – the performance of any single forecasting model or method is unlikely to be consistently superior across the entire dataset. In other words, different models tend to perform better on specific parts of the series but underperform on others. This variability highlights a common challenge in time series forecasting: no single model or method dominates in all situations. It is precisely this kind of scenario where ensemble methods like bagging come into play.

By leveraging the strengths of multiple models through bagging, we aimed to improve overall forecasting accuracy by combining their predictions. The idea was to minimize errors that might arise from relying on a single model and instead capture more robust insights by aggregating results from diverse perspectives within the ensemble. This approach has shown promise in addressing the inherent limitations of individual forecasting methods while providing a more balanced and accurate prediction across the entire time series.

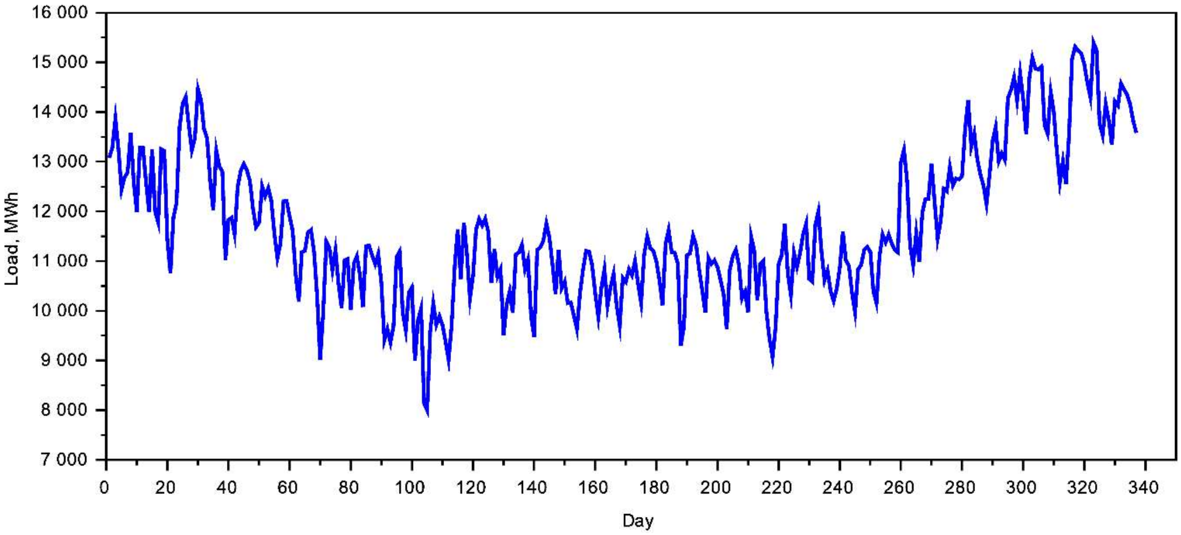

Figure 3: Daily electric load time series

In our simulation, we utilize six specialized STLF models within an ensemble framework. Each of these models has unique inputs and distinct structural differences, which collectively contribute to a diverse predictive capability. Some models focus on historical weather patterns as inputs, while others prioritize calendar events like holidays that affect electricity usage. The diverse structures ensure that each model interprets and processes these inputs differently – some may use linear regression techniques, while others employ neural networks capable of identifying complex patterns.

Deploying six specialized STLF models with varied inputs and structures allows us to comprehensively capture the multifaceted nature of electric load data. This strategic diversity ensures that our ensemble can account for a wide array of factors influencing demand across different parts of the time series.

Figure 4 illustrates the time series data and forecasts for the past 30 days. While long-term trends are reasonably captured by all models, the variability in the data within shorter timeframes is difficult for every forecasting method used. As a result, none of the models demonstrate a significant advantage over the others.

We utilized the proposed metamodel to derive an optimal combination $\hat{y}^*(k)$ of the six forecasts from the ensemble member models. By visually inspecting the plot, it becomes evident that the combined forecast $\hat{y}^*(k)$ closely aligns with the actual data series $y(k)$. This proximity is further corroborated by an in-depth error comparison presented in Table 1.

To evaluate the accuracy of our predictions, we employed the Mean Absolute Percentage Error (MAPE) criterion. Widely recognized in short-term electric load forecasting research, MAPE is a robust metric that quantifies forecast errors as a percentage of actual values, which makes a clear physical sense. The best performing ensemble member model achieved a MAPE of 4.858%. Through the application of the nonlinear bagging procedure, the MAPE was reduced to 3.3751%. This represents an approximate reduction by a factor of 1.44 times compared to the original best model's performance alone.

Table 1
1-day ahead forecasting errors for all models and the metamodel output

Forecasts	#1	#2	#3	#4	#5	#6	Metamodel
MAPE	6.8171%	7.5066%	4.8580%	5.0311%	4.8827%	5.1015%	**3.3751%**

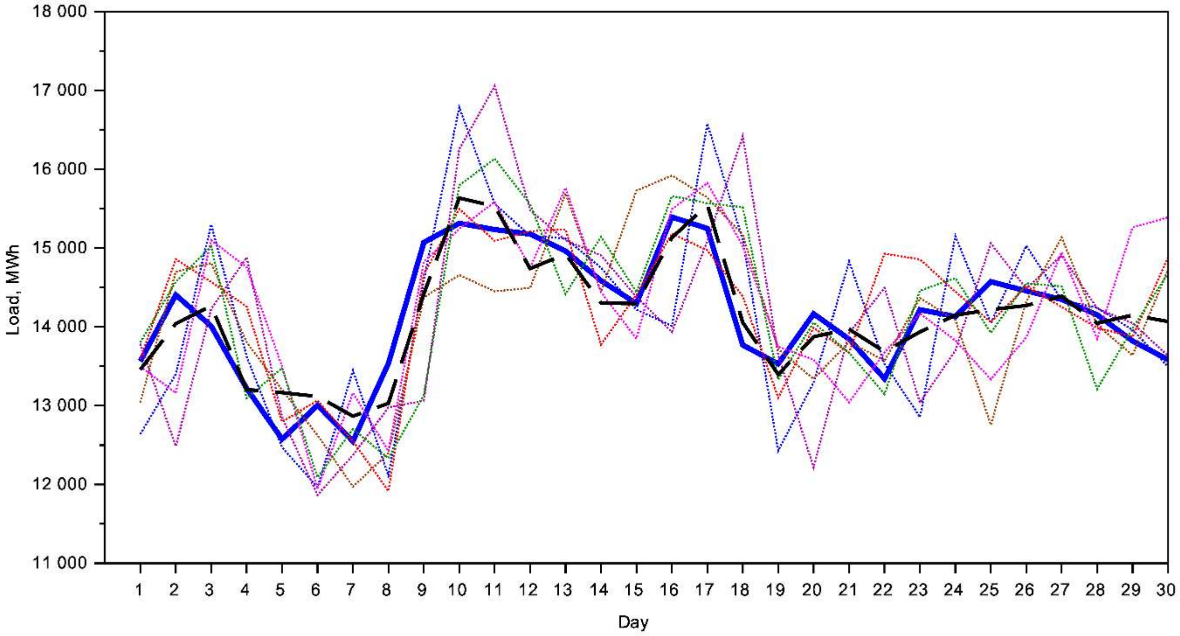

Figure 4: Forecasting results: true electric load (solid blue line), 6 independent 1-day ahead forecasts (dotted color lines), and the metamodel forecast (dashed black line).

5. Conclusions

In this study, we introduced an innovative bagging method based on an adaptive nonlinear neo-fuzzy metamodel. This system is designed to integrate the results from multiple computational intelligence systems working towards solving similar tasks, such as approximating values or making forecasts.

Our approach processes data in real-time and is particularly suited for environments where data conditions change rapidly over time (non-stationarity). This adaptability is crucial because many real-world applications, such as electric load forecasting, involve dynamic factors influencing demand. The speed of processing is also vital; ensuring timely decisions are made based on current data.

We validated our metamodel through a simulation involving short-term electric load forecasting. Utilizing an ensemble of six independent forecasting models to predict electricity consumption, we compared their results with those generated by our proposed metamodel. The outcomes demonstrated that our method outperformed the individual models, decreasing the MAPE error by a factor of 1.44 times relative to the best model in the ensemble.

Looking ahead, we aim to enhance this metamodel's structure by increasing its flexibility. This will enable it to more effectively adapt to various types of relationships in data, particularly nonlinear ones, which are common in complex real-world scenarios like electricity demand forecasting. By doing so, we expect the metamodel to become even more robust and versatile in handling diverse and dynamic data environments.

Acknowledgements

This research was partially funded by the European Union (through the EURIZON H2020 project, grant agreement 871072).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 126–140. https://doi.org/10.1007/BF00058655.

[2] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artificial Intelligence 137.1-2 (2002) 239–263. https://doi.org/10.1016/S0004-3702(02)00190-X.

[3] C. Zhang, Y. Ma (Eds) Ensemble Machine Learning. Methods and Applications, Springer, New York, NY, 2012. https://doi.org/10.1007/978-1-4419-9326-7.

[4] J. H. Friedman, P. Hall, On bagging and nonlinear estimation, Journal of Statistical Planning and Inference 137.3 (2007) 669–683. https://doi.org/10.1016/j.jspi.2006.06.002.

[5] A. Bifet, G. Holmes, B. Pfahringer, R. Gavaldà, Improving Adaptive Bagging Methods for Evolving Data Streams, in: ZH. Zhou, T. Washio (Eds), Advances in Machine Learning, Vol. 5828 of LNCS, Springer, Berlin, 2009, pp. 23–37. https://doi.org/10.1007/978-3-642-05224-8_4.

[6] 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.

[7] H. Wu, D. Levinson, The ensemble approach to forecasting: A review and synthesis, Transportation Research Part C: Emerging Technologies, 132 (2021) 103357. https://doi.org/10.1016/j.trc.2021.103357

[8] Y. Yang, Ensemble Learning, in: Y. Yang (Ed) Temporal Data Mining Via Unsupervised Ensemble Learning, Elsevier, 2017, pp. 35–56. https://doi.org/10.1016/B978-0-12-811654-8.00004-X

[9] 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, volume 2774 of LNCS, Springer, Berlin, Heidelberg, 2003. https://doi.org/10.1007/978-3-540-45226-3_132.

[10] T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi, A neo fuzzy neuron and its applications to system identification and prediction of the system behavior, in: Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, 1992, pp. 477-483.

[11] E. Uchino, T. Yamakava, Neo-fuzzy neuron based new approach to system modeling with application to actual system, in: Proc. Sixth International Conference on Tools with Artificial Intelligence, New Orlean, LA, USA, 1994, pp.564-570.

[12] T. Miki, T. Yamakawa, Analog implementation of neo-fuzzy neuron and its on-board learning, in: Computational Intelligence and Applications, WSES Press, Piraeus, 1999, pp. 144-149.

[13] I. Perfilieva, Fuzzy transforms: Theory and applications, Fuzzy Sets and Systems 157.8 (2006) 993-1023. https://doi.org/10.1016/j.fss.2005.11.012.

[14] J. Zhang, A. Knoll, Constructing Fuzzy Controllers with B-Spline Models. Principles and Applications, International Journal of Intelligent Systems 13, N. 2/3 (1998) 257–286. https://doi.org/10.1002/(SICI)1098-111X(199802/03)13:2/3%3C257::AID-INT9%3E3.0.CO;2-Z.

[15] V. Kolodyazhniy, Ye. Bodyanskiy, Cascaded multiresolution spline-based fuzzy neural network, in: Proc. International Symposium on Evolving Intelligent Systems, 2010, pp. 26-29.

[16] Ye. Bodyanskiy, O. Mykhalyov, I. Pliss, Adaptyvne vyyavlennya rozladnan' v ob'ektakh keruvannya za dopomogoyu shtuchnykh neyronnykn merezh, Systemni tekhnilogii, Dnipro, 2000. In Ukrainian.

[17] S. Kaczmarz, Approximate solution of systems of linear equations, International Journal of Control 57.6 (1993) 1269–1271. https://doi.org/10.1080/00207179308934446.

[18] B. Widrow, M.E. Hoff, Adaptive switching circuits, in: 1960 IRE WESCON Convention Record, Part 4, IRE, New York, 1960, pp. 96–104.

[19] Ye. Bodyanskiy, I. Kokshenev, V. Kolodyazhniy, An adaptive learning algorithm for a neo fuzzy neuron, in: Proc. 3rd Conference of the European Society for Fuzzy Logic and Technology, Zittau, Germany, 2003, pp. 375-379.

[20] Ye. Bodyanskiy, S. Popov, T. Rybalchenko, Feedforward neural network with a specialized architecture for estimation of the temperature influence on the electric load, in: Proc. 2008 4th International IEEE Conference Intelligent Systems, Varna, Bulgaria, 2008, pp. 7-14–7-18. https://doi.org//10.1109/IS.2008.4670444.