Enhancing Time Series Forecasting with On-the-Fly Data Augmentation

Mohamed Abdellatif Ayadi[1]

Technical University of Dortmund, Dortmund, Germany
Lamarr Institute for Machine Learning and Artificial Intelligence, Dortmund
mohamed-abdellatif.ayadi@tu-dortmund.de

Abstract. Time series forecasting is pivotal across domains such as banking, healthcare, and energy systems, where accurate predictions enable proactive decision-making and optimal resource allocation. However, forecasting remains challenging due to complex temporal dependencies, seasonal patterns, and inherent uncertainties in data, even with advancements in machine learning. This work investigates the efficacy of innovative preprocessing techniques and deep learning architectures in enhancing the predictive performance of time series models. Through comprehensive experiments across diverse domains, we demonstrate how these methods address key forecasting challenges, offering robust and scalable solutions applicable to real-world scenarios.

Keywords: Time Series Forecasting · Deep Learning · Data Augmentation.

1 Introduction

Time series forecasting plays a pivotal role in a wide array of fields, including healthcare, finance, and climate monitoring, enabling proactive decision-making and optimal resource allocation. However, the non-linear and non-stationary nature of contemporary datasets poses significant challenges for traditional statistical models, necessitating the development of reliable, scalable, and adaptable forecasting solutions.

Deep learning has emerged as a transformative approach in this domain, offering the ability to capture intricate temporal patterns in high-dimensional data. Techniques such as Convolutional Neural Networks (CNNs)[9], Long Short-Term Memory (LSTM) networks [12], and attention mechanisms have demonstrated notable performance improvements across various forecasting tasks. Despite these advancements, existing models often rely heavily on large datasets, are prone to overfitting, and lack interpretability, particularly in scenarios involving limited or sparse data.

In this paper, we address these limitations by introducing a novel framework that integrates real-time data augmentation into the forecasting pipeline. By dynamically generating augmented data during training, our approach enhances model generalization, reduces dependence on extensive labeled datasets, and maintains temporal coherence. Building upon techniques such as Multiple Block Bootstrap (MBB) [2] and generative frameworks like GRATIS (GeneRAting TIme Series) [8], we propose an optimized solution for forecasting under complex conditions.

Through comprehensive evaluations on benchmark datasets and real-world applications, we demonstrate the efficacy of our framework in bridging the gap between model robustness and data scarcity. This work contributes to advancing state-of-the-art time series forecasting and provides a foundation for future exploration and practical implementation.

1.1 Main Contribution

Building on the challenges and approaches discussed in the introduction, this study introduces a novel framework for *On-the-Fly Data Augmentation Techniques (OnDAT) [4]* aimed at advancing deep learning-based time series forecasting. The principal contributions of this work are as follows:

1. **Deep Learning for Time Series Forecasting and Data Augmentation**: Leveraging state-of-the-art deep learning architectures such as Long Short-Term Memory (LSTM) networks[12], Transformers, and Convolutional Neural Networks (CNNs)[9], this work highlights the role of data augmentation in enhancing these models'ability to capture complex temporal dependencies. The study addresses the limitations of traditional statistical methods while emphasizing the importance of augmenting data to improve robustness.
2. **Addressing Challenges in Temporal Dependencies**: The proposed framework leverages decomposition techniques, such as Seasonal and Trend Decomposition using LOESS (STL) [6], to effectively isolate intrinsic temporal patterns, including long-term trends and recurring seasonal variations. This decomposition process ensures that augmented data maintains its original temporal structure, which is crucial for preserving forecasting accuracy. By retaining these essential characteristics, the framework enhances the reliability of predictions in domains such as energy consumption forecasting and financial time series analysis.
3. **Proposed On-the-Fly Data Augmentation Framework[4]**: A dynamic data augmentation framework is proposed, generating augmented data during training to expose the model to diverse data distributions. This approach enhances model generalization, reduces overfitting, and mitigates reliance on large labeled datasets.
4. **Evaluation and Scalability:** The proposed OnDAT framework [4] is rigorously evaluated on the ETT dataset [18] to assess its forecasting accuracy and scalability. Results demonstrate its ability to improve predictive performance while maintaining computational efficiency across varying dataset sizes. By leveraging scalable augmentation techniques, the framework ensures adaptability to real-world forecasting applications.

2 Related Work

The progress in time series forecasting and data augmentation techniques has led to significant advancements in the field of predictive modeling. This section reviews recent trends in deep learning-based forecasting, explores existing data augmentation methodologies for time series, and highlights gaps that motivate the proposed framework.

2.1 Deep Learning for Time Series Forecasting and Data Augmentation

Deep learning architectures have revolutionized time series forecasting by effectively capturing complex temporal dependencies that traditional statistical models, such as ARIMA [3] and exponential smoothing, struggle to handle. Recurrent neural networks like Long Short-Term Memory (LSTM) networks [12] have shown notable success due to their ability to retain sequential dependencies through gated memory mechanisms.

More recently, Transformer-based architectures [14] have gained prominence for their ability to efficiently model long-range dependencies. Despite their advantages, these models are computationally expensive and require large-scale datasets to achieve optimal performance. This reliance on extensive training data highlights the need for data augmentation to enhance generalization and mitigate overfitting.

Data augmentation is a crucial technique for improving the robustness of deep learning models, particularly in domains where labeled data is scarce [15]. Traditional augmentation methods, such as time warping, jittering, and random cropping, introduce variability into training data but often fail to preserve intrinsic temporal structures such as seasonality and trend. [13]

To address this limitation, frequency-domain augmentation techniques , such as FrAug [5], manipulate the spectral components of time series data to introduce meaningful variations while maintaining underlying periodicity. Although effective, these approaches require domain-specific expertise and are computationally expensive.

On-the-fly data augmentation[4] provides a more adaptive approach by dynamically generating augmented samples during training. Unlike traditional methods, this strategy aligns augmentation with model training, ensuring that generated samples enhance model generalization without compromising temporal consistency.

Table 1 summarizes and compares key data augmentation techniques used in time series forecasting, highlighting their advantages and limitations.

Table 1. Comparison of Data Augmentation Techniques for Time Series Forecasting

Method	**Advantages**	**Limitations**
Time Warping	Simple implementation, enhances diversity	May distort temporal dependencies
FrAug	Preserves frequency-domain properties	Requires domain-specific knowledge, high computational cost
On-the-Fly Augmentation	Dynamically adapts to training needs	Computational overhead during training

2.2 Addressing Challenges in Temporal Dependencies

Despite advancements in time series forecasting, existing methods face challenges in preserving temporal dependencies and adapting to dynamic data distributions. Static augmentation techniques often fail to capture evolving temporal structures, resulting in suboptimal generalization. Furthermore, frequency-domain approaches like FrAug [5] require domain-specific knowledge and lack flexibility across diverse datasets.

Temporal decomposition techniques, such as Seasonal and Trend Decomposition using LOESS (STL) [6], offer a promising avenue for addressing these limitations by preserving intrinsic temporal patterns, such as trends and seasonality. However, few studies have effectively incorporated these techniques into dynamic augmentation frameworks.

For instance, in energy forecasting, capturing seasonal patterns like daily or monthly consumption fluctuations is critical. Ignoring temporal dependencies can lead to significant prediction errors. Similarly, in healthcare analytics, patient monitoring data often exhibits both short-term variability and long-term trends (e.g., circadian rhythms or disease progression [1]). Failing to consider these dependencies compromises the reliability of predictive models and their practical applicability.

These limitations highlight the need for a dynamic data augmentation framework that integrates decomposition-based techniques while ensuring adaptability to evolving time series patterns. Such an approach must enhance robustness, scalability, and predictive accuracy while preserving intrinsic temporal dependencies.

2.3 Proposed On-the-Fly Data Augmentation Framework

this paper proposes the On-the-Fly Data Augmentation Technique (OnDAT) [4]. OnDAT builds upon the strengths of traditional augmentation methods while overcoming their limitations by integrating decomposition-based augmentation and dynamic synthetic data generation into the training process. [4]

The framework leverages STL decomposition [6] to ensure that augmented data maintains temporal consistency, reducing the risk of distorted trends or unrealistic fluctuations. Furthermore, it incorporates adaptive augmentation strategies that adjust data transformations dynamically during training, improving model robustness to unseen patterns.

Table 2 outlines the key steps in the proposed framework, detailing its structured approach from data preprocessing to scalability optimization.

Table 2. Key Steps in the Proposed Framework for On-the-Fly Data Augmentation [4]

Step	**Description**	**Purpose/Outcome**
Data Preprocessing	Handle missing values, normalize datasets, and decompose time series into trend, seasonal, and residual components.	Ensures data quality and prepares for augmentation while preserving temporal patterns.
Dynamic Augmentation	Generate synthetic data variations during model training by respecting decomposed components like seasonality and trends.	Improves model generalization by exposing it to diverse patterns.
Model Training	Integrate augmented data into the training pipeline for models such as Transformer-based architectures [14]	Enhances learning robustness and predictive accuracy.
Evaluation	Assess performance using metrics like MAE [17] and RMSE [16] on benchmark datasets (e.g., ETT [18]).	Validates the effectiveness and efficiency of the framework.
Scalability Optimization	Implement efficient sampling strategies to minimize computational overhead.	Enables practical deployment in large-scale scenarios.

2.4 Evaluation and Scalability in Time Series Forecasting

Scalability is a fundamental challenge in time series forecasting, particularly when models are deployed in large-scale, real-world applications. Traditional statistical models, such as ARIMA [3] and Exponential Smoothing, have been widely used for forecasting but struggle to scale effectively due to computational limitations. Deep learning models, such as LSTMs [12] and Transformer-based architectures [14], have improved predictive accuracy but require significant computational resources, making scalability a critical concern.

Prior research has explored various techniques for evaluating forecasting performance, emphasizing the importance of selecting appropriate metrics to ensure robust model comparisons. While conventional error measures remain widely used, recent studies highlight the need for metrics that account for scale-invariance and generalization across datasets [10]. Additionally, scalability optimization techniques, such as adaptive data augmentation and efficient sampling strategies, have been proposed to enhance forecasting efficiency.

Despite these advancements, existing models often fail to balance scalability, forecasting accuracy, and computational efficiency in dynamic environments. Addressing this gap, the proposed OnDAT framework incorporates real-time augmentation and decomposition-based techniques to improve scalability while maintaining forecasting reliability. The following sections discuss the framework's validation process and experimental evaluation.

3 Methodology

The proposed methodology integrates multiple key steps into a unified and coherent workflow, which is illustrated clearly in Figure 1. The workflow begins with input time series data, followed by STL [6] decomposition to isolate trends, seasonality, and residual components. Subsequently, dynamic data augmentation [4] is applied to enhance the training set diversity and robustness. The augmented data then feeds into the training pipeline, utilizing advanced deep learning models. Finally, validation and rigorous testing ensure accurate forecasting results.

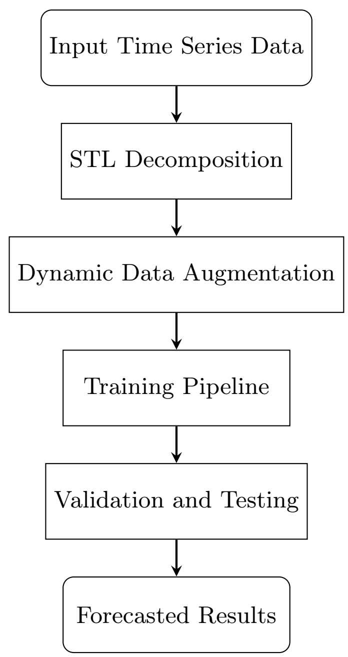

Fig. 1. Workflow of the On-the-Fly Data Augmentation Framework.

3.1 Decomposition of Time Series Data

Time series data often contains intertwined components such as trends, seasonality, and residual noise. Separating these components is essential to preserve temporal properties during augmentation and improve forecasting accuracy. The decomposition can be expressed as [6]:

$$y(t) = T(t) + S(t) + R(t) \tag{1}$$

where $y(t)$ is the original time series, $T(t)$ captures long-term trends, $S(t)$ represents repeating seasonal patterns, and $R(t)$ accounts for residual variations.

Seasonal and Trend Decomposition using Loess (STL) [6] is utilized for this purpose. STL's ability to isolate non-linear trends and seasonality makes it ideal for datasets with complex temporal patterns, such as those in energy consumption and financial forecasting.

3.2 Dynamic Data Augmentation Framework

The dynamic data augmentation framework [4] generates augmented samples in real time to enrich training datasets and improve model generalization. By introducing variability during training, the framework mitigates overfitting and enhances the robustness of deep learning models.

Dynamic augmentation begins with preprocessing, where missing values are handled, and time series are normalized and decomposed using STL. Once preprocessed, the framework applies real-time transformations, such as scaling and shifting, time warping, and noise injection. These transformations preserve the temporal structure of the data while exposing the model to diverse patterns. This approach is particularly effective in domains where labeled data is limited, as it enhances the learning process without requiring extensive manual data curation.

3.3 Training Pipeline

The training pipeline integrates decomposition and dynamic data augmentation into a cohesive workflow, ensuring robust and scalable forecasting models. It consists of three key stages:

First, the preprocessed and augmented data is fed into advanced deep learning architectures, specifically Long Short-Term Memory (LSTM) networks [12]and Transformer-based models [14]. These architectures effectively capture sequential dependencies and long-term relationships inherent in time series data.

Next, the models are trained using dynamic sampling strategies, prioritizing diverse and impactful data points. This significantly reduces computational overhead, ensuring efficient resource utilization during large-scale training.

Finally, a validation dataset is used separately during training to monitor the model's learning process, prevent overfitting, and fine-tune model hyperparameters. This validation process is crucial for achieving optimal model performance before final testing.

3.4 Validation and Testing

The proposed framework undergoes a rigorous validation and testing process designed to assess its generalization capabilities and forecasting accuracy. Specifically, the performance is quantified using widely adopted metrics in forecasting literature, namely Mean Absolute Error (MAE) [17], Root Mean Square Error (RMSE) [16], symmetric Mean Absolute Percentage Error (sMAPE) [10], and Mean Absolute Scaled Error (MASE) [7] . These metrics are selected for their interpretability, robustness against outliers, and relevance for evaluating forecasting accuracy across diverse scenarios. By clearly and thoroughly assessing the model's performance on separate validation and testing subsets, the framework ensures both reliability and applicability in practical forecasting contexts.

4 Experiments

This section presents the experimental setup, evaluation metrics, and results to validate the proposed On-the-Fly Data Augmentation Techniques (OnDAT) [4]. The experiments are designed to assess the framework's performance in addressing challenges such as data scarcity, temporal dependencies, and computational efficiency.

4.1 Experimental Setup

Dataset The experimental evaluation of OnDAT [4] was conducted using state-of-the-art datasets and models to benchmark its effectiveness. The experimental design and configurations are detailed below.

A well-established dataset was selected for evaluating the proposed framework, ensuring clarity and coherence in evaluation:

– **ETT Dataset [18]**: The ETT (Electricity Transformer Temperature) dataset contains hourly energy consumption data and is particularly suitable for evaluating short-term forecasting models. Its inherent temporal dependencies and complexity make it a challenging benchmark for assessing model scalability and forecasting accuracy.

A summary of this dataset is provided in Table 3, highlighting its characteristics and application domain clearly.

Table 3. Dataset Used in Experiments

Dataset	**Time Series**	**Granularity**	**Domain**
ETT	28,000	Hourly	Energy

Evaluation Metrics Evaluating the effectiveness of time series forecasting models requires robust metrics that quantify the accuracy and reliability of predictions. The following metrics were used:

Mean Absolute Error (MAE) [17]:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

where y_i represents the actual values, $\hat{y}_i$ represents the predicted values, and n is the total number of observations. MAE [17]is widely used for its interpretability and robustness against outliers.

Root Mean Square Error (RMSE) [16]:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

RMSE penalizes larger errors more significantly, making it suitable for applications where deviations from actual values are critical.

Symmetric Mean Absolute Percentage Error (sMAPE) [10] :

$$\text{sMAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \times 100 \tag{4}$$

sMAPE offers a scale-invariant error measure, which is particularly useful for datasets with varying magnitudes.

Mean Absolute Scaled Error (MASE) [7] :

$$\text{MASE} = \frac{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|} \tag{5}$$

MASE evaluates forecasting errors relative to a naive baseline, ensuring fairness across different datasets.

Models Used The experiment compares multiple forecasting models to evaluate the impact of On-the-Fly Data Augmentation (OnDAT) [4]. These models include well-established deep learning architectures as baselines and various OnDAT variations.

- **LSTM (Baseline)**: Long Short-Term Memory (LSTM) networks [12] are widely used in time series forecasting due to their ability to capture long-range dependencies. LSTMs employ gated mechanisms to selectively retain and forget information, making them effective for sequential modeling. However, they struggle with capturing long-term dependencies efficiently and may require extensive training data.
- **Transformer (Baseline)**: Transformer models [14] utilize self-attention mechanisms to process entire input sequences in parallel, allowing them to capture long-range dependencies more effectively than recurrent models. They have gained popularity in time series forecasting for their ability to model complex temporal patterns. Despite their advantages, Transformers require large datasets and substantial computational resources.
- **OnDAT Variants**: To assess the impact of different data augmentation strategies, we evaluate multiple OnDAT configurations [4]:

- **OnDAT(tr) - Augmentation during Training Only**: This variant applies on-the-fly data augmentation only during the training phase. It aims to improve model generalization by introducing diverse patterns while the model learns. However, the absence of augmentation in validation may lead to a mismatch between training and evaluation conditions.
- **OnDAT(vl) - Augmentation during Validation Only**: In this setting, augmentation is applied exclusively during validation. The primary objective is to test how the model adapts to unseen augmented data, simulating real-world conditions where data distributions may shift over time.
- **OnDAT(gratis) - Augmentation via GRATIS**: This method leverages the GRATIS (GeneRAting TIme Series) approach [8], which synthesizes artificial time series data by capturing underlying statistical properties. While GRATIS provides diversity, it does not guarantee temporal consistency in augmented sequences.
- **OnDAT(fixed)- Fixed-Block Augmentation**: Unlike dynamic augmentation, this variant applies augmentation in fixed, predefined segments. This ensures stable transformations but may not fully capture variations in data distribution over time.
- **OnDAT(Full) - Comprehensive On-the-Fly Augmentation**: This configuration applies augmentation in both training and validation phases with moving blocks, ensuring that augmented samples are dynamically adapted to each learning stage. It aims to maximize generalization and robustness, making it the most comprehensive implementation of OnDAT.

Training Setup All models are implemented in Python using PyTorch and trained on an NVIDIA GPU with: - Learning Rate: 0.001 - Batch Size: 64 - Training Epochs: 50

4.2 Experimental Results

The following experiments evaluate the effectiveness of On-the-Fly Data Augmentation (OnDAT) [4] for time series forecasting. We assess its impact across four key aspects: 1. Performance comparison against baseline models. 2. Effectiveness of different OnDAT variations. 3. Scalability analysis across dataset sizes. 4. Comparison with traditional data augmentation techniques.

Each experiment presents a structured breakdown, followed by a detailed analysis.

Performance Comparison Against Baseline Models

Objective This experiment examines whether OnDAT [4] improves forecasting accuracy compared to two widely used deep learning models, LSTM [12] and Transformer [14]. The goal is to determine if on-the-fly data augmentation enhances generalization and reduces errors.

Results and Analysis The results indicate that OnDAT [4] achieves lower forecasting errors than both LSTM and Transformer baselines. Table 4 shows that OnDAT attains a Mean Absolute Error (MAE)[17] of 0.721 and a Root Mean Square Error (RMSE)[16] of 1.265, outperforming the baseline models. This suggests that dynamic augmentation exposes the model to more diverse temporal patterns, particularly benefiting the Transformer architecture, which shows a marked reduction in forecasting errors.

Table 4. Performance Comparison of Models on the ETT Dataset [18]

Model	MAE	RMSE	sMAPE (%)	MASE
LSTM (Baseline)	0.865	1.435	9.5	0.45
Transformer (Baseline)	0.802	1.340	8.7	0.41
OnDAT (Full)	0.721	1.265	7.6	0.38

Impact of Different OnDAT Variations

Objective This experiment investigates how different OnDAT configurations affect forecasting accuracy. Variants include OnDAT(tr), OnDAT(vl), OnDAT(gratis), OnDAT(fixed), and OnDAT(Full) [4].

Results and Analysis Table 5 shows that models augmented with OnDAT(tr) outperform those with OnDAT(vl), suggesting that training-phase augmentation is more beneficial than augmentation during validation. OnDAT(gratis) performs competitively but does not exceed OnDAT(fixed) or OnDAT(Full). OnDAT(Full) attains the lowest errors, indicating that consistently applying augmentation in both training and validation further improves generalization.

Table 5. Performance Comparison of OnDAT Variants on ETT Dataset [18]

Model	MAE	RMSE	sMAPE (%)	MASE
OnDAT (tr)	0.789	1.318	8.3	0.41
OnDAT (vl)	0.872	1.402	9.2	0.44
OnDAT (gratis)	0.753	1.287	7.9	0.39
OnDAT (fixed)	0.723	1.269	7.7	0.39
OnDAT (Full)	0.721	1.265	7.6	0.38

Scalability Analysis Across Dataset Sizes

Objective This experiment evaluates whether OnDAT [4] remains effective and computationally feasible when applied to varying portions of the ETT dataset, reflecting scenarios of different scales.

Results and Analysis As presented in Table 6, OnDAT shows minimal degradation in accuracy as dataset size increases, suggesting robust scalability. For example the Mean Absolute Error (MAE)[17] increases only slightly from 0.945 to 0.978 when transitioning from 20% to 100% of the dataset, while training time scales proportionally with the dataset size. This indicates that OnDAT is capable of handling larger datasets without incurring a substantial performance penalty.

Table 6. Scalability Analysis of OnDAT on Different ETT Subset Sizes

Subset Size	Training Time (min)	MAE	RMSE	sMAPE (%)	MASE
20% of ETT dataset	12	0.945	1.523	10.2	0.47
50% of ETT dataset	45	0.963	1.530	10.0	0.46
100% of ETT dataset	92	0.978	1.545	9.8	0.45

Note: The OnDAT (Full) result at 100% in this table differs from the numbers in Table 4 and Table 5. Here, we report the average of multiple runs with varied random seeds, whereas the earlier tables present the best single-run performance. Hence, the reported metrics differ slightly.

Comparison with Traditional Data Augmentation

Objective This experiment contrasts OnDAT [4] with static data augmentation (DA) and a Seasonal Naïve baseline [4] to determine if dynamic augmentation provides a substantial advantage.

Results and Analysis Table 7 indicates that OnDAT surpasses both static DA and the Seasonal Naïve approach, for example achieving a lower MAE [17] of 0.721. Although static DA reduces errors compared to the naïve baseline, it does not offer the same adaptability as OnDAT. These findings confirm that real-time augmentation further enhances model generalization and forecasting accuracy.

Table 7. Comparison of OnDAT with Standard Augmentation

Model	MAE	RMSE	sMAPE (%)	MASE
Seasonal Naïve	1.102	1.804	12.4	0.55
Standard DA (Static Augmentation)	0.889	1.462	9.9	0.46
OnDAT (Full)	0.721	1.265	7.6	0.38

5 Discussion

This study presented a dynamic data augmentation framework, OnDAT [4], for time series forecasting, integrating STL-based decomposition [6] with on-the-fly synthetic sample generation. Our experiments focused on the ETT dataset [18], assessing both baseline architectures (LSTM [12], Transformer [14]) and different OnDAT variations. The findings underscore several key points:

Effectiveness of OnDAT Variations Comparisons among OnDAT(tr), OnDAT(vl), OnDAT(fixed), OnDAT(gratis), and OnDAT(Full) [4] indicate that consistently applying augmentation during both training and validation yields superior performance. In particular, OnDAT(Full) generally attains the lowest forecasting errors. While methods like GRATIS [8] provide diverse synthetic series, preserving core temporal structures (e.g., via STL decomposition and block-based bootstrapping [2]) proves advantageous for tasks with strong seasonality or trends.

Performance Gains over Static Augmentation Our results confirm that an on-the-fly augmentation approach outperforms static or pretraining-only strategies in mitigating overfitting. By exposing the model to real-time variability, OnDAT effectively boosts generalization, leading to lower MAE[17],RMSE[16],sMAPE [10] and MASE [7] values. This improvement is especially evident for Transformer architectures [14], which benefit from the increased diversity of training examples.

Scalability and Computational Overhead Although OnDAT requires additional computations during training—chiefly due to real-time augmentation—this overhead remains moderate and does not affect inference. For real-world deployments, such as energy consumption or financial forecasting, the trade-off between slightly increased training times and significantly enhanced predictive performance is frequently justified.

Limitations and Future Directions While STL-based decomposition successfully preserves seasonal and trend components, its reliance on linear or smooth patterns may not capture highly non-linear dependencies. Exploring alternative decomposition methods (e.g., wavelet transforms [11]) could extend OnDAT's applicability to more complex signals. Additionally, although this work validated OnDAT on the ETT dataset [18], it may be beneficial to investigate domain-specific considerations—particularly in healthcare or anomaly detection—where data distributions evolve in unpredictable ways.

Broader Implications By reducing overfitting and improving generalization, OnDAT offers a practical avenue for augmenting limited datasets in dynamic environments. Future work could refine synthetic generation techniques (for instance, weighting selected subsequences more heavily based on model uncertainty), or introduce explainability modules to elucidate how augmented series influence model decisions. Ultimately, the adaptive nature of OnDAT positions it as a robust and extensible framework for next-generation time series forecasting applications.

6 Conclusions

Data augmentation is often crucial when limited data hinder the training of effective deep learning models, particularly in time series forecasting. A typical approach involves creating a single augmented dataset prior to training, which can restrict the variety of synthetic samples available. In this paper, we introduced OnDAT [4], a novel method that embeds data augmentation directly into the training pipeline. By applying seasonal decomposition and moving block bootstrapping (MBB) [2] to each batch during both training and validation, OnDAT continuously diversifies the training data. This on-the-fly process yields multiple synthetic dataset variations rather than relying on a single augmented set.

Our extensive experiments demonstrate that OnDAT outperforms not only a static augmentation baseline but also the scenario without any augmentation. These findings underscore the benefits of dynamically generated synthetic data in enhancing forecasting models, providing stronger generalization and improved accuracy. By shifting data augmentation from a one-time preprocessing step to a continuous strategy throughout training, OnDAT offers a practical way to address data scarcity in time series forecasting applications.

Acknowledgments The author would like to thank Professor Dr. Saadallah from the Lamarr Institute for Machine Learning and Artificial Intelligence and TU Dortmund for their guidance and support throughout this research.

Disclosure of Interests The author declares that there are no competing interests relevant to the content of this paper.

References

1. Aschoff, J.: Circadian rhythms in man. Science **148**(3676), 1427–1432 (1965)
2. Bergmeir, C., Hyndman, R.J., Benítez, J.M.: Bagging exponential smoothing methods using stl decomposition and box–cox transformation. International Journal of Forecasting **32**(2), 303–312 (2016)
3. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1970)
4. Cerqueira, V., Santos, M., Baghoussi, Y., Soares, C.: On-the-fly data augmentation for forecasting with deep learning. arXiv preprint arXiv:2404.16918v1 (2024), available at https://arxiv.org/html/2404.16918v1
5. Chen, J., Zhang, L., Li, H., Wang, Y.: Fraug: Frequency-based data augmentation for time series forecasting. arXiv preprint arXiv:2303.10165 (2023)

6. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: Stl: A seasonal-trend decomposition. Journal of Official Statistics **6**(1), 3–73 (1990)
7. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International Journal of Forecasting **22**(4), 679–688 (2006)
8. Kang, Y., Hyndman, R.J., Li, F.: Gratis: Generating time series with diverse and controllable characteristics. Statistical Analysis and Data Mining: The ASA Data Science Journal **13**(4), 354–376 (2020)
9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition, vol. 86. IEEE (1998)
10. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The m4 competition: Results, findings, conclusion and way forward. International Journal of Forecasting **34**(4), 802–808 (2018)
11. Percival, D.B., Walden, A.T.: Wavelet Methods for Time Series Analysis. Cambridge University Press (2000)
12. Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparison of ARIMA and LSTM in forecasting time series. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 1394–1401. IEEE (2018)
13. Um, T., Pfister, F.M.J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., Kulić, D.: Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. Proceedings of the 19th ACM International Conference on Multimodal Interaction pp. 216–220 (2017)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
15. Wen, R., Torkkola, K., Narayan, A., Makila, M., Zhou, W.K.: Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478 (2021)
16. Willmott, C.J., Matsuura, K.: Statistics for the evaluation and comparison of models, vol. 90 (1985)
17. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate Research **30**(1), 79–82 (2005)
18. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI Conference on Artificial Intelligence **35**(12), 11106–11115 (2021)