Evaluating Large Language Models for Turboshaft Engine Torque Prediction

Alessandro Tronconi[1], David He[2], and Eric Bechhoefer[3]

[1,2]*University of Illinois-Chicago, Chicago, IL, 60607, USA*
atron@uic.edu
davidhe@uic.edu

[3]*GPMS International Inc., Waterbury, VT, 05677, USA*
eric@gpms-vt.com

ABSTRACT

Recent advances in deep learning (DL), particularly transformer-based architectures, have opened new possibilities for time series forecasting. Large Language Models (LLMs), initially developed for natural language processing, are emerging as promising tools for sequence modeling beyond text due to their scalability, multimodal integration, and few-shot learning capabilities. However, their application in industrial forecasting tasks, such as turboshaft engine torque prediction, remains largely unexplored.

This paper evaluates the potential of LLMs for forecasting engine torque, a key parameter for ensuring the safety, reliability, and efficiency of helicopter operations. We compare the performance of four models: two general-purpose LLMs (GPT-2 and ChatGPT), a specialized time series LLM (TimeGPT), and a standard transformer-based time series model. Using real-world Health and Usage Monitoring System (HUMS) data from a Bell 407 engine, we assess these models in terms of forecasting accuracy and classification of insufficient torque conditions. Our findings demonstrate that GPT-2 and TimeGPT achieve strong predictive performance (RMSE ~1.35–1.46), while prompt-based solutions like ChatGPT, though accessible, lag in accuracy. The results highlight the promise of LLMs for industrial time series forecasting and outline future directions for developing efficient, domain-adaptable models.

1. INTRODUCTION

The rapid advancement of LLMs has significantly transformed various industries, enabling diverse applications ranging from simple user tasks to complex industrial processes. Among these applications, researchers have begun to explore using LLMs in the critical domain of time series forecasting. Time series forecasting involves predicting future values based on historical, time-stamped observations. Based on a sequence of values $\{x_1, \dots, x_T\}$, where x_t corresponds to the time stamp t, its objective is to estimate x_{T+1}. Time series analysis and forecasting underpin decision-making in industrial contexts, where achieving high accuracy in prediction is essential, as it directly influences strategic and operational decisions.

Traditional statistical forecasting methods, such as seasonal autoregressive integrated moving average with exogenous variables (SARIMAX), have demonstrated strong performance in short-term predictions but often require significant domain expertise and manual adjustments to achieve optimal results. More recently, DL approaches like Long Short-Term Memory (LSTM) networks have gained popularity due to their effectiveness in capturing longer-term dependencies within time series data. However, existing methods typically remain specialized, with models frequently optimized for specific data characteristics and prediction horizons.

Inspired by the success of transformer-based architectures, particularly large-scale models developed initially for natural language processing (NLP), this paper investigates their potential for enhancing time series forecasting capabilities. Models such as Informer (Zhou, Zhang, Peng, Zhang, Li et al., 2021) have already demonstrated substantial success within this context, suggesting that transformer architectures possess inherent advantages in handling temporal data. Leveraging LLMs introduces the possibility of employing their advanced capabilities for forecasting purposes. This

approach offers significant promise, particularly due to the models' abilities to understand and reason over complex patterns and temporal dependencies without extensive domain-specific customization (Chang, Wang, Peng, & Chen, 2025).

Furthermore, the growing accessibility and relatively low cost associated with LLMs highlight their potential as efficient alternatives to specialized forecasting models. This paper examines explicitly whether cost-effective, readily accessible LLMs can outperform traditional forecasting approaches, thereby providing broader and more generalized solutions.

The remainder of this paper is organized as follows. Section II reviews the relevant literature and contextual background. Section III outlines the proposed methodology. Section IV presents the case study and its results. Finally, Section V concludes the study and discusses potential directions for future research.

2. BACKGROUND AND RELATED WORK

2.1. Deep Learning-Based Time Series Forecasting

The foundation of time series forecasting rests on classical statistical methods refined over decades of research and practical application. These methods include exponential smoothing, moving averages, and autoregressive models that assume relatively stable underlying patterns in the data. The autoregressive integrated moving average (ARIMA) family of models represents one of the most influential developments in classical time series analysis. Such models are capable of modeling a wide range of temporal patterns, and the integration of seasonal components has proven particularly valuable in domains where predictable cyclical patterns significantly influence outcomes.

Machine learning (ML) models such as Linear Regression, Random Forest, Gradient Boosting, and other regression-based algorithms have been widely applied to high-dimensional time series data. The advent of DL has transformed time series forecasting, introducing neural network (NN) architectures specifically designed to capture complex temporal dependencies.

Modern DL architectures for time series forecasting typically employ encoder-decoder frameworks that can handle both one-step-ahead and multi-horizon prediction tasks, and the diversity of architectural choices reflects the heterogeneous nature of time series problems across various domains. Lim and Zohren (2020) provided an extensive overview of DL methods for time series forecasting. A significant trend in recent years has been the emergence of hybrid models that combine well-established statistical techniques with NN components, leveraging both statistical methods' theoretical foundation and interpretability and the pattern recognition capabilities of DL architectures (Lim & Zohren, 2020).

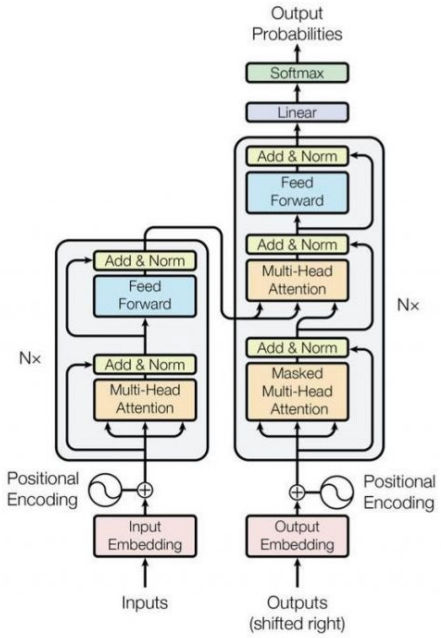

Figure 1. The transformer model architecture (extracted from Vaswani et al., 2017).

Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), and LSTM networks are among the most popular DL models, and combining CNN with LSTM offers distinct advantages for processing long input sequences (Brownlee, 2019). However, among DL methods, transformer-based architectures (Figure 1) have emerged as particularly promising and are the focus of our research, as they have marked a significant milestone in the evolution of temporal prediction methods. The success of transformers in NLP has inspired researchers to adapt these architectures for time series applications, leading to the development of specialized models. The transformer architecture is entirely based on the self-attention mechanism (Vaswani, Shazeer, Parmar, Uszkoreit, Jones et al., 2017), composed of encoder-decoder layers. The multi-head attention mechanism, which forms the core of the encoder-decoder structure, consists of multiple attention heads whose outputs, derived from scaled dot-product attention, are concatenated and linearly projected (see Figure 2). The attention function is computed as follows (Soydaner, 2022):

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (1)$$

In Eq. (1), Q, K, and V represent the matrices containing queries, keys, and values, respectively. d_K represents the dimension of the keys.

$$Z_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (2)$$

$$MultiHead(Q,K,V) = Concat(Z_i, \dots, Z_h)W^O \quad (3)$$

Equations (2) and (3) show the multi-head attention computation, where W^Q, W^K, W^V are the weight matrices for queries, keys, and values, and W^O is a learned weight matrix used to linearly project the attentions (Z_i), combining information from different heads in a learned manner.

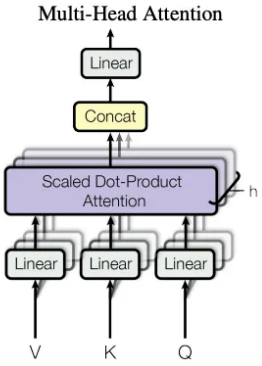

Figure 2. The basic structure of the multi-head attention mechanism (extracted from Vaswani et al., 2017).

Several advanced models have been developed to adapt transformers to the specificities and challenges of time series data, such as the issue of quadratic complexity, by revising the attention mechanism. The Informer model (Zhou et al., 2021) represents a relevant advancement in applying attention mechanisms to long sequence time series forecasting, addressing the computational limitations of Vanilla transformers for temporal data, and opening the door to multivariate time series forecasting (Simhayev, Rogge, & Rasul, 2023).

Examples of similar models are the Autoformer (Simhayev, Rasul, & Rogge, 2023) and the Temporal Fusion Transformer (Kafritsas, 2024). Overall, recent studies have provided evidence that when compared against equivalent-sized models, transformer-based architectures consistently achieve superior performance on standard test metrics.

2.2. Large Language Models and LLM-Based Time Series Forecasting

LLMs are advanced Artificial Intelligence (AI) systems designed to understand and generate human language by learning from vast textual corpora. Built on DL architectures, particularly transformers, they can capture complex linguistic patterns. Models such as GPT revolutionized NLP with a two-stage approach: pre-training on large-scale, unlabeled datasets through self-supervised learning objectives, followed by fine-tuning on specific downstream tasks (Cao & Wang, 2024).

This paradigm has led to state-of-the-art performance across diverse applications, including text generation, translation, and conversational agents. The most prominent LLM providers include OpenAI, Meta, and Google. OpenAI's GPT series, particularly GPT-2, gained early recognition for generating coherent and contextually appropriate text. Its architecture is displayed in Figure 3. Building on this foundation, ChatGPT has become one of the most popular conversational systems worldwide.

The success of LLMs in NLP has sparked interest in their potential applications beyond textual data. The pattern recognition capabilities exhibited by LLMs have proven transferable to other domains, suggesting that the underlying mechanisms for sequence modeling and pattern extraction may be more general than initially anticipated (Jin, Wang, Ma, Chu, Zhang et al. 2024). The challenge lies in effectively bridging the modality gap between numerical time series observations and the natural language tokens that LLMs are designed to process.

Multimodal LLMs (MLLMs) are models capable of processing heterogeneous and unstructured data from multiple modalities and are increasingly applied in industrial settings for anomaly detection. Recent approaches have explored their adaptation to non-textual domains such as vibration and acoustic signals. Leveraging transfer learning and few-shot learning frameworks enables effective classification and forecasting with minimal labeled data (He, He, & Taffari, 2023).

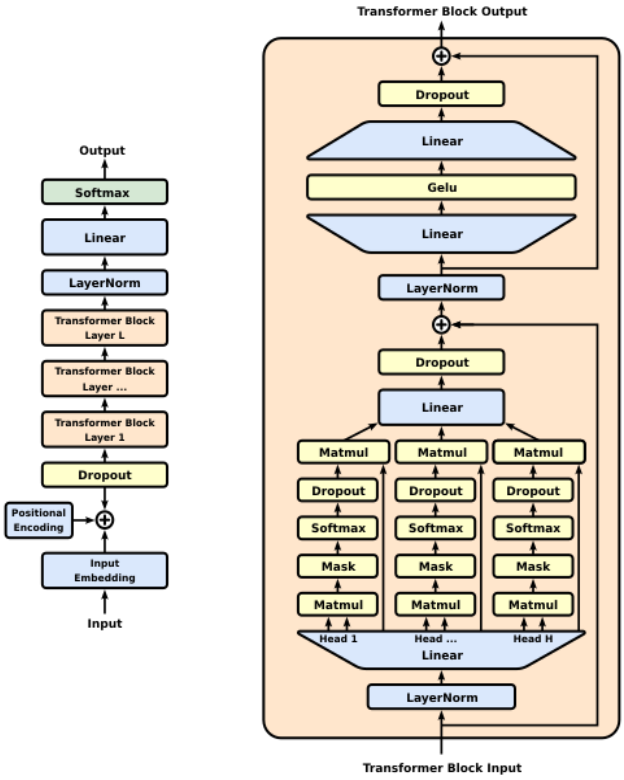

Figure 3. GPT-2 architecture (from Wikipedia).

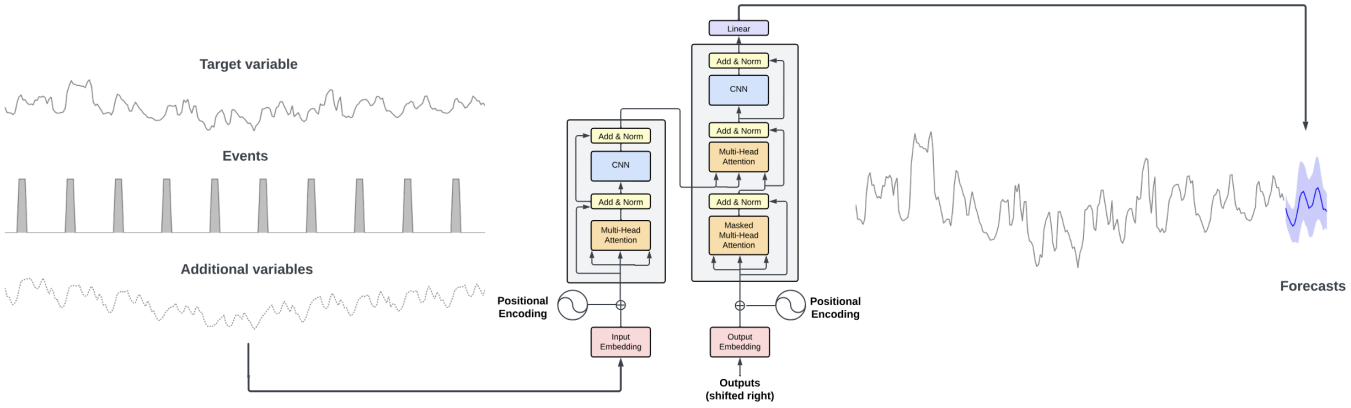

Figure 4. TimeGPT architecture overview (extracted from Garza, Challu, & Mergenthaler-Canseco, 2023).

Jiang, Li, Deng, Liu, Gao et al. (2025) proposed a benchmark for MLLMs handling images and textual data in the field of industrial inspection, while other studies have addressed adapted models for zero-shot anomaly detection (Deng, Luo, Zhai, Cao, & Kang, 2021). Through GPT4MTS, Jia, Wang, Zheng, Cao, and Liu (2023) have also explored a prompt-based solution for the integration of textual and numerical data.

Specialized GPT models for time series have been introduced, such as TEMPO (Cao, Jia, Arik, Pfister, Zheng et al., 2023), which uses a prompt-based approach and employs a transformer architecture fed with data about the trend, seasonal, and residual components of the data. In such cases, designing an optimal prompt is crucial and generally, LLMs perform well in forecasting time series with clear patterns and trends but struggle with datasets that lack periodicity or exhibit irregular behavior (Tang, Zhang, Jin, Yu, Wang et al., 2025), which requires specific adaptation to address time series forecasting and outperform the classical models effectively.

TimeGPT (Garza, Challu, Mergenthaler-Canseco, 2024), a large pre-trained time series model developed by Nixtla, is among the most promising recent advancements. It delivers state-of-the-art performance while maintaining accessibility through built-in functions and user-friendly settings. Figure 4 illustrates its architecture, closely resembling the standard transformer framework shown in Figure 2. However, instead of a Softmax layer, TimeGPT employs a linear transformation to output forecasted values directly.

2.3. Turboshaft Engine Torque Prediction

The analysis and prediction of turboshaft engine torque are critical to ensure the safety and longevity of helicopter operations. Torque measurements directly reflect mechanical load on the engine components. They are closely linked to component wear, fuel consumption, and overall performance, serving as a key indicator for proactive maintenance and failure prevention. Previous research has explored data-driven approaches utilizing HUMS sensor data, including DL models to capture complex temporal dynamics. In particular, He, Bechhoefer, and Hess (2025) applied LSTM-based models trained on HUMS data and fine-tuned them for new engines and operating conditions, demonstrating strong generalization with minimal target-domain data. Our approach builds on this foundation by investigating whether LLMs can achieve comparable or superior performance. Paniccia, Tucci, Guerrero, Capone, Sanguini et al. (2025) applied NN to Leonardo's AW189P4 prototype engine for torque prediction. However, these approaches face limitations in terms of generalization capabilities and accessibility. GPT variants and transformer-based foundation models are increasingly explored for time-series forecasting and can address these limitations through their native support for heterogeneous data types, enabling multimodal processing. Furthermore, the self-attention mechanism effectively captures long-range dependencies within torque time series, offering improved performance over traditional window-based forecasting approaches.

3. THE METHODOLOGY

Four models have been included in our analysis: (1) GPT-2 as an LLM; (2) *ChatGPT-o4-mini-high*, the most suitable OpenAI conversational model for code generation and visual reasoning, in a prompt-based configuration; (3) TimeGPT as a specialized large time-series model; (4) Time Series Transformer model as a benchmark. Figure 5 illustrates the proposed methodology for forecasting turboshaft engine torque. The process begins with collecting and preparing HUMS sensor data. The workflow then branches into four model-specific pipelines, including embedding and positional encoding when applicable, model training, and forecasting.

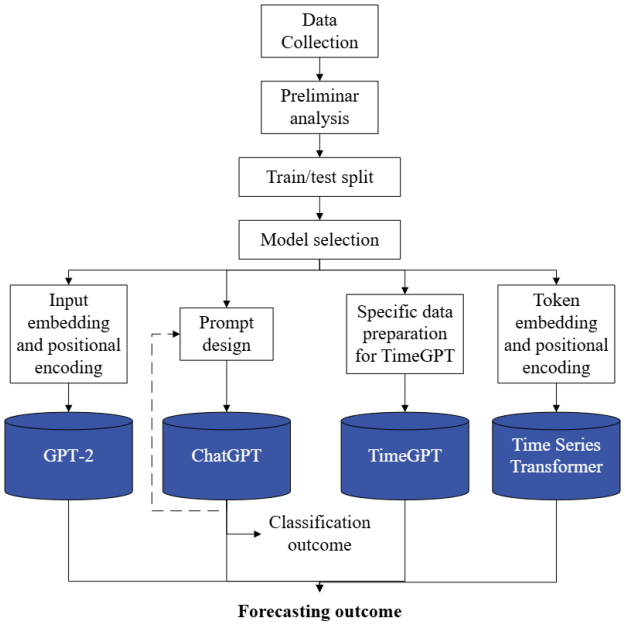

Figure 5. Flowchart of the proposed methodology.

The GPT-2 and Time Series Transformer models were implemented using the *PyTorch* library in Python. Their setup involved model initialization, parameter configuration, and the training loop. In contrast, ChatGPT operates through prompt-based interactions; it requires appropriate textual prompts and produces output in the form of a structured dataset containing predicted values. TimeGPT requires a more elaborate preprocessing phase, including proper formatting of the input sequences, to utilize its built-in fine-tuning and forecasting functions; multiple training and test splits were considered to evaluate the model's forecasting performance across different data availability scenarios. All paths converge at a common forecasting accuracy evaluation step, measured by Root Mean Squared Error (RMSE), defined in Eq. (4).

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \qquad (4)$$

ChatGPT's classification performance was also assessed, as detailed in the following section.

4. CASE STUDY

4.1. The Dataset

This study utilizes HUMS data collected from a Bell 407 helicopter engine. The dataset contains 7954 observations of engine torque, recorded alongside six exogenous variables: Outside Air Temperature (OAT), Measured Gas Temperature (MGT), Pressure Altitude (PA), Indicated Airspeed (IAS), Gas Generator Speed (Ng), and Power Turbine Speed (Np). The measurement units and corresponding value ranges for each variable are summarized in Table 1. For all the models, the dataset was partitioned into a training set (80%, 6363 observations) and a test set (20%, 1591 observations).

Table 1. The dataset parameters and value range.

Parameter	Unit	Range
Engine Torque	%	[0.102, 93.445]
OAT	°C	[-2.086, 15.684]
MGT	°C	[454.633, 812.562]
PA	feet	[3348.425, 8699.112]
IAS	knot	[-0.186, 147.156]
NG	rpm	[68.499, 99.889]
NP	rpm	[41.030, 102.988]

The target variable, Engine Torque, was recorded as a time series and expressed as a percentage of the engine's maximum torque. Data analysis revealed extended intervals during which torque values remained consistently low and stable, indicating insufficient torque output by the engine. This condition reflects a reduced capacity to generate rotational force, resulting in diminished acceleration and lower power at reduced speeds. Consequently, it is also relevant to evaluate the model's ability to forecast torque values accurately and assess its effectiveness in classifying whether the torque exceeds a defined operational threshold.

To this end, for ChatGPT, a threshold of 40% was adopted, and the model's classification effectiveness in identifying insufficient torque conditions was measured by precision, recall, and accuracy.

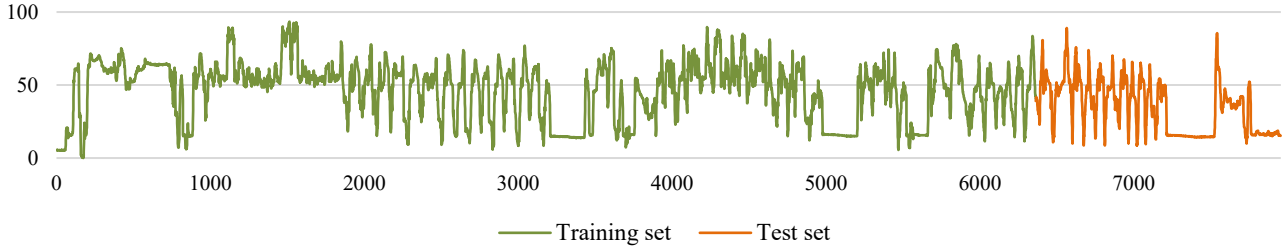

Figure 6. Engine Torque data plot.

4.2. Results

4.2.1. GPT-2

The GPT-2 model was fine-tuned for time series forecasting using all six exogenous variables. Prior to training, the input data were scaled, and numerical features were embedded with positional encoding to retain temporal structure. The pre-trained GPT-2 architecture was adapted and fine-tuned using the parameters detailed in Table 2 and the default *gpt2* configuration. After training, the model was applied to forecast the values in the test set (Figure 7), yielding an RMSE of 1.35, indicating effective learning of temporal dependencies and exogenous variable influence.

Table 2. GPT-2 model's hyperparameters.

Hyperparameter	Value
Embedding dimension	768
Batch size	32
Epochs	100
Initial Learning Rate	10^{-4}
Final Learning Rate	10^{-6}
Scheduler	Cosine Annealing
Loss function	MSE
Optimizer	AdamW

4.2.2. ChatGPT

It was observed that *ChatGPT-o4-mini-high* defaults to using a basic ARIMA approach when requested to generate forecasts without a properly formulated instruction prompt; it fails to infer the appropriate order, resulting in inaccurate predictions. Instead, when explicitly asked to perform forecasting using an LLM, it produces satisfactory results even on the first attempt (Figure 8). This demonstrates the model's effectiveness in a scenario where no additional contextual information is provided. However, the model produced a relatively high RMSE of 12.93, as the prediction consistently overestimated the true values. Nevertheless, it successfully captured the overall trend of the data over time and correctly identified periods of insufficient torque. Its classification performance—distinguishing whether values were above or below the 40% threshold—shows about 75% in recall (see Table 3).

Table 3. ChatGPT's classification performance.

Classification metric	Value
Precision	0.9897
Recall	0.7467
Accuracy	0.8517

4.2.3. TimeGPT

TimeGPT from Nixtla was first evaluated in its forecasting capabilities without exogenous variables, and it failed to produce meaningful forecasts for short and long (*timegpt-1-long-horizon*) horizons, indicating that the target variable cannot be reliably predicted from historical values alone and demonstrating intrinsic dependence of the engine torque on contextual variables. While this may appear as a limitation, it aligns with practical scenarios where such contextual data is usually available. When all six exogenous variables were included, TimeGPT (Figure 9) demonstrated strong predictive performance, achieving an RMSE of 1.46 (Table 4). In addition to accurate predictions, the model returned interpretable variable importance scores, allowing for analysis of the relative influence of each exogenous input on the forecast. To explore the effect of prediction horizon on model accuracy, the number of predicted values was progressively increased, therefore decreasing the training set size. When forecasting 6000 values, instead of the standard 1591, the RMSE rose within acceptable limits, while further increases in the forecasting horizon led to continued degradation in accuracy, with the model struggling to capture peaks.

Table 4. TimeGPT's prediction performance with various training and test splits.

Training test size	Test set size	RMSE
6363 (80%)	1591 (20%)	1.4585
1954 (25%)	6000 (75%)	2.2037
454 (6%)	7500 (94%)	4.1259

4.2.4. Time Series Transformer Model

Finally, we evaluated a time series transformer, which involves token embedding and positional encoding as the main components. The model training was performed on normalized data and using the hyperparameters in Table 5. The resulting forecasting RMSE was 5.38. As a reference, a simple SARIMAX(2,1,1) model yielded an RMSE of 6.12.

Table 5. Time Series Transformer hyperparameters.

Hyperparameter	Value
Model dimension	256
Batch size	32
Epochs	100
Initial Learning Rate	10^{-4}
Final Learning Rate	10^{-6}
Scheduler	Cosine Annealing
Loss function	MSE
Optimizer	AdamW

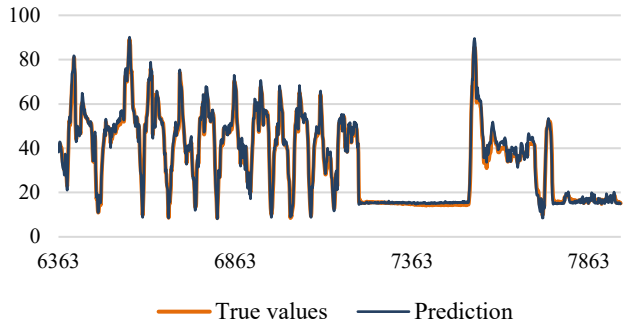

Figure 7. GPT-2 forecasting plot.

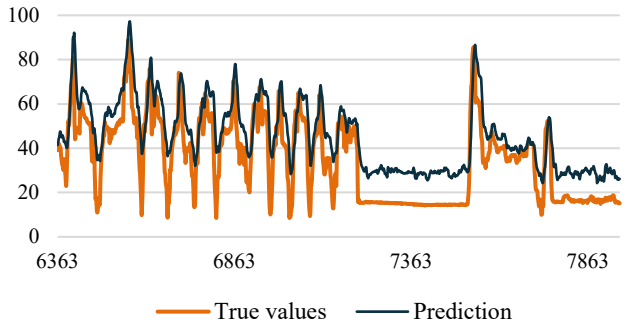

Figure 8. ChatGPT forecasting plot.

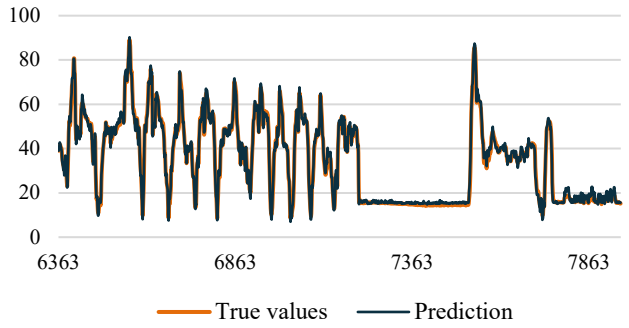

Figure 9. TimeGPT forecasting plot.

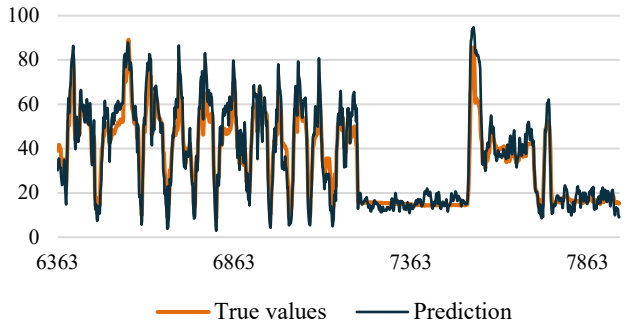

Figure 10. Time Series Transformer forecasting plot.

4.2.5. Summary

A summary of the prediction results is presented in Table 6.

Table 6. Prediction results for the models assessed.

Model	RMSE
GPT-2	1.35
ChatGPT	12.9260
TimeGPT	1.4585
Time Series Transformer	5.3787

Among the evaluated models, GPT-2, ChatGPT, TimeGPT, and a time series transformer, GPT-2 and ChatGPT represent LLMs and were the primary focus of the investigation. LLMs exhibited strong potential in this domain, with GPT-2 achieving comparable results to TimeGPT. However, constraints like computational costs and limited accessibility for larger models persist. Prompt-based approaches have emerged as a practical and user-friendly solution, as evidenced by the increasing popularity of tools like ChatGPT, but they lack accuracy and control over training procedures compared to code-based models.

For time series transformers, hyperparameter tuning becomes a critical challenge that can significantly impact performance when dealing with black-box datasets lacking contextual or domain-specific information. Specialized models further illustrate the potential of transformer-based architectures in time series forecasting. GPT-2 achieved the best performance in our study, with an RMSE of 1.35, while TimeGPT stood out for its ease of use and built-in explainability, combining strong predictive accuracy with minimal configuration effort.

In summary, the results highlight the growing potential of both general-purpose and domain-specific LLMs for accurate time series forecasting.

5. CONCLUSIONS

This study assessed the feasibility and performance of applying LLMs to the challenging task of turboshaft engine torque prediction, comparing general-purpose and specialized transformer-based architectures.

The results obtained by GPT-2 affirm the promise of LLMs for industrial forecasting tasks when properly fine-tuned. Meanwhile, the time series transformer and ChatGPT, despite their advantages in flexibility and interactivity, lagged in accuracy, underscoring the importance of model

selection and configuration for mission-critical forecasting applications.

Looking ahead, two avenues for future research emerge. First, developing compact, efficient, and domain-adaptable LLMs tailored for time series tasks could democratize access to advanced forecasting tools for non-expert users. Second, enhancing the fine-tuning and few-shot learning capabilities of general-purpose LLMs will be key to unlocking their full potential in industrial contexts, such as prognostics, quality assurance, and zero-defect manufacturing. These directions promise to accelerate the integration of LLM-based solutions into real-world aerospace and industrial health management systems, supporting more reliable and data-driven decision-making.

REFERENCES

Brownlee, J. (2019, Aug. 5). *How to get started with deep learning for time series forecasting*. Machine Learning Mastery. https://machinelearningmastery.com/how-to-get-started-with-deep-learning-for-time-series-forecasting-7-day-mini-course/

Cao, D., Jia, F., Arik, S. O., Pfister, T., Zheng, Y., Ye, W., & Liu, Y. (2023). Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

Cao, R., & Wang, Q. (2024). An evaluation of standard statistical models and llms on time series forecasting. *arXiv preprint arXiv:2408.04867*.

Chang, C., Wang, W. Y., Peng, W. C., & Chen, T. F. (2025). Llm4ts: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, *16*(3), 1-20.

Deng, H., Luo, H., Zhai, W., Cao, Y., & Kang, Y. (2024). Vmad: Visual-enhanced multimodal large language model for zero-shot anomaly detection. *arXiv preprint arXiv:2409.20146*.

Garza, A., Challu, C., & Mergenthaler-Canseco, M. (2023). TimeGPT-1. *arXiv preprint arXiv:2310.03589*.

Jiang, X., Li, J., Deng, H., Liu, Y., Gao, B. B., Zhou, Y., ... & Zheng, F. (2024). Mmad: The first-ever comprehensive benchmark for multimodal large language models in industrial anomaly detection. *arXiv preprint arXiv:2410.09453*.

He, D., Bechhoefer, E., & Hess, A. (2025). Automated rotorcraft turboshaft engine performance prediction using a transfer learning approach. *IEEE Aerospace Conference*, Mar. 1–8, Big Sky, MT, USA.

He, D., He, M., & Taffari, A. (2023). Few-shot learning for full ceramic bearing fault diagnosis with acoustic emission signals. *PHM Society Asia-Pacific Conference* (Vol. 4, No. 1), Sept. 11–14, Tokyo, Japan.

Kafritsas, N. (2024, Dec. 17). *Temporal fusion transformer: Time series forecasting with interpretability*. AI Horizon Forecast. https://aihorizonforecast.substack.com/p/temporal-fusion-transformer-time

Koti, V. (2024, Sept. 12). *From theory to code: step-by-step implementation and code breakdown of GPT-2 model*. Medium. https://medium.com/@vipul.koti333/from-theory-to-code-step-by-step-implementation-and-code-breakdown-of-gpt-2-model-7bde8d5cecda

Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200209.

Paniccia, D., Tucci, F. A., Guerrero, J., Capone, L., Sanguini, N., Benacchio, T., & Bottasso, L. (2025). A supervised machine-learning approach for turboshaft engine dynamic modeling under real flight conditions. *arXiv preprint arXiv:2502.14120*.

Simhayev, E., Rasul, K., & Rogge, N. (2023, Jun. 16). *Yes, transformers are effective for time series forecasting (+ autoformer)*. Hugging Face. https://huggingface.co/blog/autoformer

Simhayev, E., Rogge, N., & Rasul, K. (2023, Mar. 10). *Multivariate probabilistic time series forecasting with informer*. Hugging Face. https://huggingface.co/blog/informer

Soydaner, D. (2022). Attention mechanism in neural networks: where it comes and where it goes. *Neural Computing and Applications*, *34*(16), 13371-13385.

Tang, H., Zhang, C., Jin, M., Yu, Q., Wang, Z., Jin, X., ... & Du, M. (2025). Time series forecasting with llms: Understanding and enhancing model capabilities. *ACM SIGKDD Explorations Newsletter*, *26*(2), 109-118.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.

Wikipedia (2024, Mar. 22). *Full GPT architecture*. https://commons.wikimedia.org/wiki/File:Full_GPT_architecture.svg

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 12, pp. 11106-11115).