

FAE: A Multi-Agent System for Automated and Explainable Multivariate Time Series Forecasting Pipelines

Anonymous ACL submission

Abstract

This paper introduces FAE (Forecaster Agent and Explainer), a multi-agent system that automates and explains multivariate time series forecasting pipelines integrated with an AutoML tool called AutoDCE-TS. The system integrates three cooperative agents to execute pipelines, handle errors, summarize intermediate steps, and generate user-oriented explanations using Large Language Models. FAE autonomously performs data preprocessing, model configuration, forecasting, and result interpretation, while supporting interactive explanations through Retrieval-Augmented Generation. Experimental results show that AutoDCE-TS configured by FAE achieves predictive performance comparable to manual configurations across multiple datasets and horizons. A case study with different LLMs highlights trade-offs between reasoning depth, efficiency, and communication quality. The results demonstrate the potential of agent-based LLM systems to automate forecasting workflows while improving transparency and interpretability.

1 Introduction

In recent years, Machine Learning (ML) has driven the development of new architectures applied to several domains, with the goal of learning generalizable patterns capable of producing accurate predictions on unseen data (Vieira et al., 2020). However, achieving this objective depends on a complex, time-consuming, and error-prone process that involves multiple decisions, such as data preprocessing, model selection, and hyperparameter definition, which directly impact the final performance of the models.

In response to these challenges, Automated Machine Learning (AutoML) emerged with the aim of automating the construction of ML pipelines, reducing human effort, the need for task-specific modeling, and intensive data manipulation (Luo et al., 2023). In time series forecasting — a central

task for planning and control activities across different domains — AutoML approaches play a particularly relevant role, since these problems require specialized technical knowledge to handle inherent properties of temporal data, such as trend, seasonality, and temporal dependence (Paldino et al., 2021).

Despite the wide availability of AutoML tools designed for time series forecasting, such as AutoGluonTS (Alexandrov et al., 2019), PyCaret (Ali, 2020), Fedot (Nikitin et al., 2022), AutoTS (Wang et al., 2022), and AutoDCE-TS (Lucas et al., 2025), these solutions still demand considerable technical expertise for proper configuration and use, which imposes barriers to adoption by non-expert users (Fischer and Saadallah, 2024).

More recently, intelligent agents based on Large Language Models (LLMs) have introduced a new layer of automation into AutoML systems, enhancing the interaction between users and systems. These agents have been employed to support tasks such as hyperparameter tuning, explanation of the pipelines used, and, in some cases, the complete planning and execution of the ML process (Trirat et al., 2024; Guo et al., 2024a; Liu et al., 2024; Xu et al., 2024).

In the domain of time series forecasting, the only agent-based AutoML system identified in the literature is TimeSeriesScientist (Zhao et al., 2025), a multi-agent system aimed at fully automating pipelines for univariate time series. This system is composed of four agents responsible, respectively, for diagnosing the input time series, selecting models and hyperparameters, generating forecasts, and producing a final report in natural language.

Therefore, a gap in the literature can be observed regarding agent-based AutoML systems for time series forecasting, particularly in addressing multivariate problems. In this context, this work proposes the development of a multi-agent system called FAE (Forecaster Agent and Explainer), integrated with AutoDCE-TS. AutoDCE-TS automates

the stages of a MISO (Multiple Input, Single Output) pipeline, ranging from feature extraction and selection to model construction and forecast generation. However, the system presents limitations related to the handling of missing data, the configuration of time series decomposition strategies, the definition of lag window size, and the lack of a user-friendly communication interface, especially with respect to explaining the final pipeline. FAE aims to complement AutoDCE-TS by providing a more accessible user interface, automating data preprocessing and its configurations, and generating explanatory reports that facilitate the understanding and interpretation of the results by the end user.

2 Preliminary Concepts

2.1 AutoML

Automated Machine Learning (AutoML) refers to systems capable of automatically recommending optimized machine learning pipelines for specific tasks (Zöller and Huber, 2021). A pipeline consists of a sequence of operations that transforms a feature vector X into a target y , encompassing steps such as data preparation, feature engineering, and model generation, including hyperparameter selection and tuning (He et al., 2021).

In AutoML, the combination of these operations is formulated as an optimization problem, in which the pipeline structure can be represented as a Directed Acyclic Graph (DAG). In this context, let G denote the set of valid structures, A the set of algorithms, and $\vec{\lambda}$ their hyperparameters; the objective is to find, for a dataset D , the optimal pipeline P^* that minimizes the loss function \mathcal{L} , as expressed in Equation 1 (Zöller and Huber, 2021; Chen et al., 2019).

$$P_{(g, \vec{A}, \vec{\lambda})}^* = \operatorname{argmin}_{g \in G, \vec{A} \in A, \vec{\lambda} \in \Lambda} \mathcal{L}(P_{(g, \vec{A}, \vec{\lambda})}, D) \quad (1)$$

2.2 Agents Based on Large Language Models

In the field of Artificial Intelligence (AI), an autonomous agent is an entity capable of perceiving the environment, making decisions, and executing actions independently (Xi et al., 2025). The evolution of these agents has progressed from rule-based systems to approaches such as reinforcement learning, transfer learning, and meta-learning, with LLMs being the main drivers of recent advances due to their reasoning, planning, and tool-use capabilities (Xi et al., 2025).

Intelligent agents based on LLMs are able to plan and make decisions across multiple iterations, access external tools (such as databases and APIs), and use memory to store information (Masterman et al., 2024).

Agent architectures can range from single-agent systems to multi-agent configurations. While single agents are suitable for well-structured tasks, multiple agents are preferable when collaboration is beneficial. In multi-agent systems, each agent has a specific role and can cooperate in the planning and execution of tasks (Guo et al., 2024b).

3 Method

This section presents FAE (Forecaster Agent and Explainer), a multi-agent system developed to automate and explain the pipelines generated by AutoDCE-TS. The system is composed of three agents: the Executor, responsible for executing the forecasting pipeline, including data preprocessing, configuration and invocation of AutoDCE-TS, result visualization, and extraction of process-related information; the Corrector, in charge of detecting and handling errors during execution; and the Explainer, responsible for generating partial reports and the final report, as well as clarifying user questions about all stages of the workflow.

FAE is structured into four modules — pipeline execution, correction, summarization, and explanation — as illustrated in Figure 1. Based on a user request, the system automatically deploys the entire workflow, coordinating the agents in an integrated manner to execute a new forecasting task or generate explanations for a previously executed pipeline. The Explainer agent is responsible for interpreting the request and triggering the appropriate modules in each case. The operation of each module is detailed below.

3.1 Pipeline Execution

The pipeline execution module is activated after the user requests a forecasting task, which must include a description of the problem and the dataset to be used. In this module, the Executor agent is configured with a specific prompt and equipped with tools for data preprocessing, forecast generation, result visualization, and extraction of information from AutoDCE-TS.

The prompts, described in Appendix A, follow the ReAct format (Yao et al., 2022), which enhances the agent’s reasoning capability

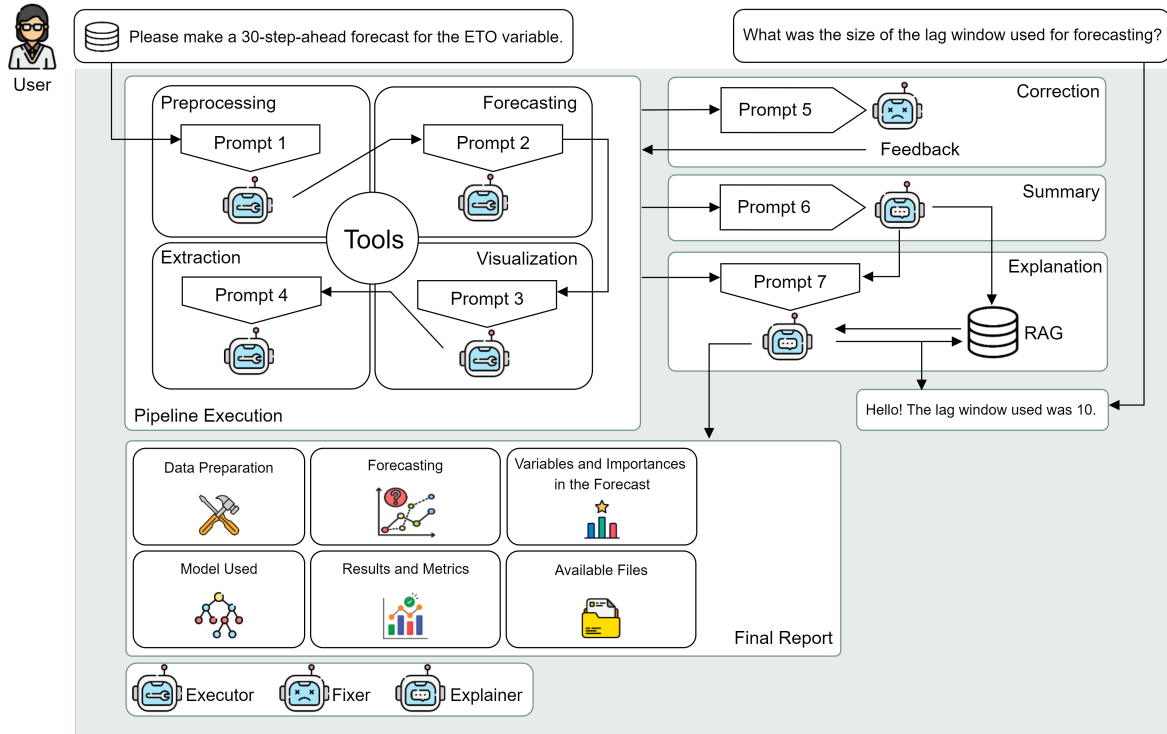


Figure 1: Illustrative diagram of FAE.

and increases the transparency of the decision-making process through iterative thought–action–observation cycles. Each prompt adopts the standard ReAct structure, followed by a description of the task corresponding to a specific stage of the pipeline.

Preprocessing: The agent is instructed, through Prompt 1, to check for the existence of missing values in the dataset and, if necessary, to automatically select an appropriate tool for imputation.

Forecasting: Through Prompt 2, the agent performs forecasting according to the user request, which must specify the target variable and the forecasting horizon. Based on this information, the agent configures AutoDCE-TS by defining the lag window size (bounded between 5 and 50) and deciding on the decomposition of the endogenous time series. Auxiliary tools may be used to analyze the input series and support these decisions.

Result visualization: In this stage, guided by Prompt 3, the agent generates a comparative plot between the AutoDCE-TS forecasts and the actual values.

Information extraction: Through Prompt 4, the agent extracts information from the forecasting process, including the causal graphs used for variable and lag selection, as well as the models, hyperparameters, and the importance of each variable in forecasting the endogenous series.

The tools available to the Executor agent are described in Appendix B. The imputation tools were defined based on a survey of the main techniques for handling missing values in time series (Mara Ribeiro and Leite de Castro, 2022; Has-sankhani Dolatabadi et al., 2024), and their descriptions are provided to the agent to support appropriate selection at each stage.

3.2 Correction

The correction module is activated after each stage of the pipeline, using the Corrector agent, guided by Prompt 5. In this way, each step carried out by the Executor agent is analyzed, and if an error occurs, the Executor receives feedback incorporated into the prompt specific to that stage, enabling the task to be re-executed. This module is essential for automating error handling.

3.3 Summarization

The summarization module is activated after the approval of each pipeline phase in the correction stage, using the Explainer agent, guided by Prompt 6. In this way, each step conducted by the Executor agent is summarized and stored, so that the Explainer agent has access to a summarized database of what was performed at each stage of the pipeline. The summaries are important because LLMs have a context limit, that is, a restriction on the amount of text they can process simultaneously. By storing only the summaries, the Explainer agent can access this database in the subsequent stage (Explanation) without the need to reprocess lengthy texts.

3.4 Explanation

When the user request involves a new forecasting task, the explanation module is activated at the end of the pipeline. The Explainer agent, guided by Prompt 7, uses the summary database and the information returned by AutoDCE-TS to generate a complete report of the process, including aspects such as value imputation, defined parameters, selected variables, chosen models, and their respective hyperparameters.

If the user requests additional explanations about a previously executed pipeline, this module is activated in isolation, allowing direct interaction with the agent. For this purpose, the summaries, the final report, and other information extracted from the AutoML system are stored in a vector database, employing the Retrieval-Augmented Generation (RAG) technique. During the dialogue, the Explainer agent retrieves relevant information through semantic search, enabling accurate responses.

The possibility of additional queries is essential, as reports generated by agents are not fully deterministic and may omit relevant details, especially in multivariate scenarios with a large number of variables and results.

4 Experiments

This section presents the evaluations conducted on the FAE multi-agent system:

- 1. Comparison between FAE and AutoDCE-TS:** The impact of LLM agent-based automation on the performance of the AutoDCE-TS method is evaluated. The comparison assesses the accuracy of point forecasts for 1, 15, and

30 steps ahead on four multivariate time series.

- 2. Case study:** The complete execution of FAE is examined based on six evaluation criteria, using four different LLM models. In addition, the overall final report generated by FAE is presented using the model that achieved the best performance in this evaluation.

4.1 Comparison between FAE and AutoDCE-TS

The experiments were conducted with the objective of comparing the results obtained by AutoDCE-TS in the publication (Lucas et al., 2025) with those obtained by AutoDCE-TS configured by FAE. In the paper, the authors present results with and without decomposition and conclude that the use of decomposition yields better performance, therefore recommending its adoption. FAE was configured with the *meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8* model¹, with temperature set to 0 and a maximum limit of 1024 tokens. The search space available for the agent to define the lag window size was the interval [1, 50]. The experiments were carried out by splitting the time series into 10 windows and forecasting horizons of 1, 15, and 30, following the same experimental methodology described in (Lucas et al., 2025). The selected datasets were: Bitcoin, Eolic, Air, and ETO. A summarized description of these datasets is presented in Table 1.

The metric used to compute forecasting accuracy in the experiments was the Normalized Root Mean Squared Error (NRMSE), calculated according to Equation (2), where y_{max} and y_{min} are the maximum and minimum values of the test dataset.

$$NRMSE = \frac{\sqrt{\sum_{t=0}^n (y(t) - \hat{y}(t))^2}}{y_{max} - y_{min}} \quad (2)$$

The results obtained in this evaluation are presented in Table 2. For these experiments, AutoDCE-TS and AutoCE-TS were configured by the authors with a window size of 20, with and without decomposition, respectively. FAE, in turn, performed the configuration according to each data window, consistently opting for the use of decomposition when the endogenous time series was stationary. In most cases, the lag window size was

¹<https://github.com/marketplace/models/azureml-meta/Llama-4-Maverick-17B-128E-Instruct-FP8>

Table 1: Description of the datasets.

Dataset	Granularity	Number of samples	Number of variables	Target
Bitcoin	daily	8,469	6	Dow Jones average index
Eolic	hourly	43,802	9	Wind power generation
Air	hourly	35,052	10	Air quality
ETO	daily	7,307	7	Evapotranspiration

defined based on a trade-off between accuracy and computational cost.

For the Bitcoin dataset, FAE chose to use decomposition in all windows, while selecting lag window sizes between 10 and 20. These changes resulted in improvements across all forecasting horizons when compared to AutoDCE-TS.

For the Eolic dataset, FAE opted to use decomposition with lag windows of size 20. Only in two windows were sizes 24 and 10 used. These changes did not lead to differences in performance for the 1-step forecasting horizon, but resulted in worse and better performance for the 15- and 30-step horizons, respectively.

For the Air dataset, FAE chose not to use decomposition and maintained, in most cases, lag windows of size 20, except for one window in which a size of 30 was adopted. This change improved only the 1-step forecasting horizon.

For the ETO dataset, FAE used decomposition in only 3 out of the 10 data windows and adopted lag window sizes between 10 and 15. These changes did not result in significant improvements, keeping the NRMSE values close to those obtained with the manual configuration.

Based on these results, it can be observed that the agent adequately fulfills its automation role, being able to trigger the appropriate tools, evaluate their respective outputs, and reason about them to configure the AutoML hyperparameters. Moreover, the performance achieved by the automated AutoML was not inferior to that obtained through manual configuration.

4.2 Case Study

In this case study, we demonstrate how FAE performs the forecasting of reference evapotranspiration (ETO) using the ETO dataset. The objective is to highlight the agents' decision-making process, from data preparation to the generation of final explanations. In addition, a qualitative evaluation of the actions taken throughout the process and of the final report presented to the user is carried out.

The evaluation was conducted through a com-

parison among four models designed for operation in agent-based systems: *meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8*² (Llama), with 17 billion parameters; *NousResearch/Hermes-3-Llama-3.1-70B*³ (Hermes), with 70 billion parameters; *anthropic/claude-4-opus*⁴ (Claude), whose number of parameters is not disclosed; and *deepseek-ai/DeepSeek-V3.1-Terminus*⁵ (DeepSeek), with 671 billion parameters.

For each model, the complete history of actions performed by the Executor agent and the report produced by the Explainer agent were simultaneously presented to ChatGPT⁶, which was assigned the task of scoring performance based on six criteria:

- Process Coherence:** does the Executor agent correctly follow the order of the pipeline stages? 378
- Technical Correctness and Decision Justification:** does the Executor agent present technically correct and well-grounded reasoning? 381
- Quality of Tool Usage:** does the Executor agent use each tool appropriately, including correct parameter configuration? 382
- Consistency of the Final Report:** does the Explainer agent faithfully describe the actions executed by the Executor agent? 384
- Quality of Communication:** does the Explainer agent produce a clear, coherent, and informative report? 385
- Operational Efficiency:** does the Executor agent avoid redundant or unnecessary steps? 386

Table 3 presents a comparative evaluation of the Claude, DeepSeek, Hermes, and Llama mod-

²<https://github.com/marketplace/models/azureml-meta/Llama-4-Maverick-17B-128E-Instruct-FP8>

³<https://huggingface.co/NousResearch/Hermes-3-Llama-3.1-70B>

⁴<https://www.anthropic.com/claude/opus>

⁵<https://api-docs.deepseek.com/news/news250922>

⁶<https://chatgpt.com/>

Table 2: NRMSE forecasting results obtained by FAE, AutoDCE-TS, and AutoCE-TS on the Bitcoin, Eolic, Air, and ETO datasets.

	Bitcoin			Eolic			Air			ETO		
	1	15	30	1	15	30	1	15	30	1	15	30
AutoDCE-TS	0.22	0.25	0.30	0.08	0.37	0.51	0.16	0.36	0.38	0.23	0.26	0.38
AutoCE-TS	0.02	0.03	0.09	2.1	6.2	2.2	0.1	0.49	0.51	1.6	2.0	0.57
FAE	0.04	0.21	0.24	0.09	0.49	0.41	0.05	0.40	0.47	0.28	0.25	0.40

els considering six criteria related to agent performance throughout the complete time series forecasting workflow: process coherence, technical correctness, tool usage, operational efficiency, consistency between reasoning and the final report, and quality of communication. These criteria reflect both the technical capabilities of the models and their ability to execute the task in a structured manner aligned with the system’s objective.

According to the results, the Claude model achieved the best overall performance, with a final average score of 8.83, standing out particularly in the criteria of technical correctness (10) and quality of communication of the final report (9). This suggests that, despite its complexity and larger number of parameters, the model is able to technically justify its decisions and produce clear and well-structured reports. According to the analysis performed by ChatGPT, the points penalized for the Claude model were mostly due to a certain level of excessive verbosity, repetition of simple actions such as printing tables, and some unnecessary result outputs.

The other three models achieved similar final average scores, but their individual scores differ considerably. Llama received a low evaluation score because, although it follows a basic pipeline flow and correctly uses the available tools, the presented reasoning is poorly structured, with imputation decisions made without prior data exploration and insufficient technical justification, including conceptual misconceptions about stationarity and a lack of grounding in the choice of the imputation method. In addition, the overall analysis is superficial and does not deeply investigate the problem.

DeepSeek exhibits issues similar to those of Llama during processing, but to a lesser extent; however, its score drops notably with respect to the final report. The model shows adequate isolated performance, but when compared to the others, its overall result is inferior, especially due to limited technical depth, restricted explanation of the results, and a less consistent reasoning structure.

The agent using the Hermes model presented

solid performance, with a clear and efficient pipeline. The technical correctness is good, demonstrating understanding of stationarity, lag structures, and different imputation methods; however, the choice of mean imputation was neither justified nor based on data exploration, which weakens the reasoning. The high operational efficiency came at the cost of reduced analytical depth.

Overall, the analysis shows that larger models tend to exhibit more solid technical reasoning and more consistent justifications, as well as greater fidelity between the process and the final report. However, smaller models demonstrate higher operational efficiency, possibly due to a lower tendency to explore unnecessary parallel paths. Another relevant aspect is that communication quality does not depend linearly on model size, which may indicate that instruction alignment mechanisms and text generation optimizations play a significant role.

The following presents the descriptions of the user requests, as well as the final report generated by the Claude model, which achieved the highest score in the evaluation. In addition, an example of a request regarding information on exogenous variables that were not present in the final report is included, but which were retrieved through RAG.

The report highlights FAE’s ability to statistically analyze time series, identify relevant patterns (such as stationarity and autocorrelation), select appropriate preprocessing techniques based on data characteristics—such as the justified choice of the imputation method—and make informed decisions regarding model configuration, including the definition of the number of lags, the need for decomposition, and the selection of the algorithm and its hyperparameters. Furthermore, the system provides detailed explanations of the selected variables, their importance, and the model’s performance, demonstrating not only the automation of the process but also the explanatory and analytical capabilities of the proposed method.

User request for the forecasting task:

Model	Process Coherence	Technical Correctness	Tool Usage	Operational Efficiency	Final Report Consistency	Final Report Quality	Final Average
Claude	9	10	9	8	8	9	8,83
DeepSeek	8	7	8	9	5	6	7,16
Hermes	7	7	8	9	6	8	7,5
Llama	6	6	7	8	7	8,5	7,08

Table 3: Scoring results of the Claude, DeepSeek, Hermes, and Llama models across the six evaluated criteria.

Perform a 5-step-ahead forecast for the ETO column. The dataset contains daily climate data. The ETO column represents daily reference evapotranspiration (mm).

Final report:

Hello! I will explain the entire analysis and forecasting process that was carried out using your evapotranspiration (ETO) data.

Data Analysis and Preparation

We began by analyzing the dataset to ensure that it was ready for modeling. During this initial analysis, it was found that there were some missing values in the maximum temperature (Tmax) column—specifically, 20 missing values out of a total of 2,000 records, which represents only 1% of the data.

To better understand how to handle these missing values, their locations were investigated, and it was observed that they were dispersed throughout the dataset rather than being consecutive. The statistical characteristics of maximum temperature were also analyzed, revealing moderate variability (approximately 13.77%) with smooth transitions between adjacent values.

Based on these characteristics—dispersed missing values, smooth transitions, and the natural gradual behavior of temperature—it was decided to apply linear interpolation to fill in the missing values. This method is appropriate for temperature data, as it preserves local trends and maintains the natural continuity of climatic data.

Forecasting Process with AutoDCE-TS

To generate the forecasts, **AutoDCE-TS** (Automated Machine Learning Based on Decomposition, Causality, and Evolutionary Multitask Optimization for Time Series Forecasting) was used, an advanced automated machine learning system specialized in time series.

Before applying the model, a detailed analysis of the ETO series was performed. First, the stationarity of the series was tested (that is, whether its statistical properties remain constant over time), and it was confirmed that the series was stationary. Next, the autocorrelation of the data was analyzed to determine the number of past values that influence future values. Significant correlation was identified up to 31 temporal lags, with the strongest correlations occurring within the first 6–7 periods.

Based on this analysis, the AutoML system was configured to use up to 15 temporal lags, seeking a balance between capturing relevant patterns and maintaining computational efficiency. As the series was stationary, decomposition was not required.

Variables and Their Importance in Forecasting

The model automatically selected the most relevant variables and lags to forecast ETO. The variables used

and their respective importance scores are presented below:

Most important variables (present across multiple horizons)

- Maximum temperature at time $t - 1$: 0,263
- Solar radiation at time $t - 1$: 0,199
- Relative humidity at time $t - 3$: 0,044

Other relevant variables

- Maximum temperature at time $t - 3$: 0,073
- Maximum temperature at time $t - 2$: 0,067
- Solar radiation at time $t - 3$: 0,062
- ETO at time $t - 2$: 0,061
- Precipitation at time $t - 1$: 0,047
- Relative humidity at time $t - 1$: 0,044
- ETO at time $t - 3$: 0,043

These results indicate that maximum temperature and solar radiation are highly influential factors in evapotranspiration forecasting, which is consistent with the physical behavior of the process.

Model Used

For forecasting the ETO variable, the system selected the **XGBoost** model with the following optimized hyperparameters:

- Number of estimators: 75
- Minimum samples per leaf: 2
- Maximum number of *features*: 0.2 (20% of the variables in each tree)

Results and Metrics

The forecasts for the next 5 periods were:

$$\text{Forecasts} = [4.615, 3.858, 4.312, 4.003, 3.815]$$

The model performance was evaluated using the following metrics:

- **RMSE**: 0.525
- **NRMSE**: 0.750 (75%)
- **MAE**: 0.485
- **MAPE**: 10.58%

These results indicate that the model achieves reasonable accuracy, with a mean percentage error of 10.58%.

Available Files

You have access to the following files:

1. **modelo.pickle**: Contains the complete model. You can access:
The trained model for each variable:
`modelo.dict_variables['variable name']`
The selected variables:
`modelo.G_list['variable name']`
2. **grafico.jpg**: Plot comparing actual and predicted values.
3. **grafo.jpg**: Causal graph displaying relationships between variables and their importance.

We hope this analysis has been useful. The results show that the model satisfactorily captured temporal patterns and interactions among climatic variables to forecast evapotranspiration.

User request for additional explanations:

Which model and hyperparameters were used to predict Tmax?

Response from the AutoDCE-TS-Agentic:

To predict Tmax, the XGBoost model was used with the following hyperparameters: 75 estimators, `min_samples_leaf=2`, and `max_features=0.2`. The variables selected for the model were: Rs (lag 3), Tmax (lags 1, 2, 3, 11), RH (lags 1, 3), and pr (lag 3).

5 Conclusion

This work presented FAE, a multi-agent system designed to automate and explain multivariate time series forecasting workflows based on AutoDCE-TS. By integrating LLM-based agents for execution, error correction, summarization, and explanation, FAE enables a complete workflow that combines automation with interpretability. The experimental evaluation showed that workflows configured by FAE achieve performance comparable to manual configurations, without degrading predictive accuracy. The case study further highlighted how different LLMs influence reasoning quality, operational efficiency, and clarity of explanations, indicating that model choice plays a critical role in agent-based AutoML systems. Overall, the results reinforce the potential of LLM-based multi-agent architectures as a promising direction for the automatic configuration and explanation of time series forecasting systems.

Limitations

This section describes the main limitations of the proposed multi-agent system, named FAE. First, the experiments showed that the quality of the analyses performed on the input time series is strongly dependent on the Large Language Model (LLM) employed. Smaller models tend to strictly adhere to the explicit instructions provided in the prompt, exhibiting limitations in decision-making when more in-depth data analyses are required. In contrast, larger models demonstrate a greater ability to explore time series data, enabling more well-founded and justified choices throughout the analytical process.

In addition, the quantitative evaluation of FAE proved to be limited by the absence of comparable agentic AutoML frameworks specifically designed for handling multivariate time series. This gap in the state of the art restricts the possibility of conducting direct and more comprehensive comparisons with existing approaches.

Acknowledgments

References

- Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. 2019. *GluonTS: Probabilistic Time Series Models in Python*. *CoRR*, abs/1906.05264.
- Moez Ali. 2020. PyCaret: An open source, low-code machine learning library in Python.
- Yi-Wei Chen, Qingquan Song, and Xia Hu. 2019. *Techniques for Automated Machine Learning*.
- Raphael Fischer and Amal Saadallah. 2024. *AutoX-PCR: Automated Multi-Objective Model Selection for Time Series Forecasting*. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 806–815, New York, NY, USA. ACM.
- Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. 2024a. *DS-Agent: Automated Data Science by Empowering Large Language Models with Case-Based Reasoning*. In *Proceedings of Machine Learning Research*, volume 235, pages 16813–16848. ML Research Press.
- Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024b. *Large Language Model Based Multi-agents: A Survey of Progress and Challenges*. pages 8048–8057. International Joint Conferences on Artificial Intelligence.

564	Sepideh Hassankhani Dolatabadi, Ivana Budinská, Rafe Behmaneshpour, and Emil Gatial. 2024. Closing the Data Gap: A Comparative Study of Missing Value Imputation Algorithms in Time Series Datasets . pages 77–90.	618
565		619
566		620
567		621
568		622
569	Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A Survey of the State-of-the-Art . <i>Knowledge-Based Systems</i> , 212.	623
570		624
571		625
572	Siyi Liu, Chen Gao, and Yong Li. 2024. Large Language Model Agent for Hyper-Parameter Optimization .	626
573		627
574		628
575	Patrícia de Oliveira e Lucas, Frederico Gadelha Guimarães, and Eduardo M.A.M. Mendes. 2025. Automated machine learning based on decomposition, causality and evolutionary multitask optimization for time series forecasting . <i>Neurocomputing</i> , 656:131569.	629
576		630
577		631
578		632
579		633
580		634
581	Yuan Luo, Song Han, and Jingjing Liu. 2023. Machine learning’s own Industrial Revolution .	635
582		
583	Silvana ID Mara Ribeiro and Cristiano ID Leite de Castro. 2022. Missing Data in Time Series: A Review of Imputation Methods and Case Study . Technical Report 1.	636
584		637
585		
586		
587	Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. 2024. The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey .	
588		
589		
590		
591	Nikolay O. Nikitin, Pavel Vychuzhanin, Mikhail Sarafanov, Iana S. Polonskaia, Ilia Revin, Irina V. Barabanova, Gleb Maximov, Anna V. Kalyuzhnaya, and Alexander Boukhanovsky. 2022. Automated evolutionary approach for the design of composite machine learning pipelines . <i>Future Generation Computer Systems</i> , 127:109–125.	
592		
593		
594		
595		
596		
597		
598	Gian Marco Paldino, Jacopo De Stefani, Fabrizio De Caro, and Gianluca Bontempi. 2021. Does AutoML Outperform Naive Forecasting? <i>Engineering Proceedings</i> , 5(1):36.	
599		
600		
601		
602	Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. 2024. AutoML-Agent: A Multi-Agent LLM Framework for Full-Pipeline AutoML .	
603		
604		
605	Sandra Vieira, Walter Hugo Lopez Pinaya, and Andrea Mechelli. 2020. Introduction to machine learning . <i>Machine Learning: Methods and Applications to Brain Disorders</i> , pages 1–20.	
606		
607		
608		
609	Chunnan Wang, Xingyu Chen, Chengyue Wu, and Hongzhi Wang. 2022. AutoTS Package .	
610		
611	Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, and 9 others. 2025. The rise and potential of large language model based agents: a survey .	
612		
613		
614		
615		
616		
617		
	Jinglue Xu, Jialong Li, Zhen Liu, Nagar Antheel Venkatesh Suryanarayanan, Guoyuan Zhou, Jia Guo, Hitoshi Iba, and Kenji Tei. 2024. Large Language Models Synergize with Automated Machine Learning .	618
		619
		620
		621
		622
	Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing Reasoning and Acting in Language Models .	623
		624
		625
		626
	Haokun Zhao, Xiang Zhang, Jiaqi Wei, Yiwei Xu, Yuting He, Siqi Sun, and Chenyu You. 2025. TimeSeriesScientist: A General-Purpose AI Agent for Time Series Analysis .	627
		628
		629
		630
	Marc-André Zöller and Marco F. Huber. 2021. Benchmark and Survey of Automated Machine Learning Frameworks . <i>Journal of Artificial Intelligence Research</i> , 70:409–474.	631
		632
		633
		634
	A Appendix A	635
	This appendix presents the prompts used to guide the FAE agents.	636
		637
	Prompt 1	
	You are an assistant specialized in time series preprocessing, with a focus on missing value imputation. Rules:	
	1. Check all columns of the dataframe to detect missing values.	
	2. If missing values exist, use the most appropriate tool for imputation.	
	Pay attention to the following observation: <i>{feedback from the Reviewer.}</i>	
	Prompt 2	
	You are an assistant specialized in time series forecasting. Your only task is to generate forecasts for the problem described by the user: user message. Rules:	
	1. Automatically define, always with a brief justification: - The maximum number of lags (<code>'max_lags'</code>), choosing between 5 and 50 to balance accuracy and computational cost. - Whether decomposition (<code>'decomposition'</code>) should be used or not.	
	2. Use the tool <code>'testar_estacionariedade'</code> if it is necessary to decide about stationarity.	
	Pay attention to the following observation: <i>{feedback from the Reviewer.}</i>	
	B Appendix B	638
	This appendix describes the tools made available to the Executor agent.	639
		640
	Linear interpolation: applies a linear function between adjacent points to estimate missing	641
		642

Prompt 3

You are an assistant specialized in time series visualization.

Rules:

1. Focus only on the data visualization task.
2. The dataset has two columns corresponding to the real and predicted values.
2. Generate a plot comparing the real and predicted values.
3. Return the figure in Base64 format ready for use, without explanations or additional text.

Pay attention to the following observation: *{feedback from the Reviewer.}*

Prompt 4

You are a model analysis assistant.

Rules:

1. Extract information generated by the AutoML model using the tool `extrair_informacao_automl`.
2. Respond only in the format of a valid Python dictionary.
3. Do not include explanations, comments, or text outside the dictionary.

The tool will return the following technical information within a dictionary:

- The dictionary 'modelo_dict' contains a key for each variable. Each variable was predicted by a specific model, with its associated hyperparameters.

- The causal graphs generated for the variables are in the dictionary 'grafo'. Each key corresponds to a variable, and the value is a dataframe where True indicates that the variable (column) was selected at the corresponding lag (row).

Your task:

Based on this information, generate a structured response in the format: `{{ "modelos": {{ "variable1": "Short text describing the model and its hyperparameters.", "variable2": "Short text describing the model and its hyperparameters." }}, "variaveis_selecionadas": "variable1": "Short text describing the selected variables and their lags.", "variable2": "Short text describing the selected variables and their lags." }}}`

Objective: Generate a compact and informative dictionary, with no extra text outside the requested structure. Pay attention to the following observation: *feedback from the Reviewer.*

Prompt 5

You are an error corrector for an agent that uses the following tools: *{list of tools}*.

During the stage stage name, the agent executed the following actions: *{log of the Executor's actions}*, but an error occurred in the last action, interrupting the execution.

Your task is:

1. Identify which was the last action performed before the error.
2. Analyze the format of the call made to that tool and verify whether it complies with the expected format.
3. Briefly explain to the agent the correct format of the tool call.

Return only a valid JSON object, without code blocks, without explanatory text, and without markdown. The format must be exactly one of the following two:

`{ "avaliacao": "no", "feedback": "Describe here what the agent did wrong and how to correct the tool call." }`
Do not add anything before or after the JSON.

Prompt 6

You are the Summarizer Agent. You will receive two blocks below: - steps: alternating lines following the pattern 'THOUGHT: ...' and 'ACTION: tool(args)'

OBJECTIVE Write a report in pt-BR exactly in the following format:

Name of the step performed in Portuguese: short title in pt-BR.

Description of the Action: faithfully copy/summarize the corresponding THOUGHT, without inventing content or using 'none'.

Chosen Tool: name of the tool indicated in ACTION or 'none', parameters indicated in Action Input or 'none'.

Result: summarize the output associated with this step, without inventing information.

RULES

- Maintain the chronological order of the steps.
- Do not invent tools, data, or results.
- Be concise and technical; avoid embellishments.
- If there is no THOUGHT or ACTION for a given step, fill the missing field with '—'.
- If there is nothing to summarize, respond only with: Sem dados para resumir.

DATA: *{Executor actions}*

values, assuming a uniform transition between them. It is indicated when there is an approximately linear relationship between neighboring points and the missing values are sporadic, without long sequences of missing data.

Spline interpolation: uses smooth polynomial functions to approximate the data and compute missing values, resulting in a more continuous curve than linear interpolation. It is recommended for series with smooth nonlinear behavior, in which preserving the shape and smoothness of the time series is desired.

Mean: fills missing values using the global mean or window-based means of the time series itself. This technique is more suitable for approximately stationary series, that is, with a stable mean over time.

Nearest neighbor: replaces the missing value with the value from an adjacent temporal point. It is appropriate when there is low variation between consecutive observations and temporal proximity is a good indicator of the missing value.

AutoDCE-TS: a tool based on the Automated

655

656

657

658

659

660

661

662

663

664

665

666

Prompt 7

You will receive the following technical information:
- Pre-processing stage: {summary 1}. - Forecasting stage: {summary 2}. - Lists of variables/lags and their importances used to predict the variable {*name of the endogenous variable*}: {lags} - The dictionary {model_dict} contains one key for each variable. Each variable was predicted by a model and its associated hyperparameters. - The forecasts generated by the model: {forecasts} - The metrics calculated using the forecasts and the true values: {metrics}.

Your task:

- Generate an explanation with a friendly tone. - Explain what was done at each pre-processing and forecasting stage in a connected manner (not just as a list). - Inform the user that the AutoML used for forecasting is AutoDCE-TS. - If EMD decomposition was applied, explain in a simple way what this means, into how many IMFs the series was decomposed, and how the components (IMFs) were used. - Inform the user that they will have access to the file modelo.pickle. In this file, they can access the model chosen to forecast each variable, its hyperparameters via modelo.dict_variables['variable name']. The selected variables can be accessed via modelo.G_list['variable name']. - List all variables and lags used to forecast the variable {*name of the endogenous variable*}, as well as their respective importances for the model. Instead of saying "variable 1 (value)", say variable at time t-1: value. Discard any variable whose importance is -1. - List the model used to forecast the variable {*name of the endogenous variable*} and its hyperparameters. - Report the calculated metrics. - Explain to the user that they will have access to two images: (grafico.jpg) a graph comparing true and predicted values and (grafo.jpg) a causal graph with the selected variables and their respective importances for the model. - Inform the user of the forecasts generated by the model as a list. - Use a fluent text, as if you were speaking directly to the user, rather than merely repeating the received data.

importance of each variable returned by the estimator is also extracted.

Auxiliary tools: are tools that can help the agent make decisions. The stationarity test tool, for example, evaluates whether the time series is stationary based on the ADF and KPSS statistical tests. This tool reports that the series is stationary only if both tests confirm this property. In addition, the agent can use functions that display statistical information about the data, detect missing values, analyze variable names, among others.

667 Machine Learning Based on Decomposition,
668 Causality, and Evolutionary Multitask
669 Optimization for Time Series Forecasting
670 (AutoDCE-TS) method. AutoDCE-TS is a
671 multiple-input multiple-output (MISO) system
672 for multi-step-ahead forecasting, structured
673 as a fixed four-layer pipeline that automates
674 the stages of feature extraction and selection,
675 model selection and generation, and
676 forecasting.

677 **Plot:** presents the predicted values of the endogenous
678 time series in a graph, enabling comparative
679 analysis with the real data.

680 **AutoDCE-TS information extraction:** extracts
681 the causal graphs, models, and hyperparameters
682 of all variables in the forecasting problem.
683 For the endogenous time series, the relative