Gazi University
Journal of Science
PART A: ENGINEERING AND INNOVATION
http://dergipark.org.tr/gujsa

A No-Code Automated Machine Learning Platform for the Energy Sector

Ezgi AVCI[1*]

[1] TED University, Applied Data Science Department, Ankara, Türkiye

Keywords
Auto-ML
No-Code Platform
AI
Energy

Abstract
This paper presents a No-Code Automated Machine Learning (Auto-ML) platform designed specifically for the energy sector, addressing the challenges of integrating ML in diverse and complex data environments. The proposed platform automates key ML pipeline steps, including data preprocessing, feature engineering, model selection, and hyperparameter tuning, while incorporating domain-specific knowledge to handle unique industry requirements such as fluctuating energy demands and regulatory compliance. The modular architecture allows for customization and scalability, making the platform adaptable across various energy sub-sectors like renewable energy, oil and gas, and power distribution. Our findings highlight the platform's potential to democratize advanced analytical capabilities within the energy industry, enabling non-expert users to generate sophisticated data-driven insights. Preliminary results demonstrate significant improvements in data processing efficiency and predictive accuracy. The paper details the platform's architecture, including data lake and entity-relationship diagrams, and describes the design of user interfaces for data ingestion, preprocessing, model training, and deployment. This study contributes to the field by offering a practical solution to the complexities of ML in the energy sector, facilitating a shift towards more adaptive, efficient, and data-informed operations.

Cite
Avci, E. (2024). A No-Code Automated Machine Learning Platform for the Energy Sector. *GU J Sci, Part A*, *11*(2), 289-303. doi:10.54287/gujsa.1473782

Author ID (ORCID Number)		Article Process	
0000-0002-9826-1027	Ezgi AVCI	**Submission Date**	25.04.2024
		Revision Date	10.05.2024
		Accepted Date	22.05.2024
		Published Date	04.06.2024

1. INTRODUCTION

The advent of machine learning (ML) has significantly transformed various sectors, with the energy industry standing out as a prime beneficiary of this technological revolution. The deployment of ML in energy encompasses a wide range of applications, from optimizing renewable energy production to predictive maintenance of infrastructure. However, the complexity of developing and deploying effective ML models can be a significant barrier, particularly in an industry that traditionally has not focused on software development. This is where Automated Machine Learning (Auto-ML) comes into play, providing tools and frameworks that automate the ML pipeline from data preprocessing to model selection and tuning.

The need for an Auto-ML platform specifically tailored for the energy sector is driven by unique challenges such as high variability in data types, stringent reliability requirements, and the critical nature of operational efficiency. An effective Auto-ML platform for this sector must not only automate the model development process but also ensure that the models are scalable, robust, and compliant with industry-specific regulations and standards.

Our research question is *Can a No-Code Auto-ML platform specifically tailored for the energy sector effectively address the unique challenges faced by energy industry professionals, thereby enhancing operational efficiency and decision-making processes?* with the hypothesis: *A No-Code Auto-ML platform*

*Corresponding Author, e-mail: ezgi.avci@tedu.edu.tr

designed with domain-specific knowledge and tailored features for the energy sector will democratize access to advanced machine learning capabilities, leading to significant improvements in data processing efficiency, predictive accuracy, and overall operational efficiency, even for users with limited technical expertise.

This paper introduces a novel Auto-ML platform designed specifically for the energy industry. We discuss the core components of the platform, including the architecture, entity-relationship (ER) diagram, data lake architecture, ML approach, and design interfaces for the ML pipeline. The objective is to offer a comprehensive tool that empowers energy sector professionals with little to no ML expertise to leverage predictive analytics, thereby enhancing decision-making and operational efficiency.

2. THEORETICAL BACKGROUND

2.1. Auto-ML Pipeline

2.1.1. Structure

The construction of an ML pipeline begins with establishing its structure. The process starts with cleaning the input data through various steps. Subsequently, feature selection and generation are tailored to the specific domain of application. Ultimately, the model is trained using the processed features. In practice, experienced data scientists often modify and expand this basic structure to suit specific needs.

Most Auto-ML frameworks utilize a fixed pipeline structure (McGushion, 2019): data cleaning, feature selection, and a modeling phase. The preprocessing phase typically involves selecting one of several well-known algorithms, such as matrix decomposition techniques. Although the data cleaning steps often include imputation and scaling, certain steps may be unnecessary depending on the dataset, such as imputation in a dataset without missing values. By adhering to a fixed structure, the complexity of designing a pipeline is reduced, as it primarily involves choosing suitable preprocessing and modeling algorithms. However, this rigidity can sometimes result in suboptimal performance for complex datasets that may require more nuanced preprocessing.

To achieve optimal results, data science experts often design highly customized pipelines for specific ML tasks, which is not feasible with rigid pipeline structures. The initial approach to generating flexible ML pipelines using genetic programming was introduced by Olson & Moore, (2016), where pipelines are treated as tree structures and modified through the combination and mutation of sub-graphs (Koza, 1994; Banzhaf, 2006).

Hierarchical task networks simplify complex problems into smaller, manageable sub-problems until only simple operations remain. This approach reduces pipeline structure design to identifying the optimal endpoint in the network (Mohr et al., 2018). Similarly, Monte-Carlo method, a heuristic search algorithm, progressively builds more complex pipelines, aiming to find the best performing node in the search tree (Kocsis & Szepesvari, 2006; Browne et al., 2012; Rakotoarison et al., 2019).

Self-play, a reinforcement learning strategy popularized by developments like AlphaZero, involves generating training data through self-competition (Lake et al., 2016; Silver et al., 2018). In pipeline structure search, this method treats the process as a game, where actions include adding, removing, or replacing nodes, and pipeline performance dictates the scoring (Drori et al., 2018).

Although these methods offer flexibility, they often overlook dependencies between different stages of the pipeline and overall constraints. To address potential issues, such as the creation of incomplete or inappropriate pipelines, restrictions can be imposed through a defined grammar, ensuring the generation of practical yet adaptable pipelines (Drori et al., 2019).

2.1.2. Automatic Data Preparation

Data collection is essential for creating or expanding datasets. Data cleaning involves filtering out noisy data to ensure it doesn't affect subsequent model training negatively. Data augmentation is crucial for increasing model robustness and performance, with further details on these stages provided in the following subsections. The significance of high-quality datasets in ML has been widely acknowledged, leading to the development of various open datasets. Finding appropriate datasets for specific tasks, such as those involving medical or privacy-sensitive data, can be challenging. Two approaches to address this are data searching and data synthesis. Web searching is common but faces issues like mismatched search results and poor data labeling. Noise in collected data can be detrimental to model training, necessitating robust data cleaning methods (Jesmeen et al., 2018). Techniques range from crowdsourcing to automated systems like Katara (Chu et al., 2015, 2016) and methods focusing on optimizing cleaning operations for efficiency (Krishnan & Wu, 2019). Continuous data generation poses challenges for ongoing data cleaning, which is crucial for enterprises managing daily data influxes. Systems that adapt from past cleaning experiences to optimize future workflows are being developed (Mahdavi et al., 2019).

Data augmentation (DA) is a significant component of data preparation, discussed separately due to its role in both generating new data and preventing model overfitting. Recent advances in DA focus on automating the selection of optimal DA policies using techniques like reinforcement learning and various optimization strategies (Cubuk et al., 2019; LingChen et al., 2020).

2.1.3. Feature Engineering

It is widely acknowledged that the potential of ML is fundamentally constrained by the quality of data and features, with models and algorithms serving merely to approximate this upper limit. Within this framework, feature engineering is critical for maximizing the utility of raw data for algorithmic and model use. Feature engineering encompasses three distinct phases:

- Feature selection involves creating a subset of features to minimize irrelevance and redundancy, which helps improve model performance and prevent overfitting. The selected features should be diverse and have a strong correlation with the target variables.
- Feature construction is the process of creating new features from basic feature spaces or raw data to improve the model's robustness and generalizability. This typically requires human expertise and includes preprocessing transformations like standardization, normalization, or feature discretization. Due to the impracticality of manually exploring all possible feature combinations, automated methods for feature construction have been developed (Gama, 2004; Zheng, 1998; Vafaie & De Long, 1998). These automated approaches help in searching and evaluating combinations of operations, achieving results that are often comparable to or better than manual techniques.
- Feature extraction focuses on reducing dimensionality by applying specific mapping functions to isolate informative and non-redundant features.

2.1.4. Model Generation and Evaluation

Model generation within AutoML is about automating the selection and configuration of machine learning algorithms. Traditional machine learning model development requires a practitioner to manually select an algorithm suitable for the problem at hand and then experiment with various hyperparameters to find the best configuration. This is a time-consuming and often technically challenging process, particularly in complex datasets and problem domains. AutoML simplifies this by using sophisticated algorithms to explore a wide range of machine learning models. This includes a variety of algorithm types such as:

- Decision Trees: These models perform classification or regression by making decisions based on the data features. They are easy to understand and interpret, making them a popular choice for many applications.
- Support Vector Machines (SVMs): SVMs are effective in high-dimensional spaces and are best known for their use in classification problems.
- Neural Networks: With their ability to model complex nonlinear relationships, neural networks are suited for tasks like image and speech recognition, where they can capture intricate patterns in data.

- Ensemble Methods: Techniques like Random Forests and Gradient Boosting Machines combine multiple models to produce one optimal predictive model, reducing the likelihood of overfitting and improving prediction accuracy.

AutoML tools typically use a search strategy such as grid search, random search, or more advanced methods like Bayesian optimization to systematically explore different models and their configurations. These strategies vary in their efficiency and effectiveness; for example, Bayesian optimization can more quickly generate optimal model parameters by building a probabilistic model of the function mapping hyperparameters to model performance.

Once models are generated, model selection stands as a pivotal stage where the optimal model, from those generated and evaluated earlier, is chosen for deployment. This process involves several intricate steps that ensure the selected model not only performs well statistically but also aligns with practical, operational criteria:

- **Ranking Models Based on Evaluation Metrics**

These metrics differ depending on the type of task—for instance, accuracy, precision, recall, and F1 score for classification tasks, or mean squared error and R-squared for regression tasks. In classification tasks, for instance, while accuracy might give a basic idea of performance, it could be misleading in cases of imbalanced classes, thereby making metrics like F1 score or AUC-ROC more reliable for assessing model quality. AutoML systems often employ sophisticated scoring systems that aggregate these metrics to provide a holistic assessment of each model's performance. This might involve weighting different metrics based on their relevance to the specific application or using statistical techniques to normalize scores across various metrics.

- **Considering Computational Efficiency and Complexity**

Beyond just raw performance metrics, AutoML systems also consider the computational efficiency and complexity of each model. This is crucial because a model that requires excessive computational resources for training or inference may not be practical, especially in environments with limited hardware capabilities.

  - **Computational Efficiency**: This involves evaluating the time and resources required for training and making predictions. A model that is slightly less accurate but much faster may be preferred over a top-performing model that is computationally expensive.
  - **Model Complexity**: Simpler models are often more robust and easier to maintain. They are also less likely to overfit the training data. AutoML systems can evaluate complexity based on the number of parameters, the depth of decision trees, or the architecture of neural networks.

- **Performance Criteria and Thresholds**

Model selection isn't merely about picking the highest-ranked model; it often involves meeting predefined performance criteria or thresholds. These criteria are set based on business objectives or the specific requirements of the task at hand. For example, in a medical diagnosis application, a high recall might be more important than achieving the highest possible accuracy.

- **Further Fine-Tuning**

Once a model or a set of top models is identified, AutoML systems might engage in further fine-tuning. This can involve adjusting hyperparameters, scaling up the model with more data, or simplifying the model to improve computational efficiency without significantly affecting performance. Techniques such as Bayesian optimization are often employed here to find the optimal configuration with fewer iterations.

- **Validation and Testing**

Before deployment, the selected model undergoes rigorous validation and testing to ensure that it performs consistently well on unseen data. This often includes testing on a separate test set that was not used during the model training or evaluation phases. It helps confirm that the model's performance metrics are genuinely indicative of its real-world capabilities.

- **Deployment Considerations**

Finally, deployment considerations such as integration with existing systems, the need for real-time versus batch processing, and the operational environment (cloud, on-premises, edge devices) are taken into account. The model's architecture might be adjusted to fit into the deployment environment, ensuring that it delivers the expected performance in the real world.

2.2. Auto-ML Tools

In this section, features and functionalities of some widely used Auto-ML platforms such as Auto-sklearn, Auto-Keras, TPOT, Auto-WEKA, H2O Auto-ML,Amazon Sagemaker, and AutoGluon (Thornton et al., 2013; Feurer et al., 2019).

2.2.1. Auto-sklearn

Auto-sklearn is an open-source tool which utilizes the established Scikit-Learn package in Python (Pedregosa et al., 2011). This tool automatically identifies the most suitable learning algorithm for a new dataset and fine-tunes its hyperparameters. Additionally, Auto-sklearn employs Bayesian optimization techniques in its operations.

2.2.2. Auto-keras

Auto-Keras, also open-source, is constructed atop the Keras library and specializes in automatic neural network design, particularly for text and image data (Jin et al., 2019). It simplifies complex tasks such as embeddings and dimensionality reduction, which are typically challenging for novices in deep learning. Auto-Keras also performs automatic neural architecture optimization, adjusting aspects like neuron counts and layer configurations. Models developed with Auto-Keras function as standard TensorFlow/Keras models, handling many preprocessing tasks automatically, such as text data vectorization and cleaning (Jin et al., 2023).

2.2.3. TPOT

TPOT (Tree-based Pipeline Optimization Tool), is an open-source which utilizes genetic programming to determine the most effective model pipelines. It symbolically represents a model's pipeline, including all stages from data preparation to modeling. TPOT emphasizes ease of use with minimal user input, generating sophisticated, high-performance ML pipelines (Olson et al., 2016).

2.2.4. Auto-WEKA

Auto-WEKA leverages the classification and regression algorithms from the WEKA platform, an established open-source ML framework known for its user-friendly interface (Kotthoff et al., 2019). It is designed to help beginners by automatically tuning WEKA's algorithms and hyperparameters using Bayesian Optimization, making it as approachable as other learning algorithms (Kotthoff et al., 2016).

2.2.5. H2O

H2O, both a commercial and open-source tool, excels in detecting data schemas and analyzing results comprehensively (Darren, 2016). It is particularly adept at handling time series data, automating feature engineering, and creating efficient pipelines. H2O Auto-ML performs iterative model comparisons to identify the best model. It also has a recommender that provides data-driven recommendations tailored to business needs (Del & Poirier, 2020).

2.2.6. Amazon sageMaker

Introduced by AWS, Amazon SageMaker Autopilot is a cloud-based platform designed to work seaML essly with well-known deep-learning methods (Iwendi et al., 2022). It automatically identifies the suitable model for any given dataset. This platform significantly simplifies the process of building ML models by autonomously searching for the best models and providing an explainability report that helps users understand and evaluate the models it develops (Park et al., 2022).

2.2.7. AutoGluon

AutoGluon is an advanced Auto-ML platform known for its automated data preprocessing, model architecture exploration, and hyperparameter tuning, which together enhance model performance and predictive accuracy (Ji et al., 2022; Erickson et al., 2020; UC., 2023). It features a diverse array of ML models. AutoGluon is designed to efficiently determine the most suitable model for specific applications, thereby optimizing performance and accuracy (UC., 2023). AutoGluon employs a multi-layer stacking strategy that uses the output of one layer of models as the input for the next layer, continually building complexity. This methodology not only uses diverse model types within each layer but also combines their outputs to create new feature sets for subsequent layers, enhancing the model’s ability to learn from the data. In its final stages, AutoGluon implements ensemble models to prevent overfitting (Ge, 2020).

3. PROPOSED AUTO-ML PLATFORM

3.1. Challenges of Developing Auto-ML Solutions in Energy Sector

Despite advances in Auto-ML, its implementation in the energy sector is still limited. This shortfall is due to specific challenges within the energy domain that obstruct the adoption of Auto-ML Technologies:

- Developing a high-quality dataset:

It's crucial for models to be trained on data that reflects the format and quality expected in real-world applications. However, applying automated feature engineering to energy data, which is often highly noisy, can be challenging and lead to less than optimal outcomes.

- Black-box property of Auto-ML systems:

They do not provide clear visibility into their decision-making processes, such as the exploration of different model configurations, which can lead to mistrust among both expert and novice users. This distrust is especially acute in the critical areas of energy operations and infrastructure, where the clarity and interpretability of algorithms are vital for their integration into existing workflows.

- Big-Data:

The current methods for optimizing ML pipelines struggle to handle the large datasets that are typical in the energy sector, further limiting the presence of Auto-ML solutions in this area.

This paper seeks to overcome these challenges by creating an Auto-ML platform specifically designed for the energy industry, capable of handling the vast and diverse data streams that characterize today's energy data. Through iterative consultations with professional traders and data scientists in the energy field, essential data sources and technical needs have been pinpointed. The ability to efficiently process these data sources at high speeds is critical for improving forecasting accuracy. The proposed platform will focus on streaming data analytics, requiring continuous data access, scalability for handling substantial data volumes, and stringent data quality management.

3.2. Platform Architecture

The platform architecture (Figure 1) describes the key components and how they interact to automate the ML process. It details the data flow from ingestion to model deployment, including the integration of domain-specific modules for handling energy data. The architecture is designed to be scalable and flexible, supporting various types of energy data and ML algorithms.

3.3. Auto-ML Entity-Relationship (ER) Diagram

The ER Diagram (Figure 2) provides a visual representation of the data relationships within the Auto-ML platform. This diagram shows how different data entities such as datasets, models, features, and results are interconnected. The ER diagram serves as a blueprint for database design and helps in ensuring that the data management component of the platform is robust and efficient, catering to the needs of model training, evaluation, and deployment.

3.4. Data Lake Architecture

Although distribution companies actively collect data from different systems and many similar data sources, most of these systems work asynchronously and are stored in independent databases. Therefore, the interaction between the collected data is extremely weak. The proposed data lake architecture (Figure 3) enables many different and discrete types of data sources can be integrated with each other. SQL and NoSQL based hybrid, integrated database infrastructure. Users easily integrate their data, can get started quickly with ML.

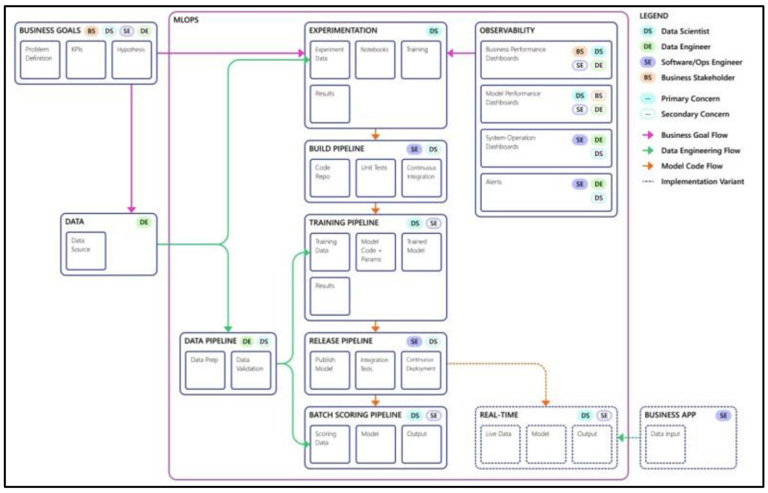


Figure 1. *Platform Architecture*

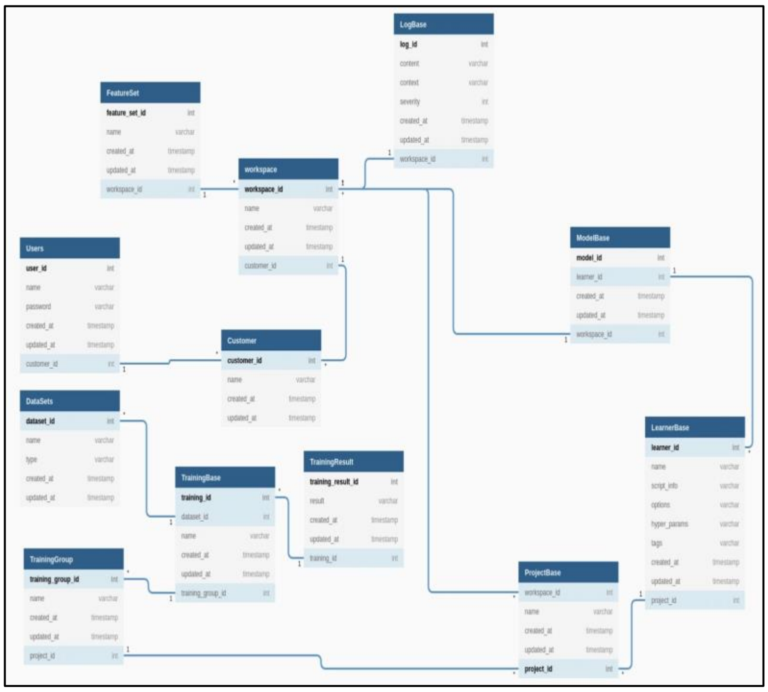


Figure 2. *ER Diagram*

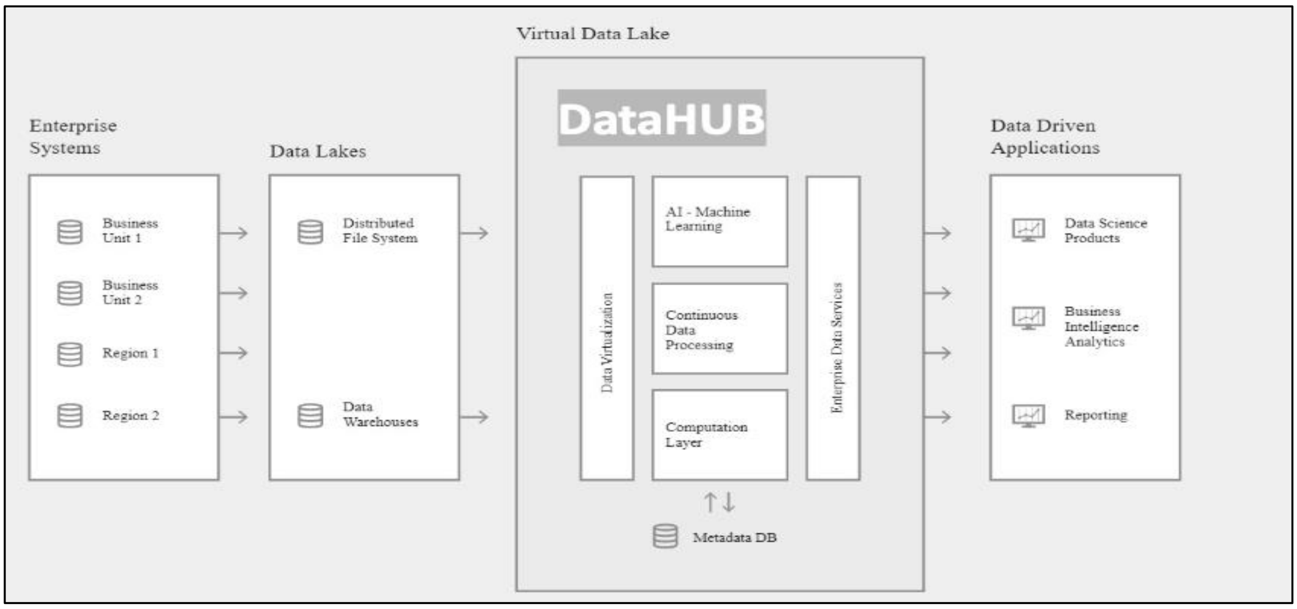
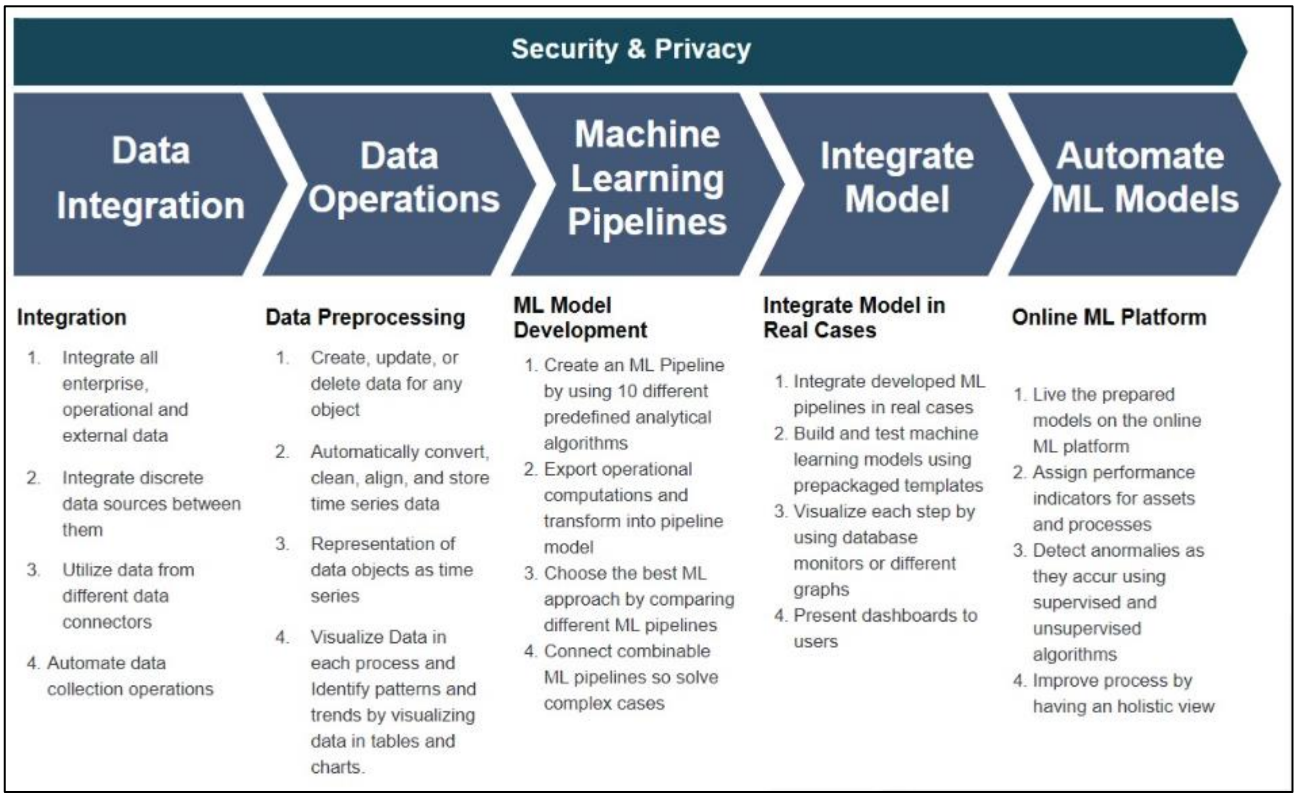


Figure 3. *Data Lake Architecture*

3.5. ML Approach

Our proposed Auto-ML platform, at the macro-level uses a model-centric strategy (Figure 4) to automate the selection, training, and optimization of ML models. This approach focuses on selecting the most appropriate models based on the characteristics of the data and the specific requirements of the energy sector tasks at hand. It includes the use of meta-learning to predict the best models and configurations, and how the platform adapts its strategies based on feedback from ongoing operations.



Integration

1. Integrate all enterprise, operational and external data
2. Integrate discrete data sources between them
3. Utilize data from different data connectors
4. Automate data collection operations

Data Preprocessing

1. Create, update, or delete data for any object
2. Automatically convert, clean, align, and store time series data
3. Representation of data objects as time series
4. Visualize Data in each process and Identify patterns and trends by visualizing data in tables and charts.

ML Model Development

1. Create an ML Pipeline by using 10 different predefined analytical algorithms
2. Export operational computations and transform into pipeline model
3. Choose the best ML approach by comparing different ML pipelines
4. Connect combinable ML pipelines so solve complex cases

Integrate Model in Real Cases

1. Integrate developed ML pipelines in real cases
2. Build and test machine learning models using prepackaged templates
3. Visualize each step by using database monitors or different graphs
4. Present dashboards to users

Online ML Platform

1. Live the prepared models on the online ML platform
2. Assign performance indicators for assets and processes
3. Detect anormalies as they accur using supervised and unsupervised algorithms
4. Improve process by having an holistic view

Figure 4. *Model-driven ML Approach (Macro-Level)*

Our ML approach, at the micro-level (Figure 5), enables Utility Users to:

- Integrate data sources from our platform or other data service platforms.
- Easily preprocess raw data without writing code by using predefined blocks.
- Built ML pipelines using predefined ML algorithms.
- Visualize each step of data, input or output of pipeline in preview.
- Deliver AI insights and develop your operations with actionable and qualitative AI insights.

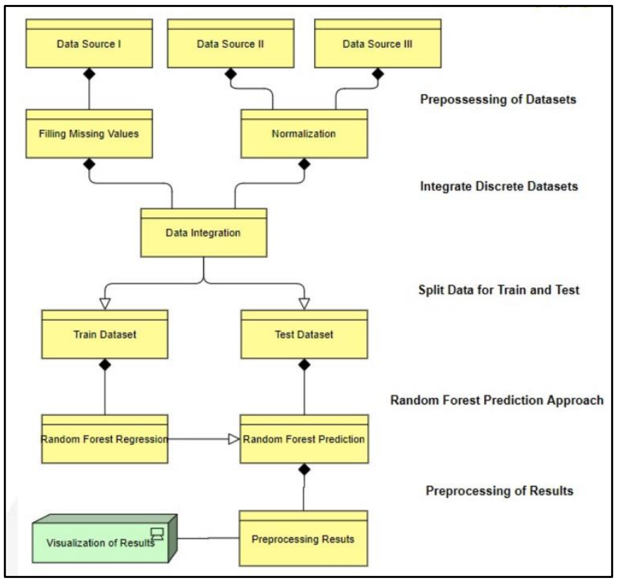
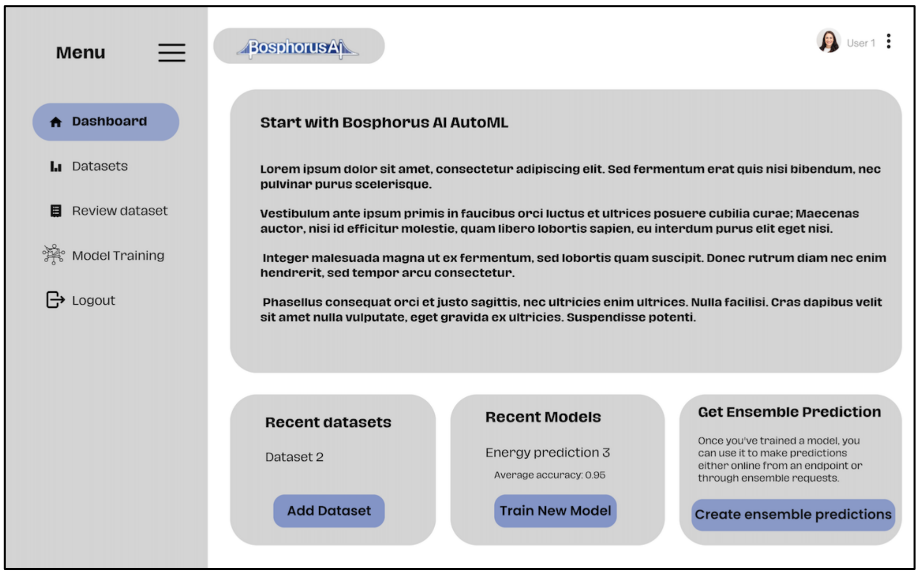


Figure 5. *ML Approach (Micro-Level)*

3.6. Design

3.6.1. Dashboard Interface

The dashboard interface (Figure 6) provides users with an overview of ongoing activities, performance metrics, and system statuses. It is designed for ease of use, allowing users to quickly assess model performances, monitor system health, and make informed decisions about adjustments or deployments.



Start with Bosphorus AI AutoML

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed fermentum erat quis nisi bibendum, nec pulvinar purus scelerisque.

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Maecenas auctor, nisi id efficitur molestie, quam libero lobortis sapien, eu interdum purus elit eget nisi.

Integer malesuada magna ut ex fermentum, sed lobortis quam suscipit. Donec rutrum diam nec enim hendrerit, sed tempor arcu consectetur.

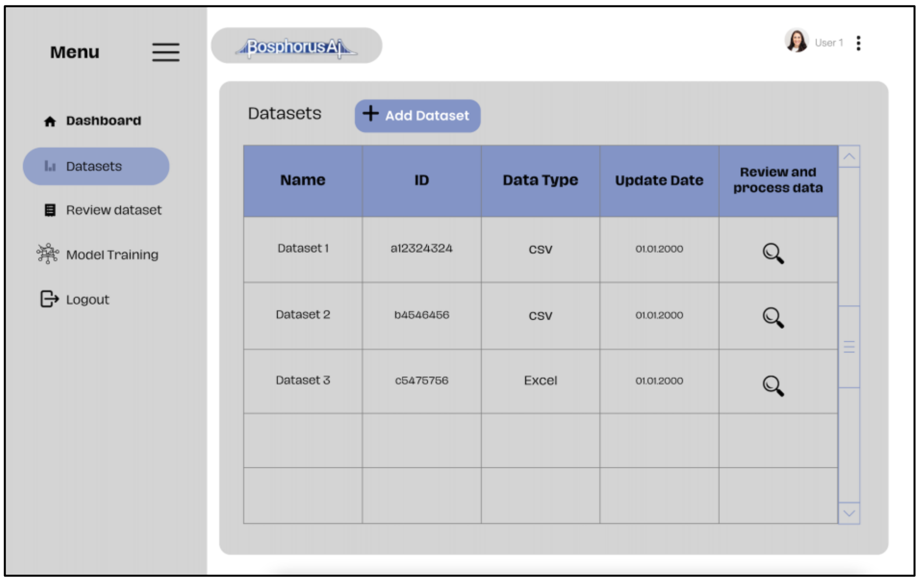
Phasellus consequat orci et justo sagittis, nec ultricies enim ultrices. Nulla facilisi. Cras dapibus velit sit amet nulla vulputate, eget gravida ex ultricies. Suspendisse potenti.

Once you've trained a model, you can use it to make predictions either online from an endpoint or through ensemble requests.

Figure 6. *Dashboard Interface*

3.6.2. Data Viewing and Addition Interface

The data viewing and addition interface (Figure 7) enables users to view existing datasets and add new ones. It explains how users interact with the data, the types of data supported, and how new data can be uploaded to the system. The interface facilitates easy management of data resources, including search, filter, and sort capabilities.

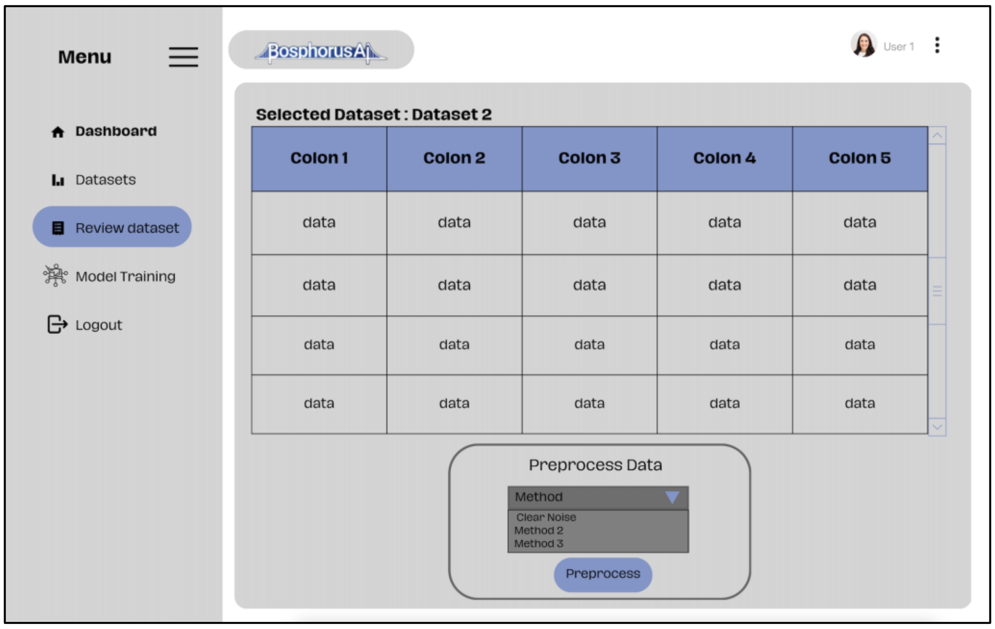


Name	ID	Data Type	Update Date	Review and process data
Dataset 1	a12324324	csv	01.01.2000	🔍
Dataset 2	b4546456	csv	01.01.2000	🔍
Dataset 3	c5475756	Excel	01.01.2000	🔍

Figure 7. Data Viewing and Addition Interface

3.6.3. Data Preprocessing Interface

The data preprocessing interface (Figure 8) enables users to preprocess data which includes options for cleaning data, handling missing values, normalizing data, and feature engineering. The user-friendly design allows non-experts to perform complex preprocessing tasks with minimal technical knowledge.



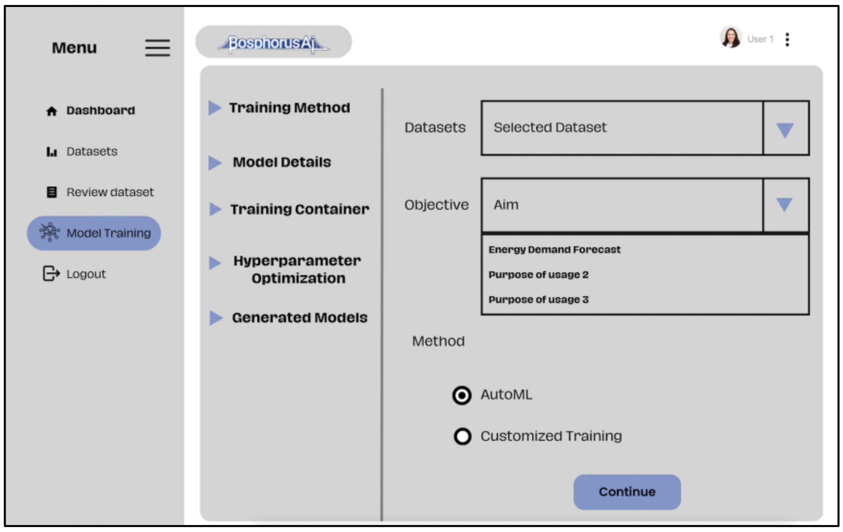
Selected Dataset : Dataset 2

Colon 1	Colon 2	Colon 3	Colon 4	Colon 5
data	data	data	data	data
data	data	data	data	data
data	data	data	data	data
data	data	data	data	data

Figure 8. Data Preprocessing Interface

3.6.4. Model Training Interface

The model training interface (Figure 9) enables users to select, configure, and train ML models. It includes tools for setting parameters, choosing algorithms, and initiating training sessions. This screen also provides feedback on model performance and diagnostics to help users optimize their models.



Datasets: Selected Dataset
Objective: Aim
Method: [✓ AutoML] Customized Training

Figure 9. *Model Training Interface*

3.7. Usability and Preliminary Results

While the proposed No-Code Auto-ML platform for the energy sector presents significant potential, it is crucial to consider and test specific challenges such as the black-box nature of Auto-ML systems and the need for high-quality datasets, providing valuable insights into how these issues may affect the platform's deployment and effectiveness.

3.7.1. Mitigating the Effect of Black-Box Nature of Auto-ML Systems

Trust is essential for the integration of AI-driven solutions into existing workflows, and the opaque nature of Auto-ML models can hinder this trust. To mitigate this issue, the platform incorporates visualization tools aimed at enhancing interpretability and transparency. Advanced visualization tools that graphically represent model behavior and decision pathways, making it easier for users to comprehend complex models.

3.7.2. Robust Data Preprocessing and Cleaning Modules

Energy companies collect data from diverse sources and systems, leading to data silos. Integrating these disparate datasets into a cohesive whole can be challenging and time-consuming. To address these challenges, the platform includes robust data preprocessing and cleaning modules. Tools for automated data cleaning that filter out noise, handle missing values, and standardize data formats to improve dataset quality. Features that facilitate the integration of data from multiple sources, including SQL and NoSQL databases, to create comprehensive and high-quality datasets for model training.

3.7.3. Preliminary Results

Empirical validation is crucial to demonstrate the practical effectiveness of the proposed No-Code Auto-ML platform in real-world energy industry settings. We did a pilot project in collaboration with key stakeholders in the energy sector. The platform is applied to improve the accuracy of energy demand forecasts, allowing better matching of supply and demand dynamics. Accurate forecasting is critical for energy providers to ensure reliability and cost-effectiveness. Initial feedback from the pilot projects has been promising. Users have reported significant improvements in data processing efficiency and predictive accuracy. Key observations include:

- **Data Processing Efficiency**: The platform's automated data preprocessing and feature engineering capabilities have streamlined workflows, significantly reducing the time required for data preparation.

- **Predictive Accuracy**: Enhanced model accuracy has been observed in applications involving high variability in energy demands, leading to more reliable and actionable insights.

- **User Experience**: Non-expert users have found the platform's no-code interface intuitive and easy to use, facilitating broader adoption of machine learning tools within their organizations.

These preliminary results provide a strong foundation for further validation and suggest that the platform can deliver substantial benefits to energy sector professionals.

4. CONCLUSION

In this paper, we presented the design and implementation of a no-code Auto-ML platform tailored specifically for the energy industry, aiming to streamline the development and deployment of ML models. The proposed Auto-ML platform addresses the unique challenges faced by the energy sector, such as high variability in data sources and strict regulatory requirements, by incorporating domain-specific knowledge into every phase of the ML pipeline. The modular architecture of the Auto-ML platform allows for extensive customization and scalability, ensuring that it can be adapted to meet the diverse needs of various energy sub-sectors. This adaptability fosters a wider adoption of advanced analytics across the industry, from large-scale utility companies to renewable energy startups. By reducing the barrier to entry for sophisticated data analysis, our platform empowers industry stakeholders to leverage predictive analytics and ML without the need for deep technical expertise in these areas. In conclusion, our work contributes a significant tool to the energy industry’s arsenal, facilitating a shift towards more adaptive, efficient, and data-informed operations. The ongoing development and refinement of such technologies will be crucial in meeting the evolving demands of global energy markets and in tackling the pressing challenges of energy sustainability and management.

While the development of the No-Code Auto-ML platform for the energy sector has shown significant promise, several limitations and challenges were encountered throughout the process. Acknowledging these issues is crucial for understanding the platform's current capabilities and identifying areas for future improvement. One of the primary challenges faced during the development of the Auto-ML platform was ensuring the quality and availability of data. The energy sector generates vast amounts of data, but this data is often noisy, incomplete, or stored in disparate systems. Developing a high-quality dataset that accurately reflects real-world conditions is essential for training effective machine learning models. This challenge was partially mitigated by integrating robust data cleaning and preprocessing modules, but inconsistencies in data sources still posed significant hurdles. Incorporating domain-specific knowledge into the Auto-ML platform to handle unique industry challenges such as fluctuating energy demand and regulatory compliance proved complex. While the platform's modular architecture allows for customization, translating expert knowledge into automated processes and algorithms required extensive collaboration with domain experts. This process was time-consuming and highlighted the need for ongoing refinement and expert input to ensure the platform remains relevant and effective.

The potential for future research and practical applications of this platform has several promising directions to further enhance its impact and utility. Collaborative and federated learning approaches can be explored to enable the platform to learn from decentralized data sources without compromising data privacy. This involves developing algorithms that allow multiple stakeholders to collaboratively train models while keeping their data local. Such approaches can enhance the platform's ability to generate robust models using diverse datasets from different organizations. While the platform is designed for the energy sector, its modular architecture allows for adaptation to other industries. Future research can explore the customization of the Auto-ML platform for different domains such as healthcare, finance, and manufacturing. This includes developing domain-specific modules and algorithms that address the unique challenges and requirements of these sectors, broadening the platform's applicability and impact.

ACKNOWLEDGEMENT

This study was supported with the project number 3220630 under the program of "TÜBİTAK TEYDEB 1501 - Industry R&D Projects".

CONFLICT OF INTEREST

The author declares no conflict of interest.

REFERENCES

Banzhaf, W. (2006). Introduction. Genetic Programming and Evolvable Machines, 7(1), 5–6. https://doi.org/10.1007/s10710-006-7015-0

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., & Colton, S. (2012). A Survey of Monte Carlo Tree Search Methods. IEEE Transactions on Computational Intelligence and AI in Games, 4(1), 1–43. https://doi.org/10.1109/tciaig.2012.2186810

Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data Cleaning. Proceedings of the 2016 International Conference on Management of Data. https://doi.org/10.1145/2882903.2912574

Chu, X., Morcos, J., Ilyas, I. F., Ouzzani, M., Papotti, P., Tang, N., & Ye, Y. (2015). KATARA. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. https://doi.org/10.1145/2723372.2749431

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). AutoAugment: Learning Augmentation Strategies From Data. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2019.00020

Darren, C. (2016). Practical ML with H2O: powerful, scalable techniques for deep learning and AI. O'Reilly Media, Inc.

Drori, I., Krishnamurthy, Y., Rampin, R., Lourenco, R. d. P., Ono, J. P., Cho, K., Silva, C., & Freire, J. (2018). AlphaD3M: Machine Learning Pipeline Synthesis. In International Conference on Machine Learning AutoML Workshop.

Drori, I., Krishnamurthy, Y., de Paula Lourenco, R., Rampin, R., Kyunghyun, C., Silva, C., & Freire, J. (2019). Automatic Machine Learning by Pipeline Synthesis using Model-Based Reinforcement Learning and a Grammar. In International Conference on Machine Learning AutoML Workshop.

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. (2020). AutoGluon-Tabular: Robust and Accurate Auto-ML for Structured Data. arXiv preprint arXiv:2003.06505.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J. T., Blum, M., & Hutter, F. (2019). Auto-sklearn: Efficient and Robust Automated Machine Learning. The Springer Series on Challenges in Machine Learning, 113-134. https://doi.org/10.1007/978-3-030-05318-5_6

Gama, J. (2004). Functional Trees. Machine Learning, 55(3), 219–250. https://doi.org/10.1023/b:mach.0000027782.67192.13

Ge, P. (2020). Analysis on Approaches and Structures of Automated Machine Learning Frameworks. 2020 International Conference on Communications, Information System and Computer Engineering (CISCE). https://doi.org/10.1109/cisce50729.2020.00106

Iwendi, C., Huescas, C. G. Y., Chakraborty, C., & Mohan, S. (2022). COVID-19 health analysis and prediction using machine learning algorithms for Mexico and Brazil patients. Journal of Experimental &amp; Theoretical Artificial Intelligence, 1–21. https://doi.org/10.1080/0952813x.2022.2058097

Z. H, J. M., Hossen, J., Sayeed, S., Ho, C., K, T., Rahman, A., & Arif, E. M. H. (2018). A Survey on Cleaning Dirty Data Using Machine Learning Paradigm for Big Data Analytics. Indonesian Journal of Electrical Engineering and Computer Science, 10(3), 1234. https://doi.org/10.11591/ijeecs.v10.i3.pp1234-1243

Ji, Z., He, Z., Gui, Y., Li, J., Tan, Y., Wu, B., Xu, R., & Wang, J. (2022). Research and Application Validation of a Feature Wavelength Selection Method Based on Acousto-Optic Tunable Filter (AOTF) and Automatic Machine Learning (AutoML). Materials, 15(8), 2826. https://doi.org/10.3390/ma15082826

Jin, H., Song, Q., & Hu, X. (2019). Auto-Keras: An Efficient Neural Architecture Search System. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining. https://doi.org/10.1145/3292500.3330648

Jin, H., Chollet, F., Song, Q., & Hu, X. (2023). Autokeras: an Auto-ML library for deep learning. Journal of Machine Learning Research, 24(6), 1-6. https://www.jmlr.org/papers/volume24/20-1355/20-1355.pdf

Kocsis, L., & Szepesvári, C. (2006). Bandit Based Monte-Carlo Planning. Machine Learning: ECML 2006, 282–293. https://doi.org/10.1007/11871842_29

Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2019). Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA. The Springer Series on Challenges in Machine Learning, 81–95. https://doi.org/10.1007/978-3-030-05318-5_4

Kotthoff, L., Thornton, C., & Hutter, F. (2017). User guide for auto-WEKA version 2.6. Department of Computer Science, University of British Columbia, BETA Lab, Tech Report 2, 1-15. Vancouver, BC, Canada.

Koza, JohnR. (1994). Genetic programming as a means for programming computers by natural selection. Statistics and Computing, 4(2). https://doi.org/10.1007/bf00175355

Krishnan, S., & Wu, E. (2019). AlphaClean: Automatic generation of data cleaning pipelines. https://doi.org/10.48550/arXiv.1904.11827

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2016). Building machines that learn and think like people. Behavioral and Brain Sciences, 40. https://doi.org/10.1017/s0140525x16001837

LeDell, E., & Poirier, S. (2020). H2o Auto-ML: scalable automatic machine learning. In 7th ICML Workshop on Automated Machine Learning. https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf

LingChen, T. C., Khonsari, A., Lashkari, A., Nazari, M. R., & Sambee, J. S. (2020). UniformAugment: A search-free probabilistic data augmentation approach. arXiv preprint arXiv:2003.14348. https://doi.org/10.48550/arXiv.2003.14348

McGushion, H. (2019). HyperparameterHunter. Available at https://github.com/HunterMcGushion/hyperparameter_hunter.

Mahdavi, M., Neutatz, F., Visengeriyeva, L., & Abedjan, Z. (2019). Towards automated data cleaning workflows. Machine Learning, 15, 16.

Mohr, F., Wever, M., & Hüllermeier, E. (2018). ML-Plan: Automated machine learning via hierarchical planning. Machine Learning, 107(8–10), 1495–1515. https://doi.org/10.1007/s10994-018-5735-z

Olson, R. S., Bartley, N., Urbanowicz, R. J., & Moore, J. H. (2016). Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. Proceedings of the Genetic and Evolutionary Computation Conference 2016. https://doi.org/10.1145/2908812.2908918

Park, J. B., Lee, K. H., Kwak, J. Y., & Cho, C. S. (2022). Deployment Framework Design Techniques for Optimized Neural Network Applications. 2022 13th International Conference on Information and Communication Technology Convergence (ICTC). https://doi.org/10.1109/ictc55196.2022.9952771

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2011). Scikit-learn: ML in python. Journal of Machine Learning Research, 12, 2825-2830.

Rakotoarison, H., Schoenauer, M., & Sebag, M. (2019). Automated Machine Learning with Monte-Carlo Tree Search. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. https://doi.org/10.24963/ijcai.2019/457

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science, 362(6419), 1140–1144. https://doi.org/10.1126/science.aar6404

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2487575.2487629

UC Irvine ML Repository. (2023). Epileptic Seizures Dataset. https://www.kaggle.com/datasets/chaditya95/epileptic-seizures-dataset.

Vafaie, H., & Jong, K. (1998). Evolutionary Feature Space Transformation. Feature Extraction, Construction and Selection, 307–323. https://doi.org/10.1007/978-1-4615-5725-8_19

Zheng, Z. (1998). A Comparison of Constructing Different Types of New Feature For Decision Tree Learning. Feature Extraction, Construction and Selection, 239□255. https://doi.org/10.1007/978-1-4615-5725-8_15