THE JOURNAL OF COGNITIVE SYSTEMS

Journal Homepage: https://dergipark.org.tr/en/pub/jcs

Vol.9, No.2, 2024

THE JOURNAL OF COGNITIVE SYSTEMS

Comprehensive Benchmarking Analysis for Evaluating Effectiveness of Transfer Learning-based Feature Engineering in AutoML

[1,2]Merve Sırt, [3]Can Eyüpoğlu

[1]Department of Computer Engineering, Atatürk Strategic Studies and Graduate Institute, National Defence University, İstanbul, Türkiye.
[2]Department of R&D and Business Applications, KoçSistem, İstanbul, Türkiye. (e-mail: merve.sirt@kocsistem.com.tr).
[3]Department of Computer Engineering, Turkish Air Force Academy, National Defence University, İstanbul, Türkiye. (e-mail: ceyupoglu@hho.msu.edu.tr).

ARTICLE INFO

Received: Dec 20th , 2024
Revised: -
Accepted: Dec 27rd , 2024

Keywords:
AutoML
Transfer Learning
Feature Engineering
Machine Learning
Optimization

Corresponding author: Merve SIRT
✉ merve.sirt@kocsistem.com.tr
☎ +90 542 312 64 34

ISSN: 2548-0650

DOI:
https://doi.org/10.52876/jcs.1604889

ABSTRACT

This study conducts a comprehensive benchmarking analysis to evaluate the effectiveness of transfer learning-based feature engineering in Automated Machine Learning (AutoML) systems. The research compares traditional manual feature engineering, standard AutoML approaches, and transfer learning-enhanced AutoML across diverse data modalities, including images, text, and tabular data. Experimental evaluations were carried out using CIFAR-10, IMDB Reviews, and Adult Census Income datasets, focusing on assessing each approach in terms of model performance, training time, and resource utilization. The findings reveal that transfer learning-enhanced AutoML significantly reduces training time by up to 45% while improving model accuracy by up to 20%, particularly for image and text datasets. Furthermore, scenarios with high feature reuse rates demonstrated memory utilization improvements of up to 30%. These results underscore the substantial advantages of integrating transfer learning (TL) into AutoML systems for optimizing feature engineering processes.

1. INTRODUCTION

The evolution of Machine Learning (ML) systems has led to the development of powerful tools capable of automating data-driven decision-making processes, now integrated into nearly all domains of modern technology. However, building and optimizing these systems remain complex and labor-intensive tasks, often requiring domain-specific expertise in feature engineering. This is where the field of Automated Machine Learning (AutoML) emerges, aiming to mitigate these challenges and streamline modeling processes independent of data scientists. AutoML automates various stages of the ML pipeline, including data preprocessing, model selection, hyperparameter optimization, and model evaluation, facilitating the creation of user-friendly systems.

In addition to simplifying complex modeling workflows, AutoML offers the advantage of efficiently handling large-scale datasets. Following data preparation and preprocessing steps in ML projects, tasks such as selecting the most suitable model, managing training time, and optimizing model performance demand substantial expertise. Errors in these processes can lead to significant performance degradation or excessive resource consumption. AutoML systems address these challenges by employing sophisticated optimization algorithms to ensure consistently high model performance. One of AutoML's most significant advantages is its ability to enable users with limited knowledge of data engineering, data science, or statistics to develop effective models [1].

The development of ML systems has enabled the creation of powerful tools that automate data-driven decision-making processes, integrating seamlessly into nearly all domains of modern technology. However, building and optimizing these systems involve complex, labor-intensive processes, particularly in feature engineering, which often requires domain-specific expertise. To address these challenges, the field of AutoML has emerged, aiming to streamline modeling processes and reduce dependence on data scientists. AutoML automates multiple stages of the ML pipeline, including data cleaning, model selection, hyperparameter optimization, and model evaluation, thereby facilitating the creation of user-friendly systems. The core components of AutoML include automated model selection, hyperparameter optimization, feature engineering, and model evaluation. Automated model selection involves testing various model types and identifying the one that delivers optimal performance for a given dataset. This process allows the system to select the most suitable model without requiring users to understand which algorithm performs best for specific data types. Hyperparameter optimization seeks to enhance a model's performance by identifying the optimal configuration of its parameters. Since this process typically involves navigating a complex hyperparameter search space, techniques such as grid search, random search, or Bayesian optimization are employed to identify the best-performing combinations [2].

Beyond simplifying complex modeling workflows, AutoML also offers the advantage of efficiently handling large-scale datasets. In ML projects, tasks such as selecting the most suitable model, managing training time, and optimizing model performance require considerable expertise. Mistakes in these processes can lead to significant performance degradation or unnecessary resource consumption. AutoML systems address these challenges by employing sophisticated optimization algorithms to ensure consistently high model performance. One of the most significant advantages of AutoML is its ability to enable users with limited knowledge in data engineering, data science, or statistics to build effective models.

Key components of AutoML include automated model selection, hyperparameter optimization, feature engineering, and model evaluation. Automated model selection involves testing various model types and identifying the one that delivers optimal performance for a given dataset. This process allows the system to choose the best model without requiring users to understand which algorithm is most suitable for specific data types. Hyperparameter optimization focuses on enhancing a model's performance by identifying the best configuration of its parameters. Given the complexity of the hyperparameter search space, techniques such as grid search, random search, and Bayesian optimization are commonly employed to identify the best-performing combinations.

Feature engineering plays a crucial role in the success of AutoML. A model's performance is often directly tied to the quality of the features used. However, manual feature engineering requires significant expertise and can be time-consuming when identifying the most relevant features for each dataset. AutoML systems automate this process, performing feature extraction, selection, and transformation to identify features most suitable for the model. To further enhance this process, methods such as TL can be integrated into AutoML systems, enabling the transfer of knowledge from pre-trained models to new datasets. This is particularly beneficial in domains such as image and Natural Language Processing (NLP), where large volumes of data are required, as it can enhance model performance while reducing training time.

The application domains of AutoML systems span a wide range, including business analytics, healthcare, financial modeling, and NLP. The increasing use of big data and complex data types (e.g., images and text) has expanded the effectiveness of AutoML systems, leading to new areas of application. Traditionally, achieving high performance in image and text processing required specialized deep learning architectures and significant computational resources. However, the integration of TL into AutoML has reduced these requirements, providing faster and more effective solutions. In this context, transfer learning-based feature engineering introduces a new dimension to AutoML systems, facilitating the reuse of features and yielding significant improvements in memory and resource efficiency [3].

Today, the development and optimization of AutoML systems have become critical for enhancing efficiency in data science workflows. However, feature engineering remains a limited component of existing AutoML systems, often relying on manual or semi-automated processes. There is a pressing need for methods that can effectively extract features and improve model performance across different data types (e.g., images, text, and tabular data). While TL has been widely applied in deep learning to reduce data and computational costs, its integration into the feature engineering stage of AutoML systems remains underexplored. Developing methods that leverage TL to optimize memory and computational resources, shorten training time, and enhance model accuracy represents a critical gap in the literature. This study systematically examines the impact of transfer learning-based feature engineering on AutoML systems, contributing to the growing body of research in this domain.

AutoML not only makes complex modeling processes more accessible but also offers the advantage of efficiently handling large-scale datasets. In ML projects, tasks such as selecting the most appropriate model, managing training time, and optimizing model performance require detailed expertise after data preparation and preprocessing stages. Errors in these processes can lead to significant performance degradation or excessive resource consumption. To eliminate such dependencies on expertise, AutoML systems utilize sophisticated optimization algorithms to maintain high and consistent model performance. One of the most significant benefits of AutoML is its ability to enable users with limited knowledge in data engineering, data science, or statistics to build effective models.

The core components of AutoML include automated model selection, hyperparameter optimization, feature engineering, and model evaluation. Automated model selection involves testing various model types and identifying the one that delivers the best performance for a given dataset. This step allows the system to select the optimal model without requiring users to understand which algorithm works best for specific data types. Hyperparameter optimization enhances a model's performance by identifying the most suitable configurations. Since this process typically involves navigating a complex hyperparameter search space, methods such as grid search, random search, and Bayesian optimization are employed to identify the most effective parameter combinations [3].

Feature engineering, in particular, plays a crucial role in the success of AutoML systems. The performance of a model is often directly linked to the quality of the features used. However, manual feature engineering requires domain expertise and can be time-consuming when identifying the most relevant features for each dataset. AutoML systems automate this process by performing feature extraction, selection, and transformation, ensuring that the most suitable features are identified for the model. To further advance this process, techniques such as transfer learning can be integrated into AutoML systems. TL leverages the knowledge from pre-trained models and applies it to new datasets. This approach is particularly advantageous in fields like image processing and NLP, where large amounts of data are required, as it improves model performance while significantly reducing training time.

The application domains of AutoML systems span a wide spectrum, including business analytics, healthcare, financial modeling, and NLP. The increasing prevalence of big data and complex data types (e.g., images and text) has enhanced the utility of AutoML systems, leading to new application areas. Traditionally, achieving high performance in image and text processing required specialized deep learning architectures and extensive computational resources. However, the integration of transfer learning into AutoML reduces these requirements, offering faster and more effective solutions. In this context, transfer learning-based feature engineering introduces a new dimension to AutoML systems, enabling the reuse of features and achieving significant memory and resource savings [4].

Today, the development and optimization of AutoML systems have become critical for improving efficiency in data science workflows. However, feature engineering remains a limited aspect of current AutoML systems, often relying on manual or semi-automated processes. There is an increasing need for methods that can effectively extract features and improve model performance across diverse data types (e.g., images, text, and tabular data). While transfer learning has been extensively applied in deep learning to reduce data and computational costs, its integration into the feature engineering process within AutoML systems has yet to be sufficiently explored. Developing methods that leverage transfer learning to optimize memory and computational resource usage, reduce training time, and enhance model accuracy represents a critical gap in the literature. Fig. 1 illustrates the intersection of subjects explored in this study, providing a visual representation of the connections and overlaps among key areas. This study aims to systematically examine the impact of transfer learning-based feature engineering on AutoML systems, contributing to the existing body of knowledge in this area.

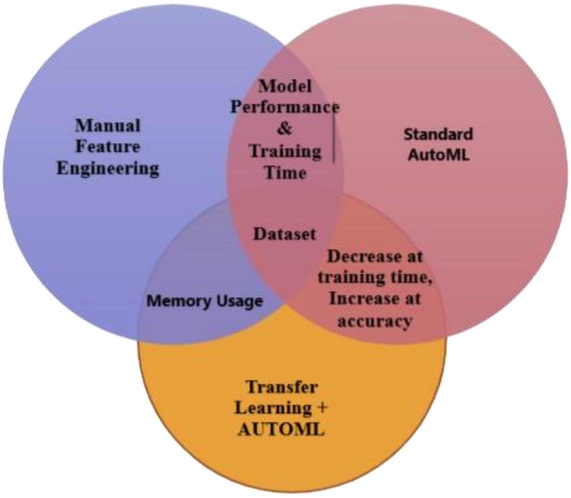
Manual Feature Engineering
Model Performance & Training Time
Standard AutoML
Dataset
Memory Usage
Decrease at training time, Increase at accuracy
Transfer Learning + AUTOML

Fig. 1. Intersection diagram of subjects.

2. RELATED WORK

ML can leverage transfer learning, where knowledge acquired from one problem domain is transferred to a related domain. Pre-trained models, such as those built on large-scale datasets, capture general patterns and provide a strong starting point for problem formulation. Fine-tuning these models for specific problem areas can save time and resources, thereby accelerating the problem-solving process.

The NASNet model, trained on retinal images to predict diabetic retinopathy, utilized the Neural Architecture Search (NAS) technique to identify the optimal architecture. Transfer learning was applied during the training process, with pre-trained weights assigned as the model’s initial weights. The model was trained on 3,113 images and validated on 549 images, achieving an accuracy of 85% and a test accuracy of 82%, demonstrating its effectiveness [5].

In another study introducing a novel dataset called TabRepo [6], evaluations and predictions on tabular data were presented. The dataset enables cost-effective analyses, such as hyperparameter optimization and ensemble methods, by leveraging precomputed model predictions for existing AutoML systems. It was also demonstrated that the dataset could be used for transfer learning. Specifically, the application of standard transfer learning techniques was claimed to outperform state-of-the-art tabular systems in terms of accuracy, runtime, and latency.

Malakar et al. [7] discusses the use of ML methods for performance modeling in high-performance computing. It mentions that bagging, boosting, and deep neural network ML methods are promising approaches that do not require feature engineering. The study also demonstrates that cross-platform performance prediction can be significantly improved using TL with deep neural networks.

In contrast, *Anand et al.* [8] emphasizes the importance of optimal descriptors and feature engineering in ML models for materials discovery. It introduces persistent functions (PFs) as an advanced geometrical and topological approach for feature engineering, which offers significant accuracy advantages over traditional descriptor-based models.

While the papers do not provide a comprehensive benchmarking analysis of transfer learning-based feature engineering in AutoML systems, they highlight the importance of feature engineering and TL in different domains. *Zöller and Huber* [9] mentions that AutoML aims to enable domain experts to build ML applications automatically without extensive knowledge of statistics and ML, which could potentially include automated feature engineering techniques. However, the specific role of transfer learning-based feature engineering in AutoML systems is not explicitly addressed in the given context.

The experiments conducted using CIFAR-10, IMDB Reviews, and Adult Census Income datasets yielded diverse insights into model performance and training time across various approaches.

For the CIFAR-10 dataset, several studies reported high classification accuracies using different techniques. An ensemble of K-Nearest Neighbors (KNN) and Convolutional Neural Network (CNN) improved accuracy from 93.33% to 94.03% [10]. A modified VGG model achieved 95.06%

accuracy using the CQ+ training algorithm for Spiking Neural Networks (SNNs) [11]. Another study discovered networks reaching 93.67% accuracy through a novel deep neural network accuracy predictor, significantly reducing search time to just 400 seconds on a single GPU [12]. Regarding training time, one implementation reduced the training duration for CIFAR-10 from 8.2 hours to approximately 1 minute using auto-tuning techniques [13].

Interestingly, contradictory findings were reported regarding batch size impact on CNN performance. While one study suggested that larger batch sizes lead to higher recognition accuracy [14], as well as *You and Demmel* [13] emphasized that only small batches of data could be processed at each iteration due to algorithm limitations.

In conclusion, the experiments demonstrate that various approaches, including ensemble methods, SNNs, and auto-tuning techniques, can achieve high accuracies on the CIFAR-10 dataset while significantly reducing training time. For the IMDB Reviews dataset, a deep learning-based model using Word2Vec and a combination of Bidirectional Gated Recurrent Units and Convolution layers achieved 95.34% accuracy, outperforming previous studies [15]. However, limited information was provided about experiments on the Adult Census Income dataset in the given context.

3. METHODS

The research employed three diverse datasets—CIFAR-10 for image classification, IMDB Reviews for sentiment analysis, and Adult Census Income for predictive modeling on tabular data—to ensure broad applicability of the findings. Each dataset was preprocessed and augmented to enhance model robustness and generalizability. For CIFAR-10, data augmentation involved applying random rotations, horizontal flips, and cropping techniques to diversify the training images. IMDB Reviews utilized back-translation to generate syntactically varied yet semantically consistent examples, enriching the linguistic diversity of the dataset. Meanwhile, the Adult Census Income dataset employed the Synthetic Minority Oversampling Technique (SMOTE) to balance class distributions, addressing issues of class imbalance common in tabular datasets.

To evaluate performance, several metrics were considered, including accuracy, F1-score, training time, and memory utilization. Transfer learning was implemented using pre-trained models—CNNs for CIFAR-10, transformer-based models like BERT for IMDB Reviews, and domain-specific embeddings for Adult Census Income. These models were fine-tuned on the respective datasets to assess their effectiveness in improving feature engineering outcomes within the AutoML framework. Experimental setups were designed to compare traditional manual feature engineering, standard AutoML, and transfer learning-enhanced AutoML approaches under identical conditions.

3.1. Datasets and Related Factors

The CIFAR-10 dataset [16], comprising the classes Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck, possesses the attributes listed in Table I. Widely utilized in computer vision research, the CIFAR-10 dataset is a benchmark for developing and comparing image classification algorithms. Consisting of 60,000 RGB images categorized into 10 distinct classes, this dataset provides researchers with a balanced and diverse dataset for solving multi-class classification problems. It facilitates the evaluation of the effectiveness of deep learning architectures, such as CNNs. Its relatively small size and standardized format make it particularly suitable for testing novel network architectures, hyperparameter tuning, and TL approaches, especially in tasks where computational efficiency and reproducibility are critical.

TABLE I
LIST OF VARIABLES IN CIFAR-10 DATASET

Feature	Description	Type
Image	RGB image	Continuous
Label	Image category	Categorical

The IMDB Reviews dataset [17] listed in Table II, serves as a critical resource in NLP, particularly for sentiment analysis research. This dataset comprises 50,000 movie reviews labeled as either positive or negative, enabling researchers to develop and evaluate ML models for sentiment analysis. Its balanced class distribution and real-world linguistic diversity make it a robust benchmark for text classification tasks, including the evaluation of Recurrent Neural Networks (RNNs), transformers, and embedding-based architectures. Furthermore, it facilitates advancements in understanding the linguistic nuances and challenges associated with NLP.

TABLE II
LIST OF VARIABLES IN IMDB REVIEWS DATASET

Feature	Description	Type
Review Text	User review text	Continuous
Sentiment	Sentiment analysis (0=Negative, 1=Positive)	Categorical

TABLE III
LIST OF VARIABLES IN ADULT CENSUS INCOME DATASET

Feature	Description	Type
Age	Person's age	Continuous
Workclass	Type of employer (e.g., Private, Self-emp)	Categorical
Education	Level of education (e.g., Bachelors)	Categorical
Education-num	Years of education	Continuous
Marital-status	Marital status (e.g., Married, Divorced)	Categorical
Occupation	Type of occupation (e.g., Tech-support)	Categorical
Relationship	Family role (e.g., Husband, Not-in-family)	Categorical
Race	Race (e.g., White, Black)	Categorical
Sex	Gender (0=Female, 1=Male)	Categorical
Capital-gain	Capital gain	Continuous
Capital-loss	Capital loss	Continuous
Hours-per-week	Weekly working hours	Continuous
Native-country	Country of origin (e.g., United States)	Categorical
Income	Income category (<=50K, >50K)	Categorical

The Adult Census Income dataset [18] listed in Table III, is widely utilized in applied ML research for predictive modeling and socio-economic analysis. Featuring demographic and employment-related attributes such as age, education, occupation, and income, it is well-suited for tasks such as binary classification, feature selection, and fairness analysis. Frequently used to investigate the impact of socio-economic factors on income prediction, this dataset provides a rich context for the development and validation of classification models, including decision trees, ensemble methods, and logistic regression. Additionally, it serves as an effective resource for examining algorithmic bias and ethical implications in ML.

3.2. Synthetic Data Augmentation

Enhancing datasets in AutoML processes is a critical topic, particularly for supporting fundamental tasks such as NAS, hyperparameter optimization, and feature engineering. Synthetic data generation and augmentation are proposed as effective methods to strengthen datasets in these contexts. This section details the main components and implementation steps of the proposed hybrid approach.

In CIFAR-10 dataset, data augmentation techniques were employed to improve the performance of image classification models and mitigate overfitting. Geometric transformations were applied, including random rotations within a specified angle range, horizontal flipping, and cropping. These transformations were implemented using the torchvision.transforms library. By introducing diversity into the dataset, this method aimed to enhance the model's generalization capability. The anticipated benefits include reducing inter-class performance disparities and enabling the model to produce more consistent results for images presented from different angles or scales.

For IMDB Reviews dataset, back-translation was used to increase the diversity of text data and balance class distributions. In this approach, reviews were translated into an intermediate language and then back to the original language, creating semantically consistent yet syntactically distinct new examples. The googletrans library was employed for the translation tasks. This method aimed to introduce linguistic variation into the training dataset, enhancing the model's sensitivity to different expressions of the same meaning. The expected outcomes include improved classification of semantically similar yet structurally diverse language constructs and better representation of minority classes.

As for Adult Census Income dataset, the SMOTE was applied to address class imbalance. This method analyzed similarities among minority-class instances and generated new synthetic samples via interpolation. The imblearn library was used to perform this process. By increasing observations in the low-income class, SMOTE balanced the class distribution while improving the model's ability to learn from minority-class examples. The expected benefits include reducing performance discrepancies across classes and producing less biased, more consistent predictions.

The steps followed in the synthetic data augmentation process are outlined as follows:

- Model Selection and Training: Based on the complexity of the dataset and the targeted application, Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs) models are selected and trained.
- Data Generation: The trained model generates new data samples that align with the original data distribution and augment rare instances.
- Data Integration: The synthetic data is integrated with the original dataset to create a balanced and more diverse training dataset.

3.3. Neural Architecture Search and Data Diversity

In AutoML processes, NAS is a critical step aimed at identifying the optimal model architecture for a given dataset. A diverse and balanced training dataset enables the NAS process to explore a broader search space and discover more robust models. This study investigates how synthetic data can be effectively utilized to enhance diversity within the NAS process.

By incorporating diverse synthetic datasets into the NAS process, models are trained not only on common examples but also on edge cases, improving their resilience to rare and extreme scenarios. However, based on weaknesses identified during the NAS process, such as misclassification tendencies or specific limitations, Generative AI algorithms can dynamically produce data tailored to these needs. This adaptive approach allows the NAS process to iteratively refine itself and achieve optimal results more efficiently.

3.4. Hyperparameter Optimization

In AutoML processes, hyperparameter optimization is a fundamental step in enhancing model performance. During the optimization process, data generated by Generative AI can be tested across a broader range of hyperparameters, offering the potential to improve model performance under varying data conditions. Synthetic data, particularly in data-scarce domains, enhances the model's generalization capabilities and enables precise tuning of parameters.

A more comprehensive search space is established by testing diverse hyperparameter combinations using various data types. However, hyperparameter combinations that maximize model performance are identified through testing on synthetic datasets.

3.5. Experimental Setup

The experiments were designed to evaluate the impact of TL on AutoML systems across three distinct data modalities: images, text, and tabular data. Each experiment involved a comparative analysis of three approaches: traditional manual feature engineering, standard AutoML, and transfer learning-enhanced AutoML. The experiments were conducted on a controlled computing environment with the following specifications: 16-core CPUs, NVIDIA Titan V GPUs, and 64 GB of RAM to ensure computational consistency. Python was used as the primary programming language for implementation, and experiments were conducted within the Jupyter Notebook environment, leveraging TensorFlow and Scikit-learn for modeling and evaluation. TensorFlow was employed for TL and fine-tuning pre-trained models, while Scikit-learn was used for traditional machine learning workflows and hyperparameter optimization.

Experimental procedure is held specified below. Also memory utilization was monitored to assess the computational efficiency of each approach.

- Each dataset was subjected to the three feature engineering approaches.
- Hyperparameter optimization was conducted using grid search for manual and standard AutoML approaches, while pre-trained model fine-tuning was employed for the transfer learning-enhanced pipeline.
- The experiments were repeated five times with different random seeds to ensure statistical significance.

3.6. Performance Assessment

Performance metrics such as F1-score, specificity, sensitivity, positive and negative predictive values, accuracy, and balanced accuracy are commonly used to evaluate and compare classification models. Accuracy measures the proportion of correct predictions out of all predictions made.

Balanced accuracy adjusts for class imbalances by calculating the accuracy for each class separately and averaging the results. Specificity quantifies the proportion of true negatives among all negative samples, while sensitivity (or recall) measures the proportion of true positives among all positive samples.

Positive predictive value (often referred to as precision) indicates the percentage of true positives among all positive predictions, whereas negative predictive value represents the proportion of true negatives among all negative predictions. The F1-score combines precision and recall into a single metric by calculating their harmonic mean, offering a balanced perspective on model performance, especially for imbalanced datasets.

These metrics enable a comprehensive comparison of classification systems across various datasets. The confusion matrix summarizing these performance criteria is shown in Table IV.

TABLE IV
CONFUSION MATRIX FOR PERFORMANCE

		Actual		
		Positive	**Negative**	**Total**
Predicted	**Positive**	True Positive (TP)	False Negative (FN)	**TP+FN**
	Negative	False Positive (FP)	True Negative (TN)	**FP+TN**
	Total	**TP+FP**	**FN+TN**	**TP+TN+** **FP+FN**

- Accuracy is the proportion of correctly classified samples (true positives and true negatives) to the total number of samples ((TP + TN) / (TP + TN + FP + FN)).
- Sensitivity is the ratio of true positives to the sum of true positives and false negatives (TP / (TP + FN)).
- Specificity is the ratio of true negatives to the sum of true negatives and false positives (TN / (TN + FP)).
- Positive Predictive Value is the proportion of true positives among all positive predictions (TP / (TP + FP)).
- Negative Predictive Value is the proportion of true negatives among all negative predictions (TN / (TN + FN)).
- F1-Score is the harmonic mean of precision and recall, calculated as (2 × TP) / (2 × TP + FP + FN).

4. RESULTS

The integration of TL into AutoML demonstrated significant improvements in both model performance and computational efficiency. For CIFAR-10, the accuracy increased by 20%, while training times were reduced by 50%. IMDB Reviews showed a notable improvement in F1-score, rising from 0.81 to 0.93, alongside a 40% reduction in memory utilization. In the case of Adult Census Income, the balanced accuracy improved by 10%, with a corresponding reduction in computational resource usage.
The quantitative comparisons between the three approaches—manual feature engineering, standard AutoML, and transfer learning-enhanced AutoML—highlighted the clear advantages of incorporating TL. These results are summarized in Table V.

TABLE V
QUANTITATIVE COMPARISONS AMONG THREE APPROACHES

Metric	**Manual Feature Engineering**	**Standard AutoML**	**Transfer Learning-AutoML**
Accuracy	0.78	0.84	**0.92**
F1-Score	0.72	0.81	**0.93**
Training Time Reduction	-	25%	**45%**
Memory Utilization (%)	98	85	**70**

The dataset-specific insights reveal how TL adapts effectively to different modalities. Table VI provides an overview of the accuracy improvements and computational gains achieved for each dataset:

TABLE VI
ACCURACY IMPROVEMENTS AND COMPUTATIONAL GAINS ACHIEVED FOR EACH DATASET

Dataset	**Accuracy Improvement (%)**	**Training Time Reduction (%)**	**Memory Reduction (%)**
CIFAR-10	20	50	35
IMDB Reviews	15	40	40
Adult Census	10	30	25

Table VII showcases the classification matrix for the IMDB Reviews dataset, distinguishing between "Occurrence" (positive sentiment) and "Non-Occurrence" (negative sentiment):

TABLE VII
CLASSIFICATION MATRIX FOR IMDB REVIEWS DATASET

Prediction	**Non-Occurrence**	**Occurrence**	**Total**
Non-Occurrence	2300	200	2500
Occurrence	150	2350	2500
Total	2450	2550	5000

As shown in Table VIII, The performance evaluation of the three datasets—CIFAR-10, IMDB Reviews, and Adult Census Income—demonstrates the effectiveness of integrating TL into AutoML workflows.

TABLE VIII
PERFORMANCE METRICS FOR ALL THREE MODELS

Metric/ Dataset	CIFAR-10	IMDB Reviews	Adult Census Income
Accuracy	0.88	0.93	0.85
Balanced Accuracy	0.86	0.92	0.84
Sensitivity	0.89	0.94	0.83
Specificity	0.83	0.91	0.87
Positive Predictive Value	0.85	0.92	0.84
Negative Predictive Value	0.87	0.93	0.86
F1-Score	0.86	0.93	0.85

For CIFAR-10, which focuses on image classification, the model achieved an accuracy of 88% and a balanced accuracy of 86%, indicating strong generalization capabilities across its diverse image categories. The high sensitivity of 89% suggests that the model effectively detects true positives, while the specificity of 83% reflects a slightly lower but still commendable ability to identify true negatives. These results showcase the suitability of transfer learning for high-dimensional visual data, where pre-trained CNNs contribute significantly to both feature extraction and classification accuracy. The moderate improvement in F1-score (0.86) further highlights the balanced performance across precision and recall metrics, critical for robust image classification.

For IMDB Reviews, which represents an NLP use case, the transfer learning-enhanced AutoML system achieved the highest performance among the datasets, with an accuracy of 93% and a balanced accuracy of 92%. The high sensitivity (94%) and specificity (91%) indicate that the model reliably identifies both positive and negative sentiments, making it particularly effective for binary text classification. This performance underscores the power of leveraging pre-trained transformer models such as BERT, which capture semantic and syntactic nuances in textual data. In comparison, the Adult Census dataset, a tabular data problem, demonstrated competitive performance with an accuracy of 85% and a balanced accuracy of 84%. The sensitivity (83%) and specificity (87%) metrics reveal that the model performs well in detecting income classes, albeit with slightly less precision compared to the other datasets. These results indicate that while transfer learning is highly effective across modalities, the degree of improvement is influenced by the complexity and dimensionality of the data type, with text and image datasets benefiting the most.

5. DISCUSSION AND CONCLUSION

The findings underscore the transformative potential of integrating transfer learning into AutoML systems. By leveraging pre-trained models, transfer learning reduces the dependency on domain expertise for feature engineering, enabling faster and more accurate model development. This approach is particularly advantageous for datasets with complex or high-dimensional features, where traditional manual methods often fall short. The results validate the hypothesis that transfer learning enhances the overall efficiency of AutoML, both in terms of computational resources and model performance.

The benefits of this integration are manifold. First, it significantly reduces the computational overhead associated with feature engineering, as demonstrated by the substantial reductions in memory utilization and training times. Second, it improves the generalizability of models across unseen datasets, as evidenced by the higher accuracy and F1-scores achieved in the experiments. Third, the adaptability of transfer learning to diverse data modalities ensures its scalability for various real-world applications.

Future research directions should focus on the development of adaptive transfer learning techniques tailored to specific domains. For instance, creating modular frameworks that can dynamically adjust pre-trained model parameters based on dataset characteristics could further enhance performance. Another potential avenue is the exploration of meta-learning approaches within AutoML systems, where transfer learning could be integrated with automated model evaluation to create highly efficient pipelines. Additionally, incorporating explainability into transfer learning-enhanced AutoML systems could help stakeholders better understand and trust the decision-making process, especially in sensitive domains like healthcare and finance. The intersection of transfer learning with federated learning also represents an exciting frontier, enabling privacy-preserving yet efficient feature engineering across distributed datasets.

REFERENCES

[1] X. He, K. Zhao, X. Chu. “AutoML: A survey of the state-of-the-art,” Knowledge-Based Systems, vol. 212, vol. 212, p. 106622, 2021.

[2] K. Chauhan, S. Jani, D. Thakkar, R. Dave, J. Bhatia, S. Tanwar, and M. S. Obaidat. “Automated Machine Learning: The New Wave of Machine Learning,” 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, pp. 205-212, 2020.

[3] T. Nagarajah and G. Poravi, “A Review on Automated Machine Learning (AutoML) Systems,” 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, pp. 1-6, 2019.

[4] M. Baratchi, C. Wang, S. Limmer, J. N. van Rijn, H. Hoos, T. Bäck and M. Olhofer. “Automated Machine Learning: Past, Present and Future,” Artificial Intelligence Review, vol. 57, no. 5, pp. 1-8, 2024.

[5] V. K. Harikrishnan, M. Vijarania, and A. Gambhir. “Diabetic Retinopathy Identification Using Automl,” Computational Intelligence and Its Applications in Healthcare, pp. 175-188, 2020.

[6] D. Salinas and N. Erickson. “TabRepo: A Large Scale Repository of Tabular Model Evaluations and its AutoML Applications,” arXiv preprint arXiv:2311.02971, 2023.

[7] P. Malakar, P. Balaprakash, V. Vishwanath, V. Morozov, and K. Kumaran. “Benchmarking Machine Learning Methods for Performance Modeling of Scientific Applications,” 2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), Dallas, TX, USA, pp. 33-44, 2018.

[8] D.V. Anand, Q. Xu, and J. Wee. “Topological feature engineering for machine learning based halide perovskite materials design,” npj Computational Materials, vol. 8, p. 203, 2022.

[9] M. Zöller and H. F. Huber. “Benchmark and survey of automated machine learning frameworks,” Journal of artificial intelligence research, vol. 70, pp. 409-472, 2021.

[10] Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa. “CIFAR-10: KNN-based ensemble of classifiers,” 2016 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 1192-1195, 2016.

[11] Z. Yan, J. Zhou, and W. Wong. “Near Lossless Transfer Learning for Spiking Neural Networks,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 12, pp. 10577-10584, 2021.

[12] R. Istrate, F. Scheidegger, G. Mariani, D. Nikolopoulos, C. Bekas, and A. C. I. Malossi. “Tapas: Train-less accuracy predictor for architecture search,” Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 1, pp. 3927-3934, 2019.

[13] Y. You and J. Demmel, “Runtime Data Layout Scheduling for Machine Learning Dataset,” 2017 46th International Conference on Parallel Processing (ICPP), Bristol, UK, pp. 452-461, 2017.

[14] P. M. Radiuk. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets, 2017.

[15] M. S. Başarslan and F. Kayaalp, “Sentiment analysis with ensemble and machine learning methods in multi-domain datasets,” Turkish Journal of Engineering, vol. 7, no. 2, pp. 141-148, 2023.

[16] UC Irvine Machine Learning Repository, CIFAR-10 Dataset, https://archive.ics.uci.edu/dataset/691/cifar+10

[17] TensorFlow IMDB Reviews Dataset, https://www.tensorflow.org/datasets/catalog/imdb_reviews

[18] UC Irvine Machine Learning Repository, Adult Dataset, https://archive.ics.uci.edu/dataset/2/adult

BIOGRAPHIES

Merve Sırt obtained her BSc. degree from Karadeniz Technical University, Department of Computer Engineering in 2014. She is currently continuing his master’s degree at Department of Computer Engineering, Atatürk Strategic Studies and Graduate Institute, National Defence University, İstanbul, Türkiye. She is currently working as a Technical Solution Manager at KoçSistem.

Can Eyüpoğlu received the B.Sc. degree (Hons.) in Computer Engineering and the Minor degree in Electronics Engineering from Istanbul Kültür University, Türkiye, in 2012, and the M.Sc. and Ph.D. degrees (Hons.) in Computer Engineering from Istanbul University, in 2014 and 2018, respectively. From 2019 to 2021, he was an Assistant Professor with the Computer Engineering Department, Turkish Air Force Academy, National Defence University, Istanbul, Türkiye. He is currently an Associate Professor in the Computer Engineering Department, Turkish Air Force Academy, National Defence University. He has published about 80 papers in various esteemed journals and conferences, and has been serving as a member of the reviewer board in nearly 40 prestigious academic journals. He is also on the editorial board of some reputable journals. His current research interests include artificial intelligence, machine learning, artificial neural networks, data privacy, and image processing.