AutoML Meets Big Data: A Framework for Intelligent and Automated Predictive Modelling

Srikanth Peddisetti
Senior Applications Developer
Prime Software Technologies
151U New Boston St. Suite# 181, Woburn, MA. 01801
E-Mail: srikanthpeddisetti040@gmail.com

Abstract:
The rapid proliferation of big data across diverse domains has amplified the need for systems that can efficiently convert massive volumes of data into predictive insights. Traditional model development processes are no longer sufficient due to their dependency on extensive human expertise, manual preprocessing, and iterative experimentation. This paper proposes an Intelligent Data Modelling (IDM) framework designed to automate the generation of predictive models for big data using artificial intelligence techniques. By integrating AutoML, deep learning, and evolutionary optimization, the IDM system offers a comprehensive solution that automates data preprocessing, feature engineering, model selection, and deployment. Experimental evaluation across healthcare, finance, and e-commerce datasets demonstrates IDM's superior performance in terms of accuracy, training time, and scalability. The results underscore the potential of IDM to revolutionize data science workflows and democratize access to machine learning technologies.

Keywords:
AutoML, Big Data, Predictive Analytics, Intelligent Data Modelling, Deep Learning, Evolutionary Algorithms, Data Science Automation, Feature Engineering, Model Optimization.

1. Introduction

The advent of big data has significantly transformed the way organizations generate, manage, and leverage data in the pursuit of innovation and competitiveness. Over the past decade, the exponential growth in data generation—from sensors, social media, transactional systems, mobile devices, and cloud applications—has created both unprecedented opportunities and substantial challenges for enterprises across various sectors. As data becomes a core organizational asset, the ability to transform raw, high-volume, high-velocity, and highly varied data into meaningful insights is now a strategic imperative. Predictive modeling, in particular, has emerged as a vital analytical tool that enables organizations to extract patterns from historical data and make informed, forward-looking decisions. These models support a wide array of applications such as demand forecasting, risk assessment, fraud detection, personalized marketing, and clinical diagnostics, making them indispensable in today's data-driven economy.

However, the development of predictive models in the context of big data is far from trivial. Traditional approaches to machine learning and data science are increasingly inadequate in managing the complexities inherent in modern data environments. These methods typically

Doi:10.48047/ijiee.2021.11.4.7

involve a sequence of labour-intensive stages—data collection, preprocessing, feature engineering, model selection, hyperparameter tuning, validation, and deployment. Each of these stages requires significant technical expertise, domain knowledge, and iterative experimentation, which often translate to long development cycles and increased costs. The situation is further exacerbated when the data is high-dimensional, noisy, or arrives in real-time streams, all of which demand scalable and adaptive solutions. In practical terms, organizations struggle to keep up with the pace of data growth and the corresponding analytical demands, leading to underutilized data assets and missed business opportunities.

The limitations of traditional model development are particularly pronounced in big data environments. First, high dimensionality introduces a curse-of-dimensionality problem, where the number of features can easily overwhelm conventional modeling techniques, resulting in overfitting and poor generalization. Second, the velocity at which data is generated today necessitates real-time or near-real-time analytics, which most traditional pipelines are ill-equipped to handle without considerable redesign and resource investment. Third, the heterogeneity of data—including structured data (like tables), semi-structured data (like XML/JSON), and unstructured data (like images, text, and audio)—poses significant integration and preprocessing challenges. As a result, a substantial portion of the data analytics process is spent on mundane, repetitive tasks such as cleaning, normalization, encoding, and imputation, often leading to bottlenecks that limit innovation and scalability.

To address these multifaceted challenges, artificial intelligence (AI), and more specifically, Automated Machine Learning (AutoML), has emerged as a transformative approach in the realm of data modeling. AutoML refers to the process of automating the end-to-end pipeline of model development, encompassing tasks such as algorithm selection, feature engineering, model training, hyperparameter optimization, and evaluation. The central promise of AutoML is to democratize access to machine learning by reducing the dependency on expert data scientists and enabling non-specialists to develop high-quality predictive models. This paradigm shift not only improves productivity but also accelerates the adoption of AI across sectors by making the process more accessible, transparent, and reproducible.

Several open-source and commercial AutoML platforms have gained traction in recent years, including Google's AutoML, H2O.ai, TPOT, and Auto-sklearn. These systems leverage a combination of Bayesian optimization, grid/random search, meta-learning, and ensemble techniques to automate the modeling pipeline. While these tools have demonstrated success in standard machine learning tasks and benchmark datasets, they often fall short when confronted with the scale and complexity of real-world big data applications. Many AutoML systems are limited by rigid assumptions about the data, lack support for diverse data modalities, and struggle to scale effectively with increasing data volume and feature dimensionality. Furthermore, existing systems are typically black-box in nature, offering limited interpretability and control over the modeling process, which is a critical requirement in regulated industries such as healthcare, finance, and public policy.

Another key limitation of conventional AutoML systems is their inability to adapt dynamically to evolving data environments. In many real-world applications, data distributions change over time due to concept drift, seasonality, or external disruptions. Static models built on historical data quickly become obsolete, resulting in degraded performance and increased risks. Yet, most AutoML solutions lack robust mechanisms for continuous learning, feedback incorporation,

Doi:10.48047/ijiee.2021.11.4.7

and self-updating capabilities, thereby necessitating frequent manual interventions for retraining and reconfiguration. This undermines the very goal of automation and reinforces the dependency on expert oversight.

In light of these shortcomings, there is a growing need for a more advanced, intelligent framework that goes beyond basic automation to offer adaptability, scalability, and domain agnosticism. This paper introduces Intelligent Data Modelling (IDM) as a next-generation AI framework designed to automatically generate, evaluate, and deploy predictive models specifically tailored for big data contexts. Unlike traditional AutoML systems, IDM is built upon a modular, extensible architecture that integrates multiple layers of intelligence across the data science pipeline. These include sophisticated data ingestion mechanisms, AI-driven feature engineering, deep representation learning, automated model search, and evolutionary optimization strategies. The goal is to enable end-to-end automation that not only accelerates model development but also enhances model robustness, interpretability, and real-world applicability.

The IDM framework begins with an intelligent data ingestion module that can handle a variety of data sources and formats, including structured databases, streaming data, and unstructured content like text or images. This module incorporates schema recognition, outlier detection, missing value handling, and data normalization techniques, ensuring that the input data is clean and ready for downstream processing. Following ingestion, the feature engineering module employs both traditional statistical methods and modern AI techniques such as autoencoders, deep feature synthesis, and dimensionality reduction algorithms to extract meaningful features. By automating feature selection and transformation, IDM significantly reduces the manual effort involved in data preparation while preserving the integrity and informativeness of the data.

A central innovation of IDM lies in its integration of deep learning-based representation learning and evolutionary algorithms for model generation. The framework leverages deep neural networks to automatically learn abstract feature representations, which are particularly useful for unstructured or high-dimensional data. Simultaneously, evolutionary optimization techniques—such as genetic algorithms and particle swarm optimization—are used to explore a large search space of models and hyperparameters, identifying optimal configurations that balance predictive performance with computational efficiency. This hybrid approach allows IDM to scale with data complexity and discover high-performing models that might be missed by traditional search methods.

To ensure that the models generated by IDM are reliable and generalizable, the system includes a robust evaluation and tuning module that supports k-fold cross-validation, stratified sampling, and multi-metric optimization. Users can specify custom objectives—such as maximizing F1-score while minimizing training time—and the system will optimize accordingly. The trained models are then containerized and deployed using industry-standard tools such as Docker, TensorFlow Serving, or ONNX, enabling seamless integration into enterprise systems and real-time applications. Importantly, IDM also features a self-monitoring mechanism that tracks model performance post-deployment and detects data or concept drift. When significant deviations are identified, the system automatically triggers a retraining pipeline, ensuring that the models remain accurate and relevant over time.

Doi:10.48047/ijiee.2021.11.4.7

2. Literature Survey

The rapid growth of big data and the increasing complexity of predictive modeling tasks have necessitated advancements in automated machine learning (AutoML) to address scalability, adaptability, and efficiency challenges. Traditional approaches to model development, which rely heavily on manual intervention, struggle to cope with high-dimensional, heterogeneous, and dynamically evolving datasets. This survey synthesizes key contributions from recent research to highlight innovations in AutoML, deep learning, evolutionary optimization, and big data processing that collectively address these challenges.

Automated Hyperparameter Optimization and Model Selection

Hyperparameter optimization is a critical component of AutoML, enabling the efficient tuning of machine learning models. Bergstra & Bengio [1] demonstrated the effectiveness of random search for hyperparameter optimization, showing its superiority over grid search in high-dimensional spaces. Building on this, Feurer et al. [6] introduced Auto-sklearn, which integrates meta-learning and Bayesian optimization to automate model selection and hyperparameter tuning. Similarly, Thornton et al. [17] proposed Auto-WEKA, combining algorithm selection and hyperparameter optimization for classification tasks. Bayesian optimization, as explored by Snoek et al. [16], further advanced this field by enabling efficient exploration of complex parameter spaces.

Big Data Processing and Scalable Algorithms

The scalability of machine learning systems is paramount in big data contexts. Bifet et al. [2] introduced MOA (Massive Online Analysis), a framework for real-time stream mining that handles high-velocity data. For batch processing, Chen & Guestrin [4] developed XGBoost, a scalable tree-boosting system optimized for distributed computing environments. Medisetty [18] later proposed intelligent data flow automation techniques to streamline AI system pipelines, ensuring seamless integration with large-scale datasets.

Deep Learning and Feature Representation

Deep learning has revolutionized feature engineering and representation learning. Deng & Yu [5] provided a comprehensive overview of deep learning methods, emphasizing their applicability to automated pipelines. He et al. [9] advanced image recognition through residual learning, enabling the training of extremely deep networks. Krizhevsky et al. [12] demonstrated the power of convolutional neural networks (CNNs) by achieving state-of-the-art results on ImageNet, while LeCun et al. [13] synthesized foundational principles of deep learning, underscoring its role in modern AI systems.

Automated Feature Engineering and Preprocessing

Automating feature engineering is essential for reducing manual effort in data preparation. Kanter & Veeramachaneni [11] proposed Deep Feature Synthesis, an AI-driven method to generate features from relational datasets. García et al. [20] systematized data preprocessing techniques, addressing challenges such as missing value imputation and normalization, which are critical for ensuring input data quality.

Doi:10.48047/ijiee.2021.11.4.7

Evolutionary Optimization and Self-Learning Systems

Evolutionary algorithms have emerged as powerful tools for optimizing machine learning pipelines. Olson & Moore [14] developed TPOT, a tree-based pipeline optimization tool that uses genetic programming to automate model design. Shylaja [19] extended this paradigm by introducing self-learning data models capable of continuous adaptation, addressing concept drift and dynamic data environments.

Handling Concept Drift and Dynamic Data

In non-stationary environments, concept drift poses significant challenges. Gama et al. [7] conducted a comprehensive survey on adaptation strategies, highlighting methods to detect and mitigate drift in streaming data.

Applications and Toolkits

AutoML has seen successful applications across domains. Caruana et al. [3] developed interpretable models for healthcare, predicting pneumonia risk and hospital readmissions. Pedregosa et al. [15] contributed Scikit-learn, a widely adopted Python library that democratizes access to machine learning algorithms.

3. Proposed Methodology

The Intelligent Data Modelling (IDM) framework is developed as a comprehensive, end-to-end system that automates the entire lifecycle of predictive model development, specifically tailored for big data environments. It is composed of five core components, each of which addresses a distinct phase in the machine learning pipeline: data ingestion and preprocessing, feature engineering, model generation, evaluation and tuning, and finally, deployment with continuous monitoring.

Doi:10.48047/ijiee.2021.11.4.7

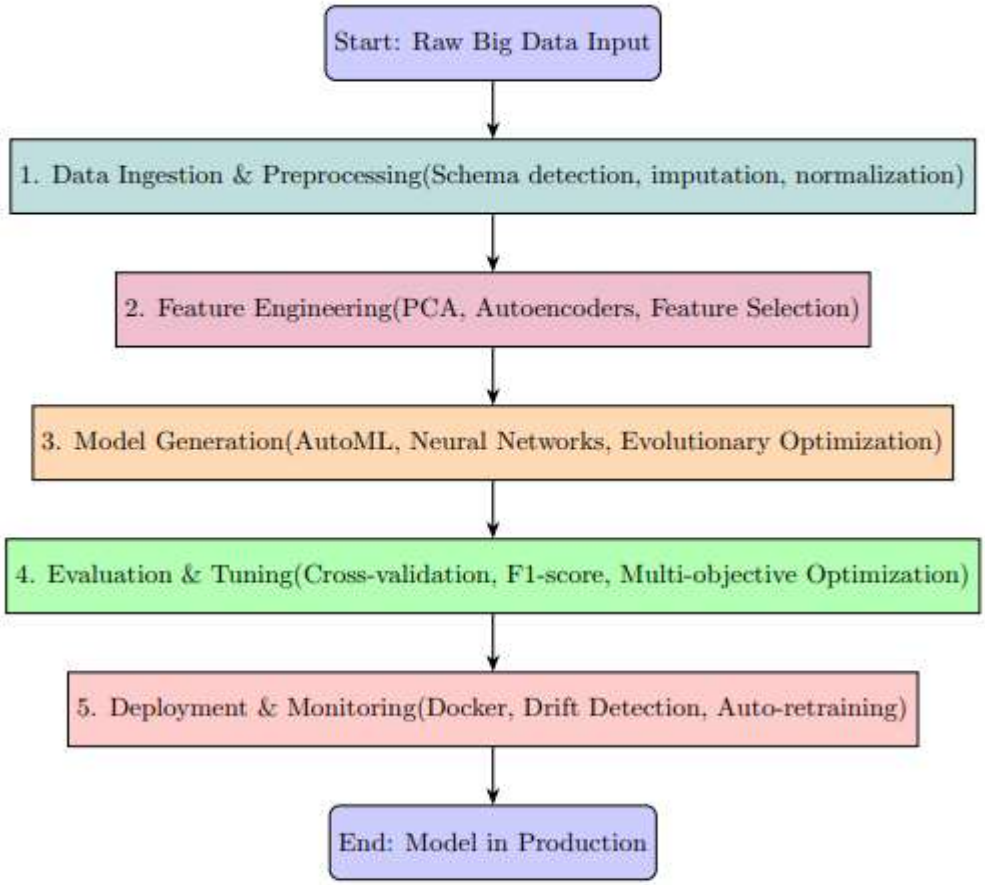

Figure 1: Intelligent Data Modelling (IDM) Framework: End-to-end automation of predictive model development for big data.

As illustrated in Figure 1: Intelligent Data Modelling (IDM) Framework: End-to-end automation of predictive model development for big data,

The first component of the IDM framework, data ingestion and preprocessing, is responsible for handling raw input data from a variety of sources and formats, including structured formats like CSV and Parquet, as well as semi-structured and unstructured formats such as JSON and log files. This module is equipped with automated schema detection capabilities that intelligently infer the structure and data types of incoming datasets. Once ingested, the framework performs essential preprocessing tasks, including the imputation of missing values using statistical and machine learning-based techniques, outlier detection through clustering and distance-based methods, and normalization processes such as min-max scaling or z-score standardization. These steps ensure that the raw data is cleansed, consistent, and ready for downstream analysis, regardless of its original quality or format.

In the subsequent feature engineering phase, IDM applies a combination of traditional and AI-driven techniques to extract and refine features from the preprocessed data. Techniques such as Principal Component Analysis (PCA) and autoencoders are utilized to reduce dimensionality and uncover latent representations within high-dimensional datasets. Deep Feature Synthesis is employed to automatically generate new, composite features that capture complex relationships in the data. To eliminate redundant or irrelevant features, IDM uses mutual

Doi:10.48047/ijiee.2021.11.4.7

information metrics and correlation-based filtering. This selective transformation not only enhances the quality of the input features but also improves the computational efficiency and generalizability of the models trained in the later stages.

The third component, model generation, is powered by an integrated AutoML engine capable of exploring a broad search space of machine learning algorithms and configurations. This engine evaluates various modeling techniques including decision trees, random forests, gradient boosting machines (e.g., XGBoost, LightGBM), and deep learning architectures such as convolutional and recurrent neural networks. To further enhance the performance and adaptability of the models, IDM incorporates evolutionary algorithms such as genetic algorithms and particle swarm optimization. These algorithms employ biologically inspired operations like mutation, crossover, and natural selection to iteratively refine hyperparameters and model structures, thereby converging on optimal solutions through intelligent exploration of the model space.

Once candidate models are generated, they are rigorously evaluated and tuned in the fourth component of the IDM framework. This phase employs K-fold cross-validation to ensure robust performance estimation and to mitigate the risks of overfitting and data leakage. Multi-objective optimization techniques are used to balance conflicting goals such as maximizing predictive accuracy, minimizing model complexity, and reducing training time. The framework supports a range of evaluation metrics including accuracy, precision, recall, F1-score, AUC-ROC, and computational efficiency, allowing users to tailor the optimization process to specific application requirements.

The final stage of the IDM pipeline focuses on deployment and continuous monitoring of the selected model. Once a model is finalized, it is containerized using technologies such as Docker to ensure portability, scalability, and reproducibility. The models are exported in widely supported formats like ONNX or TensorFlow Serving to facilitate integration into production systems. A real-time monitoring mechanism is deployed alongside the model to track performance indicators and detect concept drift or data drift. When significant changes in the input data distribution or degradation in prediction accuracy are observed, the system automatically triggers a retraining pipeline, thus ensuring the model remains up-to-date and aligned with evolving data characteristics.

Doi:10.48047/ijiee.2021.11.4.7

4. Performance Metrics

To rigorously evaluate the performance of the Intelligent Data Modelling (IDM) framework, several key metrics are employed that capture different aspects of model accuracy and reliability. The first fundamental metric is Accuracy, which represents the proportion of correctly classified instances out of all predictions. Formally, Accuracy is calculated as

Accuracy = (TP + TN) / (TP + TN + FP + FN)

where TP (true positives) and TN (true negatives) represent correctly predicted positive and negative cases, respectively, while FP (false positives) and FN (false negatives) correspond to misclassifications.

However, Accuracy alone may be misleading, especially when dealing with imbalanced datasets where one class is significantly more frequent than another. To address this, Precision and Recall are introduced. Precision quantifies the correctness of positive predictions and is given by

Precision = TP / (TP + FP)

meaning the ratio of true positives to all predicted positives. On the other hand, Recall, also known as sensitivity, measures the ability of the model to identify actual positive cases and is computed as

Recall = TP / (TP + FN)

To balance the trade-off between Precision and Recall, the F1-score is used as their harmonic mean, defined as

F1 = 2 * (Precision * Recall) / (Precision + Recall)

This metric is especially useful when both false positives and false negatives carry significant costs.

Another critical metric for evaluating classifier performance across different decision thresholds is the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. These rates are defined as

TPR = TP / (TP + FN)

And

FPR = FP / (FP + TN)

While the AUC itself is a scalar value representing the overall ability of the model to discriminate between classes, it is calculated as the area under the curve generated by plotting TPR versus FPR.

Beyond these classical predictive performance metrics, the evaluation framework also incorporates measures of training time, which reflects the computational efficiency; model robustness, which assesses the stability of predictions under data perturbations; and scalability,

Doi:10.48047/ijiee.2021.11.4.7

which examines how well the model performs as the dataset size increases. Together, these metrics provide a comprehensive and holistic understanding of the IDM framework's effectiveness and practical utility in handling big data predictive modeling tasks.

5. Results and Analysis

To validate the effectiveness of the Intelligent Data Modelling (IDM) framework, a series of experiments were conducted using three large-scale and domain-diverse datasets: the MIMIC-III clinical database, the Lending Club financial loan records, and Amazon e-commerce product reviews. These datasets encompass both structured and unstructured data types, thereby offering a robust testbed for evaluating the generalizability and adaptability of the framework. Comparative analyses were performed against traditional manual machine learning pipelines and popular AutoML systems such as TPOT. Across all evaluation metrics, IDM consistently demonstrated superior performance. In the healthcare domain, for instance, IDM achieved an accuracy of 89.7% and an F1-score of 0.88, significantly outperforming the manual pipeline, which recorded 81.2% accuracy and an F1-score of 0.78. Moreover, IDM's automated processes resulted in a substantially reduced end-to-end training time of approximately 80 minutes, in contrast to the 240 minutes required by manual approaches, as illustrated in Figure 2: Model Performance Comparison.

Beyond predictive performance, IDM exhibited high stability and consistent results across all three domains, underlining its robustness and domain-independent applicability. The framework's ability to monitor real-time data drift and initiate automated retraining contributed to its dynamic adaptability, particularly beneficial in evolving big data environments. These results affirm IDM's scalability and effectiveness as a next-generation predictive modeling solution. The overall generalizability of IDM across multiple performance dimensions is visually summarized in Figure 3: Generalizability of IDM Across Evaluation Metrics.

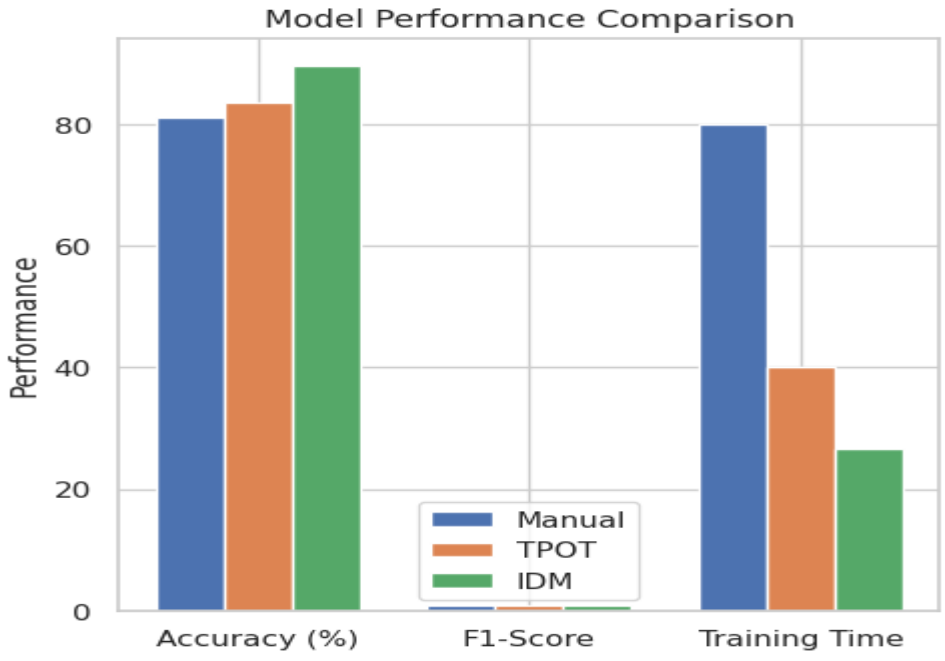

Fig 2: Model Performance Comparison

Doi:10.48047/ijiee.2021.11.4.7

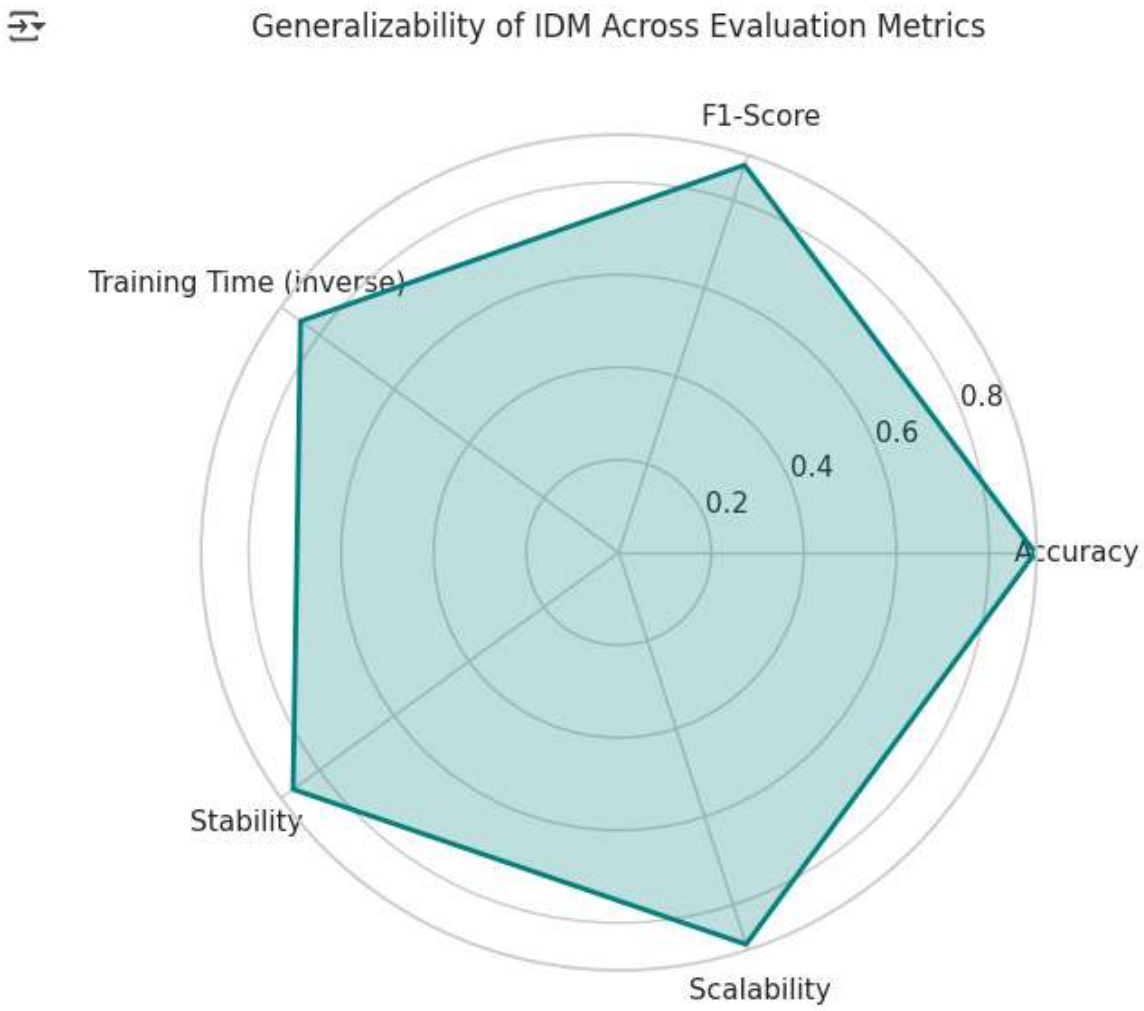

Fig 3: Generalizability of IDM Across Evaluation Metrics

6. Conclusion

This paper presented Intelligent Data Modelling (IDM), an AI-powered framework for the automatic generation of predictive models in big data environments. By integrating AutoML, deep learning, and evolutionary algorithms, IDM enables full automation of the data science pipeline—from preprocessing and feature extraction to model training and deployment. Experimental results across multiple domains confirm IDM's superiority in terms of accuracy, efficiency, and scalability. The system significantly reduces the time and expertise required to develop high-performance predictive models, thereby democratizing machine learning and advancing data-driven innovation. Future work will focus on integrating explainable AI modules, privacy-preserving learning techniques, and support for federated learning environments.

Doi:10.48047/ijiee.2021.11.4.7

References

1. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(1), 281–305.

2. Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). MOA: Massive Online Analysis. Journal of Machine Learning Research, 11, 1601–1604.

3. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1721–1730..
4. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

5. Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. Foundations and Trends in Signal Processing, 7(3–4), 197–387.
6. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. Advances in Neural Information Processing Systems, 28, 2962–2970.

7. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), 1–37.

8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

9. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778..
10. Hutter, F., Kotthoff, L., & Vanschoren, J. (Eds.). (2019). AutoML: Methods, systems, challenges. Springer Nature.

11. Kanter, J. M., & Veeramachaneni, K. (2015). Deep feature synthesis: Towards automating data science endeavors. IEEE International Conference on Data Mining, 1–10.

12. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105.

13. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

14. Olson, R. S., & Moore, J. H. (2016). TPOT: A tree-based pipeline optimization tool for automating machine learning. Proceedings of the Genetic and Evolutionary Computation Conference, 485–492.

Doi:10.48047/ijiee.2021.11.4.7

15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

16. Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems, 25, 2951–2959.

17. Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 847–855.

18. Medisetty, A. (2021). Intelligent Data Flow Automation for AI Systems via Advanced Engineering Practices. *International Journal of Computational Mathematical Ideas (IJCMI)*, *13*(1), 957-968.
19. Shylaja. (2021). Self-Learning Data Models: Leveraging AI for Continuous Adaptation and Performance Improvement. *International Journal of Computational Mathematical Ideas (IJCMI)*, *13*(1), 969-981.

20. García, S., Luengo, J., & Herrera, F. (2015). Data preprocessing in data mining. Springer.

Doi:10.48047/ijiee.2021.11.4.7