A PARAMETRIC INVESTIGATION OF THE TRAIN-TEST RATIO FOR MACHINE LEARNING ALGORITHMS IN STRUCTURAL MECHANICS APPLICATIONS

Daniel Rademan[1], and George Markou[2,3]

[1] University of Pretoria
cnr Lynwood Road and Roper Street, Hatfield, South Africa
u18049037@tuks.co.za

[2] University of Pretoria
cnr Lynwood Road and Roper Street, Hatfield, South Africa
george.markou@up.ac.za

[3]Neapolis University Pafos
Department of Civil Engineering, 2 Danais Avenue, Pafos 8042, Cyprus;
g.makrou@nup.ac.cy

ABSTRACT

This research investigates the impact of varying train-test ratios on machine learning (ML) algorithms in the context of structural mechanics. Notably, it reveals that these algorithms favour lower train-test ratios to counter overfitting and enhance robustness. The study employed five datasets with various characteristics, objectively evaluating the train-test ratios' influence when implementing the under-study ML algorithms. This research identified optimal ratios for different ML algorithms, contributing to a more tailored approach to ML algorithmic performance assessment. According to the parametric investigation, it was found that the range of train-test ratios that is optimal for most algorithms is 0.1-0.15, suggesting a need to explore a wider range of ratios with smaller intervals for a comprehensive and more detailed performance assessment. Finally, the study calls for future research to explore the response of ML algorithms when trained on numerical versus experimental datasets, potentially leading to a generalised recommendation when dealing with structural mechanics applications.

KEYWORDS: MACHINE LEARNING, PREDICTIVE MODELS, TRAIN-TEST RATIO, ERROR METRICS, STRUCTURAL MECHANICS

INTRODUCTION

In recent years, structural mechanics has undergone a transformative shift with the incorporation of machine learning (ML) techniques, offering new avenues for optimising the design, maintenance, and monitoring of complex structural systems. This transformation has been demonstrated in the work of Kicinger et al. (2005), Liao et al. (2011), and Amezquita-Sanchez et al. (2016). This shift is considered a significant advancement in the field. However, the performance of these algorithms is highly contingent on the data available for training and testing, and the train-test ratio plays a fundamental role in this regard. While guidelines exist for selecting train-test ratios (Joseph 2022), there remains a scarcity of comprehensive studies that systematically investigate the influence of varying train-test ratios on the performance of ML algorithms in the context of structural mechanics applications.

To address this research gap, this paper aims to conduct a parametric investigation into the impact of the train-test ratio on ML algorithms in structural mechanics. Through a series of experiments, this research will explore how different train-test ratios affect the algorithm's accuracy, robustness, and generalisation across various structural mechanics scenarios. The findings will provide valuable insights for practitioners and researchers in structural mechanics, aiding in the selection of optimal train-test ratios tailored to specific applications.

Furthermore, the study builds upon the work of previous researchers who have explored the integration of ML in structural mechanics (Markou et al., 2024). By shedding light on the critical role of the train-test ratio in ML applications within the field of structural mechanics, this study contributes to the ongoing efforts to harness the potential of artificial intelligence (AI) and ML algorithms for more efficient and accurate structural analysis and design.

The subsequent sections will delve into the methodology, present experimental results, and discuss the implications of the findings. By combining new research with the existing body of knowledge, this research aims to offer a comprehensive perspective on the use of ML in structural mechanics while emphasising the significance of optimal data partitioning.

MACHINE LEARNING

Machine learning in structural mechanics

Adeli and Hung (1994) introduced the pioneering concept of multiparadigm learning, emphasising the integration of various AI disciplines, including neural networks, genetic algorithms, fuzzy sets, and parallel processing. Their work demonstrated that such integration significantly enhances performance, laying the foundation for hybrid AI approaches in civil engineering applications. Kicinger et al. (2005) conducted an extensive study on the application of evolutionary computation within the context of structural design. This exploration underscores the potential of AI-based optimisation techniques in enhancing structural designs, providing valuable insights into the optimisation of civil engineering processes. Lu et al. (2012) conducted a comprehensive survey of diverse AI methods, encompassing fuzzy logic, evolutionary computation, neural networks, swarm intelligence, and expert systems, among others. This broad survey underscores the wide array of AI tools available for civil engineering applications and provides a reference for the range of methods under consideration.

Liao et al. (2011) reviewed studies focusing on metaheuristics as optimisation techniques to address the lifespan of construction and engineering projects. This work highlights the role of AI-driven metaheuristics in optimising project management. Saka and Geem (2013) conducted a survey on mathematical and metaheuristic algorithms in the design optimisation of steel frame structures, emphasising the importance of optimisation in structural design. Meanwhile, Aldwaik and Adeli (2014) reviewed the progress in optimising high-rise buildings, and Mardani et al. (2015) explored fuzzy multiple-criteria decision-making techniques. These studies collectively delve into the nuances of structural design optimisation, offering a valuable reference point for this research. Nasiri et al. (2017) conducted a survey on various AI methods, including artificial neural networks, Bayesian analysis, genetic algorithms, and case-based reasoning, in the field of fracture mechanics. This reference provides insights into the applications of AI within fracture mechanics, a critical aspect of structural analysis. Zamarron et al. (2017) reviewed the application of multi-criteria decision analysis for ageing dam management. This work exemplifies the integration of decision-making frameworks within civil engineering.

Pongiglione and Calderini (2016) presented a state-of-the-art overview of sustainable structural design within the context of green building rating systems and building codes. Their work highlights the evolving importance of sustainability within civil engineering. Varela et al. (2018) assessed the social sustainability of infrastructure using multi-criteria decision analysis, and Zavadskas et al. (2017) surveyed state-of-the-art methods applied to sustainable decision-making in civil engineering, construction, and building technology. These studies collectively underscore the importance of sustainability considerations in infrastructure and construction, aligning with our research focus.

The train-test ratio in machine learning

One of the pivotal aspects of ML algorithm development is the choice of the train-test ratio, which determines the proportion of data allocated to training and testing subsets. This selection significantly impacts algorithm robustness, generalisation, and overall performance.

Rácz et al. (2021) compared several combinations of dataset sizes and split ratios with five different ML algorithms to find the differences or similarities and to select the best parameter settings in nonbinary (multiclass) classification. Rácz et al. (2021) concluded that the size of the datasets and the train-test split ratios can greatly affect the outcome of the algorithms and thus the classification performance itself. The importance of selecting appropriate train-test ratios is paramount, especially in the context of classification tasks.

Data availability and its implications

Structural mechanics faces a significant challenge when it comes to the size and accessibility of datasets, and this challenge is central to the development of ML algorithms. Just as the choice of algorithm is crucial (as emphasised by Asteris et al., 2021), the availability and quality of the dataset play an equally critical role in algorithm training.

While numerical simulations provide a substantial and easily obtainable source of data, the practical application of ML algorithms in real-world scenarios often grapples with constraints arising from the limited availability of experimental data. As highlighted by Thai (2022), efforts have been made to establish database platforms, but these primarily rely on data collected from tests or simulations of individual structural components. The databases essential for predicting the behaviour and strength of complete structural systems remain scarce. Generating such databases would ideally involve finite element simulations, a more cost-effective alternative to expensive experimental tests, as described by Dimiduk et al. (2018).

This scarcity of comprehensive data is influenced by multiple factors, including financial considerations, the time required for data collection, and safety concerns. Consequently, the field of structural mechanics frequently necessitates the integration of both numerical and experimental datasets, as exemplified in the research of Cheraghzade and Roohi (2022). This integration, however, introduces unique challenges in handling diverse data types.

Within structural mechanics, data availability presents a multifaceted challenge. The field grapples with a significant disparity in data sizes, where numerical simulations yield vast datasets in contrast to the limited availability of experimental data from real-world structures. The ability to harmonise these disparate data sources while maintaining the reliability of algorithms stands as a critical domain for ongoing research and advancement.

METHODOLOGY AND EXPERIMENTAL DESIGN

This section outlines the approach for investigating the influence of train-test ratios on ML algorithms in structural mechanics applications that will be described below. It also includes a description of the datasets, parameter variations, and experimental setup.

Dataset description

This research employs a set of five datasets, which are accessible in Bakas et al. (2023). Detailed information regarding the datasets used is presented in Table 1. These datasets were developed for the need to propose predictive models for computing the fundamental period of reinforced concrete and steel structures, the shear strength of slender concrete beams without stirrups, and the deflection of curved steel I-beams. The selected datasets consist of different numbers of data points and thus will allow an objective investigation of the train-test ratio.

Table 1: Dataset information

Dataset	Description	no. Variables	no. Datapoints
1	The fundamental period of a reinforced concrete frame	6	790
2	The deflection of a curved steel I-beam	10	1320
3	The fundamental period of a steel frame 1	6	98308
4	The fundamental period of a steel frame 2	6	1153
5	The capacity of a slender reinforced concrete beam	10	35849

Machine learning algorithms

For the purposes of this research work, the baseline ML algorithm is linear regression (LR), serving as a reference for comparison. Additionally, four advanced ML algorithms, each with specific enhancements and hyperparameter tuning, are used. These algorithms, as detailed in Markou et al. (2024), include the Polynomial Regression with hyperparameter tuning (POLYREG-HYT), Extreme Gradient Boosting with hyperparameter tuning and cross-validation (XGBoost-HYT-CV), Random Forests with hyperparameter tuning (RF-HYT), and MPI and Horovod-based deep learning Artificial Neural Network with hyperparameter tuning (DANN-MPIH-HYT).

Parameter tuning and evaluation metrics

A range of test ratios, specifically 0.1, 0.15, 0.2, 0.25, 0.3, and 0.35, were systematically investigated. This approach enables a comprehensive assessment of the effect of data partitioning (in a relevant range) on ML algorithm performance in structural mechanics applications. Additionally, a multifaceted performance evaluation strategy was implemented, incorporating various metrics such as Pearson correlation coefficient (R), mean absolute percentage error (MAPE), mean absolute mean percentage error (MAMPE), mean absolute error (MAE), and root mean squared error (RMSE) (Markou et al., 2024). This diverse set of metrics establishes a robust framework for assessing algorithm accuracy, precision, and generalisation capabilities, ensuring a thorough examination of effectiveness in the context of structural mechanics.

RESULTS

The most important numerical characteristic of ML algorithms is the ability to produce generalised, accurate, and objective relationships between the given input and output features. Furthermore, emphasising the methods' training and testing abilities lies in the practical relevance of evaluating how well ML algorithms perform when presented with new, out-of-sample data, which is crucial for their real-world applicability. In order to evaluate any ML algorithm's performance, error metrics are used. Specifically, the average of the Pearson correlation coefficient (R) and mean absolute percentage error (MAPE) across the datasets were computed and analysed after each run. These averaged metrics offer a consolidated perspective on the overall performance of the algorithms, accounting for variations in individual datasets. As stated above, each ML algorithm was used to train on all datasets selected for the needs of this research work, and for each dataset, six different train-test ratios were used. After the completion of each analysis, the data were stored in a tabulated manner where the average error metrics were then computed. Due to brevity purposes, only the averaged R and MAPE are presented in this manuscript.

Figure 1 displays the trends in the average R and MAPE error metrics across the stipulated range of train-test ratios for the LR ML algorithm, which serves as the baseline algorithm. When analysing the trends, it is apparent that the train-test ratio of 0.15 produces the best results for the case of this ML algorithm, which resulted in the highest error.

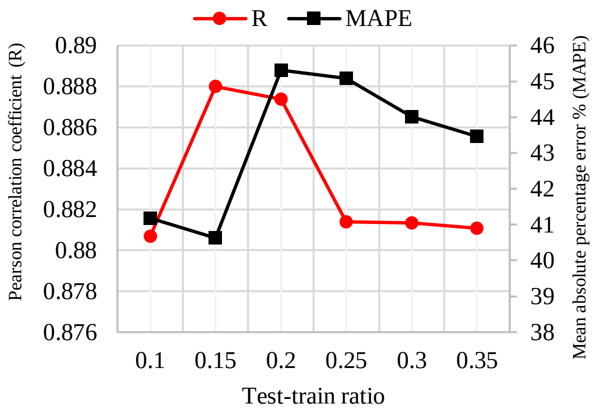

Figure 1: Average R and MAPE for the LR algorithm

Figure 2 through Figure 5 present the trends in the average R parameter and the MAPE error metric derived by: POLYREG-HYT, XGBoost-HYT-CV, RF-HYT, and DANN-MPIH-HYT, respectively. According to the numerical findings of this research work, the POLYREG-HYT and XGBoost-HYT-CV methods perform optimally when a train-test ratio of 0.1 is used. Additionally, the RF-HYT method shows optimum accuracy and correlation R at a ratio of 0.1, with 0.2 producing results that are of similar performance. Notably, DANN-MPIH-HYT achieves its peak performance at a train-test ratio of 0.15.

It is worth noting that the extent of variation in both the R and MAPE (across the range of train-test ratios) for XGBoost-HYT-CV and RF-HYT is notably minimal. In comparison, POLYREG-HYT exhibits a slightly higher degree of variation, while DANN-MPIH-HYT displays even greater variability than LR.

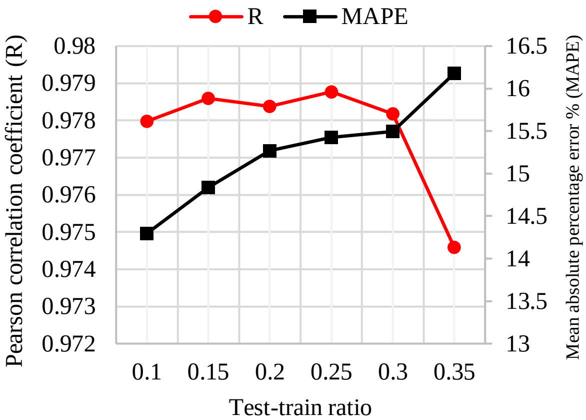

Figure 2: Average R and MAPE for the POLYREG-HYT algorithm

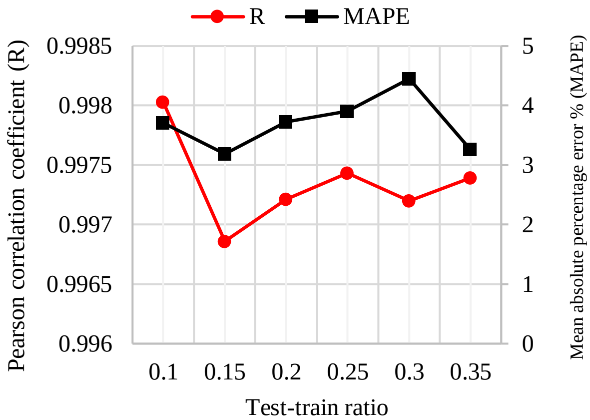

Figure 3: Average R and MAPE for the XGBoost-HYT-CV algorithm

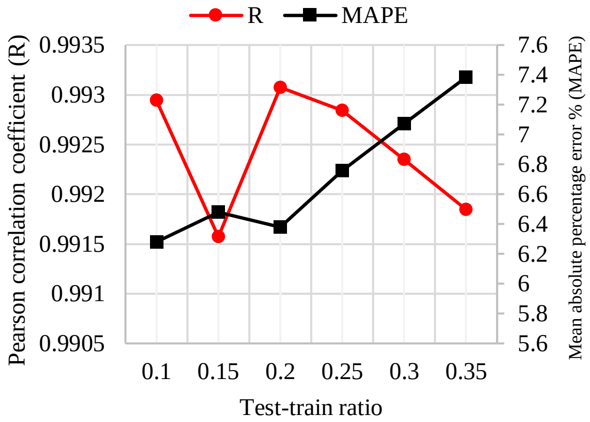

Figure 4: Average R and MAPE for the RF-HYT algorithm

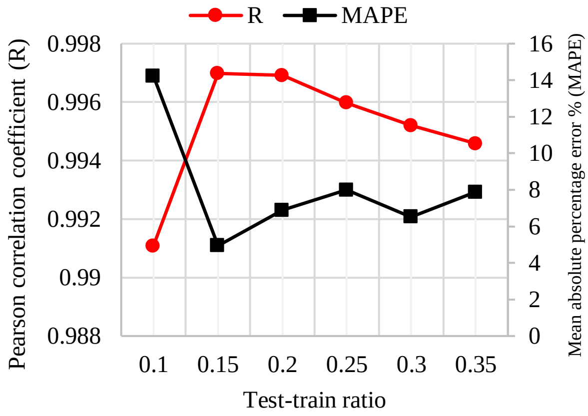

Figure 5: Average R and MAPE for the DANN-MPIH-HYT algorithm

When analysing the dataset averages of test error metrics across the entire range of train-test ratios, it becomes evident that the XGBoost-HYT-CV ML algorithm consistently exhibits the best performance, characterised by the lowest average error metrics and the highest R-value. Following closely is the DANN-MPIH-HYT algorithm, which delivers accurate results but at the expense of higher computational demand. RF-HYT performs well, landing in the middle ground. POLYREG-HYT demonstrates respectable performance, while LR, serving as the baseline algorithm, ranks as the least-performing ML algorithm among those assessed. Notably, this outcome is in alignment with findings by Markou et al. (2024). The complete details of the error metrics can be found in the appendix for a more comprehensive understanding of the algorithm's performance.

CONCLUSION

In the context of this parametric investigation, it has been found that most of the under-study ML algorithms exhibited a more accurate numerical response for low train-test ratios (0.1-0.15). This is attributed to the formulation of the ML algorithms, and it is also in line with the observations made by Markou et al. (2024). It is important to note here that the optimal choice of a specific train-test ratio is particularly beneficial when data availability is limited; thus, having to use a smaller test ratio will ensure a smaller proportion of the data to be used for testing, while the ML algorithm will still be able to generate a generalised predictive model. Low train-test ratios can be advantageous in scenarios where the dataset exhibits a high degree of homogeneity, whereas ML algorithms can effectively capture underlying patterns.

Moreover, when considering the effect of dataset size on the performance of ML algorithms, interesting variations emerge. It was found that for all the ML algorithms apart from DANN-MPIH-HYT, there is generally more variation in both R and MAPE for the smaller datasets, while the larger datasets demonstrate greater stability. However, in the case of DANN-MPIH-HYT, higher variation is observed across datasets, and there appears to be no discernible correlation between the variation (R and MAPE) and dataset size. This is attributed to the fact that deep learning algorithms require an exceptionally large amount of data to perform optimally. Nevertheless, it is shown from this research work that this advanced ML method is able to derive accurate models even in the case of small datasets.

After this parametric investigation of the train-test ratio, it is deemed necessary to expand it through the use of more datasets found in international literature. Future research should focus on analysing the performance differences between algorithms trained on numerical and experimental datasets and should consider testing a wider range of ratios with smaller intervals to allow for a more detailed performance assessment. This will culminate in the development of a more generalised recommendation for selecting the most optimal train-test ratio.

REFERENCES

Adeli, H. and Hung, S.-L. 1994. *Machine Learning: Neural Networks, Genetic Algorithms, and Fuzzy Systems*. John Wiley & Sons, Inc.

Aldwaik, M. and Adeli, H. 2014. Advances in optimization of highrise building structures. *Structural and Multidisciplinary Optimization* 50, pp. 899–919. doi: 10.1007/s00158-014-1148-1.

Amezquita-Sanchez, J., Valtierra-Rodriguez, M., Aldwaik, M. and Adeli, H. 2016. Neurocomputing in Civil Infrastructure. *Scientia Iranica* 23, pp. 2417–2428. doi: 10.24200/sci.2016.2301.

Asteris, P.G., Skentou, A.D., Bardhan, A., Samui, P. and Pilakoutas, K. 2021. Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models. *Cement and Concrete Research* 145, p. 106449. Available at: https://www.sciencedirect.com/science/article/pii/S0008884621000983.

Bakas, N., Markou, G., Langousis, A. and Savvas, C. 2023. *NBML: computer software for data analysis and predictive modelling with artificial intelligence algorithms*.

Cheraghzade, M. and Roohi, M. 2022. Deep learning for seismic structural monitoring by accounting for mechanics-based model uncertainty. *Journal of Building Engineering* 57, p. 104837. Available at: https://www.sciencedirect.com/science/article/pii/S2352710222008506.

Dimiduk, D.M., Holm, E.A. and Niezgoda, S.R. 2018. Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering. *Integrating Materials and Manufacturing Innovation* 7(3), pp. 157–172. Available at: https://doi.org/10.1007/s40192-018-0117-8.

Joseph, V.R. 2022. Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 15(4), pp. 531–538. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.11583.

Kicinger, R., Arciszewski, T. and De Jong, K. 2005. Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers & Structures* 83, pp. 1943–1978. doi: 10.1016/j.compstruc.2005.03.002.

Liao, T., Egbelu, P., Sarker, B. and Leu, S.S. 2011. Metaheuristics for project and construction management-A state-of-the-art review. *Automation in Construction* 20, pp. 491–505. doi: 10.1016/j.autcon.2010.12.006.

Lu, P., Chen, S. and Zheng, Y. 2012. Artificial Intelligence in Civil Engineering. Kang, F. ed. *Mathematical Problems in Engineering* 2012, p. 145974. Available at: https://doi.org/10.1155/2012/145974.

Mardani, A., Jusoh, A. and Zavadskas, E.K. 2015. Fuzzy multiple criteria decision-making techniques and applications – Two decades review from 1994 to 2014. *Expert Systems with Applications* 42(8), pp. 4126–4148. Available at: https://www.sciencedirect.com/science/article/pii/S0957417415000081.

Markou, G., Bakas, N., Chatzichristofis, S. and Papadrakakis, M., 2024. A General Framework of High-Performance Machine Learning Algorithms: Application in Structural Mechanics. *Computational Mechanics*, Volume 73, pages 705–729.

Nasiri, S., Khosravani, M.R. and Weinberg, K. 2017. Fracture mechanics and mechanical fault detection by artificial intelligence methods: A review. *Engineering Failure Analysis* 81, pp. 270–293. Available at: https://api.semanticscholar.org/CorpusID:115410194.

Pongiglione, M. and Calderini, C. 2016. Sustainable Structural Design: Comprehensive Literature Review. *Journal of Structural Engineering* 142, p. 04016139. doi: 10.1061/(ASCE)ST.1943-541X.0001621.

Rácz, A., Bajusz, D. and Héberger, K. 2021. Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification. *Molecules* 26, p. 1111. doi: 10.3390/molecules26041111.

Saka, M.P. and Geem, Z.W. 2013. Mathematical and Metaheuristic Applications in Design Optimization of Steel Frame Structures: An Extensive Review. Chen, S. ed. *Mathematical Problems in Engineering* 2013, p. 271031. Available at: https://doi.org/10.1155/2013/271031.

Sierra Varela, L., Yepes, V. and Pellicer, E. 2018. A Review of Multi-Criteria Assessment of the Social Sustainability of Infrastructures. *Journal of Cleaner Production* 187. doi: 10.1016/j.jclepro.2018.03.022.

Thai, H.-T. 2022. Machine learning for structural engineering: A state-of-the-art review. *Structures* 38, pp. 448–491. Available at: https://www.sciencedirect.com/science/article/pii/S2352012422000947.

Zamarron, I., Yepes, V. and Moreno-Jiménez, J.M. 2017. A systematic review of application of multi-criteria decision analysis for aging-dam management. *Journal of Cleaner Production* 147, pp. 217–230. doi: 10.1016/j.jclepro.2017.01.092.

Zavadskas, E., Antucheviciene, J., Vilutienė, T. and Adeli, H. 2017. Sustainable Decision-Making in Civil Engineering, Construction and Building Technology. *Sustainability* 10, p. Article No 14. doi: 10.3390/su10010014.

APPENDIX

Appendix A

A1: LR error metrics and computation time for dataset 1

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.987260	13.6103	7.5713	0.0371	0.0488	0.0010
	Test	0.986769	15.0681	8.4216	0.0366	0.0480	0.0000
0.15	Train	0.987618	13.4805	7.5059	0.0365	0.0484	0.0006
	Test	0.984799	14.1575	8.3323	0.0390	0.0509	0.0000
0.2	Train	0.987307	13.5695	7.5299	0.0369	0.0487	0.0010
	Test	0.986872	14.4159	8.1170	0.0375	0.0488	0.0000
0.25	Train	0.987274	13.3843	7.4979	0.0367	0.0487	0.0010
	Test	0.987054	14.8063	8.1306	0.0380	0.0490	0.0000
0.3	Train	0.987201	13.3186	7.5590	0.0367	0.0486	0.0010
	Test	0.987290	14.8681	7.8821	0.0378	0.0491	0.0000
0.35	Train	0.987384	13.3330	7.5429	0.0365	0.0483	0.0010
	Test	0.986958	14.8315	7.9507	0.0384	0.0497	0.0000

A2: LR error metrics and computation time for dataset 2

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.891378	107.3225	30.2801	2.8882	3.9892	0.0010
	Test	0.902645	71.8096	26.8191	2.7234	3.8475	0.0000
0.15	Train	0.889842	109.0105	30.5193	2.9013	4.0204	0.0010
	Test	0.907926	70.7771	26.2270	2.6564	3.7122	0.0000
0.2	Train	0.888638	107.3536	30.6024	2.9228	4.0511	0.0010
	Test	0.907620	90.9117	27.2288	2.6671	3.6633	0.0000
0.25	Train	0.888158	108.9437	30.8291	2.9272	4.0693	0.0010
	Test	0.905451	88.0989	27.3495	2.7116	3.6814	0.0000
0.3	Train	0.888016	109.3574	30.7538	2.8816	4.0347	0.0010
	Test	0.902441	82.6297	27.6881	2.8065	3.8424	0.0000
0.35	Train	0.887564	110.4007	30.8493	2.8483	4.0347	0.0010
	Test	0.899861	80.0158	27.7895	2.8578	3.8945	0.0000

A3: LR error metrics and computation time for dataset 3

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.923695	23.1002	20.1519	0.5002	0.5823	0.0050
	Test	0.923808	22.9266	20.1435	0.4985	0.5817	0.0001
0.15	Train	0.923597	23.0871	20.1530	0.5003	0.5823	0.0040
	Test	0.924316	22.9325	20.1244	0.4979	0.5819	0.0001
0.2	Train	0.923439	23.0730	20.1555	0.5007	0.5827	0.0040
	Test	0.924751	22.9912	20.1159	0.4968	0.5804	0.0001
0.25	Train	0.923691	23.0597	20.1347	0.5003	0.5822	0.0040
	Test	0.923739	23.0496	20.1807	0.4988	0.5824	0.0001
0.3	Train	0.923588	23.0694	20.1469	0.5002	0.5821	0.0040
	Test	0.923975	23.0401	20.1530	0.4995	0.5825	0.0002
0.35	Train	0.923525	23.0751	20.1648	0.5006	0.5826	0.0040
	Test	0.924042	23.0377	20.1307	0.4991	0.5816	0.0002

A4: LR error metrics and computation time for dataset 4

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.735160	64.7434	47.6502	0.6299	0.7201	0.0010
	Test	0.709729	66.2862	47.1927	0.6636	0.7390	0.0000
0.15	Train	0.731077	64.7201	47.7507	0.6326	0.7221	0.0010
	Test	0.743005	65.3191	46.7779	0.6373	0.7219	0.0000
0.2	Train	0.730667	64.9207	47.4586	0.6316	0.7214	0.0010
	Test	0.741062	67.8621	48.1772	0.6402	0.7261	0.0000
0.25	Train	0.737850	64.9206	46.9411	0.6303	0.7200	0.0010
	Test	0.717455	68.8546	49.6604	0.6422	0.7302	0.0000
0.3	Train	0.738144	64.9977	46.8626	0.6292	0.7184	0.0010
	Test	0.718624	68.8271	49.3859	0.6428	0.7348	0.0000
0.35	Train	0.737337	65.1642	46.9953	0.6280	0.7188	0.0015
	Test	0.721542	68.7006	48.7436	0.6432	0.7321	0.0000

A5: LR error metrics and computation time for dataset 5

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.872197	30.7180	22.4127	26.7088	34.9664	0.0030
	Test	0.880421	29.7941	21.6397	25.7564	33.6050	0.0001
0.15	Train	0.871806	30.7575	22.4575	26.7460	35.0212	0.0030
	Test	0.879927	29.9541	21.6692	25.8897	33.7448	0.0001
0.2	Train	0.872130	30.6627	22.4281	26.7045	35.0117	0.0030
	Test	0.876577	30.3835	21.9841	26.2709	34.1081	0.0001
0.25	Train	0.872929	30.6022	22.3687	26.6379	34.9371	0.0030
	Test	0.873240	30.6336	22.2093	26.5084	34.5209	0.0001
0.3	Train	0.872439	30.6273	22.3794	26.6307	34.9280	0.0030
	Test	0.874325	30.6780	22.2293	26.5685	34.6123	0.0001
0.35	Train	0.873019	30.5733	22.3172	26.5633	34.8565	0.0030
	Test	0.872965	30.7458	22.3511	26.6850	34.7928	0.0001

A6: POLYREG-HYT error metrics and computation time for dataset 1

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.996412	5.2681	3.9827	0.0195	0.0259	9.16
	Test	0.996934	4.7426	3.8214	0.0166	0.0230	0.00
0.15	Train	0.996741	4.7176	3.7583	0.0183	0.0248	11.02
	Test	0.996433	5.0736	3.9209	0.0183	0.0248	0.00
0.2	Train	0.995806	5.1230	4.1967	0.0205	0.0280	10.20
	Test	0.995920	5.0193	4.1987	0.0194	0.0272	0.00
0.25	Train	0.996336	5.2395	4.0406	0.0198	0.0262	8.02
	Test	0.996254	5.6807	4.1437	0.0194	0.0263	0.00
0.3	Train	0.996572	5.1551	3.9195	0.0190	0.0252	7.81
	Test	0.995734	5.7602	4.3755	0.0210	0.0285	0.00
0.35	Train	0.996534	5.1149	3.8957	0.0189	0.0253	6.49
	Test	0.995751	5.4576	4.1974	0.0203	0.0283	0.00

A7: POLYREG-HYT error metrics and computation time for dataset 2

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.988235	20.0019	9.0757	0.8657	1.3461	18.34
	Test	0.984704	14.6522	9.4318	0.9578	1.5480	0.00
0.15	Train	0.986966	19.9990	9.4859	0.9018	1.4181	20.73
	Test	0.983540	16.3398	9.9196	1.0047	1.6122	0.00
0.2	Train	0.989773	18.1104	8.4964	0.8115	1.2601	26.98
	Test	0.986301	17.2807	9.2545	0.9065	1.4695	0.00
0.25	Train	0.991645	16.7146	8.0058	0.7601	1.1423	29.50
	Test	0.987296	17.5910	9.0438	0.8967	1.3846	0.00
0.3	Train	0.989902	16.2108	8.4864	0.7952	1.2438	25.77
	Test	0.985726	19.2065	9.6280	0.9759	1.4984	0.00
0.35	Train	0.985191	21.2685	10.5488	0.9740	1.5017	14.69
	Test	0.978551	18.7898	11.5365	1.1864	1.8287	0.00

A8: POLYREG-HYT error metrics and computation time for dataset 3

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.992653	8.6176	4.8891	0.1213	0.1839	1229.64
	Test	0.992646	8.6221	4.8824	0.1208	0.1839	0.01
0.15	Train	0.992659	8.6212	4.8879	0.1213	0.1837	1121.64
	Test	0.992588	8.6627	4.8986	0.1212	0.1853	0.01
0.2	Train	0.992604	8.6396	4.8966	0.1216	0.1843	1028.16
	Test	0.992752	8.7428	4.9003	0.1210	0.1833	0.01
0.25	Train	0.992663	8.6106	4.8777	0.1212	0.1837	1140.69
	Test	0.992623	8.6293	4.9034	0.1212	0.1844	0.02
0.3	Train	0.992655	8.6136	4.8867	0.1213	0.1837	952.18
	Test	0.992640	8.6165	4.8976	0.1214	0.1845	0.02
0.35	Train	0.992622	8.6402	4.9031	0.1217	0.1842	901.18
	Test	0.992703	8.6278	4.8818	0.1210	0.1834	0.04

A9: POLYREG-HYT error metrics and computation time for dataset 4

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.970022	22.8935	13.9141	0.1839	0.2581	9.05
	Test	0.967969	23.7743	13.8147	0.1943	0.2636	0.00
0.15	Train	0.969229	22.9549	14.1123	0.1870	0.2605	7.31
	Test	0.975809	23.9196	13.3219	0.1815	0.2385	0.00
0.2	Train	0.969281	23.5803	14.0147	0.1865	0.2599	8.60
	Test	0.973665	24.5414	13.5609	0.1802	0.2463	0.00
0.25	Train	0.969008	24.2773	14.1270	0.1897	0.2635	9.54
	Test	0.972271	24.9080	13.7111	0.1773	0.2442	0.00
0.3	Train	0.968833	23.4752	13.9872	0.1878	0.2638	9.33
	Test	0.969385	23.7857	13.8715	0.1805	0.2589	0.00
0.35	Train	0.963728	23.7571	13.9987	0.1871	0.2841	8.44
	Test	0.964320	25.9418	14.1170	0.1863	0.2800	0.00

A10: POLYREG-HYT error metrics and computation time for dataset 5

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.947759	20.2660	13.6749	16.2961	22.8018	560.44
	Test	0.947612	19.6857	13.5348	16.1096	22.6377	0.01
0.15	Train	0.945905	20.5495	13.9664	16.6334	23.1951	540.53
	Test	0.944583	20.1719	13.9235	16.6355	23.3153	0.01
0.2	Train	0.944980	20.7425	14.1135	16.8045	23.4083	676.32
	Test	0.943234	20.7618	14.1362	16.8926	23.5398	0.01
0.25	Train	0.946738	20.4899	13.9208	16.5776	23.0598	565.08
	Test	0.945406	20.3338	13.8892	16.5778	23.0904	0.02
0.3	Train	0.947658	20.1974	13.7169	16.3226	22.8182	487.43
	Test	0.947409	20.1189	13.6685	16.3367	22.8246	0.02
0.35	Train	0.940993	22.3049	14.6597	17.4488	24.1878	349.65
	Test	0.941591	22.0785	14.5562	17.3787	24.0190	0.02

A11: XGBoost-HYT-CV error metrics and computation time for dataset 1

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.997870	3.9108	3.0433	0.0149	0.0200	40.49
	Test	0.997628	5.1397	3.5457	0.0154	0.0203	0.00
0.15	Train	0.998580	2.2075	2.1405	0.0104	0.0164	40.30
	Test	0.993039	3.9284	3.3121	0.0155	0.0348	0.00
0.2	Train	0.998588	2.2674	2.1316	0.0104	0.0163	42.55
	Test	0.994522	3.8769	3.4309	0.0158	0.0317	0.00
0.25	Train	0.998599	2.1998	2.1347	0.0104	0.0162	39.63
	Test	0.995448	3.7670	3.2642	0.0152	0.0292	0.00
0.3	Train	0.998632	2.2362	2.1187	0.0103	0.0159	37.76
	Test	0.996771	3.9957	3.1592	0.0152	0.0249	0.00
0.35	Train	0.998660	2.0996	2.0729	0.0100	0.0158	37.50
	Test	0.996564	3.7770	3.1770	0.0154	0.0256	0.00

A12: XGBoost-HYT-CV error metrics and computation time for dataset 2

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.999996	0.4733	0.1756	0.0167	0.0235	56.70
	Test	0.999347	5.0689	2.0257	0.2057	0.3213	0.00
0.15	Train	0.999958	0.6992	0.4484	0.0426	0.0810	52.50
	Test	0.998356	3.2565	2.7436	0.2779	0.5092	0.00
0.2	Train	0.999998	0.3207	0.1270	0.0121	0.0166	51.19
	Test	0.998536	5.5748	2.5974	0.2544	0.4773	0.00
0.25	Train	0.999994	0.5947	0.2241	0.0213	0.0296	50.01
	Test	0.998798	5.9645	2.6343	0.2612	0.4275	0.00
0.3	Train	0.999991	0.7346	0.2860	0.0268	0.0364	48.83
	Test	0.996760	7.7042	4.0555	0.4111	0.7237	0.00
0.35	Train	0.999938	0.9363	0.5614	0.0518	0.0984	47.89
	Test	0.997429	4.5025	3.6860	0.3791	0.6494	0.00

A13: XGBoost-HYT-CV error metrics and computation time for dataset 3

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.999999	0.0819	0.0457	0.0011	0.0017	854.14
	Test	0.999996	0.1626	0.0935	0.0023	0.0045	0.07
0.15	Train	0.999999	0.1083	0.0580	0.0014	0.0020	805.24
	Test	0.999993	0.2435	0.1280	0.0032	0.0059	0.12
0.2	Train	0.999993	0.1883	0.1523	0.0038	0.0056	762.71
	Test	0.999981	0.2622	0.2259	0.0056	0.0093	0.02
0.25	Train	0.999992	0.1988	0.1593	0.0040	0.0059	743.42
	Test	0.999979	0.2843	0.2436	0.0060	0.0098	0.03
0.3	Train	0.999998	0.1547	0.0861	0.0021	0.0029	683.95
	Test	0.999983	0.3745	0.2029	0.0050	0.0088	0.22
0.35	Train	0.999990	0.3485	0.1775	0.0044	0.0066	710.28
	Test	0.999982	0.4838	0.2384	0.0059	0.0092	0.12

A14: XGBoost-HYT-CV error metrics and computation time for dataset 4

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.999884	1.1072	0.5583	0.0074	0.0162	45.86
	Test	0.999489	2.2260	1.2978	0.0183	0.0346	0.00
0.15	Train	0.999880	1.1457	0.5869	0.0078	0.0164	42.89
	Test	0.999592	2.4782	1.2597	0.0172	0.0313	0.00
0.2	Train	0.999875	1.1714	0.5890	0.0078	0.0167	42.43
	Test	0.999562	2.8610	1.3873	0.0184	0.0322	0.00
0.25	Train	0.999874	1.1528	0.5874	0.0079	0.0169	42.11
	Test	0.999339	3.4808	1.6656	0.0215	0.0385	0.00
0.3	Train	0.999882	1.3435	0.6417	0.0086	0.0163	40.51
	Test	0.999053	4.0502	1.9311	0.0251	0.0462	0.00
0.35	Train	1.000000	0.0983	0.0458	0.0006	0.0010	39.86
	Test	0.999580	1.3833	0.6249	0.0082	0.0308	0.00

A15: XGBoost-HYT-CV error metrics and computation time for dataset 5

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.998209	3.5987	2.5840	3.0793	4.2881	738.62
	Test	0.993688	5.9589	4.5575	5.4245	7.9795	0.01
0.15	Train	0.998230	3.6084	2.5668	3.0570	4.2638	681.74
	Test	0.993321	6.0305	4.6068	5.5041	8.2314	0.01
0.2	Train	0.998332	3.4992	2.4986	2.9750	4.1448	642.03
	Test	0.993468	6.0727	4.6080	5.5065	8.1301	0.02
0.25	Train	0.998364	3.4508	2.4686	2.9398	4.1075	611.51
	Test	0.993595	6.0391	4.5955	5.4851	8.0355	0.02
0.3	Train	0.998435	3.3630	2.4137	2.8722	4.0108	576.13
	Test	0.993425	6.1291	4.6914	5.6072	8.2004	0.02
0.35	Train	0.998505	3.3123	2.3681	2.8187	3.9206	542.53
	Test	0.993397	6.1503	4.7185	5.6334	8.2211	0.03

A16: RF-HYT error metrics and computation time for dataset 1

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.998370	2.4689	2.3686	0.0116	0.0175	92.49
	Test	0.997937	3.5534	3.0282	0.0132	0.0190	0.03
0.15	Train	0.997756	3.6969	3.1718	0.0154	0.0211	91.88
	Test	0.989217	5.2415	4.5032	0.0211	0.0436	0.04
0.2	Train	0.998331	2.6201	2.4615	0.0120	0.0177	91.70
	Test	0.997031	3.8996	3.5882	0.0166	0.0233	0.04
0.25	Train	0.998324	2.6448	2.4707	0.0121	0.0177	91.62
	Test	0.996603	3.9891	3.6873	0.0172	0.0251	0.04
0.3	Train	0.998315	2.6556	2.4798	0.0120	0.0177	91.30
	Test	0.995926	4.2457	3.9327	0.0189	0.0279	0.04
0.35	Train	0.998325	2.6560	2.4813	0.0120	0.0177	90.67
	Test	0.996047	4.2065	3.9697	0.0192	0.0274	0.05

A17: RF-HYT error metrics and computation time for dataset 2

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.996623	3.7860	3.7753	0.3601	0.7306	94.73
	Test	0.988615	6.6994	6.5982	0.6700	1.3458	0.05
0.15	Train	0.996542	3.8731	3.8681	0.3677	0.7414	94.60
	Test	0.990247	6.3413	6.1437	0.6223	1.2537	0.04
0.2	Train	0.996519	4.0069	3.9258	0.3750	0.7486	94.73
	Test	0.990355	6.5400	6.4715	0.6339	1.2484	0.05
0.25	Train	0.996221	4.1478	4.1280	0.3920	0.7826	94.61
	Test	0.990523	6.7810	6.4274	0.6373	1.2167	0.05
0.3	Train	0.995939	4.3103	4.2880	0.4018	0.8053	93.93
	Test	0.988792	7.2517	6.9806	0.7076	1.3318	0.05
0.35	Train	0.995514	4.5377	4.5663	0.4216	0.8448	93.63
	Test	0.986856	7.6176	7.5696	0.7784	1.4355	0.06

A18: RF-HYT error metrics and computation time for dataset 3

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.997536	2.6972	3.1092	0.0772	0.1098	639.18
	Test	0.997486	2.7587	3.1598	0.0782	0.1109	0.66
0.15	Train	0.997057	2.8699	3.3563	0.0833	0.1198	616.49
	Test	0.997055	2.9175	3.3875	0.0838	0.1202	0.95
0.2	Train	0.996813	3.0463	3.5258	0.0876	0.1246	578.35
	Test	0.996775	3.0964	3.5747	0.0883	0.1256	1.25
0.25	Train	0.997114	3.0223	3.4564	0.0859	0.1190	553.77
	Test	0.997020	3.0882	3.5230	0.0871	0.1205	1.54
0.3	Train	0.996563	3.2026	3.6845	0.0915	0.1293	526.84
	Test	0.996476	3.2708	3.7579	0.0931	0.1312	1.84
0.35	Train	0.996257	3.3620	3.8543	0.0957	0.1349	504.69
	Test	0.996155	3.4286	3.9267	0.0974	0.1369	2.13

A19: RF-HYT error metrics and computation time for dataset 4

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.998777	4.6438	3.4725	0.0459	0.0623	93.06
	Test	0.996628	8.7104	6.0656	0.0853	0.1052	0.04
0.15	Train	0.998774	4.6970	3.4587	0.0458	0.0621	93.02
	Test	0.996955	8.3456	5.7427	0.0782	0.1006	0.04
0.2	Train	0.998644	4.8158	3.5317	0.0470	0.0649	92.84
	Test	0.996860	8.7276	6.0056	0.0798	0.1035	0.05
0.25	Train	0.998560	5.4046	3.6573	0.0491	0.0664	92.71
	Test	0.995582	10.3662	6.6901	0.0865	0.1126	0.05
0.3	Train	0.998522	5.6056	3.7277	0.0501	0.0678	92.62
	Test	0.995968	11.0292	6.7384	0.0877	0.1110	0.05
0.35	Train	0.998569	5.9452	3.8356	0.0513	0.0670	92.74
	Test	0.995856	12.0637	6.9854	0.0922	0.1140	0.05

A20: RF-HYT error metrics and computation time for dataset 5

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.988546	8.3045	6.6236	7.8932	10.8225	1466.34
	Test	0.984053	9.6787	7.8027	9.2871	12.6218	0.27
0.15	Train	0.988598	8.3390	6.6246	7.8896	10.8009	1391.14
	Test	0.984385	9.5543	7.6500	9.1400	12.5156	0.37
0.2	Train	0.988797	8.2691	6.5800	7.8347	10.7183	1292.09
	Test	0.984349	9.6299	7.6689	9.1642	12.5102	0.48
0.25	Train	0.988907	8.2516	6.5620	7.8144	10.6757	1187.46
	Test	0.984474	9.5732	7.6645	9.1481	12.4536	0.59
0.3	Train	0.989082	8.1739	6.5090	7.7455	10.5710	1132.30
	Test	0.984587	9.5555	7.6599	9.1551	12.4918	0.70
0.35	Train	0.989060	8.1740	6.5223	7.7632	10.5848	1024.11
	Test	0.984302	9.6042	7.7266	9.2248	12.6048	0.80

A21: DANN-MPIH-HYT error metrics and computation time for dataset 1

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.997365	9.9829	6.1221	0.0300	0.0370	1886.42
	Test	0.996802	12.1298	6.7708	0.0294	0.0352	0.00
0.15	Train	0.997675	4.0850	3.3169	0.0161	0.0214	1845.73
	Test	0.996361	4.6924	3.9624	0.0185	0.0251	0.00
0.2	Train	0.998321	3.6118	2.7219	0.0133	0.0187	1799.71
	Test	0.996921	4.7835	3.8187	0.0176	0.0241	0.00
0.25	Train	0.997780	5.5836	3.5294	0.0173	0.0220	1749.57
	Test	0.996658	6.4997	4.4082	0.0206	0.0265	0.00
0.3	Train	0.997596	4.6941	3.4558	0.0168	0.0214	1681.96
	Test	0.995663	5.5954	4.4151	0.0212	0.0292	0.00
0.35	Train	0.997569	9.6396	5.7469	0.0278	0.0360	1632.99
	Test	0.994879	11.9764	6.9677	0.0337	0.0445	0.00

A22: DANN-MPIH-HYT error metrics and computation time for dataset 2

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.997949	8.2543	4.2320	0.4037	0.5701	2427.28
	Test	0.997234	8.5315	4.6032	0.4674	0.6678	0.00
0.15	Train	0.998265	8.8736	8.1800	0.7776	1.1738	2371.74
	Test	0.996057	9.1368	8.8814	0.8996	1.4645	0.00
0.2	Train	0.998924	5.6844	3.2794	0.3132	0.4716	2259.60
	Test	0.997134	7.0297	4.3641	0.4275	0.7554	0.00
0.25	Train	0.997864	9.1299	5.4257	0.5152	0.6991	2214.47
	Test	0.995826	9.2616	5.8737	0.5824	0.8482	0.00
0.3	Train	0.998886	5.8170	3.0852	0.2891	0.4215	2145.76
	Test	0.997548	6.4518	3.9919	0.4046	0.6212	0.00
0.35	Train	0.998295	7.0161	3.8995	0.3600	0.5370	2132.17
	Test	0.995402	7.4987	5.1503	0.5296	0.8511	0.00

A23: DANN-MPIH-HYT error metrics and computation time for dataset 3

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.998479	8.3994	4.1577	0.1032	0.1214	82204.19
	Test	0.998499	8.4009	4.1622	0.1030	0.1209	0.00
0.15	Train	0.999479	2.6188	1.8159	0.0451	0.0620	76612.20
	Test	0.999480	2.6522	1.8142	0.0449	0.0619	0.00
0.2	Train	0.998825	13.2393	9.8610	0.2450	0.2792	128921.59
	Test	0.998820	13.3730	9.8737	0.2438	0.2789	0.01
0.25	Train	0.999149	10.1431	8.6224	0.2142	0.2629	67372.66
	Test	0.999147	10.1633	8.6072	0.2127	0.2616	0.01
0.3	Train	0.999736	6.0444	2.3922	0.0594	0.0692	62757.34
	Test	0.999731	6.1035	2.4008	0.0595	0.0694	0.01
0.35	Train	0.999573	5.3433	2.2097	0.0549	0.0661	104372.08
	Test	0.999572	5.3707	2.2192	0.0550	0.0662	0.02

A24: DANN-MPIH-HYT error metrics and computation time for dataset 4

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.994536	15.5010	6.8196	0.0902	0.1146	2310.94
	Test	0.993340	14.1054	7.1831	0.1010	0.1293	0.00
0.15	Train	0.999868	2.1594	1.8287	0.0242	0.0354	2272.40
	Test	0.999479	2.2739	2.0779	0.0283	0.0467	0.00
0.2	Train	0.999791	1.5648	1.1849	0.0158	0.0259	2194.28
	Test	0.999736	2.3799	1.5130	0.0201	0.0296	0.00
0.25	Train	0.999800	1.3634	0.8940	0.0120	0.0219	2076.55
	Test	0.999740	1.9617	1.2032	0.0156	0.0240	0.00
0.3	Train	0.999565	3.8565	2.8232	0.0379	0.0520	2010.29
	Test	0.999542	3.9727	2.8608	0.0372	0.0481	0.00
0.35	Train	0.999828	1.6510	1.2977	0.0173	0.0279	1963.30
	Test	0.999276	1.8574	1.5542	0.0205	0.0429	0.00

A25: DANN-MPIH-HYT error metrics and computation time for dataset 5

Train-Test ratio	Train or Test	R	MAPE (%)	MAMPE (%)	MAE	RMSE	Computation Time (s)
0.1	Train	0.968626	27.8667	22.8715	27.2555	33.3211	51697.53
	Test	0.969549	28.0178	23.3875	27.8366	33.7000	0.01
0.15	Train	0.995583	5.4581	4.1154	4.9013	6.7186	49433.92
	Test	0.993535	5.9577	4.5578	5.4455	8.0671	0.01
0.2	Train	0.994376	6.3444	5.4181	6.4511	9.0713	45846.40
	Test	0.991945	6.8458	5.8182	6.9527	10.1466	0.01
0.25F	Train	0.989866	11.9005	9.9695	11.8723	15.9620	42781.25
	Test	0.988501	12.1077	10.1967	12.1705	16.4945	0.02
0.3	Train	0.984736	10.3018	7.9582	9.4699	13.2557	22062.03
	Test	0.983535	10.5106	8.1647	9.7585	13.7166	0.01
0.35	Train	0.984521	12.4428	10.6888	12.7224	17.6987	20765.28
	Test	0.983797	12.6741	10.8728	12.9811	18.0754	0.01

Appendix B

B1: Average error metrics across datasets and train-test ratios

	LR	POLYREG-HYT	XGBoost-HYT-CV	RF-HYT	DANN-MPIH-HYT
R	0.8833	0.9777	0.9974	0.9924	0.9951
MAPE (%)	43.2802	15.2506	3.7076	6.7255	8.0772
MAMPE (%)	25.1602	9.2980	2.4903	5.6210	5.7225
MAE	6.0396	3.5934	1.1726	2.0104	2.6513
RMSE	7.8732	5.0521	1.7445	2.8196	3.5591

B2: Variation in error metrics across datasets and train-test ratios

	LR	POLYREG-HYT	XGBoost-HYT-CV	RF-HYT	DANN-MPIH-HYT
R	0.0073	0.0042	0.0012	0.0015	0.0059
MAPE (%)	4.6848	1.8837	1.2633	1.1040	9.2945
MAMPE (%)	0.8800	0.7607	0.5040	0.7047	4.9627
MAE	0.2090	0.2987	0.0795	0.0533	4.4201
RMSE	0.2478	0.3367	0.1287	0.0780	4.9975